4. Classification

Classification은 regression과는 달리, 반응변수 Y가 qualitative 즉, 질적인 자료일 때 사용하는 방법이다. 관측값을 특정한 class나 category로 분류하는 작업이다. 이 때, 분류가 될 때에는 확률을 통해서 '예측'하기 때문에, regression 기법과 비슷한 특성이 있다.

우리는 이 책에서는 3개의 classifier을 다룰 것이다; logistic 'regression', linear discriminant analysis, quadratic discriminant analysis, K-nearest neighbors.

4.1 An Overview of Classification

balance(카드대금), income(소득)이라는 설명변수들을 통해 default(신용불량) 여부를 알아내는 예시이다. 소득은 큰 상관관계가 없고, 대체로 balance가 많을수록 default가 yes로 나타났다.

4.2 Why Not Linear Regression?

그렇다면 왜 Linear Regression에서는 반응변수 Y가 질적변수일 때는 적합하지 않은가? 예를 들어, stroke일 때는 1, drug overdose일 때는 2, epileptic seizure일 때는 3이라고 해보자. 그러나 각각의 변수들을 수치화해서 표현했을 때, 이들간의 관계는 수치화될 수 없다. 또한, 최소제곱법(least square)을 사용하여, 회귀분석을 하면, 설명변수에 따라 확률값이 음수가 될 수도 있기 때문에, 질적변수로 회귀분석을 하는 것은 적절하지 않다. 물론, 질적변수를 이용해서 회귀분석을 할 수 있다. Binary변수일 경우에는 0과 1을 놓고, 0.5를 기준으로 0 또는 1 중 어디에 더 가까운 지 '해석'할 수 있다. 물론 이때에도 확률값'처럼' 나온다는 것이지, 그것이 곧 확률이라고 보기는 어렵다. X의 계수가 [0,1]의 범위에 없다면, 더더욱 해석은 어려워진다.

4.3.1 The Logistic Model

p(X)의 함수를 선형회귀와 같은 식으로 한다면, X의 값에 따라, 0과 1사이에 없는 상황을 초래할 수 있다. X의 값이 크면, 확률값이 1을 초과할 수 있고, X의 값이 지나치게 작으면, 확률값이 음수가 나올 수 있다는 것이다. 그러나 확률의 값은 항상 0이상 1이하이기 때문에, 우리는 X의 값에 어떤 값이 들어가더라도 항상 그 output은 0과 1사이에 있는 값이 나오도록 하는 함수를 써야한다. 로지스틱 회귀에서는 이 함수를 로지스틱 함수(logistic function)이라고 한다.

$$p(X) = \frac{e^{\beta_0+\beta_1 X}}{1+e^{\beta_0+\beta_1 X}}.$$

이 로지스틱 함수를 조금 변형하면, 다음과 같다.

$$\frac{p(X)}{1-p(X)} = e^{\beta_0+\beta_1 X}.$$

이를 odds라고 한다. 0 < p(X) < 1이기 때문에, 0 < odds ratio < ∞ 이다. 즉, 오즈비의 의미는 쉽게 말하면, '성공할 확률이 실패할 확률보다 몇 배 더 높은가'이다. 위의 두번째 식에다가 log를 씌우면, 다음과 같다.

$$\log\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + \beta_1 X.$$

좌변을 log-odds 또는 logit이라고 부른다. 우리는 위의 세번째 식에서 'logistic regression 모델은 X에 대하여 logit이 선형관계에 있다는 것을 알 수 있다.

선형회귀에서는 X의 단위변화(one unit change)는 β_1만큼 Y를 변화시켰다. 하지만, 로지스틱 회귀에서의 X의 단위변화는 logit을 β_1만큼 변화시킨다. 이는, odds를 $e^{\beta 1}$만큼 변화시킨다. 그런데, 결국 X와 p(X)의 관계는 직선의 관계가 아니기 때문에, p(X)의 변화는 X에 달려있다. 다만, 상관계수(Coefficient)는 양과 음의 방향성을 나타낸다는 점에서 의미가 없지는 않다.

4.3.2 Estimating the Regression Coefficients

선형회귀분석에서 상관계수를 추정할 때, 우리는 최소제곱법(least squares)를 썼다. 로지스틱 회귀분석에서는, 그것 대신에, 최대우도측정(maximum likelihood estimation)이라는 방법을 사용한다. 우리는 상관계수인 β_0 와 β_1를 추정하는데, 이 두개는 likelihood function을 최대화하는 값이라고 보면된다. likelihood function은 아래와 같지만, 이것에 대한 수학적인 설명은 이 책의 논의를 벗어난다.

$$\ell(\beta_0, \beta_1) = \prod_{i:y_i=1} p(x_i) \prod_{i':y_{i'}=0} (1-p(x_{i'})).$$

이 함수를 적용시켜서 상관계수를 얻는 과정은 R과 같은 프로그램에서 자동적으

로 해주기 때문에, 구체적으로 알 필요는 없다.

	Coefficient	Std. error	Z-statistic	P-value
Intercept	−10.6513	0.3612	−29.5	<0.0001
balance	0.0055	0.0002	24.9	<0.0001

4.3.3 Making Predictions

$$\hat{p}(X) = \frac{e^{\hat{\beta}_0+\hat{\beta}_1 X}}{1+e^{\hat{\beta}_0+\hat{\beta}_1 X}} = \frac{e^{-10.6513+0.0055\times 1{,}000}}{1+e^{-10.6513+0.0055\times 1{,}000}} = 0.00576,$$

이렇게 상관계수를 추정하고 나면, logistic function에 대입해주면 된다. 그리고 X에 값을 입력하게 되면 확률값이 나오게 된다. 예를 들어 balance가 1000달러인 사람은 default일 확률이 0.576%이다. 2000달러인 사람은 58.6%이게 된다.

4.3.4 Multiple Logistic Regression

$$p(X) = \frac{e^{\beta_0+\beta_1 X_1+\cdots+\beta_p X_p}}{1+e^{\beta_0+\beta_1 X_1+\cdots+\beta_p X_p}}.$$

$$\log\left(\frac{p(X)}{1-p(X)}\right) = \beta_0+\beta_1 X_1+\cdots+\beta_p X_p,$$

변수와 상관계수를 더 놓고, 상관계수를 추정한 뒤에 확률을 예측하면 된다.

4.4 Linear Discriminant Analysis

로지스틱 회귀와는 달리, 우리는 좀 더 간접적인 classification모델인 LDA를 사용한다.

그렇다면, 왜 이런걸 쓰는가?

1. 클래스가 잘 나누어져 있을 때, 로지스틱 회귀에서의 상관계수 추정치는 매우 불안정하다. 그러나 LDA는 그러한 문제를 겪지 않는다. (예를 들어 class가 극단적으로 잘 나누어져 있다고 할 때, logistic function의 베타값을 추정하는 것은 매우 어렵다, 즉 variance가 매우 크다. 이를 상관계수 추정치가 매우 불안정하다고 하는 것이다.)

2. n이 작고, X의 분포가 각각의 클래스에 대해 normal에 가깝다면, LDA는 로지스틱 회귀보다 더 안정한 모델이 된다.

3. 마지막으로는, 앞서 언급했듯이, LDA가 logistic regression보다 좀 더 대중적이고 대표적인 모델이다.(2개 이상의 반응 클래스(response classes)가 있을 시에...)

4.4.1 Using Bayes' Theorem for classification

우리는 우리의 관측치들을 k개의 클래스들로 나누려고 한다. 이때, k는 2개이상이다. π_k를 사전확률(prior probability)라고 하자. $f_k(x)$는 Pr(X=x | Y=k)라고 하자. 이는 k번째 클래스일 때의 X에 대한 밀도함수이다(density function). 이를 Bayes Theorem으로 정리하면, 아래와 같다

$$\Pr(Y = k|X = x) = \frac{\pi_k f_k(x)}{\sum_{l=1}^{K} \pi_l f_l(x)}.$$

우리가 위의 식에서 π_k 를 추정하는 것은 쉽다. 왜냐하면 k번째 클래스의 일부를 살펴보면 되기 때문이다. 하지만, $f_k(x)$는 추정하기 어려운데, 그 이유는 우리가 밀도에 대해서 단순한 형태로 가정하기 어렵기 때문이다. 따라서 $f_k(x)$를 최대한 정확하게 추정할수록, 오차율이 가장 적다고 알려진 Bayes Classifier에 가까워진다.

4.4.2 Linear Discriminant Analysis for p =1 (Only 1 predictor)

우리는 $f_k(x)$를 추정하고자 한다. 그리고 이를 통해 Pr(Y = k | X= x) (이하 $p_k(x)$)를 예측하려고한다. 이때, 두 개의 가정이 필요하다.

1. Pr(X=x | Y=k)가 정규분포를 따른다고 가정한다.(normal, Gaussian)

$$f_k(x) = \frac{1}{\sqrt{2\pi}\sigma_k} \exp\left(-\frac{1}{2\sigma_k^2}(x - \mu_k)^2\right),$$

2. k classes들에 공통적인 분산이 있다고 가정한다.(단순화하기 위해서) 그리고 위의 식을 $p_k(x)$에 대입하면, 아래와 같다.

$$p_k(x) = \frac{\pi_k \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2\sigma^2}(x - \mu_k)^2\right)}{\sum_{l=1}^{K} \pi_l \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2\sigma^2}(x - \mu_l)^2\right)}.$$

그리고 classifier는 $p_k(x)$가 가장 큰 k번째 class에 x를 assign한다. 위의 식에 log를

취하면, 아래와 같다.

$$\delta_k(x) = x \cdot \frac{\mu_k}{\sigma^2} - \frac{\mu_k^2}{2\sigma^2} + \log(\pi_k)$$

그러나 실제 상황에서 우리는 파라미터들을 알 수 없다.(평균, 파이값...들의 모수) 따라서 추정을 해야한다. 그래서 평균과 분산값에 대한 표본값을 구한다.

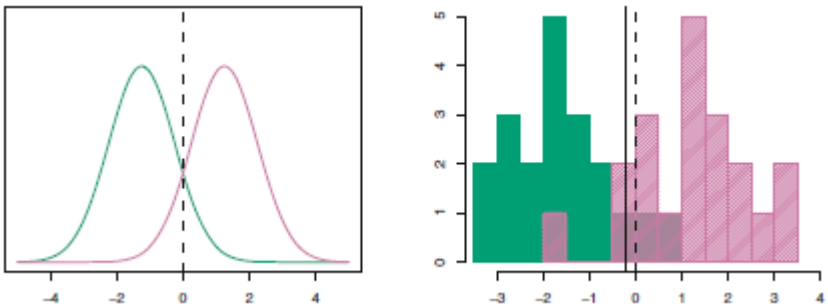
$$\begin{aligned}
\hat{\mu}_k &= \frac{1}{n_k} \sum_{i:y_i=k} x_i \\
\hat{\sigma}^2 &= \frac{1}{n-K} \sum_{k=1}^{K} \sum_{i:y_i=k} (x_i - \hat{\mu}_k)^2
\end{aligned}$$

$$\hat{\pi}_k = n_k/n.$$

표본평균은 표본에서의 x값의 평균으로 구한다. 표본분산은 표본에서의 편차의 제곱의 합을 평균을 낸다. (여기서 n-K는 자유도이다.) 그리고 파이값의 표본값은 k번째 클래스의 표본의 크기를 total number로 나누어준다. 그리고 이 값들을 다시 대입해서 풀면

$$\hat{\delta}_k(x) = x \cdot \frac{\hat{\mu}_k}{\hat{\sigma}^2} - \frac{\hat{\mu}_k^2}{2\hat{\sigma}^2} + \log(\hat{\pi}_k)$$

이렇게 되고 이 값이 가장 큰 클래스에 assign하게 된다. 즉 위에서 말한 것은 Bayes classifier이고 밑에 것이 LDA이다.



왼쪽에 있는 것이 Bayes classifier로 분류한 것이고, 오른쪽에 있는 것이 LDA로 분류한 것이다. 오른쪽에 있는 것만 보자면, 두개의 클래스 각각 표본의 크기는 20이다. 즉 둘의 파이값은 동일하다. 그리고 우리는 위에서 언급한 식대로 표본평균, 표본분

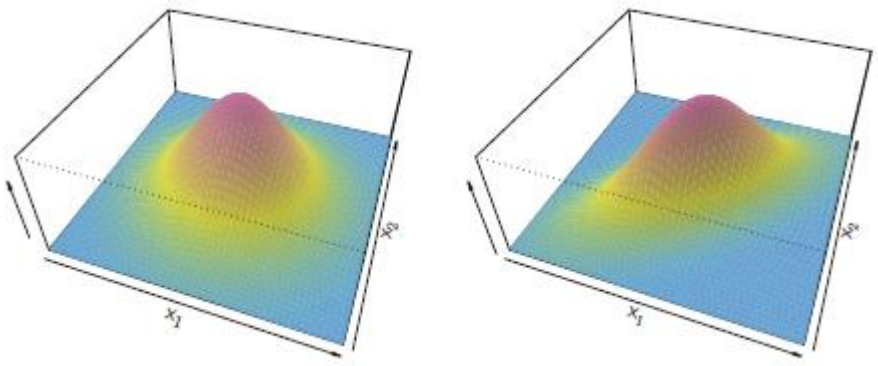
산을 계산한다. 결과적으로 파이값이 둘이 동일하므로 경계는 두 클래스의 평균의 중간값이 된다. 얼마나 잘 예측했는지 Bayes classifier와 LDA의 error rate를 계산해봤다. 각각 10.6%, 11.1%였다. LDA가 꽤 괜찮게 작동한다는 의미이다!!

정리하자면, 우리는 x의 밀도함수가 정규분포를 따르며, 각각의 클래스는 공통의 분산을 갖는다고 가정하고, 모수의 추정치인 표본값들을 Bayes classifier에 대입해서 p값이 가장 큰 클래스에 관측치를 assign했다.

4.4.3 Linear Discriminant Analysis for p > 1

이번에는 각각의 클래스에 대한 predictor가 2개 이상인 경우를 다루려고 한다. 이때는 아까와는 다른 가정이 필요하다. x들의 밀도함수는 multivariate Gaussian distribution을 따르고 클래스에는 각각의 평균값이 존재하며, 모든 클래스에는 공통의 공분산(Covariance)을 가진다.

이때, multivariate Gaussian distribution에 대해서 설명하자면, 각각의 predictor들은 1차원의 정규분포를 다르고 predictor들 간에는 상관관계가 있을 수 있다. 즉, 2개의 정규분포를 3차원으로 나타낸 것이다.



위의 왼쪽의 그림은 x1과 x2의 분산이 같고 공분산이 0인 경우이다. 즉, 상관관계가 없는 경우이다. 오른쪽의 경우에는 x1과 x2의 분산이 같지 않고, 공분산이 0이 아닌 경우이다. 우뚝 솟은 모양의 단면이 상관관계가 있으면 타원에 가깝고 없으면 원에 가깝다. 그리고 표면의 높이는 x1과 x2가 그 부분에 있을 확률이다.

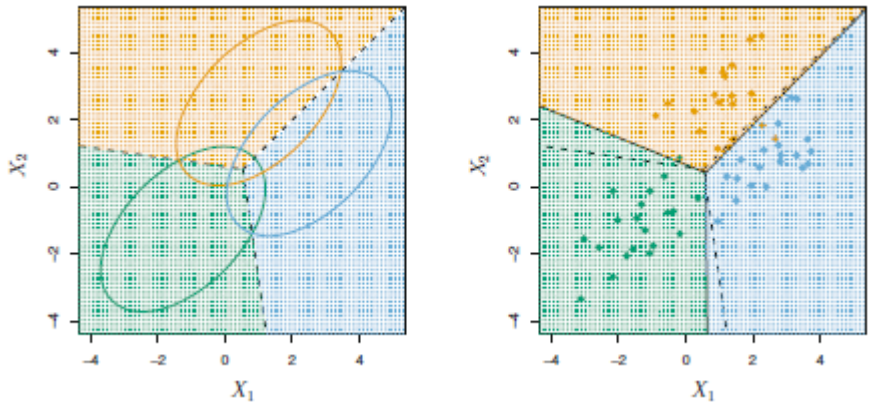
이제, 위의 가정들을 수식으로 표현한다.

$$X \sim N(\mu, \mathbf{\Sigma}).$$

$$f(x) = \frac{1}{(2\pi)^{p/2}|\mathbf{\Sigma}|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu)^T\mathbf{\Sigma}^{-1}(x-\mu)\right)$$

$$\delta_k(x) = x^T\mathbf{\Sigma}^{-1}\mu_k - \frac{1}{2}\mu_k^T\mathbf{\Sigma}^{-1}\mu_k + \log\pi_k$$

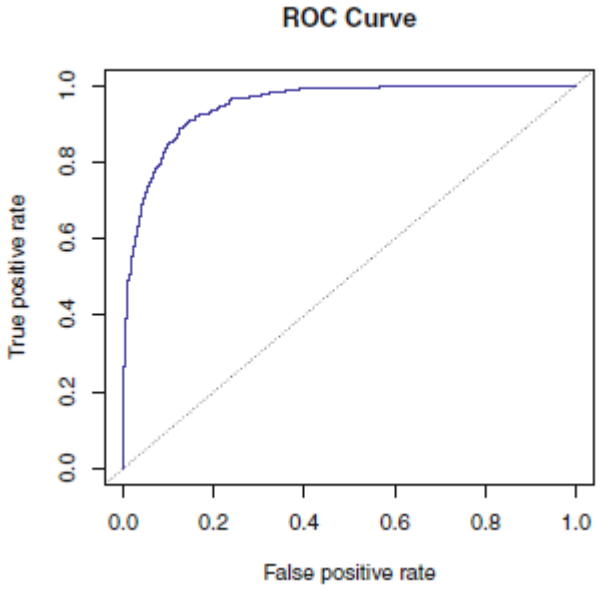
가정과 같이 X는 평균이 μ이고 공분산이 sigma인 multivariate Gaussian분포를 따른다. 이러한 X에 대한 밀도함수가 f(x)이고 이에 log변환을 한 것이 δk(x)이다. 이 값이 큰 클래스에 값을 assign하게된다. 3개의 클래스인 경우를 예로 살펴보자.



그림에서의 점선은 Bayes decision boundary이다. 클래스 1과2, 2와3, 3과1 사이의 경계이다.

우리는 여기에서도 앞서와 마찬가지로 평균값, 공분산의 표본값을 추정한다. 그리고 Bayes classifier와의 error rate을 비교한다.

이러한 classifier의 성능을 측정하는 것이 있다. 바로 ROC Curve이다.



가로축의 False positive rate은 맞는 것을 틀리다고 할 확률으로, 1종오류에 해당한다. 세로축의 True positive rate은 맞는 것을 맞다고 할 확률로 (1-2종오류;검정력)이다. 2종오류가 True negative이기 때문이다.(틀린 것을 맞다고 할 확률) 따라서 이러한 ROC Curve가 좌상향일수록 classifier의 성능이 좋은 것이고 곡선 아랫부분의 넓이인 AUC가 1에 가까울수록 성능이 좋다고 할 수 있다.

4.4.4 Quadratic Discriminant Analysis

LDA에서의 가정을 보면, 각각의 클래스에서의 predictor들은 multivariate Gaussian distribution을 따르고 공통의 공분산을 가진다. 하지만 QDA에서는 모든 클래스들이 공통의 공분산을 가지는게 아니라, 각각의 클래스별로의 공분산을 가진다.

따라서 X는 다음을 따르고

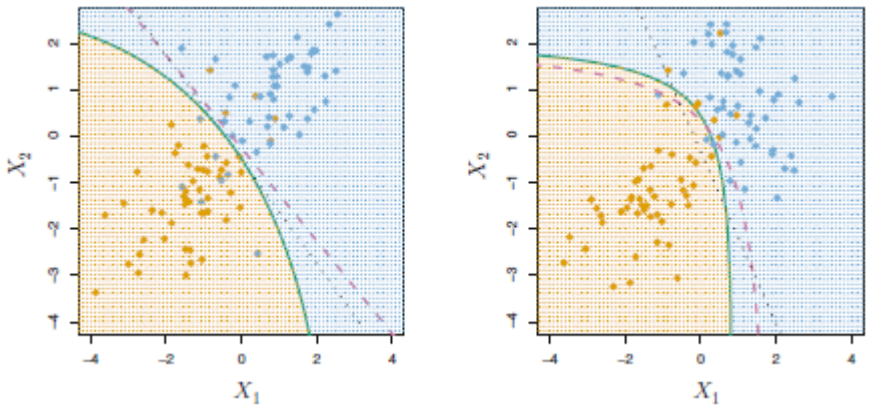
$$X \sim N(\mu_k, \Sigma_k)$$

Bayes classifier는 다음의 값이 큰 값에 observation을 assign한다.

$$\begin{aligned}\delta_k(x) &= -\frac{1}{2}(x-\mu_k)^T\Sigma_k^{-1}(x-\mu_k) - \frac{1}{2}\log|\Sigma_k| + \log\pi_k \\ &= -\frac{1}{2}x^T\Sigma_k^{-1}x + x^T\Sigma_k^{-1}\mu_k - \frac{1}{2}\mu_k^T\Sigma_k^{-1}\mu_k - \frac{1}{2}\log|\Sigma_k| + \log\pi_k\end{aligned}$$

앞서 LDA와는 달리 QDA는 2차함수(quadratic function)이다. 따라서 QDA라고 한다. 그렇다면 어떨 때 LDA와 QDA를 쓰는가. 결론부터 이야기하자면, LDA는 bias가 크고 variance는 작고, QDA는 bias가 작고 variance는 크다. 따라서 데이터의 양과 bias-

variance의 trade-off를 고려해서 잘 선택해야한다. 또한 QDA는 공통의 공분산이 쓰이기 어려울 때 쓴다.



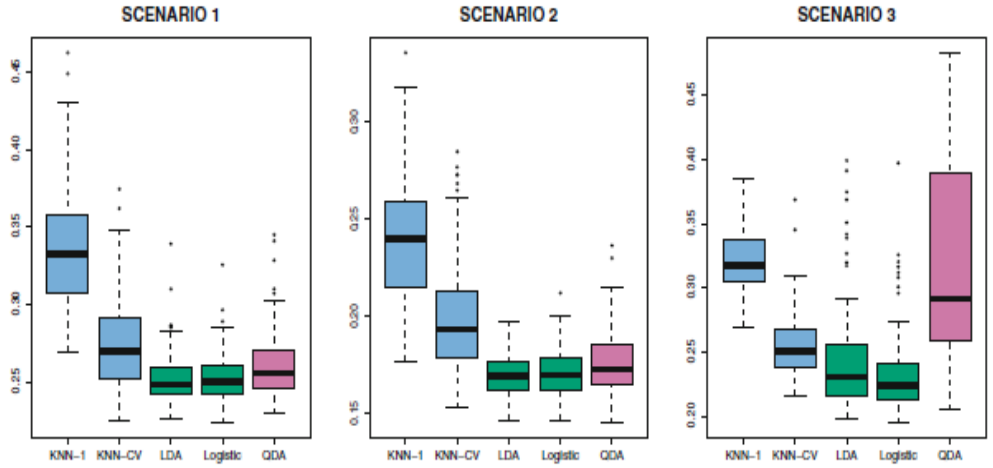
위의 그림에서 살펴보자면, 보라색선은 Bayes이고 검정색 점선은 LDA이다. 초록색선은 QDA이다. 왼쪽에서는 클래스 1과 2의 공분산이 동일한 경우다. 이 경우에는 Bayes decision boundary가 선형이기 때문에, LDA가 더 우수하다. 오른쪽의 경우에는 클래스들의 공분산이 다른 경우이다. 이때는 Bayes decision boundary가 비선형이기 때문에 QDA가 더 우수하다.

4.5 A Comparison of Classification Methods

이번에는 logistic regression, LDA, QDA, 그리고 KNN에 대해서 비교해보려고 한다. 우선 logistic regression과 LDA는 x에 대해서 선형이라는 점에서 공통적이다. 그러나 차이점이 있다면, parameter를 추정하는 방법이다. logistic regression은 maximum likelihood라는 방식을 이용하는 반면에, LDA는 평균값, 분산값 등을 추정한다. 두 개의 방법이 비슷해서 결과가 비슷하게 나오는 경우도 있지만, 꼭 그렇지는 않다. 정규분포를 만족시킬 때에는 LDA가 좋지만, 그렇지 않을 때에는 logistic regression을 쓰는 것이 더 좋다.

한편, KNN은 비모수적인 방법에 속한다(non-parametric). 즉 decision boundary의 모양에 대한 가정이 없다. 따라서, decision boundary가 매우 비선형적(highly non-linear)일 때에는 KNN이 좋다.

QDA는 이 세 개의 중간점이라고 생각하면 된다. 이차함수의 경계선을 가정하기 때문에 선형인 LDA나 logistic regression보다 더 넓은 범위의 문제를 다룰 수 있고, KNN보다는 덜 유연하지만 제한된 수의 training observation에서는 경계의 모양에 대한 가정이 있기 때문에 더 우수하다.

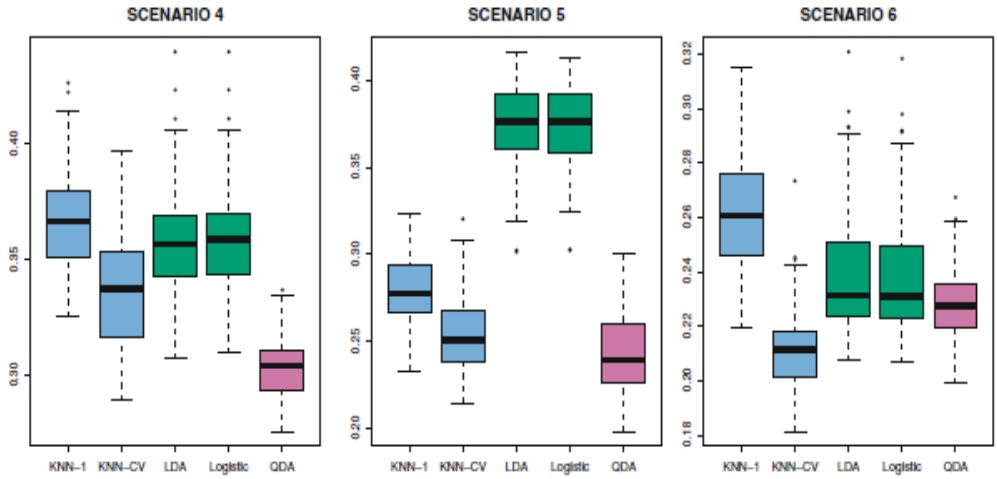


위의 그림은 classifier들을 각 시나리오별로 성능을 측정하기 위한 것이다. 세로축은 error rate을 나타낸다. 이 세 개는 Bayes decision classifier가 linear한 경우이다.

첫번째 시나리오: 20개의 관측치, 2개의 predictor, 그리고 각각의 클래스는 uncorrelated하다. 각각의 클래스의 predictor들은 정규분포를 따른다. 그 결과 역시 LDA와 logistic regression의 error rate이 가장 작다. 경계선이 선형이기 때문이다.

두번째 시나리오: 가정들은 첫번째와 같고, 다만, 각각의 클래스의 predictor들 간의 상관계수는 –0.5이다. 별 차이는 없다.

세번째 시나리오: 이번에는 각각의 predictor을 정규분포가 아닌 t분포를 따른다고 했다. 이렇게 되면, LDA와 QDA는 가정을 만족하지 못하기 때문에, error rate이 커지고, logistic regression이 가장 낮은 error rate을 기록한다.



다음으로 위의 세 개는 bayes decision boundary가 non-linear한 경우이다.

네번째 시나리오: 데이터들은 정규분포를 따르며 각각의 클래스는 서로 다른 correlation을 갖는다. bayes decision boundary가 non-linear하고 위의 조건들이 QDA

에 부합하므로 QDA가 가장 우수한 성능을 보인다.

다섯번째 시나리오: 이 경우에도 위의 경우와 조건이 같다. 다만 반응변수가 logistic function에서 추출되었다.(predictor = X_1^2, X_2^2, X_1 x X_2)그래서 QDA의 성능은 더욱더 좋아졌다.

여섯번째 시나리오: 조건은 위와 동일한데, 이번에는 반응변수가 더 복잡한 비선형적인 함수에서 추출되었다. 이 경우에는 QDA보다는 KNN에서 더 우수한 성능을 보인다. 그런데, 이때 KNN-1은 KNN-CV와는 달리 가장 높은 에러율을 보였다. (이 때, KNN-1은 K=1일 경우인데, 이 경우에는 bias가 매우 작은 대신에 variance가 매우 크다. KNN-CV는 뒤에서 배울 Cross-Validation을 통해서 K값을 정한 경우이다.) 이를 통해, KNN이 유리한 경우라도, K값을 잘 정하는 것이 중요하다는 점을 알 수 있다.