강의 계획 및 평가 방법

주차	날짜	강의 내용	과제 주제	대면/비대면	평가
1	03/06	강의 소개		Online	
2	03/13	데이터 마이닝 절차		A704	
3	03/20	데이터 탐색 및 시각화		B224	
4	03/27	차원 축소	과제 1	Online	
5	04/03	예측성능 평가		Online	과제 1 (10%)
6	04/10	다중 선형 회귀분석		A704	
7	04/17	중간 프로젝트 발표		A704	30%
8	04/24	k-최근접이웃 알고리즘 나이브 베이즈 분류		Online	
9	05/01 보강: 06/15(목)	휴업일(근로자의 날) 동영상 강의		Online	
10	05/08	분류와 회귀 나무	과제 2	Online	
11	05/15	로지스틱 회귀분석		A704	과제 2 (10%)
12	05/22	신경망 판별 분석		Online	
13	05/29 보강: 06/02(금) 19시	대체 공휴일(부처님 오신 날) 연관 규칙		Online	
14	06/05	군집 분석		A704	
15	06/12	기말 프로젝트 발표		B224	40%

Data Mining for Business Analytics

Ch. 12 Discriminant Analysis

2023.05.22.

Dept. of Industrial AI

INDUSTRIAL AI RESEARCH CENTER

Contents

12.1 Introduction

- 판별 분석
- 분류기법. 로지스틱 회귀분석과 같이 분류와 프로파일링에 사용되는 전통적인 통계 기법
- 여러 클래스의 측정값을 활용하여 새로운 항목을 이들 클래스 중의 하나로 분류
- ex) 동물과 식물을 서로 다른 종과 그 하위 종으로 분류

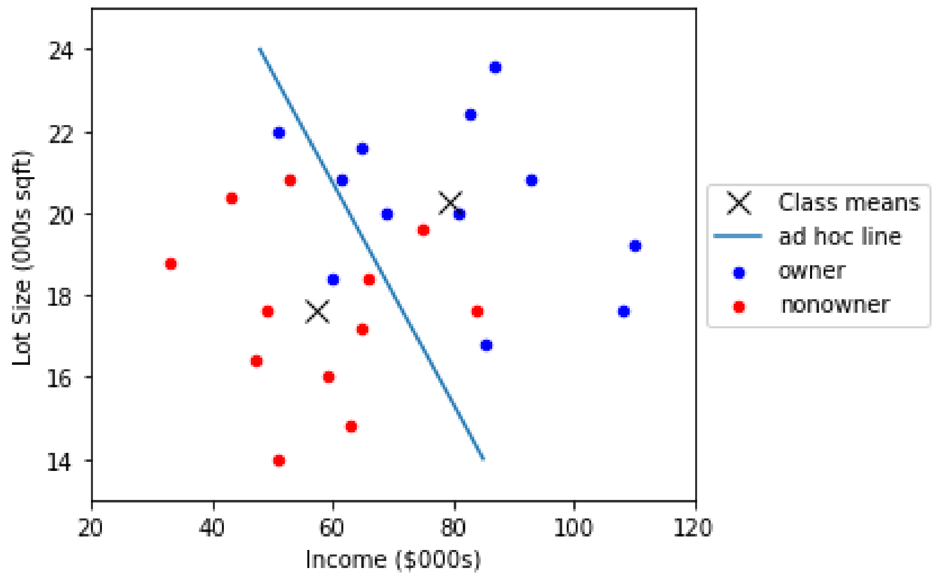
  대출, 신용카드, 보험가입 신청자들을 저위험군, 고위험군으로 분류

  신제품이 출시되었을 때, 제품에 대한 수용시기에 따라 고객 그룹을 나누는데 사용

12.1 Introduction

Example 1: Riding Mowers(승차식 잔디깎는 기계)

- 한 도시의 가구들을 대상을 승차식 잔디깎는 기계를 구입할 가구와 구입하지 않을 가구에 대한 분류
- 소유 가구: 12개 / 비소유 가구: 12개
- 선형 분류규칙: 24가구 중 4개 가구 오분류

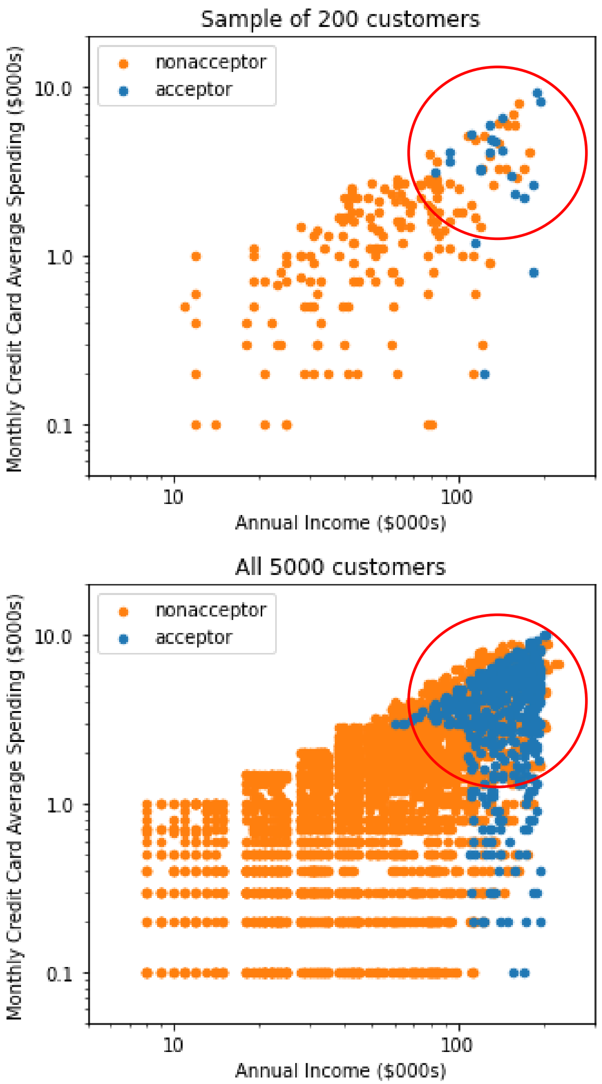


[실습] Figure 12.2

Example 2: Personal Loan Acceptance(개인대출 수락)

- 유니버셜(Universal) 은행은 채무가 있는 고객들을 개인대출 고객으로 전환하는 방법을 찾고자 함.
- 목적: 은행은 대출 제안을 받아들일 가능성이 높은 새로운 고객 파악
- 예제) 입력변수: 고객의 연간 소득(Income, 1000달러 단위), 월별 신용카드 평균사용액(CCAVG, 1000달러 단위)

- Income, CCAVG → 개인대출수락 여부
  - ✓ [위 그림, 200명 샘플] 소득이 높고 CCAVG가 많은 영역에 많은 데이터가 조밀하게 분포 → 대출 제안을 수락한 사람과 거절한 사람들 사이의 분리가 모호
  - ✓ [아래 그림, 5000명 전체 고객] 데이터가 많을 경우 적은 수의 데이터를 다룰 때에 비해 상대적으로 문제의 복잡성 증가



12.2 Distance of a Record from a Class

- 분류에 대한 일반적인 개념: 데이터를 그와 가장 가까운 클래스 속하게 하는 것 → 클래스와 관측값 사이의 거리 측정 필요
- Assume: 신규 고객의 연간 소득(Income: x)을 기반으로 개인대출 제안에 대한 고객의 수락 여부 결정
  - ✓ 수락자들의 평균 연간소득: \$144.75K / 비수락자들의 평균 연간소득: \$66.24K
- 유클리드 거리 법칙(Euclidean distance rule)
  - ✓ x 가 비수락자 클래스의 평균 연간소득보다 수락자 클래스의 평균 연간 소득에 더 가까우면 → 수락자로 분류
  - ✓ $if\ |x - 144.75| < |x - 66.24| \quad then\ class = acceptor \quad otherwise\ \ nonacceptor$
  - ✓ 예측변수가 2개 이상으로 증가하게 되면, 클래스의 중심(centroid) 사용
  - ✓ 클래스의 평균 벡터: $\bar{X} = [\bar{x}_1, \cdots, \bar{x}_p]$
  - ✓ p 개의 입력변수를 갖는 관측값 $X = [x_1, \cdots, x_p]$ 와 클래스의 중심 $\bar{X}$ 사이의 유클리드 거리

$$D_{Euclidean}(X, \bar{X}) = \sqrt{(x_1 - \bar{x}_1)^2 + (x_2 - \bar{x}_2)^2 + \cdots + (x_p - \bar{x}_p)^2}$$

12.2 Distance of a Record from a Class

- 유클리드 거리 사용의 단점

1. 입력변수들에 대한 측정 단위 선택에 따라 거리 측정이 달라짐
   - ✓ 연간 소득을 1000달러 기준이 아닌 달러 기준으로 측정한다면 산출된 유클리드 거리가 달라짐
2. 변수의 변동성을 반영하지 못 함
   - ✓ 두 클래스의 연간 소득의 변동성 비교: 수락자들의 표준편차(31.6K) 〈 비수락자들의 표준편차(40.6K) → 신규 고객의 연간소득이 수락자들의 평균소득에 더 가깝게 측정될 수 있음. 하지만 비수락자들 연간소득의 변동성이 크기 때문에, 이 고객은 비수락자로 분류하는 것이 타당할 수 있음
   - ✓ 표준편차를 고려한 측정 필요 → 표준화된 값인 z 값($z-score$)을 사용한 거리 측정
3. 변수들 간의 상관관계 무시
   - ✓ 분류하는 데 유용한 변수들이라고 할지라도 이들 간에 상관관계가 있다면, 한 변수가 갖는 효과를 다른 변수들이 포함하기 때문에 실질적으로는 그 변수가 중요하지 않을 수 있음 → **통계적 거리(Statistical distance) 또는 마할라노비스 거리(Mahalanobis distance) 사용**

12.2 Distance of a Record from a Class

- 통계적 거리(Statistical distance) 또는 마할라노비스 거리(Mahalanobis distance)
  - ✓ p : 예측변수의 개수　　　S : 공분산 행렬
  - ✓ $[\quad]^T$: transpose operation, 열벡터를 행벡터로 변환
  - ✓ S^{-1} : S 의 역행렬, 일차원에서의 나눗셈이 p 차원으로 확장된 것을 의미

$$D_{Statistical}(X,\bar{X}) = ([X-\bar{X}]^T S^{-1}[X-\bar{X}])^{\frac{1}{2}}$$

$$= \left([(x_1-\bar{x}_1),(x_2-\bar{x}_2),\cdots,(x_p-\bar{x}_p)] S^{-1} \begin{bmatrix} x_1-\bar{x}_1 \\ x_2-\bar{x}_2 \\ \cdots \\ x_p-\bar{x}_p \end{bmatrix} \right)^{\frac{1}{2}}$$

  - ✓ If $p=1$ (1개 예측변수), 예측변수에서 평균을 빼고, 이를 표준편차로 나누기 때문에 → z 값($z-score$)
  - ✓ If $S=I$ (모든 예측변수가 서로 상관관계 없음), $D_{Statistical}(X,\bar{X}) = D_{Euclidian}(X,\bar{X})$
  - ✓ 통계적 거리는 예측변수의 평균값, 값의 퍼짐 정도, 변수들 간의 상관관계 고려
- 판별분석: 분류 함수(Classification Function) 사용
  - ✓ 통계적 거리를 기반으로 분리선(예측변수가 3개 이상인 경우에는 분리 초평면(hyperplane))을 찾음
  - ✓ 분리선은 모든 클래스들의 평균들로부터 동일 거리에 위치하게 됨

12.3 Fisher's Linear Classification Functions

- 선형 분류 함수(Linear Classification Function)의 기본 개념 [R. A. Fisher(1936)]

$$Maximize: \frac{between - class\ variability\ (\text{클래스 사이의 변동성})}{within - class\ variability\ (\text{클래스 내 변동성})}$$

- 각 클래스 내의 관측값들은 매우 동질적이나 다른 클래스들과 비교했을 때는 서로 이질적인 클래스들을 얻고자 함
- 선형 분류 함수는 각 클래스 별 1개 씩 생성
- 관측값이 주어졌을 때, 각 클래스까지의 근접성을 선형 분류 함수의 점수로 계산 → 이들 중 가장 높은 분류 점수(가장 가까운 통계적 거리)를 주는 클래스로 관측값 분류

잔디 깎는 기계에 대한 판별 분석 결과
(선형 분류 함수 추정)

```
Coefficients [[0.1002303  0.78518471]]
Intercept [-21.73876167]
Decision function
 = 0.10*Income + 0.79*Lot_Size + -21.74

Example Income=$60K Lot_Size=18.4Kft2
  0.10*60 + 0.79*18.4 + -21.74 = -1.28
  negative => nonowner
```

- 분류 함수 Score
  - ✓ If 'owner(1)' 분류 함수 score 〉 'nonowner(0)' score → 'owner'
  - ✓ Decision Function(= 클래스 '1' 점수 – 클래스 '0' 점수) : (–) → "0" class / (+) → "1" class
- 레코드 #1: Income=60K, Lot size=18.4K
- Decision Function = $(0.1)(60) + (0.79)(18.4) + (-21.74) = -1.28$ → 'nonowner(0)'로 분류

※ Scikit-learn's Function: "LinearDiscrimitiveAnalysis()"
– 2개의 클래스에 대한 분류 함수의 차이 반환: Decision Function

12.3 Fisher's Linear Classification Functions

- 분류의 다른 방법: 관측값이 각 클래스에 속할 확률을 계산하고, 이 중에서 가장 높은 확률을 갖는 클래스로 관측값 할당
- 장점: 데이터에 대해 랭킹을 매길 수 있음 → 관측값을 확률값에 따라 내림차순으로 정렬하여 향상 곡선(lift curve)을 그릴 수 있음
- 클래스 개수: m , 분류 점수: $c_1(i), c_2(i), \cdots, c_m(i)$
- 관측값 i 가 클래스 k 에 속할 확률

$$P[\text{record } i \text{ (측정값 } x_1, x_2, \cdots, x_p \text{ 을 가짐)가 클래스 } k\text{에 속할 경우}]$$

$$= \frac{e^{c_k(i)}}{e^{c_1(i)} + e^{c_2(i)} + \cdots + e^{c_m(i)}}$$

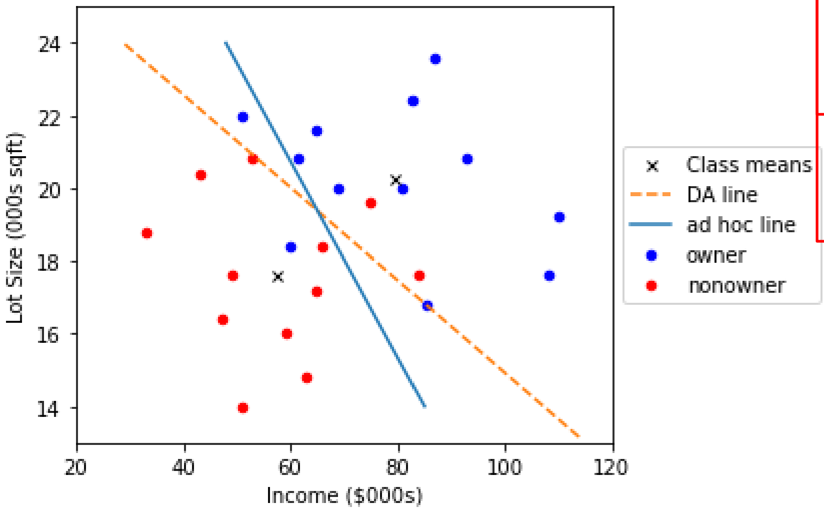
12.3 Fisher's Linear Classification Functions

- 클래스 개수: m , 분류 점수: $c_1(i), c_2(i), \cdots, c_m(i)$
- 관측값 i 가 클래스 k 에 속할 확률

$$P[\text{record } i \text{ (측정값 } x_1, x_2, \cdots, x_p \text{ 을 가짐)가 클래스 } k\text{에 속할 경우}] = \frac{e^{c_k(i)}}{e^{c_1(i)} + e^{c_2(i)} + \cdots + e^{c_m(i)}}$$

✓ 분류 score 차이: Dec. Function
✓ 분류 클래스: Prediction
✓ 확률: p(Owner)

※ 오분류 개수: 원래 선형 분류: 4개 → 판별 분석: 3개



	Income	Lot_Size	Ownership	Dec. Function	Prediction	p(Owner)
1	60.0	18.4	Owner	-1.277545	Nonowner	0.217968
2	85.5	16.8	Owner	0.022032	Owner	0.505508
3	64.8	21.6	Owner	1.716152	Owner	0.847632
4	61.5	20.8	Owner	0.757244	Owner	0.680755
5	87.0	23.6	Owner	5.511634	Owner	0.995977
6	110.1	19.2	Owner	4.372141	Owner	0.987533
7	108.0	17.6	Owner	2.905362	Owner	0.948111
8	82.8	22.4	Owner	4.148445	Owner	0.984456
9	69.0	20.0	Owner	0.880823	Owner	0.706993
10	93.0	20.8	Owner	3.914499	Owner	0.980440
11	51.0	22.0	Owner	0.647047	Owner	0.656345
12	81.0	20.0	Owner	2.083587	Owner	0.889298
13	75.0	19.6	Nonowner	1.168131	Owner	0.762807
14	52.8	20.8	Nonowner	-0.114760	Nonowner	0.471342
15	64.8	17.2	Nonowner	-1.738661	Nonowner	0.149483
16	43.2	20.4	Nonowner	-1.391044	Nonowner	0.199241
17	84.0	17.6	Nonowner	0.499835	Owner	0.622420
18	49.2	17.6	Nonowner	-2.988180	Nonowner	0.047963
19	59.4	16.0	Nonowner	-3.222126	Nonowner	0.038342
20	66.0	18.4	Nonowner	-0.676163	Nonowner	0.337118
21	47.4	16.4	Nonowner	-4.110816	Nonowner	0.016130
22	33.0	18.8	Nonowner	-3.669689	Nonowner	0.024851
23	51.0	14.0	Nonowner	-5.634430	Nonowner	0.003560
24	63.0	14.8	Nonowner	-3.803519	Nonowner	0.021806

12.4 Classification Performance of Discriminant Analysis

- 판별 분석 방법에서 분류 점수(classification score)에 이르는 2가지 가정

1. 모든 클래스의 측정값들은 다변량 정규분포로부터 얻어진다고 가정
   - ✓ 이 가정이 만족된다면, 다른 분류 방법들보다 더 효과적인 분류 방법
   - ✓ 로지스틱 회귀에 비해 30% 정도 더 효과적인 성능을 보임[Efron(1975)] → 동일한 결과를 도출하는데 있어서 로지스틱 회귀에 비해 30% 정도 적은 데이터 만을 필요로 함
   - ✓ 예측변수들이 정규분포를 따르지 않고 더미변수를 포함하더라도 비교적 강건한 결과를 보임(가장 작은 클래스의 데이터 수가 충분히 크다면(대략 20개 이상))
   - ✓ 단변량 공간(단일 예측 변수 사용)이나 다변량 공간 모두에서 이상치에 민감함 → 탐색적 분석을 통해 이상치를 찾고 이를 제거할지를 미리 결정하는게 좋음
2. 예측변수들 간의 상관관계는 특정 클래스 내에서나 다른 클래스들 내에서 동일하다고 가정
   - ✓ 각 클래스 내에서 예측변수들 간의 상관관계 행렬(correlation matrix)을 통해 확인 가능
   - ✓ 만약, 클래스 별로 상관관계가 상당히 다르다면, 분류기는 변동이 큰 클래스로 분류하는 경향을 가짐 → 이차 판별분석(quadratic discriminant analysis) 사용 추천

- 분류 정확도 평가
  - ✓ 검증 데이터에 대해 Confusion matrix와 Lift curve 활용

12.5 Prior Probabilities

- 지금까지는 각 클래스들이 특정한 레코드를 만나게 될 확률이 클래스 별로 동일하다고 가정 (즉, 각 클래스 별로 제시되는 관측값의 빈도가 동일하다고 가정)
- 만약, 어떤 한 레코드를 만날 확률이 클래스 별로 다르다면? → 분류 함수의 수정 필요
  - ✓ p_j : 클래스 j 에 속할 사전확률(Prior or future probability)
  - ✓ 각 클래스 별 분류 함수에 $\log(p_j)$ 를 더해 줌
- ex) Assume: 전체 모집단에서 소유자 비율: 15% (표본에서는 50%였음)
  - ✓ 좀 더 적은 수의 가구를 소유자 클래스로 분류해야 함
  - ✓ 분류 함수의 상수 조정(원래 값 + $\log(p_1) - \log(p_0)$)

$$(-21.74) + \log(0.15) - \log(0.85) = -22.4933$$

$$\log(0.15) - \log(0.85) = -0.75333$$

※ Decision Function = 클래스 '1' 점수 – 클래스 '0' 점수

```
Coefficients [[0.1002303  0.78518471]]
Intercept [-21.73876167]
Decision function
 = 0.10*Income + 0.79*Lot_Size + -21.74

Example Income=$60K Lot_Size=18.4Kft2
  0.10*60 + 0.79*18.4 + -21.74 = -1.28
  negative => nonowner
```

17	84.0	17.6	Nonowner	0.499835	Owner	0.622420

- ✓ #17: 0.622420 + (-0.75333) = -0.13091
- ✓ Owner (오분류) → Nonowner (정분류)

12.5 Prior Probabilities

- ex) Assume: 전체 모집단에서 소유자 비율: 15% (표본에서는 50%였음)
  - ✓ 좀 더 적은 수의 가구를 소유자 클래스로 분류해야 함
  - ✓ 분류 함수의 상수 조정(원래 값 + $\log(p_1) - \log(p_0)$)

$$(-21.74) + \log(0.15) - \log(0.85) = -22.4933$$

$$\log(0.15) - \log(0.85) = -0.75333$$

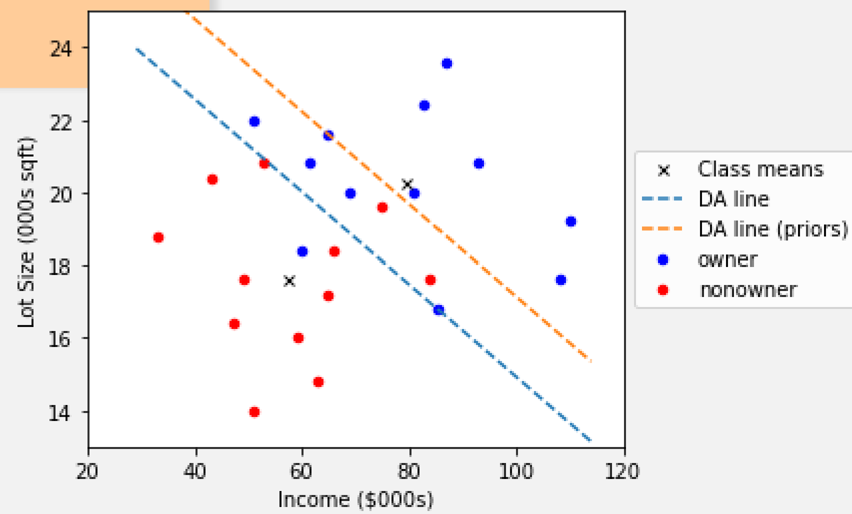
※ Decision Function = 클래스 '1' 점수 − 클래스 '0' 점수

```
Coefficients [[0.1002303  0.78518471]]
Intercept [-21.73876167]
Decision function
 = 0.10*Income + 0.79*Lot_Size + -21.74

Example Income=$60K Lot_Size=18.4Kft2
  0.10*60 + 0.79*18.4 + -21.74 = -1.28
  negative => nonowner
```

17	84.0	17.6	Nonowner	0.499835	Owner	0.622420

- ✓ #17: 0.622420 + (−0.75333) = −0.13091
- ✓ Owner (오분류) → Nonowner (정분류)



12.6 Unequal Misclassification Costs

- 오분류 비용이 클래스 별로 현저하게 다르다면, 단순 오차율 보다는 오분류의 기대비용을 최소화시켜야 함
- 2개의 클래스로 분류하는 경우 → 분류 함수의 수정 필요
  - ✓ q_1 : 클래스 '1'에 속한 레코드를 클래스 '2'로 오분류한 비용
  - ✓ q_2 : 클래스 '2'에 속한 레코드를 클래스 '1'로 오분류한 비용
  - ✓ 이 비용들을 각 클래스 별 분류 함수의 상수에 각각 $\log(q_1)$, $\log(q_2)$를 더해 줌
  - ✓ 사전 확률과 오분류 비용 함께 반영: 각 클래스 별 분류 함수의 상수에 각각 $\log(p_1 q_1)$, $\log(p_2 q_2)$를 더해 줌
- 각각의 클래스에 오분류 비용 q_1, q_2 를 함께 다루기 어려움 → 오분류 비율 $\frac{q_2}{q_1}$ 추정
  - ✓ ex) 채무 불이행자를 오분류하는 비율은 채무 이행자를 오분류하는 비용보다 10배 더 크다
  - ✓ 분류 함수의 조정: $q_1 = 1$, $q_2 = ratio$ 로 설정하고, 클래스 '2'의 상수에 $\log(q_2/q_1)$를 더해 줌

12.7 Classifying More Than Two Classes

Example 3: Medical Dispatch to Accident Scenes (사고현장으로 의료진 파견)

- 119 응급 구조대에서는 신고를 통해 주어진 대략적인 정보를 바탕으로 어떤 팀을 보내야 할 지 결정해야 함
- 데이터: 11개 예측 변수 (사고 정도(무상해, 중상, 사망 등), 요일, 기상 상태, 도로 유형 등)
- 목적: 예측 변수들이 부상 유형을 분류하는데 얼마나 유용하게 사용될 수 있는가 검토

Accident #	RushHour	WRK_ZONE	WKDY	INT_HWY	LGTCON	LEVEL	SPD_LIM	SUR_COND	TRAF_WAY	WEATHER	MAX_SEV
1	1	0	1	1	dark_light	1	70	ice	one_way	adverse	no-injury
2	1	0	1	0	dark_light	0	70	ice	divided	adverse	no-injury
3	1	0	1	0	dark_light	0	65	ice	divided	adverse	non-fatal
4	1	0	1	0	dark_light	0	55	ice	two_way	not_adverse	non-fatal
5	1	0	0	0	dark_light	0	35	snow	one_way	adverse	no-injury
6	1	0	1	0	dark_light	1	35	Wet	divided	adverse	no-injury
7	0	0	1	1	dark_light	1	70	wet	divided	adverse	non-fatal
8	0	0	1	0	dark_light	1	35	wet	two_way	adverse	no-injury
9	1	0	1	0	dark_light	0	25	wet	one_way	adverse	non-fatal
10	1	0	1	0	dark_light	0	35	wet	divided	adverse	non-fatal

12.7 Classifying More Than Two Classes

Example 3: Medical Dispatch to Accident Scenes (사고현장으로 의료진 파견)

- 데이터: 600개 (학습 데이터와 검증 데이터 분류)
- 결과 변수: 3개의 클래스 (fatal, no-injury, non-fatal)
- 학습 데이터에 대한 판별 분석 수행
- 분류 함수: 각 3개의 클래스 별로 존재
- Confusion matrix: 3x3 구조를 갖는 행렬
- 분류 규칙: 가장 높은 분류 점수를 주는 클래스로 분류
- ex) #1의 각 클래스 별 분류 점수

  'no-injury': $-0.89 + (0.03)(1) + (0.22)(0) + \cdots + (0.08)(1) = -0.46$

  'non-fatal' = −0.96

  'fatal' = −5.94

  → 'no-injury'로 분류

Accident #	RushHour	WRK_ZONE	...	WEATHER
1	1	0	...	adverse

Coefficients and intercept

	fatal	no-injury	non-fatal
constant	-1.972659	-0.891172	-0.610471
RushHour	-0.996411	0.033430	-0.015774
WRK_ZONE	-0.457188	0.220012	-0.204480
WKDY	-1.471777	0.165707	-0.135404
INT_HWY	0.755344	-0.075816	0.060599
LGTCON_day	0.009515	-0.031421	0.030124
LEVEL	0.976626	-0.082717	0.063598
SPD_LIM	0.048033	0.004381	-0.005014
SUR_COND_dry	-5.999809	-0.164874	0.257895
TRAF_two_way	0.752985	-0.012844	-0.000048
WEATHER_adverse	-6.596690	0.079166	0.032564

Confusion Matrix (Accuracy 0.5283)

	Prediction		
Actual	fatal	no-injury	non-fatal
fatal	1	1	3
no-injury	6	114	172
non-fatal	6	95	202

12.7 Classifying More Than Two Classes

Example 3: Medical Dispatch to Accident Scenes (사고현장으로 의료진 파견)

- ex) #0의 각 클래스 별 분류 점수

  'fatal' = −5.94

  'no-injury': −0.46

  'non-fatal' = −0.96

  → 'no-injury'로 분류

	Classification	Actual	Score fatal	Score no-injury	Score non-fatal
0	no-injury	no-injury	-5.94	-0.46	-0.96
1	no-injury	non-fatal	-1.05	-0.46	-1.04
2	no-injury	no-injury	-7.88	-0.63	-0.77
3	no-injury	no-injury	-8.38	-0.54	-0.84
4	no-injury	non-fatal	-9.84	-0.50	-0.85

	Propensity fatal	Propensity no-injury	Propensity non-fatal
0	0.00e+00	0.62	0.38
1	2.63e-01	0.47	0.27
2	0.00e+00	0.54	0.46
3	0.00e+00	0.57	0.43
4	0.00e+00	0.59	0.41

- 각 교통사고가 3개의 클래스에 속할 확률 계산
- ex) #0의 'non-fatal'에 속할 확률

$$\frac{e^{-0.96}}{e^{-5.94}+e^{-0.46}+e^{-0.96}} = 0.38$$

  'no-injury'에 속할 확률 = 0.62

  → 'no-injury'로 분류

✓ Remind

분류 점수: $c_1(i), c_2(i), \cdots, c_m(i)$

$P[\text{record } i$ 가 클래스 k에 속할 경우$]$

$$= \frac{e^{c_k(i)}}{e^{c_1(i)}+e^{c_2(i)}+\cdots+e^{c_m(i)}}$$

12.8 Advantages and Weaknesses

- 판별 분석은 데이터 마이닝 방법이라기 보다는 통계적 분류 방법
- 사회과학 분야에서 유용하게 사용되고 있음
- 판별 분석과 다중선형 회귀분석의 장점
  - ✓ 예측변수의 최적 가중치를 찾는 과정을 포함하고 있음
    - 선형 회귀분석에서의 가중치: 종속변수와 관련이 있음
    - 판별 분석에서의 가중치: 클래스들을 분리시키는 것과 관련이 있음
  - ✓ 두 분석 모두 최소제곱법(least square) 사용 → 결과로 얻은 추정값들은 국소 최적값(local optima)에 영향을 받지 않고 강건함(robustness)
  - ✓ 두 방법의 기본 가정: 정규성(normality)
    - 판별분석: 예측변수들이 다변량 정규분포를 따른다고 가정 → 현실적 상황에서는 많은 경우 지켜지지 못 함에도 불구하고 이 문제에 대해 강건함 → 그 이유는 판별분석이 데이터를 선형 분리선, 즉 단순한 형태의 분리 경계를 찾는 데에만 사용하기 때문[Hastie(2001)]
- 분류 방법으로서의 장점 (다중 선형 회귀 분석과 함께)
  - ✓ 단일 예측변수의 기여도에 대한 추정치 제공 → 예측변수의 중요도에 대한 순위를 정할 수 있으므로 변수 선택에 유용하게 활용 가능
  - ✓ 계산과정이 단순하고 간결한(parsimonious) 모델
  - ✓ 데이터를 최대한 이용하여 패러미터를 추정하므로 데이터의 개수가 작을 때 유용