Contents

discriminant analysis

- Discriminant analysis는 class의 구분에 적합하도록 feature들의 선형결합으로 생성한 latent variable을 이용하여 dimension reduction이나 classification에 활용하는 모형이다.
- Discriminant function은 feature들을 latent variable에 대응시키는 선형함수이다.

- 이후 소개할 모형 중 Fisher’s linear discriminant은 projection으로 feature의 차원을 축소시키는 방법이고, linear discriminant analysis와 quadratic discriminant analysis는 feature의 분포를 이용해 class를 분류한다.
- 이 모형들은 아주 다른 접근방법을 사용하고 있지만 여러 가정으로 인해 최종 '분류기'는 동일한 형태가 된다.

- Discriminative, generative를 막론하고 지금까지 소개한 대부분의 classification 모형은 선형 decision boudary를 갖는다.
- 아무런 제약이 없는 tree모형이나, 조건부 공분산행렬이 서로 다르고 독립적이 아닌 Gaussian NB, 그리고 자료에 비선형변환을 적용한 SVM 정도가 예외이다.

- Discriminant 모형과 generative 모형들의 접근방식은 완전히 다르지만 대부분 평균과 분산에 대한 정보만을 (독립적으로) 이용하여 모형을 구성하므로 분포에 대한 가정에 독립성이 추가되면 최종적으로 feature의 선형결합(linear combination, weighted sum of features)에 비선형 변환을 적용하여 확률이나 class를 예측하는데 사용한다.

- 평균과 분산(이차적률)이 독립적으로 결정되는 분포는 정규분포와 double exponential distribution 정도이므로 MSE나 MAE의 극소화문제로 환원되는 경우가 많다.
- 따라서 정규분포를 명시적으로 가정하지 않더라도 표본으로 계산한 평균과 분산에 대한 정보만 이용하는 모형들은 외견상 정규분포를 가정한 모형과 크게 차별화되지는 않는다.

Contents

1 Linear Classification

- 분류모형의 대부분은 decision boundary가 선형이지만 feature의 분포에 대한 가정과 구조에는 상당한 차이가 있다. 간단한 비교와 예제는 여기 (http://www.stat.cmu.edu/~larry/=sml/linearclassification.pdf)를 참고한다.

- **Fisher's linear discriminant**
  - 자료의 분포에 별다른 제약을 가하지 않지만, 평균과 분산에 대한 정보만 이용하는 '선형'모형으로 독립적인 정규분포를 명시적으로 가정하는 모형들과 '동일'하다.
- Logistic regression
  - log-odd가 선형함수이다.
- **Linear Determinant Analysis**
  - Feature들은 정규분포를 따르며 class에 대한 조건부 분산은 동일하다.
- **Support Vector Machine**
  - 분포에 대한 가정을 없으며, 선형으로 분류하긴 하지만 표본공간을 비선형으로 변환하여 사용하기도 한다.

- 다음 모형들은 feature들이 정규분포를 따르며 class에 대한 조건부 분산이 다를 수 있다고 가정한다. 이 경우 decision boundary는 비선형이다.
  - Gaussian Naïve Bayes
  - Quardratic Determinant Analysis

Notation

- 여기서 다루는 모형들을 유도할 때 대부분 vector 연산을 사용하므로 모든 vector를 column vector로 사용한다. 계량경제와 같이 row vector로 사용하는 경우도 많지만 sklearn에서는 column vector를 기본자료 format이다.

- Feature X의 차원은 따라서 $K \times n$으로 사용하며 개별 표본 $x_i \in \mathbb{R}^n$ 은 X의 각 column을 표시한다.

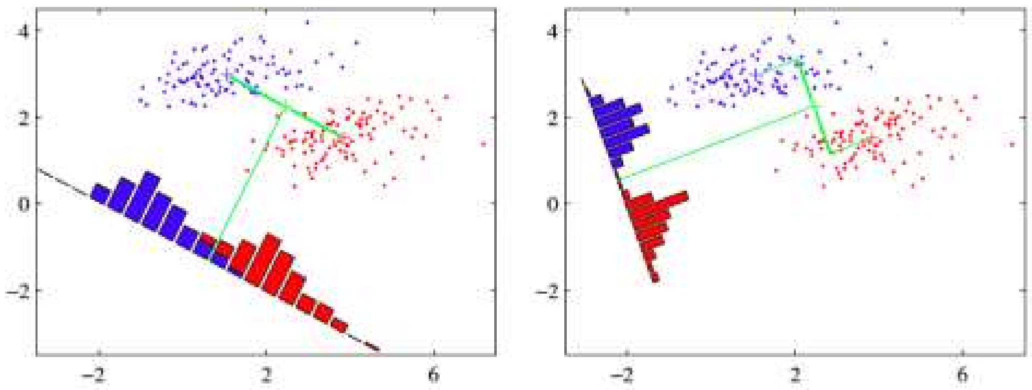
Contents

2 Fisher (linear) discriminant analysis

2.1 개략적인 모형

- 자료를 subspace에 linear mapping하는 방법으로 1936년 Fisher가 dimension reduction을 위해 고안

Figure 1. Fisher linear discriminant



- 각 class를 가장 잘 '분리'하는 새로운 축을 찾아 feature projection으로 새로운 "index" 생성
  두 분포 사이의 거리는 멀게, 각 분포의 분산은 작게
- 표본의 특징을 유지할 수 있어 dimension reduction에 주로 사용
  Feature의 정보를 최대한 유지하면서 다중공선성 문제를 완화

- 원리를 그림으로 설명한 블로그 (https://sthalles.github.io/fisher-linear-discriminant/)

Contents

2.2 Projection vector

- Fisher 모형은 각 class 평균의 차이(평균 사이의 거리)를 극대화하고 개별 class의 분산을 극소하는 projection vector를 찾는 문제이다.

$$\mathbf{w}^\intercal(\mathbf{x} - a_0) = 0$$
$$\mathbf{w}_1 x_1 + \cdots + \mathbf{w}_K x_K = a_0$$

- 최적 초평면의 계수를 각 feature의 가중값으로 사용하여 K차원의 정보를 낮은 차원으로 변환한다. $\mathbf{w}$는 projection/weight vector라고 부른다.
- 자료 $X|y = c$는 평균이 μ_c이고 공분산이 Σ_c이다.

$$\mu_c = \frac{1}{n_c}\sum_{i=1}^{n_c} x_{i|c}, \qquad \mu_c \in \mathbb{R}^K$$

$$\Sigma_c = \frac{1}{n_c}\sum_{i=1}^{n_c}(x_{i|c} - \mu_c)(x_{i|c} - \mu_c)^\intercal, \qquad \Sigma_c \in \mathbb{R}^{K\times K}$$

- 한편 각 유형의 feature $x_{i|c}$의 선형결합인 projection scalar $\mathbf{w}^\intercal x_{i|c}$의 평균과 분산은 각각 $\mathbf{w}^\intercal \mu_c$와 $\mathbf{w}^\intercal \Sigma_c \mathbf{w}$이다.
- Fisher 모형은 각 class의 projection 평균의 차이 $\mathbf{w}^\intercal(\mu_c - \mu_{c'})$를 가장 크게 하면서 projection의 분산 $\mathbf{w}^\intercal \Sigma_c \mathbf{w}$의 합을 극소화하는 projection vector, $\mathbf{w}$를 이용하여 class들의 차이를 가장 잘 보여주는 latent variable을 생성한다.

2.3 제곱합(Total Sum of Squares)의 분할

- Class 수가 2개 이상일 경우 일반적인 scatter matrice 유도

  $$\begin{eqnarray} \sum^n_{i=1} \left(x_i - \mu \right) \left(xi - \mu \right)^\intercal &=& \sum^C{c=1} \sum^{nc}{i=1} \left(x{i|c} - \mu \right)\left(x{i|c} - \mu \right)^\intercal \ &=& \sum^C_{c=1} \sum^{nc}{i=1} \left(x_{i|c} - \mu_c +\muc - \mu \right) \left(x{i|c} - \mu_c +\muc - \mu \right)^\intercal \ &=& \sum^C{c=1} \sum^{nc}{i=1} \left(x_{i|c} - \mucright)\left(x{i|c} - \mu_c\right)^\intercal
- \sum^C_{c=1} \sum^{nc}{i=1} \left(\mu_c - \mu\right)\left(\mu_c - \mu\right)^\intercal
- 2 \sum^C_{c=1} \sum^{nc}{i=1} \left(x_{i|c} - \mu_c\right)\left(\mu_c - \mu\right)^\intercal \end{eqnarray}

$$< br > \$ \sum_{i=1}^{n_c}\left(x_{i|c} - \mu_c\right) = 0\$\text{이므로} < br >$$

\begin{eqnarray} \sum^n_{i=1} \left(x_i - \mu \right)\left(xi - \mu \right)^\intercal &=& \sum^C{c=1} \sum^{nc}{i=1} \left(x_{i|c} - \mucright)\left(x{i|c} - \mu_c\right)^\intercal
- \sum^C_{c=1} \sum^{nc}{i=1} \left(\mu_c - \mu\right)\left(\muc - \mu\right)^\intercal \ &=& \sum^C{c=1} \sum^{nc}{i=1} \left(x_{i|c} - \mucright)\left(x{i|c} - \mu_c\right)^\intercal
- \sum^C_{c=1} n_c \left(\mu_c - \mu\right)\left(\muc - \mu\right)^\intercal \ &=& \underbrace{\sum^C{c=1} n_c \Sigmac}{\text{within-class scatter}}
- \underbrace{\sum^C_{c=1} n_c \left(\mu_c - \mu\right)\left(\muc - \mu\right)^\intercal}{\text{between-class scatter}} \end{eqnarray}$$

Contents

2.4 통계량의 계산

- Fisher 모형의 핵심은 class 평균의 분산(the variance between the classes, between-class scatter matrix)과 각 class의 분산(the variance within the classes, within-class scatter matrix)의 비율을 극대화하는 subspace를 찾는 것으로, projection vector $\mathbf{w}$를 이용하여 feature set을 해당subspace에 대응시킨다.

- S_B, S_W는 scatter of the original samples/feature, $\tilde{S}_{\text{between}}$ $\tilde{S}_{\text{within}}$은 scatter of the projected samples/feature로 구분하며, 각각은 between-class scatter matrix와 within-class scatter matrix로 다시 구분한다.

- 다음은 class가 둘 인 경우이며 상세한 유도과정은 다음 슬라이드 참고

$$
\begin{aligned}
\tilde{S}_{\text{between}} &= n_0\left(\mathbf{w}^\top\mu_0 - \mathbf{w}^\top\mu\right)\left(\mathbf{w}^\top\mu_0 - \mathbf{w}^\top\mu\right)^\top + n_1\left(\mathbf{w}^\top\mu_1 - \mathbf{w}^\top\mu\right)\left(\mathbf{w}^\top\mu_1 - \mathbf{w}^\top\mu\right)^\top \\
&= (n_0 \times n_1)\mathbf{w}^\top(\mu_0 - \mu_1)(\mu_0 - \mu_1)^\top\mathbf{w} \\
&= (n_0 \times n_1)\mathbf{w}^\top S_{\text{B}}\mathbf{w}
\end{aligned}
$$

$$
\tilde{S}_{\text{within}} = \sum_{c=1}^{C} n_c\mathbf{w}^\top\Sigma_c\mathbf{w} = \mathbf{w}^\top\left(\sum_{c=1}^{C} n_c\Sigma_c\right)\mathbf{w} = \mathbf{w}^\top S_{\text{W}}\mathbf{w}
$$

- Binary classification 문제에선 feature들의 class에 대한 조건부평균의 분산인 between-class scatter을 각 feature들의 조건부평균의 차이로 계산할 수 있다.

$$
\hat{S}_{\text{B}} = (\hat{\mu}_0 - \hat{\mu}_1)\,(\hat{\mu}_0 - \hat{\mu}_1)^\top
$$

- Within-class scatter는 평균이 다르고 분산은 동일하다고 가정한 pooled variance로 불편표본분산은 다음과 같다.

$$
\hat{S}_{\text{W}} = \frac{(n_0 - 1)\hat{\Sigma}_0 + (n_1 - 1)\hat{\Sigma}_1}{(n_0 - 1) + (n_1 - 1)}
$$

- 이 값은 pooled variance로 관찰값이 더 많은 class의 분산에 대한 가중값이 더 크다. Unbalanced sample에 취약하다.

- $C = 2$라면 between-class scatter은 간단한 형태로 정리할 수 있다.

$$
\begin{aligned}
\tilde{S}_{\text{between}} &= \alpha(\mathbf{w}^\top\mu_0 - \mathbf{w}^\top\mu)^2 + (1-\alpha)(\mathbf{w}^\top\mu_1 - \mathbf{w}^\top\mu)^2 \\
&= \alpha\mathbf{w}^\top(\mu_0 - \mu)(\mu_0 - \mu)^\top w + (1-\alpha)\mathbf{w}^\top(\mu_1 - \mu)(\mu_1 - \mu)^\top\mathbf{w} \\
&= \alpha(\mathbf{1-\alpha})^{\mathbf{2}}\mathbf{w}^\top(\mu_0 - \mu_1)(\mu_0 - \mu_1)^\top w + \alpha^{\mathbf{2}}(1-\alpha)\mathbf{w}^\top(\mu_1 - \mu_0)(\mu_1 - \mu_0)^\top\mathbf{w} \\
&= \alpha(1-\alpha)\mathbf{w}^\top(\mu_0 - \mu_1)(\mu_0 - \mu_1)^\top\mathbf{w}
\end{aligned}
$$

여기서

$$
\begin{aligned}
\mu_0 - \mu &= \{\mu_0 - \alpha\mu_0 + (1-\alpha)\mu_1\} = (1-\alpha)(\mu_0 - \mu_1) \\
\mu_1 - \mu &= \{\mu_1 - \alpha\mu_0 + (1-\alpha)\mu_1\} = \alpha(\mu_0 - \mu_1)
\end{aligned}
$$

Contents

2.5 최적화 문제

- Projection vector는 다음 최적화문제의 해가 된다.

$$\max_{\mathbf{w}} \frac{\tilde{S}_{\text{between}}}{\tilde{S}_{\text{within}}} = \max_{\mathbf{w}} \frac{\mathbf{w}^\intercal S_{\text{B}} \mathbf{w}}{\mathbf{w}^\intercal S_{\text{W}} \mathbf{w}}$$

- 일계조건

$$\frac{2\left(\mathbf{w}^\intercal S_{\text{W}} \mathbf{w}\right) S_{\text{B}} \mathbf{w} - 2\left(\mathbf{w}^\intercal S_{\text{B}} \mathbf{w}\right) S_{\text{W}} \mathbf{w}}{\left(\mathbf{w}^\intercal S_{\text{w}} \mathbf{w}\right)^2} = 0$$

여기서 $\mathbf{w}^\intercal S \mathbf{w}$ 는 scalar이므로

$$S_{\text{B}} \mathbf{w} - \frac{\mathbf{w}^\intercal S_{\text{B}} \mathbf{w}}{\mathbf{w}^\intercal S_{\text{W}} \mathbf{w}} S_{\text{W}} \mathbf{w} = 0$$

2.6 최적화문제의 해

- $\frac{\mathbf{w}^\intercal S_{\text{B}} \mathbf{w}}{\mathbf{w}^\intercal S_{\text{W}} \mathbf{w}}$ 은 목적함수이다. 이를 λ로 표시하면

$$S_{\text{B}} \mathbf{w} - \lambda S_{\text{W}} \mathbf{w} = 0$$

- S_{w}가 full rank를 갖는다면 일계조건은 다음과 같이 정리할 수 있다. 만일 S_{w}가 full rank를 갖지 않는다면 generalized eigenvalue 문제를 풀어야 한다.

$$S_{\text{W}}^{-1} S_{\text{B}} \mathbf{w} = \lambda \mathbf{w}$$

따라서 위 극대화문제의 해는 $S_{\text{W}}^{-1} S_{\text{B}}$의 eigenvector 중 하나이다. 한편 최적화 문제는 λ를 극대화시키는 것이므로 가장 큰 eigenvalue에 해당하는 eigenvector가 극대화 문제의 해가 된다.

- 재미있는 사실은 두 번째나 세 번째로 '적합한' projection vector 혹은 weight vector를 eigenvector를 이용해 구할 수 있다는 것이다.
- 한편 $S_{\text{W}}^{-1} S_{\text{B}}$는 대칭행렬이므로 eigenvector들은 orthogonal basis이다.

Contents

2.7 최적 weight vector의 계산

- 만일 S_W가 full rank를 갖는다면 weight vector는 eigenvector를 굳이 계산하지 않아도 간단히 찾을 수 있다.

$$S_\mathrm{W}^{-1} S_\mathrm{B} \mathbf{w} = \lambda \mathbf{w}$$

- 위 일계조건에서 $S_\mathrm{B}\mathbf{w}$은 $\mathbf{w}$와는 무관하게 $(\mu_0 - \mu_1)$과 동일한 방향을 갖는 vector임을 보일 수 있다.

$$S_\mathrm{B}\mathbf{v} = (\mu_0 - \mu_1)(\mu_0 - \mu_1)^\intercal \mathbf{v} = (\mu_0 - \mu_1)\overbrace{(\mu_0 - \mu_1)\cdot \mathbf{v}}^{\text{constant}} = c(\mu_0 - \mu_1)$$

이를 일계조건에 대입하면,

$$S_\mathrm{W}^{-1} S_\mathrm{B} \underbrace{S_\mathrm{W}^{-1}(\mu_0 - \mu_1)}_{\mathbf{v}} = S_\mathrm{W}^{-1} c(\mu_0 - \mu_1) = c \times \underbrace{S_\mathrm{W}^{-1}(\mu_0 - \mu_1)}_{\mathbf{v}}$$

- Eigenvector는 방향만 식별이 가능하므로 weight vector는 다음과 같이 계산할 수 있다.

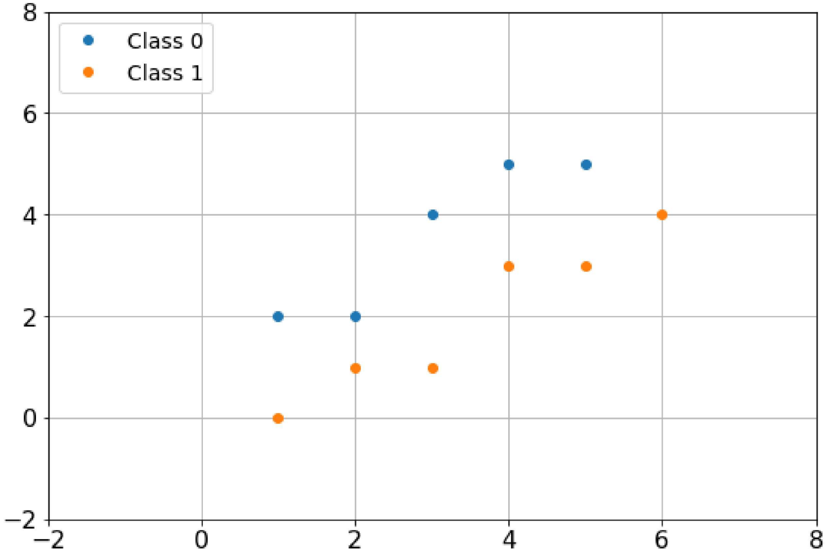
$$\mathbf{w} = S_\mathrm{W}^{-1}(\mu_0 - \mu_1) = (\alpha\Sigma_0 + (1-\alpha)\Sigma_1)^{-1}(\mu_0 - \mu_1)$$
$$\hat{\mathbf{w}} = S_\mathrm{W}^{-1}(\hat{\mu}_0 - \hat{\mu}_1)$$

```
w = np.linalg.inv(s_w).dot(mean_0 - mean_1)
w = w / np.linalg.norm(w)
```

3 예제

Contents

Figure 3-1. Observations



3.1 Fisher discriminant analysis classifier

- Fisher discriminant anlaysis는 dimension reduction에 주로 사용하지만 분류에 사용하기도 한다.
- 분류에 필요한 임계값을 계산하는 일반적인 방법은 없지만 보통은 두 유형의 비율이 비슷하다면 각 분포의 평균에 해당하는 projection의 중간값을 사용한다.

$$\text{threshold} = \frac{\hat{\mathbf{w}}^\intercal(\hat{\mu}_0 + \hat{\mu}_1)}{2}$$

- 최적 Threshold는 class의 비율이나 분산에 대한 가정 등에 영향을 받으므로 분류가 목적이라면 GaussianNB나 아래 소개할 LDA, QDA를 이용한다. 이 모형들은 posterior를 이용하므로 제한적이긴 하지만 언급한 문제들을 명시적으로 고려할 수 있다.

- 앞서 유도한 바와 같이 $\hat{\mathbf{w}} = S_W^{-1}(\hat{\mu}_0 - \hat{\mu}_1)$이므로,

$$\text{threshold} = \frac{\hat{\mathbf{w}}^\intercal(\hat{\mu}_0 - \hat{\mu}_1)}{2} = \frac{(\hat{\mu}_0 - \hat{\mu}_1)^\intercal S_W^{-1}(\hat{\mu}_0 + \hat{\mu}_1)}{2}$$

3.2 Scatter matrices

- MLE나 불편추정량을 이용하여 각 유형별 평균과 분산을 계산하고, 이를 이용하여 S_W와 S_B를 계산한다.

Contents

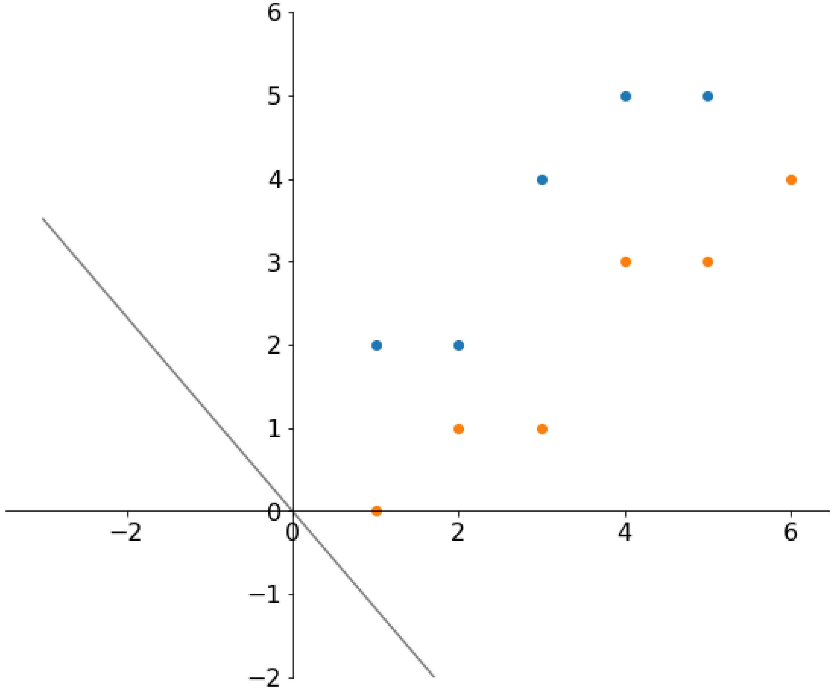

```
variance between classes:
[[ 0.56 -1.78]
 [-1.78  5.69]]
variance within class
[[27.5 23. ]
 [23.  21.2]]
```

3.3 Weight vector

- Eigenvalue 중 가장 큰 값을 찾고 이에 해당하는 eigenvector를 weight vector로 사용한다.
- Scatter는 모두 대칭행렬이므로 모든 eigenvector들은 서로 orthogonal하다.

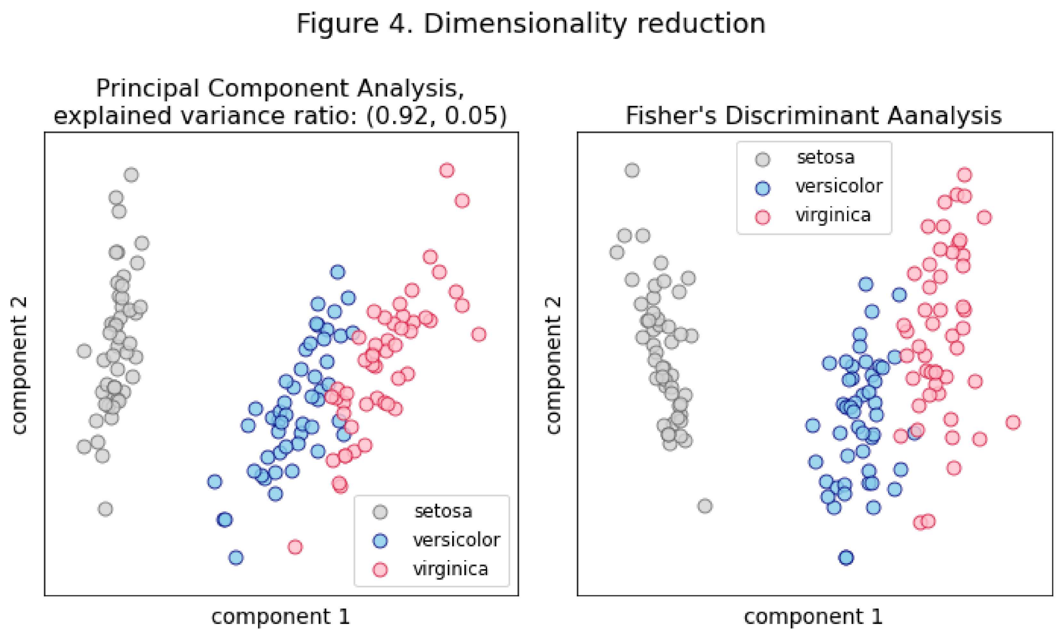
```
eigenvalues:  [4.6315427 0.        ]
eigenvectors
[[ 0.64943634 -0.95447998]
 [-0.76041597 -0.29827499]]
principal weight vector [ 0.64943634 -0.76041597]
secondary weight vector [-0.95447998 -0.29827499]
```

Figure 3-2. Optimal Projection



Contents

4 Dimensionality reduction



과제 1

1. 다음 Titanic 자료를 Training/Test dataset 으로 나누어 training dataset으로 계산한 2개의 가장 의미있는 weight vector를 선택한 후 생존여부를 다른 색으로 표시한 scatter plot을 그린다.

```
titanic = pd.read_csv("https://github.com/k5yi/econ2005/blob/master/datasets/titanic_processed.csv?raw=True")
```

1. FisherDiscriminantAnalysis는 분류목적으로 고안된 것은 아니지만 분류에 사용하기도 한다. 문제 1에서 계산한 1차원 FisherDiscriminantAnalysis를 test dataset에 적용하여 confusion matrix를 만들어 본다.

- '분포'를 가정하는 discriminant 모형에는 보통 정성변수를 사용하지 않는다. Machine learning algorithm 대부분은 가정에 크게 민감하지 않지만, 이러한 이유로 일반적인 자료를 분석할 때는 logistic regression을 더 선호한다.

Contents

5 Titanic dataset

- Fisher's discriminant analysis를 classifier로 적용한 결과와 dimension reduction으로 정리해서 logistic regression의 feature로 사용한 모형을 비교해 보자.
- Fisher's discriminant analysis 를 분류기로 사용하려면 target의 분포를 고려해서 적정 threshold를 구해야 한다. 아래 분석은 참고만 한다.

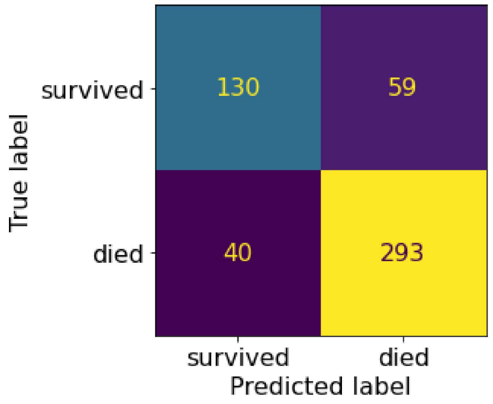
5.1 Logistic regression with the components of Fisher's discriminant analysis

- "age", "fare", "sibsp", "parch" 이상 네 개의 변수를 2차원 공간으로 project한 후 projection을 feature로 사용
- "sibsp"와 "parch"는 categorical로 취급할지 연속변수로 취급할지 선택이 간단하지 않다. 둘 모두 값이 .... 이므로 discriminant analysis에 사용할 수는 있지만 여전히 정규분포와는 거리가 멀다.
- FisherDiscriminantAnalysis는 정규분포를 따르는 변수에만 적용할 수 있다. 하지만 Titanic dataset에서는 `age` 를 제외하고는 정규분포와 비슷하게라도 볼 수 있는 변수가 없다.
- 특히 dummy는 categorical 분포를 따르므로 적용이 곤란하지만 generative 모형들은 적어도 예측 성능은 가정에 크게 영향을 받지 않는 것 같다.

```
correlation coefficient of original features
            age      fare     sibsp     parch
age    1.000000  0.180014 -0.213613 -0.128648
fare   0.180014  1.000000  0.159131  0.220711
sibsp -0.213613  0.159131  1.000000  0.373178
parch -0.128648  0.220711  0.373178  1.000000

correlation coefficient of two projections
              0             1
0  1.000000e+00  5.789219e-18
1  5.789219e-18  1.000000e+00
```

Contents


```
              precision    recall  f1-score   support

    survived       0.76      0.69      0.72       189
        died       0.83      0.88      0.86       333

    accuracy                           0.81       522
   macro avg       0.80      0.78      0.79       522
weighted avg       0.81      0.81      0.81       522
```



5.2 Fisher's discriminant analysis classifier

- 모든 categorical variable을 ordinal variable로 변환하여 추정

Contents


```
threshold: [[1.54+0.j]]

              precision    recall  f1-score   support

    survived       0.69      0.75      0.72       189
        dead       0.85      0.80      0.83       333

    accuracy                           0.79       522
   macro avg       0.77      0.78      0.77       522
weighted avg       0.79      0.79      0.79       522
```

6 Multiclass linear discriminant 모형

$$S_{\mathrm{B}} = \frac{1}{C}\sum_{c=1}^{C}(\mu_c - \mu)(\mu_c - \mu)^{\intercal}$$

$$S_{\mathrm{W}} = \sum_{c=1}^{C}\sum_{i=1}^{n_c}(x_{i|c} - \mu_c)(x_{i|c} - \mu_c)^{\intercal} = \sum_{c=1}^{C} n_c \Sigma_c$$

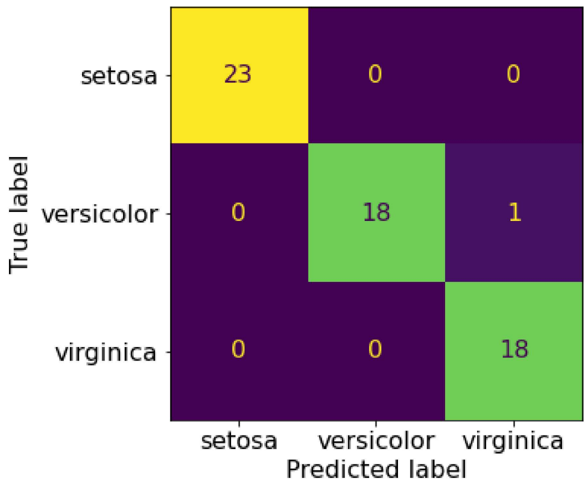
- $C = 2$인 경우와 마찬가지로 $S_{\mathrm{W}}^{-1}S_{\mathrm{B}}$의 eighenvalue 중 큰 값들에 해당하는 eigenvector를 weight vector로 사용한다.

Contents

- decision region을 구하기 위해서는 projection의 순서를 확인해야 한다.
- target label 순서대로 배열이 되어 있으므로 subspace를 3구간으로 나누어 각각에 label을 순서대로 배정한다.

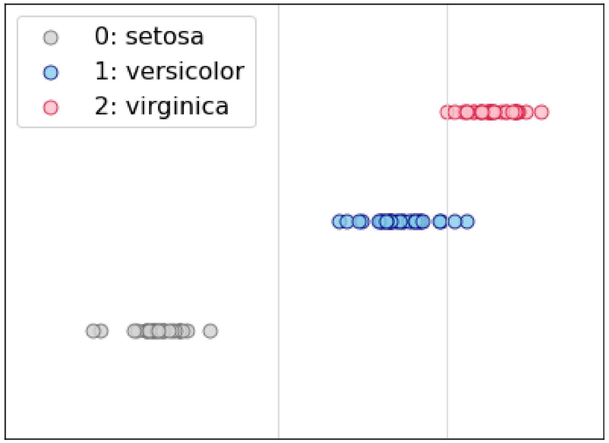
- 앞서 projection 결과를 보면 setosa, versicolor, virginica 순으로 latent varable의 평균이 커지므로 setosa와 versicolor, versicolor와 virginica 사이의 중간값을 기준으로 분리한다.
- 세 품종의 비율이 동일하므로 threshold는 각 class의 feature들의 중심의 가운데 값을 사용해도 별다른 문제가 없다.

```
best threshold: [-0.2508099950842616, 1.427604010472]

              precision    recall  f1-score   support

      setosa       1.00      1.00      1.00        23
  versicolor       1.00      0.95      0.97        19
   virginica       0.95      1.00      0.97        18

    accuracy                           0.98        60
   macro avg       0.98      0.98      0.98        60
weighted avg       0.98      0.98      0.98        60
```



Contents

Figure 5. FDA projection of iris training features



7 Linear Discriminant Analysis (LDA)

- LDA는 Fisher의 linear discriminant와 결과는 동일하지만 접근방식은 상당히 다르다.
  - LDA는 feature들의 결합확률분포에 근거하여 class를 구분하므로 최적화문제를 풀 필요없이 몇가지 통계량 계산으로 충분하다.
  - Fisher's LDA를 class를 가장 잘 분리하는 projection vector를 구하는 것으로 dimension reduction이 주목적이다.
  - [각 모형의 특징과 관련 모형을 잘 비교해 놓은 노트 (https://arxiv.org/pdf/1906.02590.pdf)](https://arxiv.org/pdf/1906.02590.pdf)
- LDA는 posterior를 분류에 사용하는 generative 모형으로 기본가정이 조금 다를뿐, 유도과정이나 특성은 Gaussian naive Bayes 모형과 더 유사하다.
- LDA의 기본가정
  - 각 유형별 feature의 분포는 **동일한** 공분산 행렬을 갖는 **정규분포**
  - Pooled variance-covariance 를 사용하므로 balanced sample 여부에 따라 performance의 차이가 큰 것으로 알려져 있다.

Contents

7.1 Posterior

- $\Pr(X|Y=c)$ 는 다변수정규분포를 따르며 밀도함수는 $f_c(x)$이다.

$$f_c(x) = f(X=x|Y=c) = \frac{1}{(2\pi)^{K/2}|\Sigma_c|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu_c)^\intercal \Sigma_c^{-1}(x-\mu_c)\right)$$

- Bayes rule로 계산한 posterior는 다음과 같다.

$$\Pr(Y=c|X) = \frac{\Pr(Y=c)\Pr(X|Y=c)}{\Pr(X)} = \frac{\pi_c f_c(x)}{\sum_{c=1}^{C} \pi_c f_c(x)}$$

- 로그를 취하고 공분산이 동일하다는 조건을 사용하면 posterior는 다음과 같이 정리할 수 있다.

$$\ln \Pr(Y=c|X) \propto \ln \pi_c - \frac{1}{2}(x-\mu_c)^\intercal \Sigma^{-1}(x-\mu_c)$$

- $(x-\mu_c)^\intercal \Sigma^{-1}(x-\mu_c)$은 Mahalanobis distance로 X의 공분산행렬을 단위행렬로 만들어 계산한 각 점과 분포의 중심인 평균과의 Euclidean distance 이다.

$$\left(\Sigma^{-1/2}(x-\mu_c)\right)^\intercal \left(\Sigma^{-1/2}(x-\mu_c)\right)$$

- LDA는 각 class의 비율을 감안하여 각 관찰값을 거리가 가장 가까운 분포에 해당하는 class로 분류한다.

Contents

7.2 선형분류기

- 분류기는 다음의 기준을 따른다.

$$\Pr(y=1|x) \geq \Pr(y=0|x)$$

- Gaussian Naive Bayes 모형에서 보았듯이 분산이 동일하면 decision boundary는 선형이다.

$$\begin{aligned}\ln \frac{\Pr(Y=1|X)}{\Pr(Y=0|X)} &= \ln \frac{\pi_1 f_1(x)}{\pi_0 f_0(x)} \\ &= \ln \frac{\pi_1}{\pi_0} + \left(-\frac{1}{2}(x-\mu_1)^\intercal \Sigma^{-1}(x-\mu_1)\right) - \left(-\frac{1}{2}(x-\mu_0)^\intercal \Sigma^{-1}(x-\mu_0)\right) \\ &= \ln \frac{\pi_1}{\pi_0} + \frac{(\mu_1-\mu_0)^\intercal \Sigma^{-1} x - (\mu_1-\mu_0)^\intercal \Sigma^{-1}(\mu_1-\mu_0)}{2}\end{aligned}$$

- $\Sigma_c = \sigma^2 I_K$ 이라면, decision boundary는 다음과 같다.

$$\Pr(Y=1|x) \geq \Pr(Y=0|x) \Leftrightarrow \ln \frac{\pi_1}{\pi_0} + \frac{(\mu_1-\mu_0)}{2\sigma^2} \cdot x - \frac{(\mu_1-\mu_0)^\intercal(\mu_1-\mu_0)}{2\sigma^2} \geq 0$$

8 Implemenation

```
sklearn.discriminant_analysis.LinearDiscriminantAnalysis(solver='svd', shrinkage=None, priors=None, n_components=None, store_covariance=False, tol=0.0001, covariance_estimator=None)
```

- n_components: int, default=None
Number of components (<= min(n_classes - 1, n_features)) for dimensionality reduction. If None, will be set to min(n_classes - 1, n_features). This parameter only affects the transform method.

- `predict()` method로 예측에 사용하거나 `transform()` method로 dimension reduction에 사용할 수 있다.

Contents

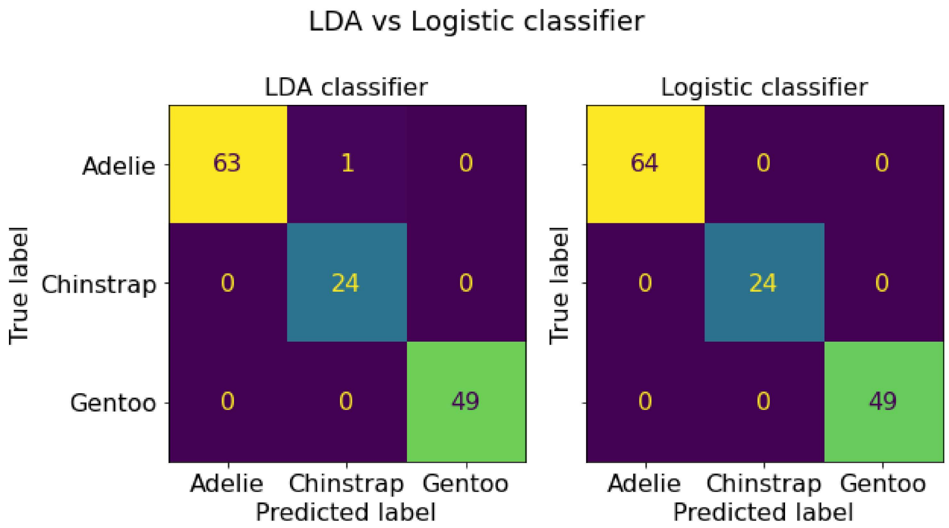
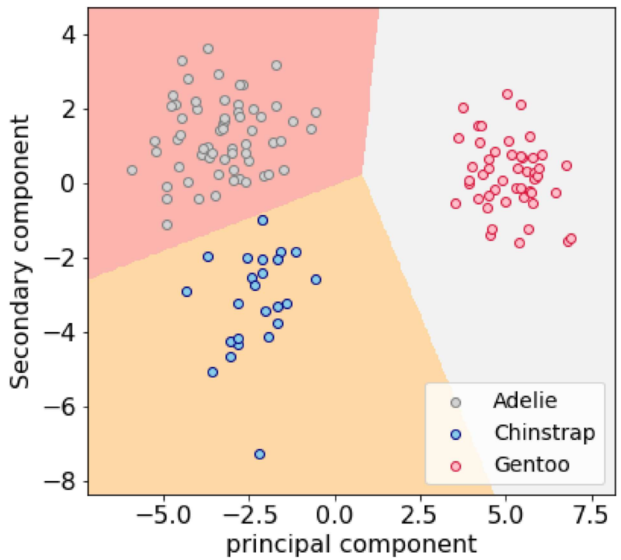


Figure 6. Decision Region of LDA for Penguin dataset



Contents

LDA와 Logistic regression

	LDA	Logistic regression
implementation	OLS based	MLE based
Number of classes	not restrictive	strong assumptions are required
Feature variables	continuous	no restriction
Probability	discriminant	estimate probability
Confidence interval	intervals for predictors	no distribution for predictors
Covariance	identical within group covariance matrix (to add them)	no restriction
Group size	similar number of observations in each group	not sensitive to group sizes
Outliers	sensitive	not so sensitive
Efficiency	if all the requirements are met, it is known about 50% more efficient	robust
	With a large sample size, LDA is superior	
	performs better with samples with high dimensional but a small sample size	

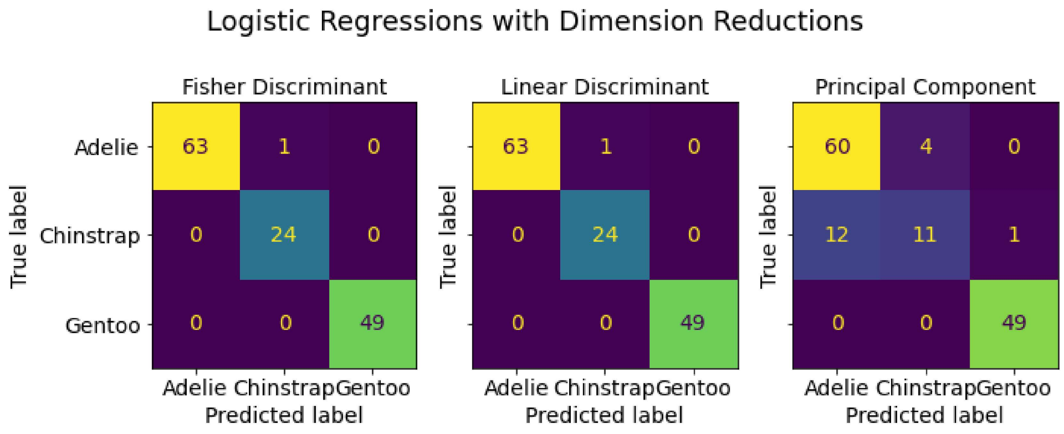
9 LDA Dimension Reduction

- Linear determinant analysis 역시 dimension reduction 용도로 사용할 수 있다.
- Whitening/sphering transformation 후 Euclidean distance를 계산하므로 $\mathbf{x}$를 해당 subspace에 (최대 차원은 K) 대응시켜 거리를 구하는 것과 같은 결과를 얻는다.
- 한편 $\mathbf{x}$를 LDA의 조건을 만족하는 subspace로 projection 하면 더 낮은 차원으로 (최대 class의 수) 축소할 수 있다.

- LinearDiscriminantAnalysis 의 `n_components` keyword는 해당 subspace의 차원을 결정한다.
  n_components $\leq$ min(n_classes - 1, n_features)

- `transform` 이나 `transform_fit` method를 사용하면 가장 효과적으로 class를 분리할 수 있는 project data로 변환할 수 있다.

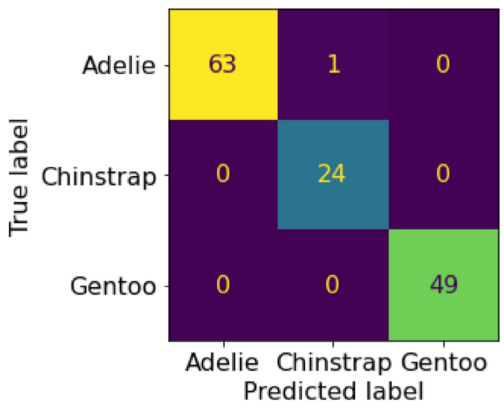
Contents



10 Quadratic Discriminant Analysis

- 공분산행렬이 주대각행렬이라고 가정하면 모든 설명변수들은 조건부독립으로 가정하게 되며 이는 Gaussian Naive Bayes와 동일하다.
- 공분산행렬에 특별한 제약을 가하지 않으면 decision boundary는 비선형이 된다.

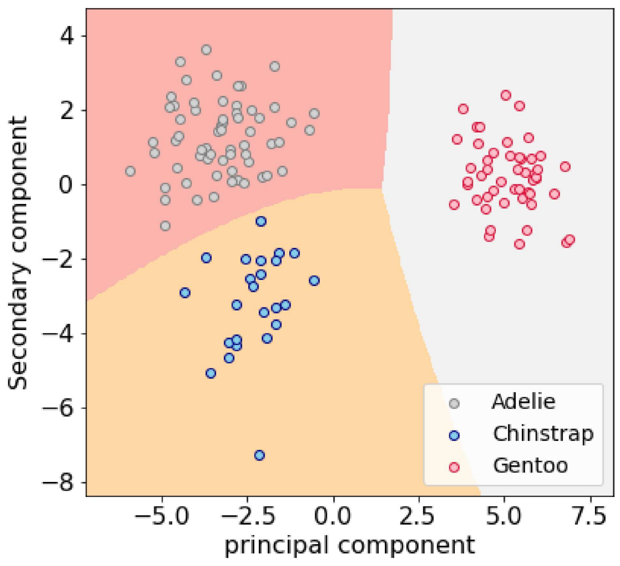
$$\ln \Pr(Y=c|X) \propto \ln \pi_c - \frac{1}{2}\ln|\Sigma_c| - \frac{1}{2}(x-\mu_c)^\intercal \Sigma_c^{-1}(x-\mu_c)$$




```
▾ QuadraticDiscriminantAnalysis
QuadraticDiscriminantAnalysis()
```

Contents

Figure 6. Decision Region of QDA for Penguin dataset



11 Generative 모형 비교

	GaussianNB	FisherDiscriminantAnalysis	LinearDiscriminantAnalysis	QuadraticDiscriminantAnalysis
Distribution of features	Gaussian	None but effectively Gaussian	Gaussian	Gaussian
Conditional independence	Yes	No	No	No
Identical Covariances	No	No	Yes	No
linear decision boundary log posterior	No	Yes decision boundary is 1D	Yes	No

Contents

	binary logistic regression	**linear discriminant analysis**
estimation	maximum likelihood Estimation	least squares estimation
estimator	probability	discriminant, prediction
feature distribution	no explicit assumption	multivariate normal
categorical features	no restriction on data types	only continuous variables
conditional variance	no requirement	independent and identical across groups
imbalance	robust	sensititve
outliers (overfitting)	a bit senstive	quite sensitive
sample size		ralatively well with large K/n

- LDA의 multivariate normal 가정은 optimality를 위한 중요한 가정으로, 가정이 만족하지 않더라고 괜찮은 성과를 보인다.
- Logistic regression과 linear discrimant analysis 모두 normality 와 homoscedasticity를 만족하면 성과가 눈에 띄게 좋아지는 것으로 알려져 있다.
- Homogeneity는 outlier와 관련이 있는 경우가 많다. LDA를 사용할 때는 (정규성 등을 위한) feature의 비선형변환이나 outlier 처리가 크게 도움이 된다.
- LDA에는 categorical feature를 사용할 수 없지만 category의 종류가 충분히 많다면 정규분포를 따른다고 가정할 수 있다.
- Logistic regression은 LDA와 비교해 decision region의 '바깥쪽에' 위치한 표본들 보다 threshold 주위의 표본에 더 큰 가중값을 사용한다. (Sigmoid의 도함수는 정규분포의 밀도함수와 비슷한 bell shape이다.) SVM은 극단적인 모형으로 그런 표본을 아예 무시해 버린다.
- The asymptotic efficiency of LDA (https://www.uio.no/studier/emner/matnat/math/STK9200/h21/efron_jasa1975.pdf) is 1.5 times better if all its requirements met, but practically there is no significant difference (https://hastie.su.domains/Papers/ESLII.pdf).

	Gaussian NB	**LDA**	**remark**
features	conditional independence	identical covariances	pixels of an image
statistical method	Bayesian	Frequentists	