



저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)

이학석사 학위논문

빅 데이터의 차원축소 기법인
Relief 알고리즘의 민감도 분석

A Sensitivity Analysis of Relief Algorithm
for Reducing the Dimensionality of Big Data




2016년 2월

한국해양대학교 대학원

데이터정보학과

금 호 연

본 논문을 금호연의 이학석사 학위논문으로 인준함.



위원장 이학박사 박찬근 (인)
위원 공학박사 김재환 (인)
위원 이학박사 김익성 (인)

2015년 11월 24일

한국해양대학교 대학원

목 차

목차	i
그림 목차	iii
표 목차	iv
초록	v
제 1 장 서 론	
1.1 연구 배경 및 목적	1
1.2 연구 내용	2
제 2 장 데이터의 축소 기법	
2.1 빅 데이터의 차원	3
2.2 특징 축소 기법	4
2.2.1 특징 선택	4
2.2.2 특징 추출/변형	6
제 3 장 Relief 알고리즘	
3.1 Relief 알고리즘의 개요	7
3.2 예제	11
3.2.1 Iris 데이터	11
3.2.2 Mushroom 데이터	16
3.3 Relief 알고리즘의 특징 및 장단점	18

제 4 장 Relief 알고리즘의 민감도 분석	
4.1 서포트 벡터 머신	19
4.2 Relief의 2단계(two-stage) 알고리즘	21
4.3 예제	24
제 5 장 결 론	27
참고 문헌	28
부록	30



그림 목차

그림 2.1	필터 방법(Filter method)	5
그림 2.2	래퍼 방법(Wrapper method)	6
그림 3.1	Relief 알고리즘	8
그림 3.2	Relief 알고리즘의 단계	9
그림 3.3	ReliefF 알고리즘의 단계	10
그림 3.4	Iris 꽃잎과 꽃받침	11
그림 3.5	Iris의 범주	11
그림 3.6	Iris의 Relief 알고리즘 R코드	13
그림 3.7	Iris의 기본 Relief 알고리즘의 실행 결과	13
그림 3.8	Iris의 ReliefF 알고리즘의 실행 결과(k=2, m=10)	14
그림 3.9	Iris의 ReliefF 알고리즘의 실행 결과(k=3, m=10)	14
그림 3.10	Mushroom 데이터	16
그림 3.11	Mushroom 기본 Relief 알고리즘의 R 코드	17
그림 3.12	Mushroom ReliefF 알고리즘의 R 코드	17
그림 4.1	2단계로 구성되는 민감도 분석 알고리즘	22
그림 4.2	2단계의 구체적인 절차	22
그림 4.3	2단계의 변수제거	23
그림 4.4	단계1의 Relief 알고리즘의 R 코드	23
그림 4.5	정확도 평가를 위한 R 코드	25

표 목차

표 3.1 기본 Relief 알고리즘 예제	9
표 3.2 Iris 데이터	12
표 4.1 예제	21
표 4.2 SVM에 의한 단계1과 단계2의 정확도 비교 결과	25



A Sensitivity Analysis of Relief Algorithm for Reducing the Dimensionality of Big Data

Kum, Ho Yeun

Department of Data Information
Graduate School of Korea Maritime and Ocean University

Abstract

Most of the real-world data mining applications are characterized by high dimensional data, where not all of the features are important. High dimensional data can contain a lot of irrelevant and noisy information that may greatly degrade the performance of a data mining process. Feature selection methods are the techniques that select a subset of relevant feature for building robust learning models by removing most irrelevant and redundant features from the data. Many feature selection methods have been developed to reduce the dimensionality of big data. Among them, the Relief algorithm is general and successful attribute estimator. The main idea of Relief algorithm is to compute ranking scores for every feature indicating how well this feature separates neighboring samples. In this study, we do perform the sensitivity analysis to find the optimal number of features and also suggest the two-stage method to design the optimal feature subset.

KEY WORDS: Big data, Feature selection, Dimensionality, Relief algorithm

제 1 장 서 론

최근 많은 인터넷 기기들이 보급됨에 따라 막대한 양의 데이터가 매일 쏟아져 나오고 있으며 이를 ‘빅 데이터’라고 부른다. 이전의 소규모의 데이터와 달리 막대한 양의 데이터를 처리하는 것은 쉬운 일이 아니며 많은 시간과 비용이 필요하다. 이러한 빅 데이터를 효율적으로 분석하고 사용하기 위해서는 데이터의 규모를 축소시키거나 불필요한 데이터를 제거하는 등 다양한 사전 처리 작업이 필요하다.

1.1 연구 배경 및 목적

특징 선택 방법(feature selection method)은 필터 방법(filter method), 래퍼 방법(wrapper method), 임베디드 방법(embedded method) 세 가지 개념의 체계로 분류할 수 있다. 필터 방법에는 Kononenko 등이 범주(class) 간의 거리를 이용한 Relief 알고리즘[1]을 제안하였고, Hall과 Smith가 정보 엔트로피(information entropy)를 사용하여 각 특징변수의 정보 획득량(Information gain)의 순위를 산정하는 정보 획득량 알고리즘[2]을 제안하였다. 이러한 방법들은 단일 변수(univariate variable)에 대한 순위를 제안하는 것이다. 이에 반해 Feng과 Ding 등은 다변량 변수(multivariate variable)간의 정보의 상호연관성(mutual information)의 개념을 도입하여 최소 중복성(minimum redundancy)을 고려한 특징 변수 선택 기법 mRMR[3]을 발표하여 화제를 모았다.

래퍼 방법은 필터 방법과는 달리 각 특징 변수 선택의 부분집합(subset)에 대해 서포트 벡터 머신[4] 등과 같은 분류기(classifier)로 정확도를 매번 평가하여 최적 특징 선택 변수를 결정하는 방법이다. 래퍼 방법은 이에 따라 필터 방법보다 컴퓨터 계산 시간이 많이 소요된다. 많은 래퍼 방법들이 개발되었으며 대표적인 래퍼 방법으로는 순차적 전진 선택법(sequential forward selection)과 후진 선택법(backward selection) 등이 있다(Inza[5], Sharma[6]). 또한, Wanderley와 Braga 등은[7] 유전자 해법(genetic algorithm)을 이용한 진화적인(evolutionary) 방법을 제시하였다. 임베디드 방법은 계산량이 많은 래퍼 방법을

보완하기 위해 분류기(classifier)를 탑재하여(embedded) 계산량을 줄이는 특징 선택기법이다. 대표적인 임베디드 기법으로는 Guyon 등이[8] 제안한 SVM-RFE(Support Vector Machine based on Recursive Feature Elimination) 방법과 Wang 등이[9] 제안한 FOIL(First Order Inductive Learner) 방법 등이 있다.

본 논문에서는 위에서 설명한 세 가지의 특징 선택 방법 중 필터 방법에 대해 다루고자 한다. 특히, 필터 방법 중에 효율적인 것으로 알려진 Relief 알고리즘을 적용하여 데이터를 분류하고, 서포트 벡터 머신을 활용하여 분류된 데이터의 정확도를 분석한다. 또한, Relief 알고리즘에 대한 민감도 분석(sensitivity analysis)을 수행하여 최적 특징 변수 집합을 설계하는 2단계 방법을 제안한다.

1.2 연구 내용

본 논문의 구성은 다음과 같다.

2장에서는 특징 축소 기법에 대해 간략히 언급한다. 특징 선택 기법 중 필터 방법과 래퍼 방법에 대해 설명한다.

3장의 Relief 알고리즘은 분류기를 사용하지 않는 필터 방법으로서 많은 특징 변수들로 이루어진 큰 규모의 실제 문제에서 특징들을 평가할 수 있는 효율적인 알고리즘 중의 하나이다. Relief 알고리즘은 일반적인 모델을 학습하기 전에 사전준비 단계에서 현재까지 가장 성공적인 사전 처리 알고리즘의 하나로 간주된다.

4장에서는 서포트 벡터 머신을 활용하여 Relief 알고리즘의 민감도를 분석한다.

기본 Relief 알고리즘과 개량된 Relief 알고리즘(ReliefF)을 사용하여 적합한 τ 값의 결정을 위한 민감도 분석을 한다. 또한, Relief 알고리즘에 의해 특징 변수를 설계한 후, 중복된 특징 변수를 제거하는 2단계의 최적 특징 변수 설계 방법을 제안한다.

끝으로 5장에서는 결론 및 추후 연구에 대해 언급한다.

제 2 장 데이터의 축소 기법

2.1 빅 데이터의 차원

데이터의 표현과 선택, 축소 혹은 특징 변형이 데이터 마이닝(data mining) 기법의 질을 높이는 가장 중요한 이슈가 되고 있다. 많은 양의 특징들은 사용 가능한 데이터의 샘플들을 분석할 때에 방해 요소가 되기도 한다. 예를 들어 특징의 개수가 수백 개라고 했을 때 분석할 샘플들이 오직 몇 백 개라도 현실적으로 사용하거나 데이터 마이닝을 위한 신뢰성 있는 모델이 되려면 차원 축소가 반드시 필요하다. 높은 차원으로 인한 데이터 과부하는 데이터 마이닝 알고리즘을 사용할 수 없게 만든다.

사전 처리 작업이 이루어지지 않은 데이터셋 에서 세 가지 중요한 차원은 행(특징들), 열(사례 혹은 샘플들), 그리고 특징들의 값 이다. 따라서 데이터 축소 과정에서 필요한 세 가지 기본적인 작업들은 행을 줄이고, 열을 줄이고, 그리고 행의 값의 수를 줄이는 것이다. 이런 세 가지 방법들을 사용하여 데이터 마이닝의 사전 처리 작업을 하였을 때 얻는 것과 잃는 것은 다음과 같다.

- 1) 계산 시간 (Computing Time)
- 2) 예측 가능한/기술적인 정확도 (Predictive/Descriptive Accuracy)
- 3) 데이터 마이닝 모형의 표현 (Representation of the Data Mining Model)

따라서 데이터의 차원을 축소하면서 시간을 줄이고, 정확도를 높이며, 간단하게 표현하는 것이 가장 이상적이다.

2.2 특징 축소 기법

데이터 마이닝에서 데이터를 다룰 때, 대부분 고차원의 데이터로 이루어져 있는데 모든 특징들이 중요한 것은 아니다. 고차원의 데이터는 연관성이 높지 않고, 잡음을 가진 정보를 가지고 있어서 데이터 마이닝 작업의 효율을 현저히 떨어뜨린다. 축소된 데이터셋들의 향상된 성능과 데이터의 질에 대해서 이야기 할 때 잡음이 섞여있고 오염된 데이터 뿐 아니라 상관없는, 불필요한 데이터의 제거 또한 이슈가 되고 있다. 데이터 마이닝에서는 데이터와 함께 반응하는 특징들은 단독으로 수집되지 않는다. 따라서 적절한 특징 하나만을 잘 활용하더라도 효율적일 수 있다.

여기서 우리가 데이터 마이닝 작업을 시행 하는데 있어 적절하게 원하는 특징을 선택하는 것을 통해 최대의 성과와 최소의 측정 및 노력을 보장해준다.

특징 축소기법은 결과적으로 다음과 같은 효과를 가져와야 한다.

- 1) 적은 데이터로 데이터 마이닝 알고리즘을 빠르게 학습해야한다.
- 2) 모델이 데이터로부터 상향표준화 되기 위한 데이터 마이닝 과정의 높은 정확도를 갖추어야 한다.
- 3) 이해하기 쉽고 사용하기 쉽도록 하기 위한 데이터 마이닝 과정의 간단한 결과를 산출해야 한다.
- 4) 상관없는 특징을 제거하여 다음 단계의 데이터 수집과 분석이 용이해야 한다.

2.2.1 특징 선택

특징 선택[10]은 데이터 마이닝 성과를 바탕으로 분석가가 초기 데이터셋에서 찾은 특징들의 부분집단을 선택해야한다. 특징 선택은 수동으로 하거나 몇 가지 지원된 자동화 과정을 통해 실행할 수 있다.

특징 선택 방법들은 필터 방법[11]과 래퍼 방법[12] 등의 개념의 체계로 분류될 수 있다. 각 방법들은 특징들을 평가하고 선택하는데 학습 알고리즘이 어떻게 포함되는가에 따라 다르다.

1) 필터 방법

필터 방법은 단일 변수에 대한 순위를 제안하는 방법으로 Relief 알고리즘, 정보 엔트로피, 정보 획득량 알고리즘 등이 존재하는데, 이러한 방법들은 효율적이고 계산이 빠르다. 반면 필터 방법은 자기 자신일 때는 쓸모가 없지만 다른 특징들과 연결되어 있을 때 매우 효과적일 수 있는 특징을 지나칠 수 있다. 이를 보완하기 위해 다변량 변수간의 정보의 상호연관성 개념을 도입하여 최소 중복성을 고려한 변수선택 기법인 mRMR[3]이 발표된 바 있다.

그림 2.1은 필터 방법의 시각적 표현모형이다.

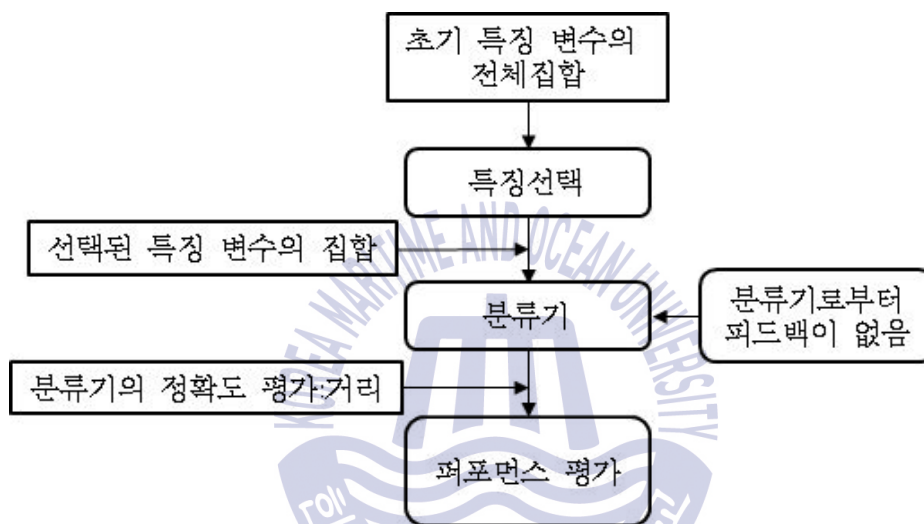


그림 2.1 필터 방법(Filter method)

2) 래퍼 방법

필터 방법과 래퍼 방법은 평가 기준으로 구분 할 수 있다. 래퍼 방법은 부분 집합 평가를 위해 그림 2.2와 같은 학습(learning or training) 알고리즘을 사용한다. 래퍼 방법은 학습 알고리즘에 가장 잘 맞는 최적의 부분집합을 선택하므로 래퍼 방법이 필터 방법보다 성과가 더 좋다. 래퍼 방법은 선택된 학습 알고리즘 주변을 “래핑” 하여 특징들을 선택하고 각 특징 부분집합 후보들을 데이터 마이닝 방법 중 학습 성과를 토대로 평가한다.

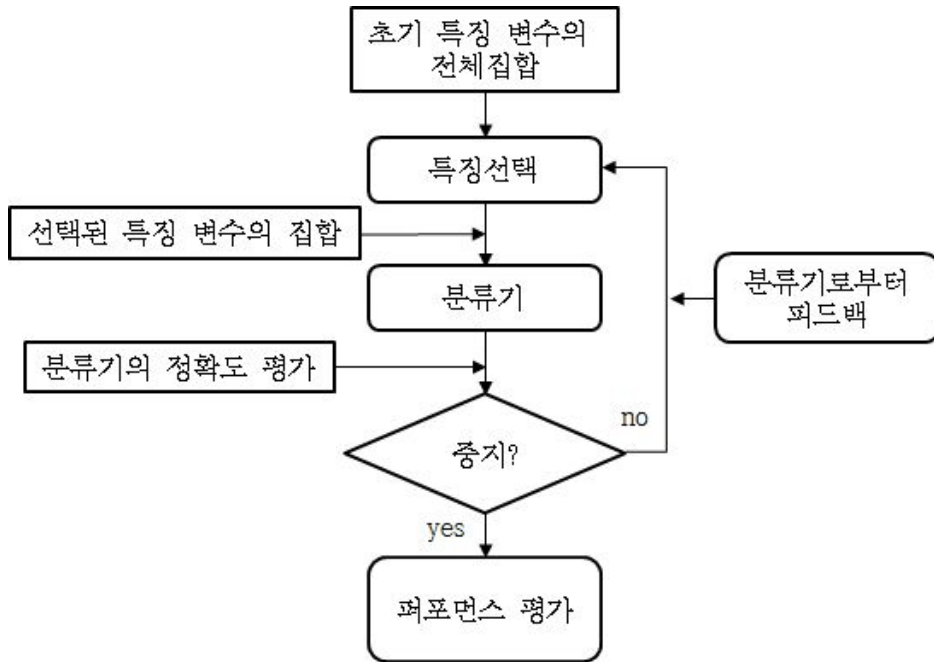


그림 2.2 래퍼 방법(Wrapper method)

2.2.2 특징 추출/변형

데이터 마이닝 방법의 결과에 아주 강력한 효과를 줄 수 있는 데이터의 변형이 존재한다. 특징들의 창작 혹은 변형이 데이터 마이닝 결과의 질을 결정하는 요소가 된다. 대부분의 경우에, 특징 창작은 응용된 지식에 의존하고 학제간의 특징 창작에 대한 접근은 데이터의 사전 처리에 두드러지는 발전을 가져다준다. 그리고 PCA[13]와 같은 몇몇 일반적 기술들은 매우 성공적으로 사용되기도 한다.

일반적으로 사람들이 특징들의 원래 의미를 간직하고 싶고 그들 중 어떤 특징이 중요한지를 결정하고 싶을 때는 특징 추출/변형보다 특징 선택이 선호된다.

제 3 장 Relief 알고리즘

3.1 Relief 알고리즘의 개요

Relief 알고리즘은 특징 선택을 위해 인스턴스 기반 학습에서 영감을 받은, 특징에 무게중심을 둔 알고리즘이다. 샘플들이 분류된 트레이닝 데이터셋에서 주어진 각 특징들의 타당성 평가에 중점을 둔다. Relief 알고리즘의 주된 관점은 하나의 특징이 얼마나 이웃의 샘플들과 잘 나뉘느냐를 나타내기 위해 모든 특징들의 랭킹 스코어를 계산하는 것이다. Relief 알고리즘의 저자들인 Kira와 Rendell은 상관있는 특징들의 랭킹 스코어가 크고 상관없는 특징들은 랭킹 스코어가 작다는 것을 증명하였다. Relief 알고리즘의 핵심은 우리가 정한 값이 서로 가까이 있는 샘플들을 얼마나 정확하게 구별하는지를 통해 특징의 질을 추정하는데 있다.

식 (3.1)을 보면 주어진 학습 데이터 S에서 알고리즘은 무작위로 샘플들의 사이즈 m의 부분 집합을 선택하는데 이때 m은 사용자가 지정한다. Relief는 선택된 샘플들의 부분 집합을 토대로 각 특징을 분석한다. 학습 데이터셋 으로부터 각각의 무작위로 선택된 샘플 X에 대해 Relief는 자신의 두 가장 가까운 이웃을 찾는다. 하나는 같은 클래스에서, 이를 nearest hit H라하고, 다른 하나는 다른 클래스에서 이를 nearest miss M이라 한다. 2차원 데이터의 예는 다음 그림 3.1과 같다.

$$W_n(A_i) = W_o(A_i) - (diff(X[A_i], H[A_i])^2 + diff(X[A_i], M[A_i])^2) / m \quad (3.1)$$

Relief 알고리즘은 자질 점수 $W(A_i)$ 를 샘플 X, M, H의 차이 값에 따라 모든 특징 A_i 에 대해 업데이트 한다. 이 과정은 랜덤하게 선택된 트레이닝 데이터셋의 샘플로부터 m번 반복되고 $W(A_i)$ 의 스코어들은 각 샘플로부터 모아진다. 마지막으로 적합성의 한계점인 τ 를 사용하여 알고리즘은 통계적으로 타겟 분류에 유의한 특징들을 찾아내고 이 특징들은 식(3.2)를 만족해야한다.

$$W(A_i) \geq \tau \quad (3.2)$$

Relief 알고리즘

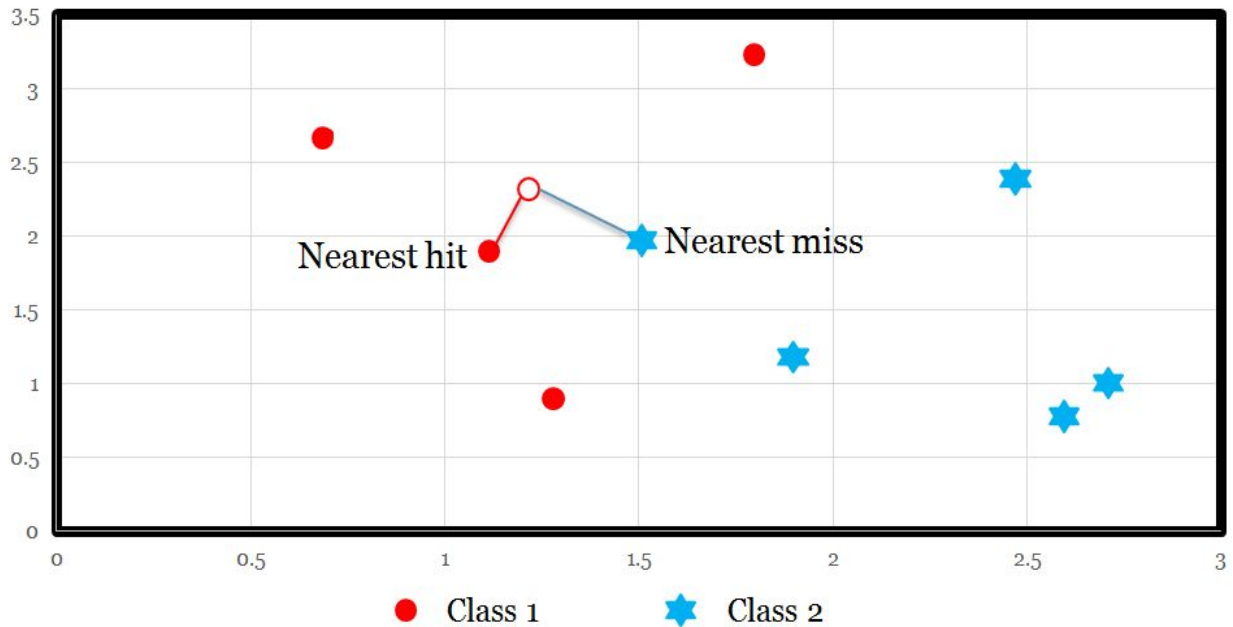


그림 3.1 Relief 알고리즘

Relief 알고리즘의 단계는 그림 3.2와 같이 의사 코드(pseudo code)로 나타낼 수 있다.

```

초기화 :  $W(A_i)=0; i=1, \dots, p$  (p는 특징들의 개수)
For i = 1 to m
  샘플 X를 학습 데이터셋 S로부터 랜덤하게 선택한다.
  자기 클래스에서 가장 가까운 H 와 다른 클래스에서 가장 가까운 M 샘플을 찾는다.
  For j = 1 to p
     $W_n(A_i) = W_o(A_i) - (diff(X[A_i], H[A_i])^2 + diff(X[A_i], M[A_i])^2) / m$ 
  End.
End.
출력 :  $W(A_i) \geq \tau$  를 만족하는 특징의 부분 집합
    
```

그림 3.2 Relief 알고리즘의 단계

예를 들어, 만약 세 가지 특징과 네 개의 샘플들로 구성된 학습 데이터셋이 표 3.1과 같을 때, Relief 알고리즘을 적용하여 F1과 F2에 대한 자질 점수 (quality score) W를 계산하였다.

표 3.1 기본 Relief 알고리즘 예제

sample	F1	F2	Class
1	3	4	C1
2	2	5	C1
3	6	7	C2
4	5	6	C2

$$W(F_1) = (0 + [-1+4] + [-1+9] + [-1+9] + [-1+4])/4 = 5.5$$

$$W(F_2) = (0 + [-1+4] + [-1+1] + [-1+9] + [-1+4])/4 = 3.5$$

표 3.1과 같은 예제에서는 샘플들의 개수가 적어 특징들의 자질 점수를 계산하는데 모든 샘플을 사용하였다. 이전의 결과를 살펴보면 F_1 이 특징을 분류하는데 있어 F_2 보다 훨씬 적절함을 알 수 있다. 사용자가 한계 값 $\tau=5$ 라고 정한다면, 특징 F_2 를 제거하는 것이 가능하고 F_1 으로만 분류 모델을 만드는 것이 가능해진다.

Relief 알고리즘을 개량한 ReliefF 알고리즘에서는 이웃한 k개를 고르며 다중 범주(multi-class)를 다루고 있다. 그림 3.3은 ReliefF 알고리즘의 단계를 의사 코드로 나타낸 것이다.

삽입 : 각 트레이닝 공간에서 자질 값과 클래스 값의 벡터

1. set all weight $W[A] := 0.0$;
2. for $i := 1$ to m do begin
3. randomly select an instance R_i ;
4. find k nearest hits H_j ;
5. for each class $C \neq class(R_i)$ do
6. from class C find k nearest misses $M_j(C)$;
7. for $A := 1$ to a do
8. $W[A] := W[A] - \sum_{j=1}^k diff(A, R_i, H_j) / (m \cdot k) +$
9. $\sum_{C \neq class(R_i)} \left[\frac{P(C)}{1 - P(class(R_i))} \sum_{j=1}^k diff(A, R_i, M_j(C)) \right] / (m \cdot k)$;

End.

출력 : 자질들의 질을 평가하는 벡터 W

그림 3.3 ReliefF 알고리즘의 단계

3.2 예제

3.2.1 Iris 데이터

Iris 데이터를 기본 Relief 알고리즘과 ReliefF 알고리즘으로 실험하여 비교해 보았다. Iris는 그림 3.4와 같이 꽃잎(Petal), 꽃받침(Sepal)의 길이와 너비로 이루어져 있으며, 그림 3.5와 같이 Versicolor, Virginica, Setosa 세 가지 종류의 붓꽃에 대한 150개의 데이터이다.



그림 3.4 Iris 꽃잎과 꽃받침



그림 3.5 Iris의 범주

표 3.2는 Iris 데이터를 요약하여 나타낸 것이다.

표 3.2 Iris 데이터

No.	Sepal. Length	Sepal. Width	Petal. Length	Petal. Width	Species
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3.0	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
4	4.6	3.1	1.5	0.2	setosa
5	5.0	3.6	1.4	0.2	setosa
6	5.4	3.9	1.7	0.4	setosa
7	4.6	3.4	1.4	0.3	setosa
8	5.0	3.4	1.5	0.2	setosa
9	4.4	2.9	1.4	0.2	setosa
10	4.9	3.1	1.5	0.1	setosa
11	7.0	3.2	4.7	1.4	versicolor
12	6.4	3.2	4.5	1.5	versicolor
13	6.9	3.1	4.9	1.5	versicolor
14	5.5	2.3	4.0	1.3	versicolor
15	6.5	2.8	4.6	1.5	versicolor
16	5.7	2.8	4.5	1.3	versicolor
17	6.3	3.3	4.7	1.6	versicolor
18	4.9	2.4	3.3	1.0	versicolor
19	6.6	2.9	4.6	1.3	versicolor
20	5.2	2.7	3.9	1.4	versicolor
21	6.3	3.3	6.0	2.5	virginica
22	5.8	2.7	5.1	1.9	virginica
23	7.1	3.0	5.9	2.1	virginica
24	6.3	2.9	5.6	1.8	virginica
25	6.5	3.0	5.8	2.2	virginica
26	7.6	3.0	6.6	2.1	virginica
27	4.9	2.5	4.5	1.7	virginica
28	7.3	2.9	6.3	1.8	virginica
29	6.7	2.5	5.8	1.8	virginica
30	7.2	3.6	6.1	2.5	virginica

Iris 데이터를 기본 Relief 알고리즘으로 실행한 R 코드와 결과는 각각 그림 3.6와 그림 3.7에 나타냈다.

```

library(FSelector)
data(iris)
weights <- relief(Species~., iris, neighbours.count = 1, sample.size = 10)
print(weights)
subset <- cutoff.k(weights, 2)
f <- as.simple.formula(subset, "Species")
print(f)

```

그림 3.6 Iris의 Relief 알고리즘 R코드

```

library(FSelector)
data(iris)
weights <- relief(Species~., iris, neighbours.count = 1, sample.size = 10)
print(weights)

```

	attr_importance
Sepal.Length	0.1722222
Sepal.Width	0.0750000
Petal.Length	0.2881356
Petal.Width	0.2583333

```

subset <- cutoff.k(weights, 2)
f <- as.simple.formula(subset, "Species")
print(f)
Species ~ Petal.Length + Petal.Width

```

그림 3.7 Iris의 기본 Relief 알고리즘의 실행 결과

cutoff.k(weights, 2) 부분에서 가장 높게 나온 두 가지 특징을 선택함을 알 수 있는데, Sepal.Length, Sepal.Width, Petal.Length, Petal.Width 네 가지 특징 중에 Petal.Length가 가장 높게 나오고 그 다음이 Petal.Width 순서로 높게 나타났다.

그림 3.8은 ReliefF 알고리즘으로 그림 3.6의 코드를 실행한 것이다.

```

weights <- relief(Species~., iris, neighbours.count = 2, sample.size = 10)
print(weights)
      attr_importance
Sepal.Length      0.1388889
Sepal.Width       0.1656250
Petal.Length      0.3279661
Petal.Width       0.3354167
subset <- cutoff.k(weights, 2)
f <- as.simple.formula(subset, "Species")
print(f)
Species ~ Petal.Width + Petal.Length

```

그림 3.8 Iris의 ReliefF 알고리즘의 실행 결과 (k=2, m=10)

그림 3.9는 샘플 사이즈를 50으로 늘리고 이웃해 개수 k를 3개씩 선택하도록 한 ReliefF 알고리즘의 실행 결과이다.

```

library(FSelector)
data(iris)
weights <- relief(Species~., iris, neighbours.count = 3, sample.size = 50)
print(weights)
      attr_importance      attr_importance
Sepal.Length      0.1540741  Sepal.Width      0.1444444
Petal.Length      0.3436158  Petal.Width      0.3529167

subset <- cutoff.k(weights, 2)
f <- as.simple.formula(subset, "Species")
print(f)
Species ~ Petal.Width + Petal.Length

```

그림 3.9 Iris의 ReliefF 알고리즘의 실행 결과 (k=3, m=50)

그림 3.8에서는 기본 Relief 와 반대로 Petal.Width가 Petal.Length보다 높다는 결과를 얻었다. 그림 3.9는 더 큰 샘플 사이즈와 이웃 해를 k개 늘려 분석한 것이다. Petal.Width, Petal.Length 순으로 결과가 나타났는데, 여기서 알 수 있는 것은 이웃 해의 개수를 다르게 했을 때 결과가 다르게 나타났다는 점이다.



3.2.2 Mushroom 데이터

그림 3.10과 같은 Mushroom 데이터를 기본 Relief 알고리즘과 ReliefF 알고리즘으로 실험하여 비교해 보았다. Mushroom 데이터셋은 북미의 Audubon 사회 분야 가이드에서 가져온 버섯에 관한 데이터이다. 핀 불이 버섯 23종에 해당하는 가상의 샘플들을 포함하고 있고 각각의 종은 확실하게 식용 혹은 독성으로 분류된다. 각 특징들은 cap의 모양, 표면, 색, gill의 붙임성, 위치, 크기, 색, 줄기의 모양, 색, veil의 종류, 색 등이 포함되어 있다.

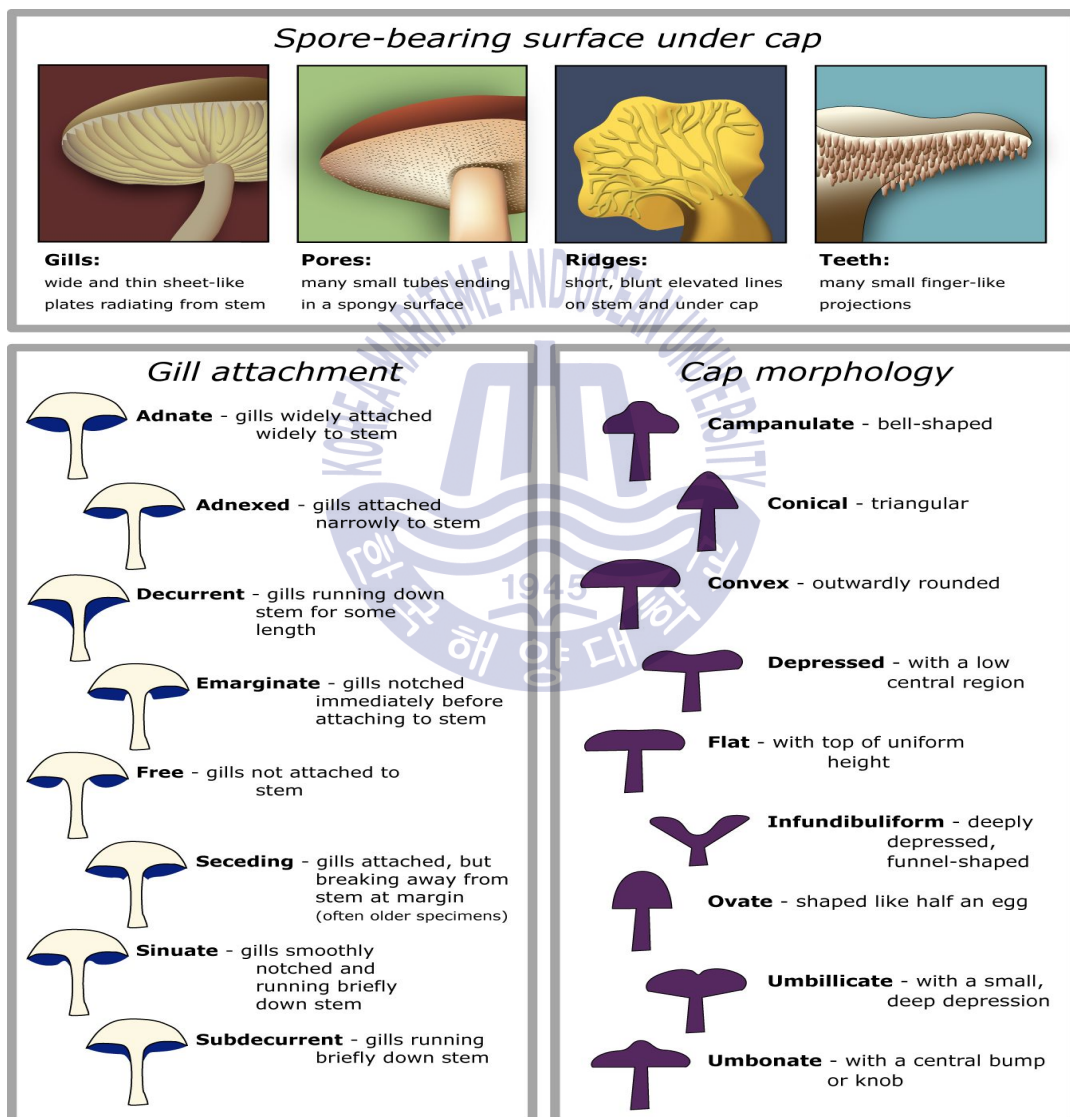


그림 3.10 Mushroom 데이터

Mushroom 데이터셋을 실험한 Relief 알고리즘의 R코드는 그림 3.11과 같다.


```

library(FSelector)
d<-read.csv("mushroom.csv")
weights <- relief(y~., d, neighbours.count = 1, sample.size = 20)
print(weights)
subset <- cutoff.k(weights, 5)
f <- as.simple.formula(subset, "Species")
print(f)

```

그림 3.11 Mushroom 기본 Relief 알고리즘의 R 코드

Relief 알고리즘을 실행한 자세한 결과는 부록에 나타나 있다. 요약한 결과는 다음과 같으며, 92개의 변수 중에서 24, 43, 44, 19, 20이 순서대로 선택됨을 확인하였다.

```

> subset <- cutoff.k(weights, 5)
> f <- as.simple.formula(subset, "Species")
> print(f)
Species ~ x24 + x43 + x44 + x19 + x20

```

그림 3.12는 ReliefF 알고리즘의 R 코드이다.

```

library(FSelector)
d<-read.csv("mushroom.csv")
weights <- relief(y~., d, neighbours.count = 3, sample.size = 20)
print(weights)
subset <- cutoff.k(weights, 5)
f <- as.simple.formula(subset, "Species")
print(f)

```

그림 3.12 Mushroom ReliefF 알고리즘의 R 코드

ReliefF 알고리즘을 실행한 자세한 결과는 부록에 나타나 있으며 간단히 요약하면 다음과 같다.

```

> subset <- cutoff.k(weights, 5)
> f <- as.simple.formula(subset, "Species")
> print(f)
Species ~ x24 + x43 + x44 + x19 + x20

```

기본 Relief 알고리즘을 실행 하였을 때와 마찬가지로 τ 값을 5로 정하였을 때 92개의 변수 중에서 24, 43, 44, 19, 20이 순서대로 선택됨을 확인하였다.

3.2 Relief 알고리즘의 특징 및 장단점

Relief 알고리즘의 시간 복잡도는 $O(mpn)$ 이고, 넓은 특징공간과 큰 숫자의 샘플들로 이루어진 실제 문제에서 특징들을 평가 할 수 있는 효율적인 필터방법 중 하나이다. Relief는 또한 소음을 잘 견디고 특징 상호작용에 영향을 끼치지 않으며 특히 어려운 데이터 마이닝 응용에 중요하다.

반면 Relief 알고리즘은 쓸모없는 특징들을 없애는 데는 아무런 도움이 되지 않는다. 클래스 개념에 특징들이 유의하다고 여기면 그들이 상관성이 높다 하더라도 선택될 것이다.

Relief 알고리즘의 문제점중 하나는 적당한 τ 값을 선택하는 것이다. 이론적으로 이른바 체비셰프(Cebysev)의 불균등 τ 가 사용되는 것으로 추측된다. τ 의 계산식은 식 (3.3)과 같다.

$$\tau \ll 1/\sqrt{\alpha m} \quad (3.3)$$

식 3.3의 α 와 m 에 관하여 τ 를 결정하는 동안, 실험들은 유의한 특징들과 그렇지 않은 특징들의 점수를 뚜렷하게 대조적으로 보여주고 τ 가 조사를 통해 쉽게 결정된다.

Relief 알고리즘은 다중 클래스 문제, 소음, 불필요함, 빠진 값 등을 해결하기 위해 확장되었고 최근에는 추가적으로 특징 선택 방법을 기반으로 한 특징 가중치 계산이 제안되었는데, ReliefF[14], RRelief-F[15], Simba, 그리고 I-Relief 알고리즘 등을 예로 들 수 있다.

제 4 장 Relief 알고리즘의 민감도 분석

본 논문에서는 Relief 알고리즘에 의해 구한 최적특징 변수의 정확도를 서포트 벡터 머신(SVM: Support Vector Machine) 분류기(classifier)를 이용하여 평가하였다.

4.1 서포트 벡터 머신

서포트 벡터 머신은 기계 학습의 분야 중 하나로 패턴 인식, 자료 분석을 위한 지도 학습 모델이며, 주로 분류와 회귀 분석을 위해 사용한다. 두 카테고리 중 어느 하나에 속한 데이터의 집합이 주어졌을 때, SVM 알고리즘은 주어진 데이터 집합을 바탕으로 하여 새로운 데이터가 어느 카테고리에 속할지 판단하는 비확률적 이진선형 분류 모델을 만든다. 만들어진 분류 모델은 데이터가 사상된 공간에서 경계로 표현되는데 SVM 알고리즘은 그 중 가장 큰 폭을 가진 경계를 찾는 알고리즘이다. SVM은 선형 분류와 더불어 비선형 분류에서도 사용될 수 있다. 비선형으로 분류하기 위해서 주어진 데이터를 고차원 특징 공간으로 사상하는 작업이 필요한데, 이를 효율적으로 수행하기 위해 커널 트릭을 사용하기도 한다.

일반적으로, 서포트 벡터 머신은 분류 또는 회귀 분석에 사용 가능한 초평면(hyperplane) 또는 초평면들의 집합으로 구성되어 있다. 직관적으로, 초평면이 가장 가까운 학습 데이터 점과 큰 차이를 가지고 있으면 분류 오차(classifier error)가 작기 때문에 좋은 분류를 위해서는 어떤 분류된 점에 대해서 가장 가까운 학습 데이터와 가장 먼 거리를 가지는 초평면을 찾아야 한다. 일반적으로 초기의 문제가 유한 차원 공간에서 다루어지는데, 종종 데이터가 선형 구분이 되지 않는 문제가 발생한다. 이러한 문제를 해결하기 위해 초기 문제의 유한 차원에서 더 높은 차원으로 대응시켜 분리를 쉽게 하는 방법이 제안되었다. 그 과정에서 계산량이 늘어나는 것을 막기 위해서, 각 문제에 적절한 커널 함수 $k(x,y)$ 를 정의한 SVM 구조를 설계하여 내적 연산을 초기 문제의 변수들을 사용해서 효과적으로 계산할 수 있도록 한다. 높은 차원 공간의 초평면은 점들의

집합과 상수 벡터의 내적 연산으로 정의된다. 초평면에 정의된 벡터들은 데이터베이스 안에 나타나는 이미지 벡터 매개 변수들과의 선형적 결합이 되도록 선택된다. 이 선택된 초평면에서, 초평면에 대응된 점 x 는 다음 식(4.1)과 같은 관계가 성립한다.

$$\sum_i \alpha_i k(x_i, x) = c \text{ (단, } c \text{는 상수)} \quad (4.1)$$

만약 $k(x, y)$ 가 x 와 y 가 점점 멀어질수록 작아진다면, 각각의 합은 테스트 점 x 와 그와 대응되는 데이터 점 x_i 의 근접성의 정도를 나타내게 된다. 이러한 방식으로, 위 커널식의 합은 구별하고 싶은 집합 안에 있는 데이터 점과 테스트 점간의 상대적인 근접성을 측정하는데 사용될 수 있다.

초기 공간에서 블록하지 않는 집합안의 점 x_i 가 높은 차원으로 대응되었을 때 오히려 더 복잡하고 어려워질 수도 있는데 이런 부분을 주의해야 한다.

데이터를 분류 하는 것은 기계학습에 있어서 일반적인 작업이다. 주어진 데이터 점들이 두 개의 클래스 안에 각각 속해 있다고 가정했을 때, 새로운 데이터 점이 두 클래스 중 어느 곳에 속하는지 결정하는 것이 목표이다. 서포트 벡터 머신에서, 데이터 점이 p -차원의 벡터(p 개의 숫자 리스트)로 주어졌을 때, 이러한 데이터 점을 $(p-1)$ -차원의 초평면으로 분류할 수 있는지를 확인하고 싶은 것이다. 이러한 작업을 선형 분류라고 말한다. 데이터를 분류하는 초평면은 여러 경우가 나올 수 있다. 초평면을 선택하는 타당한 방법 중 하나는 두 클래스 사이에서 가장 큰 분류 또는 마진(margin)을 가지는 초평면을 선택하는 것이다. 그래서 우리는 초평면에서 가장 가까운 각 클래스의 데이터 점들 간의 거리를 최대로 하는 초평면을 선택한다. 만약 그런 초평면이 존재할 경우, 그 초평면을 최대-마진 초평면(maximum-margin hyperplane) 이라 하고 선형 분류기를 최대-마진 분류기(maximum margin classifier) 라고 한다.

4.2 Relief의 2단계(two-stage) 알고리즘

현재까지 진행해 왔던 R프로그램에서의 Relief 알고리즘과 ReliefF 알고리즘은 최종적인 특징 선택 단계에서 τ 값을 사용하여 선택하는 것이 아닌, 개수를 지정하거나 일정 비율을 선택하도록 설정 되어 있다.

예제는 다음 표 4.1에 나타나 있다.

표 4.1 예제

4.7	6.0
4.5	5.1
4.9	5.9
5.8	5.6
4.6	5.0
4.7	6.6
5.8	4.5
3.9	5.3
4.9	5.8

`subset <- cutoff.k(weights, 5)`를 실행하면 높은 숫자부터 6.6, 6.0, 5.9, 5.8, 5.8 이 선택되게 되고, `subset <- cutoff.k.percent(weights, 0.2)`을 실행 하면 총 18개 특징 중 상위 20퍼센트를 선택하게 되는데, 6.6, 6.0, 5.9, 5.8 이 선택되므로 앞의 경우에는 5.8 인 특징이 세 개인 상황에서 두 개만 선택되고, 뒤의 경우에는 한 개만 선택되므로 어떤 특징이 선택되는지 모호하다는 단점이 있다.

또한 같은 특징이 여러 개가 있다는 것이 효율적이라는 것은 아니며, 오히려 특징을 선택하는데 혼란을 야기할 수 있다. 그러므로 본 논문에서는 2단계로 구성되는 Relief 알고리즘을 제안한다.

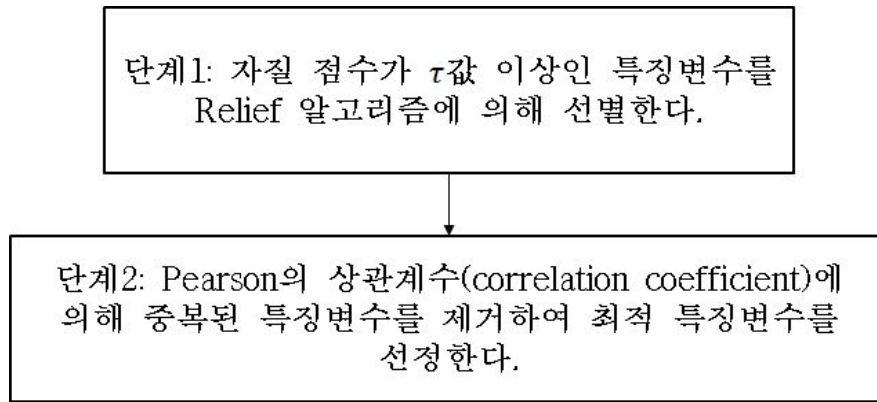


그림 4.1 2단계로 구성되는 민감도 분석 알고리즘

그림 4.1의 단계 2의 구체적인 절차를 그림 4.2에 도식화 하였다. 이 절차는 FCBF[16](Fast Correlation-based Filter) 방법과 유사하다.

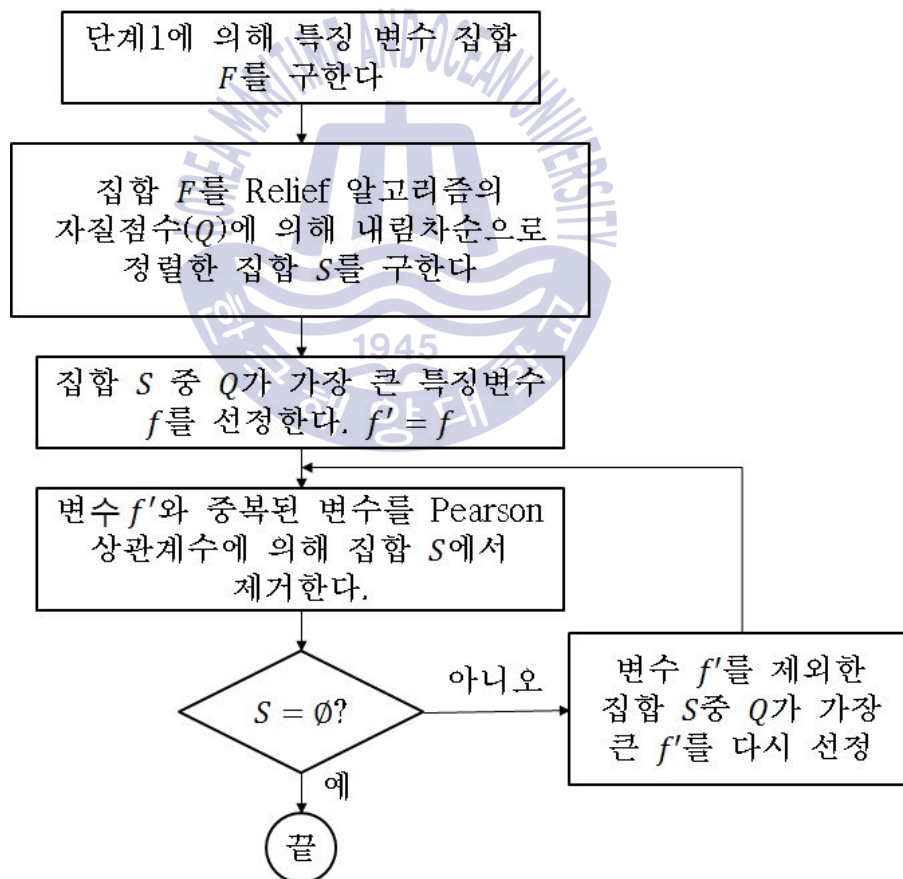


그림 4.2 2단계의 구체적인 절차

2단계의 절차를 예를 통해 설명하면, $S = \{f_1, f_2, f_3, f_4, f_5, f_6\}$ 일 때 f_1 과 상관성이 있는 특징변수가 f_2 와 f_4 라고 가정하면, 그림 4.3과 같이 f_2 와 f_4 가 제거된다. 그 다음 f_3 과 중복되는 f_5 를 제거하여 최적 특징변수 f_1, f_3, f_6 을 선정한다.

$f_1, f_2, f_3, f_4, f_5, f_6$

그림 4.3 2단계의 변수제거

4.3 예제

본 논문에서 제안하는 알고리즘 분석에 사용된 데이터셋은 UC Irvine MLR 사이트에 게시된 Mushroom, Waveform, Isolet, lungcancer, wine, ticdata2000 이다. 단계 1의 Relief 알고리즘의 R코드와 서포트 벡터 머신의 R 코드는 그림 4.4와 같다.

```
library("e1071")
w<-read.csv("C:/Users/GOLD/Desktop/lungcancer.csv")
w$y<-as.factor(w$y)
w<-data.frame(y=w$y,w[,-ncol(w)]); set.seed(5)
weights <- relief(y~.,w, neighbours.count =3, sample.size =30)
r<-weights[,1]; names(r)<-rownames(weights)
m(r) ; x<-f(r,0.02)
i<-sample(1:nrow(w),0.7*nrow(w))
train.x<-w[i,-1]; train.y<-w[i,1]
model<- svm(train.x,train.y)
pred <- predict(model, test.x)
table(pred,test.y)
```

그림 4.4 단계1의 Relief 알고리즘의 R 코드

단계 2의 R코드는 그림 4.5와 같다.

```

Leuk<-read.csv("Leuk.csv")
Leuk$y<-as.factor(Leuk$y)
r<-relief(y~,Leuk,,)
r<-weights[,1]
names(r)<-rownames(weights)
m(r)
x<-f(r,0.2)
d<-w[,-1]
x1<-g(r[x],d[,x],0.5)
    
```

그림 4.5 정확도 평가를 위한 R 코드

표 4.2 SVM에 의한 단계1과 단계2의 정확도 비교 결과

	단계1	단계2	오차	제거된 개수
Wine	0.9629	0.9629	0	0
Isolet	0.7823	0.7823	0	0
Waveform	0.8593	0.8287	0.0306	3
Mushroom	0.9976	0.9964	0.0012	1
Lungcancer	0.2	0.30	-0.1	4
Ticdata2000	0.9342	0.9382	-0.004	4

모든 데이터셋을 $n=30$, $k=3$ 을 선택하도록 고정하고 서로 상관성이 있는 변수를 제거하기 전(단계 1)과 제거 후(단계 2)의 정확도의 차이를 비교해 보았다. τ 값은 특징 변수들의 평균보다 약간 높은 값을 설정하였다.

Isolet과 Wine데이터에서는 단계 1과 단계 2의 결과가 동일하게 나타났는데, 이는 서로 상관성이 있는 변수의 개수가 적음을 의미하며 여기서는 제거된 변수가 없다는 것을 알 수 있다.

Mushroom과 Waveform데이터를 실험하였을 때는 단계 2가 단계 1보다 Mushroom과 Waveform 데이터에 대해 각각 1개, 3개 적게 구성되는 것을 관측하였다. 오차역시 0.0012, 0.0306으로 매우 낮게 나타났다.

Lungcancer 와 Ticdata2000 데이터의 경우는 단계 2에서 각각 4개의 변수가 제거되었고, 정확도는 오히려 본 논문에서 제시한 단계 2의 경우가 Relief 알고리즘 만으로 구성하는 단계 1의 경우보다 각각 0.1, 0.004만큼 높은 것을 관측하였다. 이는 본 논문에서 제안하는 2단계 알고리즘이 효율적이라는 것을 의미한다.



제 5 장 결 론

최근 빅 데이터가 이슈화 되면서 빅 데이터의 축소와 처리 그리고 분석에 관한 연구의 필요성이 강조되고 있다. 빅 데이터의 차원축소 문제는 데이터 마이닝에서 중요한 문제이다. 이를 위해 다양한 특징 선택 방법들이 연구 및 개발되었으며, 특히 Relief 알고리즘이 효율적인 것으로 알려져 있다. 따라서 본 논문에서는 Relief 알고리즘과 Relief 알고리즘의 개량형인 ReliefF 알고리즘에 대한 민감도 분석을 수행하여 최적 특징 변수를 설계 하였다. 또한, 기존의 ReliefF 알고리즘 기반의 분석 결과에 추가적으로 본 논문에서 제안한 2단계 알고리즘을 적용하여 최적 특징 변수를 선정하였다.

특징 변수의 최적 설계를 위해 본 논문에서는 2단계 알고리즘을 적용하였으며, 그 결과 적은 개수의 특징 변수로 경제적일 뿐 아니라 정확도 또한 비교적 높게 나타났다. Ticdata2000과 Lungcancer 데이터셋에 대해서는 중복 변수를 제거한 후(2단계)의 정확도가 오히려 더 높게 나오는 것을 관측하였다. 이는 특징변수의 개수를 줄여주면서 정확도는 높아지므로 상당히 효과적인 결과라 할 수 있다.

추후 연구로서는 본 논문에서 제시하는 2단계 알고리즘의 효율성 성능평가를 위해 다양한 데이터셋을 실험해 볼 필요가 있다. 그리고 Relief 알고리즘을 R프로그램으로 실행할 경우 데이터셋의 크기가 클수록 상당한 계산 시간이 소요되므로 시간을 단축시킬 수 있는 방법을 고안하거나 R 프로그램 외의 다른 프로그램을 활용하여 Relief 알고리즘의 실험을 위한 방법의 모색이 필요하다.

참 고 문 헌

- [1] K. Kira and C. A. Rndell (1992), "The feature selection problem : Traditional methods and a new algorithm." , Proceedings of the Tenth National Conference on Artificial Intelligence, pp. 129-134.
- [2] J. R. Quinlan (1986), "Induction of Decision Trees" , Machine Learning, vol. 1, pp. 81-106.
- [3] H. Peng, F. Long, and C. Ding (2005), "Feature Selection Based on Mutual Information : Criteria of Max-Dependency, Max-Relevance, and Min-Redundancy" , IEEE Transactions On Pattern Analysis And Machine Intelligence, vol. 27, pp. 1226-1238.
- [4] C.J.C. Burges, (1998), "A Tutorial on Support Vector Machines for Pattern Recognition" , Knowledge Discovery and Data Mining, vol. 2, pp. 1-43.
- [5] I. Inza, B. Sierra, R. Blanco, P. Larranaga, (2002), "Gene selection by sequential search wrapper approaches in microarray cancer class prediction" ,J. Intell. FuzzySyst, vol. 12, pp. 25-33.
- [6] A. Sharma, S. Imoto, S. Miyano, (2012), "A top-r feature selection algorithm for microarray gene expression data" , IEEE/ACM Trans. Comput. Biol. Bionformatics, vol. 9, pp. 754-764.
- [7] M. Wanderley, V. Gardeux, R. Natowicz, A. Braga, (2013), "Ga-kde-bayes:an evolutionary wrapper method based on non-parametric density estimation applied to bioinformatics preblems, in:21st European Symposium on Artificial Neural Networks-ESANN, pp. 155-160.
- [8] I. Guyon, J. Weston, S. Barnhill, V. Vapnik, (2002), "Gene selection for cancer classification using support vector machines" , Machine Learning, vol. 46, pp. 389-422.

- [9] G. Wang, Q. Song, B. Xu, Y. Zhou, (2013), “Selecting feature subset for high dimensional data via the propositional foil rules” ,Pattern Recognition, vol. 46, pp. 199-214.
- [10] A.K. Jain and D. Zongker (1997), “Feature Selection : Evaluation, Application, and Small Sample Performance” , IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 19, pp. 153-158.
- [11] V. Kumar and S. Minz (2014), “Feature Selection : A Literature Review” ,Smart Computing Review, vol. 4, pp. 211-229.
- [12] R. Kohavi and G. John (1997), “Wrappers for feature selection” , Artificial Intelligence, vol. 97, pp. 273-324.
- [13] H. Liu and H. Motoda (1998), “Feature Selection for Knowledge Discovery and Data Mining” , Kluwer Academic Publishers, London, GB
- [14] I. Konononko (1994), “Estimating attributes: Analysis and extensions of RELIEF” ,European Conference on Machine Learning, pp. 171-182.
- [15] M. Robnik Sikonja and I. Kononenko (1997), “An adaptation of relief for attribute estimation in regression” , Machine Learning: Proceedings of the Fourteenth International Conference, pp. 296-304.
- [16] H. Liu and L. Yu (2003), “Feature Selection for High-Dimensional Data: A Fast Correlation-Based Filter Solution” , In Proceedings of The Twentieth International Conference on Machine Learning , pp. 856-863.

부 록

표 1 Mushroom 데이터셋의 기본 Relief 실행 결과

특징변수	Relief 결과	특징변수	Relief 결과
x1	-0.03333333	x25	0.43333333
x2	0.06666667	x26	0.13333333
x3	0.00000000	x27	0.00000000
x4	-0.10000000	x28	0.00000000
x5	0.00000000	x29	0.00000000
x6	0.00000000	x30	0.16666667
x7	0.20000000	x31	0.16666667
x8	0.06666667	x32	0.46666667
x9	0.23333333	x33	0.46666667
x10	0.03333333	x34	0.03333333
x11	0.06666667	x35	-0.10000000
x12	0.13333333	x36	0.06666667
x13	0.10000000	x37	0.03333333
x14	0.03333333	x38	0.03333333
x15	0.03333333	x39	0.10000000
x16	0.03333333	x40	0.03333333
x17	0.06666667	x41	0.00000000
x18	0.00000000	x42	0.00000000
x19	0.50000000	x43	0.50000000
x20	0.50000000	x44	0.50000000
x21	0.36666667	x45	0.26666667
x22	0.06666667	x46	0.20000000
x23	0.20000000	x47	0.26666667
x24	0.66666667		

표 1 Mushroom 데이터셋의 기본 Relief 실행 결과(계속)

특징변수	Relief 결과	특징변수	Relief 결과
x48	0.06666667	x71	0.00000000
x49	0.23333333	x72	0.00000000
x50	0.06666667	x73	0.00000000
x51	0.20000000	x74	0.03333333
x52	0.03333333	x75	0.03333333
x53	0.20000000	x76	0.00000000
x54	0.03333333	x77	0.43333333
x55	0.10000000	x78	0.23333333
x56	0.20000000	x79	0.20000000
x57	0.30000000	x80	0.00000000
x58	0.10000000	x81	0.20000000
x59	0.10000000	x82	0.06666667
x60	0.03333333	x83	0.00000000
x61	0.13333333	x84	0.43333333
x62	0.00000000	x85	0.03333333
x63	0.00000000	x86	0.06666667
x64	0.36666667	x87	0.13333333
x65	0.10000000	x88	0.06666667
x66	0.13333333	x89	0.16666667
x67	0.06666667	x90	0.33333333
x68	0.06666667	x91	0.20000000
x69	0.00000000	x92	0.03333333
x70	0.00000000		

```

> subset <- cutoff.k(weights, 5)
> f <- as.simple.formula(subset, "Species")
> print(f)
Species ~ x24 + x43 + x44 + x19 + x20

```

표 2 Mushroom 데이터셋의 ReliefF 실행 결과

특징변수	Relief결과	특징변수	Relief결과
x1	-6.666667e-02	x27	0.000000e+00
x2	6.666667e-02	x28	0.000000e+00
x3	1.111111e-02	x29	0.000000e+00
x4	-1.000000e-01	x30	1.666667e-01
x5	0.000000e+00	x31	1.666667e-01
x6	0.000000e+00	x32	4.444444e-01
x7	2.000000e-01	x33	4.444444e-01
x8	5.555556e-02	x34	1.850372e-18
x9	2.000000e-01	x35	-3.333333e-02
x10	1.111111e-02	x36	5.555556e-02
x11	7.777778e-02	x37	-1.111111e-02
x12	1.000000e-01	x38	8.888889e-02
x13	7.777778e-02	x39	1.000000e-01
x14	-3.333333e-02	x40	2.222222e-02
x15	4.444444e-02	x41	-2.222222e-02
x16	5.555556e-02	x42	0.000000e+00
x17	4.444444e-02	x43	4.777778e-01
x18	1.111111e-02	x44	4.777778e-01
x19	4.777778e-01	x45	2.666667e-01
x20	4.777778e-01	x46	2.000000e-01
x21	3.666667e-01	x47	2.777778e-01
x22	1.111111e-01	x48	7.777778e-02
x23	1.222222e-01	x49	2.444444e-01
x24	6.666667e-01	x50	6.666667e-02
x25	4.555556e-01	x51	2.000000e-01
x26	1.222222e-01	x52	2.222222e-02

표 2 Mushroom 데이터셋의 ReliefF 실행 결과(계속)

특징변수	Relief결과	특징변수	Relief결과
x52	2.222222e-02	x73	0.000000e+00
x53	2.333333e-01	x74	2.222222e-02
x54	6.666667e-02	x75	2.222222e-02
x55	1.000000e-01	x76	0.000000e+00
x56	2.000000e-01	x77	4.333333e-01
x57	2.888889e-01	x78	2.333333e-01
x58	1.111111e-01	x79	2.000000e-01
x59	8.888889e-02	x80	0.000000e+00
x60	1.111111e-02	x81	1.888889e-01
x61	1.222222e-01	x82	1.000000e-01
x62	0.000000e+00	x83	0.000000e+00
x63	0.000000e+00	x84	4.555556e-01
x64	3.777778e-01	x85	3.333333e-02
x65	1.111111e-01	x86	6.666667e-02
x66	1.666667e-01	x87	1.777778e-01
x67	6.666667e-02	x88	5.555556e-02
x68	7.777778e-02	x89	1.555556e-01
x69	0.000000e+00	x90	2.000000e-01
x70	0.000000e+00	x91	1.666667e-01
x71	0.000000e+00	x92	2.222222e-02
x72	0.000000e+00		

```

> subset <- cutoff.k(weights, 5)
> f <- as.simple.formula(subset, "Species")
> print(f)
Species ~ x24 + x43 + x44 + x19 + x20

```


감사의 글

대학원 생활을 마무리하며 지난 시간들을 돌이켜보니 많은 아쉬움과 후회가 남습니다. 항상 주변에서 저에게 힘을 주시고 방향을 잡아주셨던 많은 분들께 감사의 말씀을 전하고자 합니다.

먼저 본 논문이 완성되기까지 세심한 지도와 많은 격려로 이끌어 주신 김재환 교수님께 진심으로 감사드립니다. 또한 논문 심사 과정에서 아낌없는 지도로 많은 가르침을 주신 박찬근 교수님, 김익성 교수님께 감사드리며 배재국 교수님, 장길웅 교수님, 홍정희 교수님, 손미정교수님, 조교님께도 그동안의 지도에 대해 감사드립니다.

마지막으로 항상 사랑으로 키워주시고 부족한 자식을 믿어주신 부모님께 감사의 말씀을 드립니다. 언제나 제 편이 되어 힘을 주시고 바르게 생각하고 행동할 수 있도록 가르쳐주신 부모님께 누가 되지 않는 아들이 되기 위해 더욱 성실하게 살도록 노력하겠습니다.

