Journal of The Institute of Electronics and Information Engineers Vol.52, NO.10, October 2015 http://dx.doi.org/10.5573/ieie.2015.52.10.073
ISSN 2287-5026(Print) / ISSN 2288-159X(Online)

논문 2015-52-10-9

최근접 이웃 규칙 기반 프로토타입 선택과 편의-분산을 이용한 성능 평가

(Nearest-neighbor Rule based Prototype Selection Method and Performance Evaluation using Bias-Variance Analysis)

심 세 용*, 황 두 성*

(Se-Yong Shim and Doo-Sung Hwang©)

요 약

이 논문은 프로토타입 선택 방법을 제안하고, 편의-분산 분해를 이용하여 최근접 이웃 알고리즘과 프로토타입 기반 분류 학습의 일반화 성능 비교 평가에 있다. 제안하는 프로토타입 분류기는 클래스 영역 내에서 가변 반지름을 이용한 다차원 구를 정의하고, 적은 수의 프로토타입으로 구성된 새로운 훈련 데이터 집합을 생성한다. 최근접 이웃 분류기는 새 훈련 집합을 이용하여 테스트 데이터의 클래스를 예측한다. 평균 기대 오류의 편의와 분산 요소를 분해하여 최근접 이웃 규칙, 베이지안 분류기, 고정 반지름을 이용한 프로토타입 선택 방법, 제안하는 프로토타입 선택 방법의 일반화 성능을 비교한다. 실험에서 제안하는 프로토타입 분류기의 편의-분산 변화 추세는 모든 훈련 데이터를 사용하는 최근접 이웃 알고리즘과 비슷한 편의-분산 추세를 보였으며, 프로토타입 선택 비율은 전체 데이터의 평균 약 27.0% 이하로 나타났다.

Abstract

The paper proposes a prototype selection method and evaluates the generalization performance of standard algorithms and prototype based classification learning. The proposed prototype classifier defines multidimensional spheres with variable radii within class areas and generates a small set of training data. The nearest-neighbor classifier uses the new training set for predicting the class of test data. By decomposing bias and variance of the mean expected error value, we compare the generalization errors of k-nearest neighbor, Bayesian classifier, prototype selection using fixed radius and the proposed prototype selection method. In experiments, the bias-variance changing trends of the proposed prototype classifier are similar to those of nearest neighbor classifiers with all training data and the prototype selection rates are under 27.0% on average.

Keywords : 최근접 이웃 규칙, 프로토타입 선택, 그리디 알고리즘, 편의-분산 분해

Ⅰ. 서 론

최근접 이웃 규칙(nearest-neighbor rule)은 테스트 데이터의 클래스를 가장 근접한 훈련 데이터의 클래스로 예측하는 알고리즘으로써 구현이 단순하나, 높은 활용도를 갖는 학습 방법이다[1]. 그러나 대량의 데이터로 구성된 학습 데이터의 처리는 데이터의 저장 공간, 데이터 간의 유사도 계산 시간, 정렬 시간 등이 급격히 증가하게 된다[2]. 분류 학습에서 프로토타입(prototype)은 훈련 데이터의 클래스 내 데이터를 대표(represent) 또는 포함(cover)하는 데이터이다. 프로토타입 선택 알고리즘(prototype selection algorithm)을 사용하여 훈련 데이터의 클래스 부분 영역을 포함할 수 있는 적은 수

* 정회원, 단국대학교 컴퓨터과학과
(Dept. of Computer Science, Dankook University)
© Corresponding Author(E-mail: dshwang@dankook.ac.kr)
Received ; April 17, 2015 Revised ; June 17, 2015
Accepted ; September 30, 2015

의 프로토타입들을 선택하여 새로운 훈련 데이터로 이용한다. 그러므로 프로토타입 선택 방법은 최근접 이웃 알고리즘의 단점인 공간과 시간의 복잡도를 줄이는데 활용되고 있다[3].

분류 학습 모델의 성능은 학습 알고리즘과 테스트 데이터에 대한 오류율(error rate) 또는 정확율(accuracy rate)을 측정하여 평가하는 것이 일반적이다. 그러나 모델 파라미터 또는 준비된 훈련 데이터의 분포 등에 일반화 성능이 의존된다. 학습 모델의 평균 기대 오류(mean expected error)의 노이즈(class noise), 편의(bias), 분산(variance) 분해 평가가 지도 학습 모델의 평가 척도로 제안되었다[4~6]. 최적의 학습 모델은 낮은 편의와 낮은 분산이 측정된 모델로 선정한다. 편의-분산을 이용한 분석은 모델 복잡도 조절이 가능한 샘플링(sampling), k-교차검증(k-fold cross-validation), 부트스트래핑(boostrapping) 훈련 전략 등으로부터 새로운 학습 데이터를 생성시켜 테스트 데이터의 예측율을 평가한다.

본 논문에서는 새로운 프로토타입 선택 방법을 제안하고, 편의-분산 분해를 이용한 프로토타입 기반 학습(prototype based learning)의 일반화 성능을 분석하는데 있다. 제안하는 프로토타입 선택 방법은 훈련 데이터로부터 선택된 프로토타입으로 구성되는 새로운 훈련 집합을 구성하여, 표준 학습 알고리즘을 이용하여 테스트 데이터의 클래스를 예측한다. 본 논문에서 테스트 데이터의 클래스는 k-최근접 이웃 규칙에 따라 프로토타입의 클래스로 예측한다. 전체 훈련 데이터를 이용한 최근접 이웃 학습과 프로토타입 선택 기반 최근접 이웃 학습의 편의-분산 분석은 최근접 이웃 규칙의 모델 파라미터 k에 따라 측정한다.

이 논문의 구성은 다음과 같다. Ⅱ장에서는 관련 연구에 대해서 살펴보고, Ⅲ장서는 최근접 이웃 규칙을 이용한 제안하는 프로토타입 선택 방법을 기술한다. Ⅳ장에서는 부트스트래핑 훈련 전략을 이용한 평균 기대 오류값의 편의-분산 분해 분석 식을 유도한다. Ⅴ장에서는 알려진 분류 학습 문제에 대한 실험 결과를 보이고, Bien 등의 프로토타입 선택 방법[7], k-최근접 이웃 알고리즘, 베이지안(Bayesian) 알고리즘 등과 제안하는 프로토타입 선택 알고리즘과의 실험 결과를 비교 토의한다. 마지막으로 Ⅵ장에서는 편의-분산 분석에 따른 프로토타입 학습 전략의 문제점과 개선 방향에 대해서 논의한다.

Ⅱ. 관련 연구

프로토타입 선택 알고리즘은 클래스 경계에 위치한 프로토타입 선택, 클래스 영역을 대표하는 프로토타입의 선택, 그리고 클래스를 대표하는 새로운 데이터의 생성 방법 등이 있다. 클래스 경계에 위치한 프로토타입 선택 알고리즘은 Tomek link를 이용한 클래스 분리 경계에 위치한 학습 데이터들로 구성된 새로운 학습 데이터를 생성시켜 분류 예측을 수행하는 프로토타입 선택 알고리즘이 제안되었다[8~9]. 이 프로토타입 선택 알고리즘은 분리 경계 영역에 위치한 데이터들로 구성된 학습데이터를 선별하여, 이미 선택된 데이터, 클래스와 거리 관계를 분석한 정보를 이용하여 프로토타입 집합에 추가할 것인지 여부를 결정한다. 클래스 영역을 대표하는 프로토타입의 선택 방법은 훈련 데이터를 중심으로 상수 거리에 위치한 학습 데이터를 포함하는 영역을 대표하는 프로토타입을 선택한다[7, 10~11]. 또한 클래스를 대표하는 새로운 데이터의 생성 방법은 마할라노비스 거리(Mahalanobis distance)를 이용하여 학습 데이터 분산 구조를 고려한 새로운 프로토타입 선택 방법 제안되었다[12].

Marchette는 동일한 클래스들만으로 구성시킬 수 있는 반지름 계산에 최근접 이웃 규칙을 이용하였다. 최단 거리에 위치한 상이한 클래스의 데이터까지 거리를 계산하여 선택된 프로토타입이 커버할 클래스 영역으로 간주하였으며, 임의의 선택에 따라 모든 학습데이터를 포함시키는 프로토타입 집합을 구성시켰다[10]. Younsi 등은 프로토타입 영역 내 포함되는 동일 클래스 데이터 수를 고려하며 잠재적 분류 경계면에 위치한 잡음 데이터를 조절시켰다[11].

Bien 등은 고정 상수 반지름을 이용하여 프로토타입이 대표하는 데이터 영역을 구들로 나누고, 가능한 모든 학습 데이터를 포함하는 소수의 프로토타입을 선택하는 그리디 알고리즘(greedy algorithm)을 제안하였다[7]. 프로토타입 선택 문제를 집합 덮개 최적화 문제로 정형화시켰으며, 독립된 클래스마다 프로토타입을 선택하는 알고리즘을 제안하였다. 그러나 잠재적 프로토타입이 포함하는 데이터 집합은 상이한 클래스에 속한 데이터들도 포함될 수 있으며 사전 실험을 이용하여 구의

반지름을 선택해야하는 단점이 있다.

기 연구된 방법은 클래스 영역 내 위치한 데이터 간의 유사도를 계산하여 조절된 상수 거리 내에 포함된 데이터로 구성되는 집합들은 다차원 공간의 구(sphere)로 구성되며 클래스 영역을 분할시킨다. 대표 학습 데이터 선택 방법은 상수 반지름 거리 내에 데이터를 가장 많이 포함하는 데이터가 프로토타입으로 우선 선별되는 그리디 알고리즘 또는 경험적 알고리즘(heuristic algorithm)으로 설계하였다.

Ditterich 등은 학습모델의 평균 기대 오류 값에서 편의를 차감하여 분산을 계산하는 모델의 평가를 제안하였다[13]. 제안 평가 식은 편의 값이 크면 음수의 분산 값을 갖는다. 이 문제점을 해결하기 위해 Kohavi 등은 0/1 손실함수을 이용한 확장된 베이지안 분포(extended Bayesian distribution)을 가정하여 평균 오류 제곱식(mean squared error)으로부터 잡음, 편의, 분산을 유도하였고 최근접 이웃 학습을 이용하여 테스트를 수행하였다[5]. Domingos는 주 예측 값(main prediction)과 Bayes 평가 가정 하에 제곱 손실(squared loss) 함수, 절대 손실(absolute loss) 함수, 0/1 손실 함수 등에 대해 통합된 클래스 잡음, 편의, 분산 등의 분리 식을 증명하였으며, 부트스트랩핑 훈련 전략 하에서 최근접 이웃 학습과 결정 트리(decision tree)에서 테스트를 수행하였다. 분산은 편의(biased) 분산에서 비편의(unbiased) 분산이 차감되었다. 수행된 편의-분산 분석으로부터 단순한 학습모델의 경우 편의가 높게 나타나고, 복잡한 모델에서는 과적합(overfitting) 학습으로 인하여 분산이 높게 나타나는 경향을 보고하였다. Domingos는 Dietterich 등[13]의 분산 값이 0보다 작아지는 경우를 설명하였고, 편의(biased) 분산의 차감으로 인해 기대 오류율이 낮아지는 현상이 설명되었다[6].

Ⅲ. 실 험

주어진 분류 문제 $\chi=\{(x_i,y_i)\,|\,i=1,\cdots,n\}$ 의 x_i는 d-차원의 벡터($x_i\in R^d$)이며 $y_i\in\{1,\cdots,C\}$ 이다. χ는 C개의 훈련 데이터 집합으로 구성되어 $\chi=\{\chi^1\cup\chi^2\cup\cdots\cup\chi^C\}$ 가 되며 $\chi^c=\{(x_i,c)\,|\,i=1,\cdots,n_c\}$ 는 클래스 c의 훈련 데이터이다($n=\sum_{c=1}^{C}n_c$). 제안하는 프로토타입 선택 방법은 패턴 분류 문제 χ로부터 각 클래스 내 데이터를 대표할 수 있는 적은 수로 구성되는 데이터 집합인 프로토타입 $P=\cup_{l=1}^{C}P^l$을 선택한다.

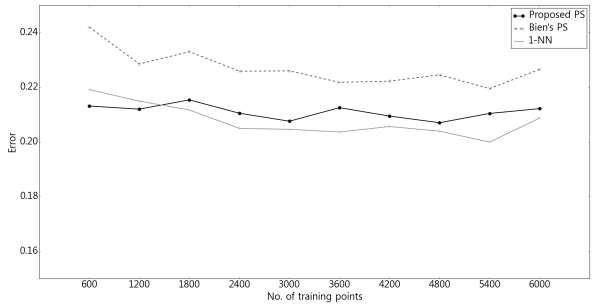
분류 문제의 각 데이터가 대표하는 같은 클래스 내 데이터의 부분 집합은 유클리디안(Euclidean) 거리로 계산한다. 데이터 $(x,c)\in\chi^c$가 대표하는 거리 r_x내에 위치한 c 클래스 데이터 집합 $S(x)=\{z\,|d(x,z)<r_x,\ l(z)=c\}$ 이다. 거리 r_x는 훈련 데이터와 거리를 구하여 동일 클래스 내 가장 큰 거리 값 r_1과 다른 클래스 중 가장 작은 거리 값 r_2을 갖는 거리의 중간 값 $r_x=(r_1+r_2)/2$으로 결정한다. 고정 반지름을 이용하여 잠재적 프로토타입이 포함하는 영역을 결정하는 Bien 등 방법[7]과 비교 시 r_x는 동일 클래스 데이터만을 대표하는 영역을 결정하며 훈련 전 r_x의 결정이 필요하지 않다. Marchette 등[10]과 Younsi 등 방법[11]과 비교 시 분류 경계에 위치한 서로 다른 클래스에 속한 두 훈련 데이터의 중간 값의 반지름으로 결정한다. 그림 1은 C 클래스 분류문제 χ의 학습 데이터에 대해 각 데이터의 프로토타입 집합 P를 계산하는 그리디 알고리즘이다. S는 훈련 데이터가 커버하는 동일 클래스의 집합이다.

알고리즘의 출력은 클래스 별 선택된 프로토타입 집합 $P=\cup_{l=1}^{C}P^l$이며 $|P^c|<<|\chi^c|$이다.

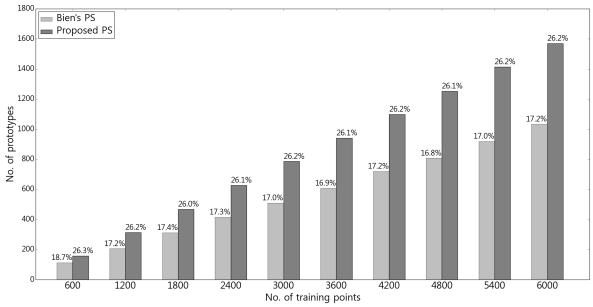
$$\triangle obj(x_j)=|S(x_j)\setminus\cup_{x_i\in P}S(x_i)| \tag{1}$$

```
procedure select_prototype (S, χ, C)
// S(x) = {z | d(x_j, z) ≤ r_x and l(x_j) = l(z)}
// χ = {(x_i, c) | i = 1, ..., n and c ∈ {1, ..., C}}
// C: 클래스 수
// P^c, c = 1,2, ..., C
P = Φ
for c = 1 to C do
  P^c = Φ;  χ^c = { x_i | (x_i, c) ∈ χ }
  while △obj (x_j) > 0 do
    x_j = argmax_{x_i ∈ χ^c} △obj (x_i)
    P^c = P^c ∪ {x_j}
  end while
  P = P ∪ P^c
end for
return P
```

그림 1. 프로토타입 선택 알고리즘
Fig. 1. Prototype selection algorithm.



(a) 테스트 오류율 비교
(a) The comparison of test error rates



(a) 테스트 오류율 비교
(a) The comparison of test error rates

그림 2. 훈련 데이터 수의 변화에 따른 에러와 프로토타입 수
Fig. 2. Error and prototype size with the size of training points.

수식 (1)의 $\triangle obj\ (x_j)$는 $x_j \in \chi^c$가 프로토타입으로 선택 시 클래스 c영역을 대표할 수 있는 데이터의 크기이다. 잠재적 프로토타입은 $\triangle obj\ (x_j)$를 최대로 하는 (x_j, c)이다.

그림 2는 4개의 클래스 분류 문제를 $N(\mu, I)$로부터 발생시켜 훈련 데이터 수의 변화에 따른 에러와 선택 프로토타입 수의 변화를 보여주고 있다. μ는 (-1, -1), (-1, +1), (+1, -1), (+1, +1)이다. 제안하는 프로토타입을 사용한 방법은 훈련 데이터 전체를 사용하는 1-최근접 이웃 알고리즘의 에러와 유사한 결과를 나타낸다. 그러나 고정 반지름 기반 프로토타입 선택의 에러는 제안하는 프로토타입을 사용한방법보다 높게 나타났다. 선택된 프로토타입 수는 훈련 데이터가 증가하지만 선택 비율은 비슷하게 나타났다. 이 실험은 제안하는 프로토타입 선택 방법은 일반화 성능을 유지하면서 훈련 데이터를 축소하는 효과를 보이고 있다.

Ⅳ. 편의-분산 분석

제안하는 편의-분산 분석은 Ditterich 등의 방법[13]과 동일하게 클래스 분포에 대한 가정을 하지 않고, 0/1 손실함수를 이용한 평균 기대 오류율를 정의한다. Domingos 분석[6]에서는 다중 분류 문제의 평균 기대 오류율을 정의하며, 중복을 허용하는 다중 샘플링 훈련 전략에서 주 예측 클래스의 정의를 이용한 분산의 구성 요소를 유도한다. 평균 기대 오류율은 편의와 분산으로 이루어진다[13]. 편의는 예측값과 사실 클래스가 다를 경우 발생된다.

표 1. 부트스트랩핑 전략으로부터 발생된 m개 훈련 집합의 예측
Table 1. Prediction results over m training sets under boostrapping learning.

χ_{te}		D_1	D_2	$\cdots$	D_m	y^*
x_1	t_1	$y_1^{(1)}$	$y_1^{(2)}$	$\cdots$	$y_1^{(m)}$	y_1^*
				$\cdots$		
x_i	t_i	$y_i^{(1)}$	$y_i^{(2)}$	$\cdots$	$y_i^{(m)}$	y_i^*
				$\cdots$		
x_n	t_n	$y_n^{(1)}$	$y_n^{(2)}$	$\cdots$	$y_n^{(m)}$	y_n^*

분산은 편의 분산(biased variance)과 비편의 분산(unbiased variance)으로 구성된다. 기대 오류율의 예측 클래스(predicted class)가 사실 클래스(true class)와 다르면 편의 값이 증가하나, 예측 클래스가 사실 클래스와 동일한 경우 편의 분산이 증가한다. 반대로 주 클래스가 사실 클래스와 같으면 편의의 증가는 나타나지 않으나, 예측 클래스가 사실 클래스와 다르면 비편의 분산이 차감되어야 한다. 그러므로 비편의 분산은 기대 오류율에 차감되는 효과가 있으며, 편의 분산은 가산되는 효과가 있다.

표 1은 부트스트랩 전략으로부터 발생된 m개의 데이터 집합의 훈련 모델로부터 테스트 데이터의 예측 분류 값을 나타낸다. $y_i^{(j)}$는 i번째 테스트의 훈련 모델 j번째의 예측 값으로 0/1-손실 함수로 다음과 같다.

$$1(y_i^{(j)} = t_i) = \begin{cases} 1 & \text{if } y_i^{(j)} \neq t_i \\ 0 & Otherwise \end{cases} \quad (2)$$

여기서 t_i는 i번째 훈련 데이터의 사실 클래스이다.

Domingos와 같이 m개의 데이터 집합의 훈련 모델 전체에서 나타난 예측 값 중에서 가장 빈번한 클래스 예측값(main prediction) $y_i^* = \arg\max_j y_i^{(j)}$라 정의하자. 데이터 집합 $D_i, i=\{1,2,\cdots,m\}$ 에 대한 $(x_i,t_i)\in X_{te}$의 오류 값은 다음과 같다.

$$\begin{aligned} L[y_i^{(j)} \neq y_i^* | (x_i,t_i)] &= \frac{1}{m}\sum_{j=1}^{m} 1(y_i^{(j)} \neq y_i^*) \\ &= 0 + \frac{1}{m}\sum_{j=1}^{m} 1(y_i^{(j)} \neq y_i^*) \\ &= 1(y_i^* \neq t_i) + \frac{1}{m}\sum_{j=1}^{m} 1(y_i^{(j)} \neq y_i^*) \end{aligned} \quad (3)$$

$$\begin{aligned} L[y_i^{(j)} = y_i^* | (x_i,t_i)] &= 1 - \frac{1}{m}\sum_{j=1}^{m} 1(y_i^{(j)} = y_i^*) \\ &= 1(y_i^* \neq t_i) - \frac{1}{m}\sum_{j=1}^{m} 1(y_i^{(j)} = y_i^*) \end{aligned} \quad (4)$$

$L[y_i^{(j)} \neq y_i^* | (x_i,t_i)]$에서는 앞에서 제시된 조건에 따라 $y_i^{(j)} = t_i$에서는 오류로 나타나지 않으며 0으로 계산되어 수식 (3)와 같다. $L[y_i^{(j)} = y_i^* | (x_i,t_i)]$은 조건에 따라 $y_i^{(j)} = t_i$이면 예측 오류는 발생하지 않으므로 m개의 테스트 데이터에 대한 분류 예측 값은 수식 (4)이 된다. 이 때 $y_i^{(j)} \neq t_i$인 x_i은 예측 오류를 발생시키므로 전체 에러에 포함되어야 한다. 테스트 데이터에 대한 전체 오류는 수식 (3)와 (4)의 합이다. n 개의 테스트 데이터에 대한 전체 오류식 E는 다음과 같다.

$$\begin{aligned} E &= \frac{1}{n}\sum_{i=1}^{n}\left\{L[y_i^{(j)} \neq y_i^* | (x_i,t_i)] + L[y_i^{(j)} = y_i^* | (x_i,t_i)]\right\} \\ &= \frac{1}{n}\sum_{i=1}^{n} 1(y_i^* \neq t_i) \quad \cdots\ B\,(\text{편의}) \\ &\quad + \frac{1}{nm}\sum_{i=1}^{n}\sum_{j=1}^{m} 1(y_i^{(j)} \neq y_i^*) \quad \cdots\ V_u\,(\text{비편의 분산}) \\ &\quad - \frac{1}{nm}\sum_{i=1}^{n}\sum_{j=1}^{m} 1(y_i^{(j)} = y_i^*) \quad \cdots\ V_b\,(\text{편의 분산}) \end{aligned} \quad (5)$$

수식 (5)의 첫 번째 항은 주요 예측 값에 대한 오류로 편의 B이며, 두 번째 항은 주 예측 클래스는 같으나 오류가 없는 경우의 분산(unbiased variance) V_u, 그리고 마지막 항은 오류가 있으나 경우의 분산(biased variance) V_b이다. 그러므로 $V_b > V_u$면 E의 분산은 음수 값을 갖는다. 유도된 B, V_u, V_b는 클래스 분포를 가정하지 않는 0/1 손실함수의 경우 Domingos의 식과 동일하다.

V. 실험 결과

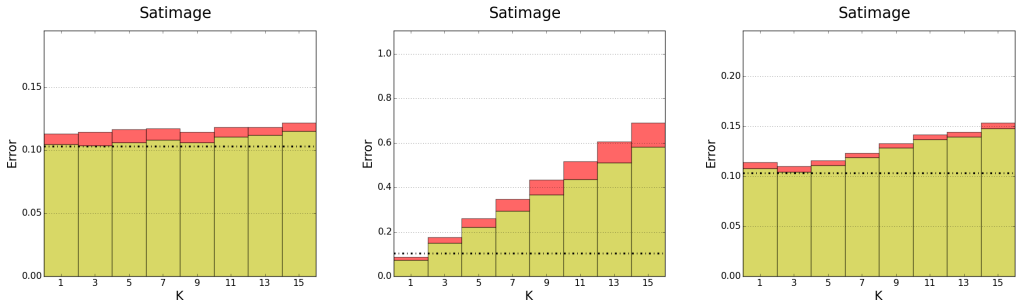
프로토타입 선택 방법의 일반화 성능 비교는 전체 훈련 데이터를 사용한 최근접 이웃 알고리즘과 베이지안 학습, 고정 반지름을 이용한 Bien 방법[7], 그리고 제안하는 프로토타입 선택 방법을 비교한다. 선택된 벤치마크 문제는 표 2에 제시되었다[14~16]. 고정된 반지름 기반 프로토타입 선택에서 사용된 반지름 값은 사전 실험을 통해 각 분류 문제마다 적절한 반지름을 선택하였다. 각 실험은 데이터를 부트스트랩 방법으로 훈련 데이터와 테스트 데이터를 나누어 실험하였고 그 실험을 5번씩 실행한 결과에 대해서 평균을 계산하였다. 선택된 문제에서 k를 1부터 15사이에 홀수의 수만 선택하여 측정된 평균 기대 오류율의 기대되는 값, 바이어스 그리고 분산을 계산하였다. 각 문제의 측정된 편의와 분산을 중첩 막대 그래프(stacked bar graph)로 제시하였다. 막대의 아래 부분은 편의이고 나머지는 분산을 나타낸다. 최적의 에러(optimal error)를 나타내는 베이지안(Bayesian) 알고리즘의 측정된 값을 점선으로 표현하였다.

실험 결과, 각 문제마다 k값을 증가시키며 측정된 바이어스와 분산으로 표현한 결과를 보면 세 가지 경향을 보였다. k값이 증가할 때 에러값이 낮아지는 경향, 에러값이 높아지는 경향과 중간 k값에서 최소 에러값이 나타나는 추세이다.

그림 3은 편의-분산 변화의 세 가지 경향의 예이다. Satimage, Twonorm, 그리고 Car 문제에 대해 최근접 이웃 알고리즘, 고정 반지름을 사용한 Bien 등의 방법, 그리고 제안하는 방법을 나타내었다. 그림 3-1의 Satimage는 k값이 증가 시 에러값이 높아지는 변화를 나타냈으며, Pendigits, Segment, USPS, Vowel, Wine 등에서 유사한 패턴을 보였다. 그림 3-2의 Twonorm은 k값이 증가할 때 에러값이 낮아지는 경향을 보였으며 DNA도 비슷한 경향을 보였다. 마지막으로 그림 3-3의 Car는 중간 k값에서 최소 에러값을 보이는 경향을 나타내었고 Heart 등에서 유사하게 나타났다. 그러나 고정 반지름을 사용하는 Bien 등[7]의 선택 방법은 k값이 1인 경우에서만 최소 오류값을 보였으며, k가 증가하면서 오류값도 증가하였다. 이는 클래스 프로토타입 내 상이한 클래스의 데이터를 포함시키기 때문이다.

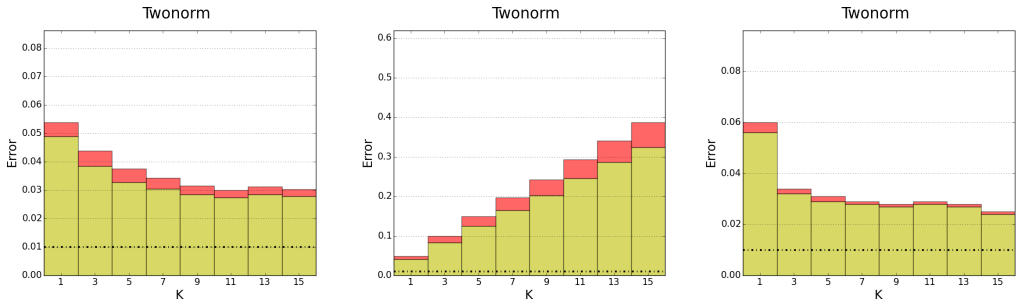
표 2. 선택된 벤치마크 분류 문제
Table 2. Selected benchmark classification problems.

데이터	크기	속성 수	Numeric 속성 수	Nominal 속성 수	클래스 수	고정 반지름
Car	1,728	6	0	6	4	1.2
DNA	2,000	180	0	180	3	0.3
Heart	270	13	13	0	2	1.0
Pendigits	7,494	16	16	0	10	1.0
Satimage	6,435	36	36	0	7	0.3
Segment	2,310	19	19	0	7	0.3
Twonorm	7,400	20	20	0	2	2.7
USPS	9,298	256	256	0	10	1.0
Vowel	990	13	13	0	11	0.3
Wine	178	13	13	0	3	1.0



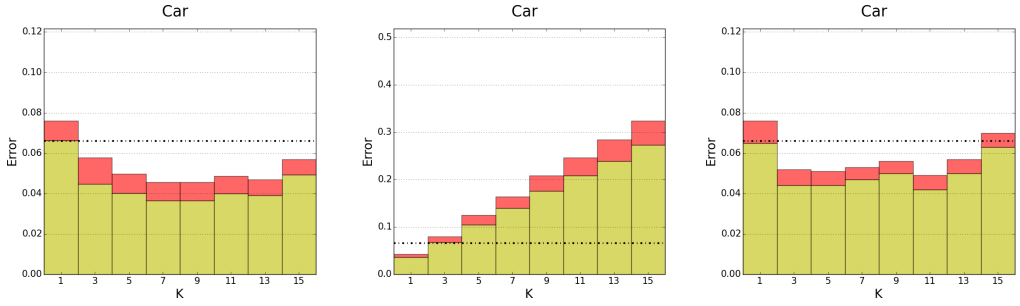
(a) 최근접 이웃 알고리즘 (b) 고정 반지름 방법 (c) 제안하는 방법

그림 3-1. Satimage의 경우



(a) 최근접 이웃 알고리즘 (b) 고정 반지름 방법 (c) 제안하는 방법

그림 3-2. Twonorm의 경우



(a) 최근접 이웃 알고리즘 (b) 고정 반지름 방법 (c) 제안하는 방법

그림 3-3. Car의 경우

그림 3. 편의-분산 분석 결과에서 변화 추세
Fig. 3. The changing trends in bias-variance analysis.

표 3. 선택 문제에 대한 최적의 일반화 성능을 나타내는 모델
Table 3. Model representing optimal generalization performance for the selected problems.

데이터	베이지안	최근접 이웃				고정 반지름 기반 프로토타입 선택				제안하는 프로토타입 선택			
		k	에러	편의	분산	k	에러	편의	분산	k	에러	편의	분산
Car	0.066	7	0.05	0.036	0.009	1	0.04	0.033	0.007	11	0.05	0.042	0.007
DNA	0.045	15	0.23	0.202	0.026	1	0.08	0.068	0.010	15	0.17	0.163	0.010
Heart	0.088	13	0.16	0.152	0.012	1	0.06	0.046	0.009	9	0.18	0.176	0.006
Pendigits	0.061	1	0.01	0.007	0.001	1	0.10	0.082	0.016	1	0.02	0.013	0.002
Satimage	0.103	5	0.11	0.105	0.010	1	0.09	0.073	0.013	3	0.11	0.103	0.006
Segment	0.108	1	0.04	0.041	0.004	1	0.09	0.074	0.015	1	0.05	0.046	0.005
Twonorm	0.010	11	0.03	0.027	0.003	1	0.05	0.041	0.008	15	0.03	0.024	0.001
USPS	0.109	5	0.03	0.026	0.001	1	0.10	0.081	0.016	1	0.04	0.035	0.004
Vowel	0.188	1	0.06	0.048	0.014	1	0.11	0.091	0.017	1	0.04	0.033	0.007
Wine	0.011	11	0.04	0.037	0.004	1	0.08	0.069	0.011	1	0.03	0.030	0.004

표 4. 선택 문제에 대한 프로토타입 선택 시간과 프로토타입 수
Table 4. Time of prototype selection and number of prototype for the selected problems.

데이터	데이터 크기	프로토타입 선택 시간(초)		프로토타입 수	
		고정 반지름 기반 프로토타입 선택	제안하는 프로토타입 선택	고정 반지름 기반 프로토타입 선택	제안하는 프로토타입 선택
Car	1,728	16.1	1.6	4.2%	6.7%
DNA	2,000	22.9	1.5	2.6%	73.8%
Heart	270	0.2	0.2	3.9%	25.1%
Pendigits	7,494	656.8	68.6	9.7%	5.8%
Satimage	6,435	164.4	18.4	0.4%	24.8%
Segment	2,310	31.8	7.2	10.5%	12.1%
Twonorm	7,400	627.6	51.7	4.4%	65.5%
USPS	9,298	609.9	48.7	0.9%	21.8%
Vowel	990	1.2	0.2	9.8%	15.0%
Wine	178	0.2	0.1	10.1%	25.8%
프로토타입 수 평균				5.65%	27.64%

표 4는 선택된 벤치마크 문제에 대한 프로토타입 선택 시간과 선택된 프로토타입의 수이다. 선택 시간에서는 고정 반지름 기반 방법이 제안하는 방법보다 높은 시간을 나타냈다. 이는 고정 반지름 기반 방법에서 프로토타입 구성 시 프로토타입 내 포함되는 상이한 클래스들의 데이터들을 포함할 수 있어 프로토타입 선택 과정에서 적절한 프로토타입들을 선택하는 시간이 많이 걸린다고 분석된다. 프로토타입 수에서는 제안하는 방법에서 Pendigits을 제외하고 대부분의 문제에서 프로토타입 수가 높게 나타났고 이는 주어진 훈련 데이터의 분포를 고려하여 동일 클래스 데이터들만으로 구성된 프로토타입을 구성하기 때문이다. 또한 분류 경계면에 위치한 데이터는 프로토타입으로 선택될 가능성이 높기 때문이다.

전체 데이터를 사용한 최근접 이웃 알고리즘과 제안하는 프로토타입을 사용한 최근접 이웃 알고리즘을 비교해보면 Car, DNA, Satimage, Twonorm, Vowel, 그리고 Wine에서 제안하는 알고리즘의 분류 예측율이 높았다. 그리고 제안하는 프로토타입을 사용한 최근접 이웃 알고리즘의 분산이 전체 데이터를 사용한 최근접 이웃 알고리즘보다 대부분의 문제에서 낮게 나타났다. 이러한 이유는 제안 방법은 데이터의 분포를 고려한 프로토타입을 선택하기 때문으로 분석된다.

VI. 결 론

본 논문에서는 최근접 이웃 규칙을 이용하는 프로토타입 선택 방법을 제안하고 표준 학습 모델과 일반화 성능을 비교하였다. 클래스 분포를 반영하는 제안하는 프로토타입 분류기가 모든 훈련 데이터를 이용한 최근접 이웃 규칙의 편의-분산 분석과 비슷한 학습 경향을 보여 효과적인 학습 전략이 될 수 있었다. 그러나 고정

반지름을 이용하는 프로토타입 선택 알고리즘에 비해 선택된 프로토타입 수의 비율이 높다는 단점이 있다. 프로토타입 선택 시간은 적게 분석되었다..

실험 분석에서 k값의 증가에 따라 3가지 유형의 평균 기대 오류율의 추세가 나타났다. 제안하는 프로토타입을 사용한 학습에서는 전체 데이터를 사용한 학습보다 낮은 분산이 나타나, 과적합 학습의 가능성이 낮아 높은 일반화 성능을 나타낼 수 있는 학습 모델 선택이 가능하다. k의 변화에 따라 오류율의 편의-분산 분해는 학습데이터의 변화에 따른 편의와 분산의 변화 추세 분석, 모델 파라미터 k 값 설정, 과적합 분석, 훈련 프로토타입 집합 선택 등에 이용될 수 있다.

REFERENCES

[1] X. Wu et al., "*The top ten algorithms in data mining,*" CRC Press, 2009.

[2] T. Hastie, R. Tibshirani, and J. Friedman, "*The Elements of Statistical Learning: Data Mining*," Inference, and Prediction, Springer Series in Statistics, 2001.

[3] J. Arturo Olvera-Lopez, J. Ariel Carrasco-Ochoa, J. Francisco Martinez Trinidad, and J. Kittler, "*A review of instance selection methods*," Artif. Intell. Rev Vol. 34, No. 2, pp. 133-143, Aug. 2010.

[4] P. Flach, "*Machine Learning, The Art and Science of Algorithms that Make Sense of Data,*" Cambridge University Press, 2012.

[5] R. Kohavi, D. H. Wolpert, "*Bias Plus Variance Decomposition for Zero-One Loss Functions*," In Proceedings of the Thirteenth International Conference on Machine Learning, 275-283, 1996.

[6] P. Domingos, "*A United Bias-Variance Decomposition for Zero-One and Squared Loss*," In Proceedings of the Seventeenth National Conference on Artificial Intelligence, 231-238, 2000.

[7] J. Bien and R. Tibshirani, "*Prototype selection for interpretable classification*," The Annuals of Applied Statistics Vol. 5, No. 4, pp. 2403-2424, Dec, 2011.

[8] D. S. Hwang, "*Performance Improvement of Nearest-neighbor Classification Learning through Prototype Selection*," Journal of The Institute of Electronics Engineers of Korea, Vol. 49(2)-CI, pp. 53-60, Mar. 2012.

[9] F. Angiulli, "*Fast Nearest Neighbor Condensation for Large Data Sets Classification*," IEEE Transactions on Knowledge and Data Engineering, Vol. 19, No. 11, pp. 1450-1464, Nov. 2007.

[10] D. Marchette, "*Class cover catch digraphs*," Wiley Interdisciplinary Reviews : Computational Statistics Vol. 2, No. 2, pp. 171-177, Mar. 2010.

[11] R. Younsi, and A. Bagnall, "*An efficient randomised sphere cover classifier*," Int. J. of Data Mining, Modelling and Management, Vol. 4, No. 2, pp.156-171, Jan. 2012.

[12] S. W. Kim, "*Relational Discriminant Analysis Using Prototype Reduction Schemes and Mahalanobis Distances*," Journal of The Institute of Electronics Engineers of Korea, Vol. 43(1)-CI, pp. 9-16, Jan. 2006.

[13] Dietterich, T. G and Kong, E. B., "*Machine learning bias, statistical bias, and statistical variance of decision tree algorithms*," Technical report, Department of Computer Science, Oregon State University, 1995.

[14] UCI Machine Learning Repository, http://archive.ics.uci.edu/ml/

[15] The DELVE Manual, http://www.cs.utoronto.ca/~deve/

[16] Stalog project, http://www1.maths.leed.ac.uk/~charles/statlog/indexdos.html

저 자 소 개

심 세 용(정회원)
2013년 단국대학교 멀티미디어공학과(공학사)
2015년 단국대학교 전자계산학과(공학석사)

<주관심분야 : Machine Learning, Image Processing, Parallel Processing>

황 두 성(정회원)
1986년 충남대학교 계산통계학과 이학사
1990년 충남대학교 계산통계학과 이학석사
2003년 Wayne State University 컴퓨터공학과 공학박사
현재 단국대학교 컴퓨터과학과 부교수
현재 단국대학교 운동의과학과 부교수

<주관심분야 : Machine Learning, Parallel Processing, Semantic Web>