논문 2012-49CI-2-8

프로토타입 선택을 이용한 최근접 분류 학습의 성능 개선

(Performance Improvement of Nearest-neighbor Classification Learning through Prototype Selections)

황 두 성*

(Doosung Hwang)

요 약

최근접 이웃 분류에서 입력 데이터의 클래스는 선택된 근접 학습 데이터들 중에서 가장 빈번한 클래스로 예측된다. 최근접 분류 학습은 학습 단계가 없으나, 준비된 데이터가 모두 예측 분류에 참여하여 일반화 성능이 학습 데이터의 질에 의존된다. 그러므로 학습 데이터가 많아지면 높은 기억 장치 용량과 예측 분류 시 높은 계산 시간이 요구된다. 본 논문에서는 분리 경계면에 위치한 학습 데이터들로 구성된 새로운 학습 데이터를 생성시켜 분류 예측을 수행하는 프로토타입 선택 알고리즘을 제안한다. 제안하는 알고리즘에서는 분리 경계 영역에 위치한 데이터를 Tomek links와 거리를 이용하여 선별하며, 이미 선택된 데이터와 클래스와 거리 관계 분석을 이용하여 프로토타입 집합에 추가 여부를 결정한다. 실험에서 선택된 프로토타입의 수는 원래 학습 데이터에 비해 적은 수의 데이터 집합이 되어 최근접 분류의 적용 시 기억장소의 축소와 빠른 예측 시간을 제공할 수 있다.

Abstract

Nearest-neighbor classification predicts the class of an input data with the most frequent class among the near training data of the input data. Even though nearest-neighbor classification doesn't have a training stage, all of the training data are necessary in a predictive stage and the generalization performance depends on the quality of training data. Therefore, as the training data size increase, a nearest-neighbor classification requires the large amount of memory and the large computation time in prediction. In this paper, we propose a prototype selection algorithm that predicts the class of test data with the new set of prototypes which are near-boundary training data. Based on Tomek links and distance metric, the proposed algorithm selects boundary data and decides whether the selected data is added to the set of prototypes by considering classes and distance relationships. In the experiments, the number of prototypes is much smaller than the size of original training data and we takes advantages of storage reduction and fast prediction in a nearest-neighbor classification.

Keywords : Prototype Selection, Nearest Neighbor Rule, Tomek Link

Ⅰ. 서 론

k개 최근접 이웃 분류(k-nearest-neighbor classification)알고리즘은 입력 데이터의 k개 가까운 학습 패턴들의 클래스 중에서 가장 많이 나타나는 클래스로 분류하는 단순 규칙을 사용한다[1]. k개 최근접 분류 규칙의 적용은 학습 단계가 필요 없는 비 모수 학습을 수행하나, 준비된 데이터가 모두 예측 분류에 참여하므로, 일반화 성능이 준비된 학습 데이터의 질에 따라 다르다. 그러나 데이터마이닝 영역에서 지지벡터 기계와 함께 최근접 알고리즘의 응용은 높이 평가되어 전문가들이 선정한 Top 10개의 주요한 알고리즘으로 인식되고 있다[2]. 준비된 학습 데이터의 수가 비교적 적거나, 차원이 높은 경우, 또는 다중 클래스 분포인 경우 신경망, SVM, 베이지안 알고리즘 등 모수 학습에 대응할 만한 일반화 성능이 보고되었다. 최근접 알고리즘의 적용 시 최적의 k 선택과 거리 측정 방법 등

* 정회원, 단국대학교 컴퓨터과학과
(Dept. of computer science, Dankook University)
접수일자: 2011년2월15일, 수정완료일: 2012년3월2일

에 따라 분류 성능이 달라진다[3]. 그러나 구현이 단순하나 높은 분류 성능을 보장하기 때문에 많은 응용에서 선택되고 있다. 언급된 이러한 문제점과 더불어 높은 기억 장치 용량의 필요하며, 분류 시 많은 계산 시간의 소요, 잡음 패턴 또는 outlier에 대한 낮은 예측율 등은 단점으로 나타났다.

최근 이러한 단점을 극복하려는 새로운 학습 전략 또는 개선된 알고리즘 등이 연구되었다. 샘플링 기반 학습은 데이터 제거 또는 클래스의 분포에 영향이 없는 새로운 데이터의 추가 등 기법을 이용한 클래스 간 균등한 데이터 분포를 이루어 학습 성능의 개선을 도모한다. 그러나 기억 장치와 계산 시간의 축소 등의 효과는 미비하나, 샘플링을 이용한 오 분류가 높은 패턴에 대한 예측율은 높일 수 있다. 프로토타입 선택(prototype selection) 학습은 준비된 학습 패턴으로부터 분류 경계 영역에 위치한 패턴을 선택하여 테스트 패턴의 클래스 예측에 이용한다[3]. 선택된 프로토타입으로 구성된 학습 데이터는 원래의 학습 데이터의 수보다 적은 수의 데이터로 구성된다. 프로토타입 선택 전략의 도입은 낮은 기억 장치 용량과 빠른 계산 시간 등의 장점을 제공하나, 잡음 패턴에 대한 예측율 높이지 못하며 준비된 학습 데이터의 분포가 학습 성능에 영향을 준다. 다양한 거리 계산 방법과 커널 함수(kernel function)의 도입 등도 최근접 알고리즘의 성능을 높일 수 있는 방법으로 보고되었으나 기억 장치와 계산 시간의 축소에 영향은 미비하였다[4].

본 논문에서는 분리 경계에 위치한 학습 데이터들로 구성되는 새로운 학습 집합을 만들어 학습하는 프로토타입 선택 알고리즘을 제안한다. 경계면에 위치한 데이터 쌍은 Tomek links[5]를 이용하여 선별하며 이미 선택된 프로토타입 데이터와 클래스와 거리 관계를 분석하여 프로토타입 집합에 추가 여부를 결정한다. 선택된 프로토타입의 수는 원래 학습 데이터에 비해 적은 수의 데이터 집합이 되어 최근접 규칙 분류 시 테스트 기억 장소의 축소와 빠른 예측 시간을 제공할 수 있다. 2절에서는 선행 연구된 프로토타입 선택 알고리즘을 살펴본다. 3절에서는 제안하는 Tomek links 기반 프로토타입 선택 알고리즘을 제안하고 프로토타입의 선별 방법에 대하여 논의한다. 4절에서는 선택된 벤치 마킹 분류 문제에 대한 제안하는 알고리즘과 최근점 분류 학습의 성능과 비교 결과를 토의한다. 마지막으로 5절에서는 정리와 제안된 알고리즘의 응용 방법 등을 논한다.

Ⅱ. 선행연구

학습 데이터 집합으로부터 선택된 프로토타입 학습 집합의 일반화 예측율이 원래 학습 데이터의 성능과 대등할 때 프로토타입 데이터는 일관된 학습 집합(consistent training set)을 구성 한다. 일관된 학습 집합은 분류 경계 면에 가까운 데이터를 선별하여 생성한다. 분류 경계면에 가까운 데이터의 평가에는 최근접 분류, 거리, 클래스 정보 등이 사용되었으며, 프로토타입 선택 알고리즘은 집합 연산을 이용한 반복적 알고리즘으로 기술되었다. 생성된 일관된 학습 집합은 테스트 데이터의 분류 규칙을 학습하는데 사용되며 원래 학습 데이터보다 적은 수로 구성된다. 일관된 학습 집합의 일반화 성능은 학습 데이터의 축소율, 분류 예측율, 데이터의 예측 시간 등으로 평가되며 축소 전 학습 문제의 성능과 비교되었다. S. García et. al.은 제안된 프로토타입 선택 전략에 대한 분석를 진행하여 알고리즘의 특성과 성능 비교 등을 분석하였다[3].

p개의 데이터로 구성된 분류 문제 $T=\{(x_i, c_i)|i=1,...,p\}$가 n개의 d-차원 학습데이터 $x_i \in R^d$의 집합 TR과 m개의 테스트 집합 TS를 이루며 c_i는 클래스 정보라 하자(p=n+m, T=TR∪TS). 함수 c=nn(x,k,S)는 S의 멤버에 대해 x의 k-최근접 분류를 수행하여 예측 클래스 c를 계산하며, d(x,y)는 x와 y의 Euclidean 거리이다. 일관된 학습 집합을 계산하는 알고리즘은 TR로부터 프로토타입 집합 S을 생성하여 TS에 대해 예측 분류를 수행한다. CNN(condensed nearest neighbor)는 최근접 규칙을 이용하여 데이터 요약을 수행하는 제안된 첫 번째 알고리즘이다[6]. 초기 S는 TR로부터 서로 다른 클래스에 속한 데이터를 임의로 선택하여 초기화한다. 각 $x_i \in$TR에 대해 최근접 분류 $c=nn(x_i,1,S)$를 수행한다. 만약 $c \neq c_i$이면 (x_i,c_i)를 S에 추가하고 $c = c_i$이면 TR의 다른 데이터에 대해 S의 추가 여부를 결정한다. 이 S의 멤버 생성 단계는 TR의 데이터에 대해 오 분류가 발생되지 않을 때까지 반복한다. CNN은 TS의 성능이 TR로부터 선택되는 데이터의 순서에 의존될 가능성 높아 경계면에 위치하지 않은 프로토타입으로 구성되는 S를 계산할 가능성이 높다.

CNN을 개선하는 MCNN(Modified Condensed Nearest Neighbor), GCNN(Generalized Condensed Nearest Neighbor), FCNN(Fast Condensed Nearest Neighbor)등이 연구되었다[3]. MCNN는 오분류된 동일

클래스의 중간 데이터를 S에 새로운 프로토타입으로 삽입하며, GCNN은 S내 동일 클래스와 오분류 클래스와 거리를 계산하여 새로운 프로토타입의 삽입을 결정한다. FCNN(Fast Condensed Nearest Neighbor)는 S는 각 클래스의 중간 값을 계산하여 초기화 후 CNN 단계를 수행한다[3,7]. 각 선택된 프로토타입 xk ∈S의 Voronoi enemy xi ∈TR을 새로운 프로토타입으로 선택한다. 이 알고리즘은 비교 실험에서 일반화 성능도 비교적 높으며 가장 빠른 알고리즘으로 보고되었다.

RNN(Reduced Nearest Neighbor) 알고리즘은 수행은 S=TR 초기화 후 각 xi ∈TR을 제외하고 S만을 가지고 최근접 분류 c=nn(x_i,1,S)를 반복하여 c==c_i이면 (x_i,c_i)를 S에서 삭제한다[8]. 실험에서 RNN으로부터 계산된 S는 일반적으로 CNN이 선택한 요약된 집합의 부분집합이 된다고 보고되었고 TR의 크기가 증가하면서 S을 얻는 계산 비용이 너무 크다는 단점이 있다. RNN을 단점을 개선하는 SNN(Selective Nearest Neighbor), MCS(Minimal Consistent Set) 등이 연구되었다[3].

NPPS(Neighbor Property based Pattern Selection) 알고리즘은 이웃 엔트로피(neighbor entropy), 최근접 규칙과 매치율(match rate)로 부터 이웃 매치 정도를 계산하여 분리 영역 주변에 있는 데이터를 선택한다.[9] NPPS의 성능은 선택된 k와 매치율에 의존된다. 최단거리를 갖은 서로 다른 클래스를 이루는 데이터 쌍은 분리 영역에 위치할 가능성이 높다. 서로 다른 클래스를 이루는 데이터 쌍은 Tomek links를 구성하며 최단거리를 갖는 Tomek links를 Hausdorff 쌍이라 한다[5, 10]. 경계 혹은 노이즈 데이타를 제거 함으로서 학습성능 향상을 위해 Tomek links의 사용이 연구되었다. Cooper et. al.은 Hausdorff 쌍을 이용한 데이터 선택의 효과를 단순 샘플링의 실험과 비교분석을 수행하였다. 데이터 선택 비율에 따라 단순 샘플링의 효과는 단순 샘플링의 Hausdorff 쌍은 1-최근접 규칙에 의해 발견될 수 있으나 선택할 데이터의 적정 수를 결정하는 단계가 요구된다[10].

살펴본 연구들은 대용량 학습데이터의 분석에서 빠른 예측을 위해 경계 영역에 위치한 데이터들로 구성되는 학습데이터 축소 방법을 제안하고 있으며, 제안된 알고리즘들은 데이터 선택에서 최근접 규칙, 거리 행렬, Tomek links 등을 기반으로 데이터 선택을 수행하고, 알고리즘의 기술은 집합 연산을 기반으로 기술되었다. 또한 일반화 성능 평가는 이진 분류 문제에 대해 비교평가를 하고, Bayes, SVM, 신경망 등을 통해 데이터 축소비율에 따른 예측율을 비교하여 우수성을 보였다.

Ⅲ. 프로토타입 선택 알고리즘

제안하는 프로토타입 선택 알고리즘 PSNN(Prototype Selection by k-Nearest Neighbors)은 Tomek link와 최근접 규칙을 이용한다. 이진 분류에서 1-최근접 이웃 알고리즘의 Tomek links 탐색을 구현하는데 이용된다. 데이터의 클래스와 그의 1개 최근접 이웃의 클래스가 다르면 그 쌍은 Tomek links이다. 프로토타입 선택 알고리즘은 학습데이터 T로부터 선택된 프로토타입 집합 S을 반복 단계을 이용하여 생성한다. 알고리즘의 단계는 Hausdorff 쌍의 계산, S의 초기화, 그리고 반복 단계를 통해 S에 새로이 선택되는 프로토타입이 추가된다. 다음 Hausdorff 쌍에 대해 S의 데이터와의 관계를 고려하여 추가 여부를 결정하며, 모든 Hausdorff 쌍을 검사 후 알고리즘은 종료한다.

이진 분류 문제의 프로토타입 선택 방법은 그림 1와 같이 3가지이다. 원은 긍정 클래스 +1에 속하고 사각형은 부정 클래스 -1에 속한다. 빈 점들은 이전 반복에서 S의 멤버로서 선택된 프로토타입이다. 잠정적 결정경계는 점선으로 Tomek links는 직선으로 표시하였다. n_p와 n_q는p와 q의 최근접 이웃이며 점선화살표로 표시된다. H의 다음 쌍 (p,q)과 이웃 n_p와n_q는 p 또는 q가 s에 속하지 않으면 S의 새로운 후보자로서의 여부가 검사된다. p와q의 최근접 이웃이 S의 멤버 여부에 따라 다음 3가지 경우로 세분화된다. 두 최근접 이웃이 S의 멤버, 하나의 최근접 이웃 만이 S의 멤버 , 마지막으로 두 최근접 이웃이 모두 S에 존재하지 않는 경우이다.

두 최근접 이웃이 S의 멤버가 되는 그림 1(a)에서 S={q, p=n_p}일 때 (p, q) ∈H 제시되었다. q는 S의 멤버이고, p의 최근접 이웃 n_p가 이미 S의 멤버이다. p와 n_p가 동일 클래스이면 이미 선택된 프로토타입 n_p가 p의

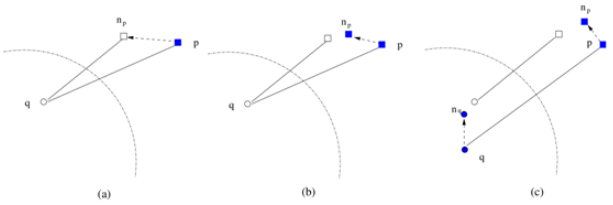

그림 1. 프로토타입 선택 규칙
Fig. 1. Prototype Selection Rules

클래스를 예측하는데 이용될 수 있으므로 S의 새 멤버가 되지 않는다. 그러나 p와 n_p가 다른 클래스이면 p를 S에 추가한다.

그림 1(b)는 하나의 최근접 이웃만이 S의 멤버인경우이다. S={q,p}와 p의 최근접 이웃 n_p이다. 만약 n_p와 p가 동일 클래스이면 분리경계에 가까운 데이터가 새로운 프로토타입으로 S에 추가된다. $d(q,n_p)>d(q,p)$이면 n_p을 새로운 프로토타입으로 선택하고 그렇지 않으면 p가 새 프로토타입이 된다. 그러나 p가 np와 이질적 클래스라면 p와 n_p를 S에 추가한다.

최근접 이웃이 모두 S에 나타나지 않는 경우는 그림 1(c)이다. 경계면에 위치될 가능성이 높은 두 데이터를 프로토타입으로 추가한다. 최근접 이웃과 동일 클래스이고 $d(n_p,q) > d(n_q, p)$이면 n_q와 p를 S에 추가한다. 그렇지 않으면 n_p와 q를 새 프로토타입으로 선택한다. n_p와 p 그리고 n_q와 q가 다른 클래스에 속하는 경우는 발생할 수 없다. 최단 거리 쌍으로 정렬된 H의 멤버가 차례로 검사되기 때문이다. 알고리즘은 n_p 와 n_q 가 S내에 있을 경우는 체크하지 않는 이유는 최근접 이웃 규칙에 따른 예측 분류 시 n_p와 n_q가 p와 q보다 높은 분류 정보를 갖고 있기 때문이다. 언급된 경우들를 제외한 다른 경우는 발생하지 않는다. 왜냐면 정렬된 Tomek links의 멤버만이 S내의 새 멤버로써 검사되기 때문이다. 알고리즘은 모든 H의 순서쌍이 검사되면 종료된다.

그림 2는 인위적으로 발생시킨 분류 문제에서 제안 알고리즘 PSNN과 최근접 이웃 학습 NN의 예측율 비교이다. 이진 분류 문제는 학습 데이터의 차원은 2이며 총 9,000개의 데이터 중 6,299개 학습 데이터와 2,701개

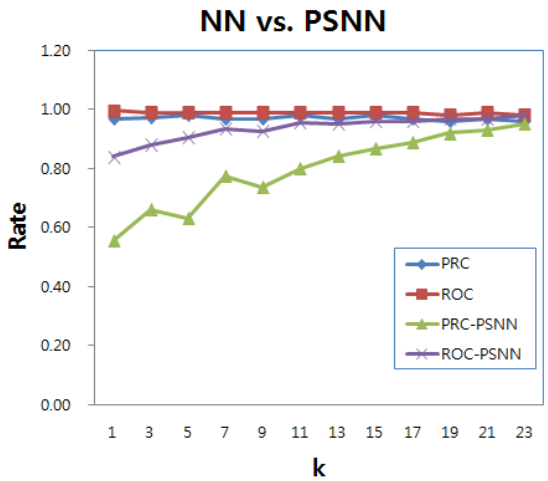

그림 2. NN과 PSNN의 예측율 비교
Fig. 2. Performance comparison of NN and PSNN.

의 테스트데이터이다. 부정과 긍정 클래스의 평균은 (-2,+2)와 (2,-2), 대각 행렬의 값이 1.5인 분산으로 준비되었다. 5-way 교차 검증을 수행하고 예측율 평가는 PRC(Precision-Recall Curve)와 ROC(Receiver Operating Characteristic)이다. NN은 6,299개의 프로토타입이 테스트데이터의 예측에 이용되나, PSNN은 선택된 프로토타입 집합을 이용하여 테스트 데이터의 분류를 수행한다. 선택된 프로토타입 집합의 크기는 k 값에 따라 9~89의 범위로 나타나 0.15%내 적은 프로토타입이 선택되었다. 작은 k 값의 경우 NN의 학습 효과가 PSNN보다 우세하나 프로토타입의 수가 많아 높은 예측 시간이 필요하다. k 값에 관계없이 PSNN는 작은 분류 시간이 소요되나 NN의 분류 성능에 도달하기 위해 큰 k값의 선택이 필요하다. k=23일때 NN과 PSNN의 성능이 비슷하게 나타난다.

Ⅳ. 실 험

1. 학습 데이터의 준비

UCI[11]로부터 선택된 분류 문제가 표 1에 기술되었다. 선택된 분류 문제가 학습과 테스트 집합으로 구성되지 않은 경우 주어진 데이터의 20%를 테스트 집합으로 추출하였으며, 다중 분류 문제 Letter와 Digit은 새 이진 분류 문제로 생성시켰다. 각 실험은 5-way 교차 검증을 진행하여 비교 평가를 하였다. 표 1은 평가에 선택된 분류 문제를 학습과 테스트 데이터의 크기, 속성의 수 등으로 기술되었다.

표 1. 선택된 분류 문제
Table 1. Selected classification problems.

문제	학습	테스트	속성
Heart	191	79	13
Breast Cancer	546	137	10
Australian	551	139	14
Pima	614	154	8
Liver Disorder	276	69	6
Ionosphere	280	71	34
Vowel	528	462	10
Letter-BG	759	425	16
Car Evaluation	1,382	346	6
Svmguide1	3,089	4,000	4
Digit-12	2,199	623	256
Shuttle	30,450	14,500	9
Splice	2,175	1,000	60

2. 성능 평가

분류 결과는 긍정 클래스와 부정 클래스로 구분된다. 두개의 분류 클래스로 예측 결과가 나타나는 경우에 표 3의 교차 테이블[12]의 평가가 사용된다. TP(True Positives)는 긍정 클래스의 입력 데이터를 긍정 클래스로 정확히 분류된 데이터의 수, FP(False Positives)는 부정 클래스의 입력 데이터가 긍정 클래스로 오 분류된 데이터의 수, FN(False Negatives)는 부정 클래스의 데이터를 긍정 클래스로 잘못 예측한 데이터의 수, 그리고 TN(True Negatives)는 부정 클래스의 입력 데이터를 부정 클래스로 정확히 분류한 데이터의 수이다. 이들 측정으로부터 오류율(error rate), 정확률(accuracy rate), ROC, PRC 등 다양한 평가가 이용된다[12~14].

오류율과 정확률은 클래스 불균형으로 인한 편향된 학습에 높은 예측율을 보이는 단점을 극복하기 위해 사전 데이터 분포의 비율를 반영하는 클래스 내에서 예측율 평가에 ROC와 PRC 평가가 이용될 수 있다. ROC 평가 방법은 정확히 예측된 긍정 데이터의 비율 TPR(True Positive Rate, 식 (1))과 긍정 클래스로 분류한 부정 클래스의 비율 FPR(False Positive Rate, 식 (2))은 두 클래스의 데이터로부터 예측된 긍정 클래스에 참여하는 비율이 된다.

$$TPR = \frac{TP}{TP + FN} \tag{1}$$

$$FPR = \frac{FP}{FP + TN} \tag{2}$$

PRC 평가는 정확하게 예측한 긍정 데이터의 *REC* 비율(Recall rate)과 긍정 클래스로 예측한 데이터 중에서 올바른 긍정 데이터의 비율 PRE(Precision rate, 식 (3))로 계산되며 REC는 ROC 측정 방법의 *TPR* 식 (1)과 같다.

$$PRE = \frac{TP}{TP + FP} \tag{3}$$

데이터의 학습 평가에서 ROC 평가의 가시화는 사전 데이터의 분포와 부정 클래스의 데이터가 긍정 클래스로 오 분류된 FP의 비율과 긍정 클래스 내에서 정확히 예측된 TP의 비율로부터 측정된다. FP의 변화는 *FPR* 계산에는 비례하지만 TPR 측정에는 고려되지 않는다. 그러므로 높은 불균형 데이터 문제의 학습 평가에 ROC 평가는 사전 데이터 분포와 클래스 별 오류율을 이용하여 나타난다. 소수의 긍정 데이터의 비율이 사전 데이터 분포에 반영되나 오 분류된 부정 데이터의 비율이 반영되지 않는다. PRC 평가의 PRE는 오 분류된 부정 클래스의 데이터 수 FP를 고려한 TP의 비율을 계산한다. 문제가 학습 시 대다수의 데이터로 구성된 부정 클래스의 학습에 치우치면 FN은 작아지고 FP는 커지게

표 2. 교차테이블
Table 2. Confusion table.

실제 / 예측	긍정	부정
긍정	TP	FP
부정	FN	TN

표 3. k vs 선택된 프로토타입의 수
Table 3. k vs the number of prototypes.

문제	NN	PSNN		
	No.	k=1	k=3	k=5
Heart	191	81.1(42.4)	88.3(46.1)	89.1(46.6)
Breast Cancer	546	177.0(32.4)	209.0(38.3)	311.3(57.0)
Australian	551	223.1(40.5)	225.0(40.8)	225.7(40.8)
Pima	786	315.5(40.1)	375.0(47.7)	387.3(49.2)
Liver Disorder	276	146.7(52.9)	141.0(51.1)	146.2(52.9)
Ionosphere	281	90.3(32.0)	122.0(43.4)	150.0(53.4)
Vowel	423	105.0(24.8)	91.0(21.5)	118.0(27.9)
Letter-BG	608	297.2(48.8)	366.0(60.2)	358.4(58.9)
Car Evaluation	1,383	960.0(69.0)	985.0(71.2)	851.2(61.5)
Svmguide1	3,089	1,728.3(55.9)	2,620.0(84.8)	2,864.3(92.7)
Digit-12	2,199	60.3(2.7)	81.0(3.7)	81.5(3.7)
Shuttle	30,450	3,433.1(11.3)	3,549.0(11.7)	3798.6(12.5)
Splice	2,175	489.0(22.5)	506.0(23.3)	1,180.2(54.3)

되어 *PRE*는 상대적으로 작아지게 된다. 그러므로 PR 평가는 소수의 긍정 데이터의 분포를 성능 평가에 반영한다.

학습 알고리즘의 예측 결과는 클래스 멤버십 또는 클래스 참여 확률로 고려될 수 있다. 데이터 집합의 예측 결과를 클래스 멤버십에 따라 정열하면 참여 정도에 따라 ROC와 PRC 평가가 2차원 그래프로 가시화될 수 있다. 가시화된 2차원의 ROC와 PRC 평가로부터 AUC(Area Under the Curve) 계산은 학습 알고리즘의 기대 성능(expected performance)으로 측정된다[13, 14].

일반적으로 ROC와 PRC의 AUC는 학습 알고리즘의 성능 비교와 분류 문제의 객관적인 학습 성능 평가로 채택되었다. AUC-ROC는 불균형 비율이 낮아지면서 나타나는 성능 향상을 제시하지 못한다. AUC-PRC 평가가 불균형 데이터 문제의 학습 성능 평가에 더 적절하다는 연구가 수행되었으며 비선형 접근법(nonlinear interpolation)을 이용한 PRC 평가의 AUC 계산 알고리즘이 제시되었다[13~15].

객관적 평가를 위해 표 1의 문제의 실험은 ROC과 PRC 평가로 가시화하여 기대 성능 AUC를 측정하였다. 학습 성능 비교에서 ROC-AUC 계산은 C. Ferri가 제시한 알고리즘[13], PRC-AUC 계산은 Jesse Davis의 알고리즘[15]을 구현하여 사용되었다.

3. 실험 결과

학습 데이터의 축소와 예측율에 대해 PSNN의 성능이 주어진 문제에서 최근접 이웃 학습 NN과 비교를 수행하였다. 표 3은 이웃 프로토타입의 수 k 값이 1, 3, 5로 주어질 때 학습 PSNN의 데이터 축소 비율을 보이고 있다. 최대 축소 비율은 Digit-12의 2.7%(k=1)이며 최대 비율은 Svmguide1의 92.7%(k=3)으로 나타났다. 대부분 약 50.0% 정도에서 데이터 축소 비율이 나타났다. 데이터 증가 비율은 k 값의 변화에 Vowel, Letter-BG, Car Evaluation, Svmguide1, Splice 등에서 증가 비율이 높으나 그 외에서는 미비하였다. PSNN의 학습 데이터의 축소는 NN의 분류 예측 시간을 줄일 수 있다.

표 4는 벤치마킹 문제의 NN과 PSNN의 PRC와 ROC 성능 비교를 제시하고 있다. Letter-BG, Car Evalutation, Splice에서 PSNN의 효과는 나타나지 않았다. k의 증가에도 선택된 프로토타입이 테스트 데이터에 대한 충분한 분류 정보를 가지지 않기 때문이다. 그 외 선택 문제에서는 NN과 대등하거나 높은 PRC와 ROC 값을 보이고 있어 PSNN의 데이터 축소 효과를 입증하였다. Digit과 Shuttle 문제는 높은 데이터 축소 비율에도 불구하고 NN과 대등한 성능을 보이고 있어 PSNN의 데이터 축소 효과가 예측율에 크게 나타났다.

표 4. NN과 PSNN의 학습 성능 비교
Table 4. Performance comparisons of NN and PSNN.

문제	k											
	1				3				5			
	NN		PSNN		NN		PSNN		NN		PSNN	
	PRC	ROC	PRC	ROC	PRC	ROC	PRC	ROC	PRC	ROC	PRC	ROC
Heart	0.66	0.73	0.67	0.67	0.68	0.76	0.72	0.77	0.70	0.78	0.70	0.76
Breast Cancer	0.97	0.98	0.96	0.97	0.96	0.97	0.97	0.98	0.97	0.98	0.97	0.98
Australian	0.73	0.82	0.71	0.77	0.77	0.84	0.71	0.79	0.76	0.83	0.76	0.83
Pima	0.75	0.64	0.75	0.61	0.78	0.70	0.78	0.68	0.80	0.76	0.75	0.70
Liver Disorder	0.50	0.62	0.38	0.54	0.50	0.62	0.34	0.50	0.50	0.63	0.40	0.60
Ionosphere	0.85	0.88	0.84	0.85	0.82	0.85	0.86	0.89	0.81	0.84	0.84	0.87
Vowel	0.20	0.69	0.29	0.77	0.27	0.70	0.18	0.66	0.27	0.68	0.27	0.75
Letter-BG	0.97	0.98	0.95	0.97	0.95	0.98	0.71	0.84	0.95	0.99	0.74	0.85
Car Evaluation	0.42	0.85	0.64	0.80	0.84	0.94	0.47	0.70	0.92	0.98	0.43	0.66
Svmguide1	1.00	0.50	1.00	0.50	1.00	0.50	1.00	0.50	1.00	0.50	1.00	0.50
digit12	1.00	1.00	0.99	0.99	0.98	0.99	0.99	0.99	0.99	0.99	0.99	0.99
Shuttle	1.00	0.99	0.95	0.90	1.00	1.00	0.97	0.98	1.00	1.00	0.99	1.00
Splice	0.72	0.73	0.61	0.65	0.79	0.77	0.62	0.65	0.80	0.77	0.76	0.73

특히 Shuttle의 경우 NN의 프로토타입의 약 10% 정도의 프로토타입 만을 가지고 동등한 학습 성능을 보이고 있다. Liver Disorder와 Vowel 문제에서 PSNN의 데이터 축소 효과는 있으나 학습 복잡도를 최근접 이웃 규칙이 모델링하는데 충분하지 않았다.

V. 결 론

최근접 이웃 분류 학습은 단순한 구현에 비해 높은 성능을 보였으나 학습 데이터가 많아지면 높은 기억 용량과 높은 계산 시간의 필요는 문제점으로 인식되었다. 본 논문에서 제안한 프로토타입 선택 알고리즘은 분리 경계에 근접한 데이터들로 구성되는 새로운 학습 데이터를 생성시켜 분류 예측을 수행한다. Tomek links 기반 데이터 선택은 데이터 선택 시 순서에 독립적이며 분리 경계에 근접한 쌍의 선택을 고려한다. 이미 선택된 프로토타입의 클래스와 거리 관계 분석을 이용하여 데이터의 프로토타입 집합에 추가 여부를 결정한다. 최근접 이웃 규칙을 프로토타입 선택과 분류 학습에 같이 사용한 벤치마킹 문제의 비교 실험에서 제안하는 프로토타입 선택 알고리즘은 원래의 학습 데이터 수에 비해 2~60%이내의 데이터 축소를 보였으나 예측율은 거의 동등하였다. 응용성을 높이기 위해 프로토타입의 선택 시 최적의 k 값, 거리계산 방법의 연구, 다중 분류 문제에 적용 등과 더불어 보다 높은 일반화 성능을 위한 다른 학습 알고리즘과 앙상블 전략에 대한 연구가 진행될 수 있을 것이다.

참 고 문 헌

[1] Christopher M. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006.

[2] X. Wu and V. Kumar, Eds, *The Top Ten Algorithms in Data Mining*, Chapman & Hall/CRC Data Mining and Knowledge Discovery, 2009.

[3] S. García, J. Derrac, J.R. Cano, F. Herrera, "Prototype Selection for Nearest Neighbor Classification: Taxonomy and Empirical Study," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 34, No. 3, pp. 417-435, 2012.

[4] K. Yu, L. Ji, and X. Zhang, "Kernel nearest-neighbor algorithm", Neural Processing Letters, Vol.15, pp.147 - 156, 2002.

[5] P. Jeatrakul, K.W. Wong and C.C. Fung, "Data cleaning for classification using misclassification analysis," Journal of Advanced Computational and Intelligent Informatics, Vol.14, No.3, pp. 297 - 302, 2010.

[6] T.M. Cover and P. E. Hart, "Nearest neighbor pattern classification," IEEE Trans. on Information Theory, Vol. 13, No. 1, pp. 21-27, 1967.

[7] F. Angiulli, "Fast nearest neighbor condensation for large data sets classification," IEEE Trans. Knowledge and Data Engineering, Vol.19, pp. 1450 - 1464, 2007.

[8] H. A. Fayed and A. F. Atiya, "A Novel Template Reduction Approach for the K-Nearest Neighbor Method," IEEE Trans. on Neural Networks, Vol.20, No. 5, pp.890-896, 2009.

[9] H. J. Shin and S. Z. Cho, "Response modeling with support vector machines," Expert Systems with Applications, Vol.30, No.4, pp.746 - 760, 2006.

[10] J. Wang, P. Neskovic, and L. N. Cooper, "Neighborhood size selection in the k-nearest-neighbor rule using statistical confidence," Pattern Recognition, Vol.39, No.3, pp.417 - 423, 2006.

[11] UCI machine learning repository, http://archive.ics.uci.edu/ml/.

[12] Ian H. Witten and Eibe Frank, *Data Mining: Practical Machine Learning Tools and Techniques*, 2nd edition, Elsevier, 2005.

[13] C. Ferri, P. Flach and J. Hernndez-Orallo, "Learning Decision Trees Using the Area Under ROC Curve," Proceedings of the 19th International Conference on Machine Learning(ICML-2002), pp. 139-146, 2002.

[14] Jin Huang and Charles X. Ling, "Using AUC and Accuracy in Evaluating Learning Algorithms," IEEE Trans. on Knowledge and Data Engineering, Vol. 17, No. 3, pp. 299-310, 2005.

[15] Jesse Davis and Mark Goadrich, "The relationship between Precision-Recall and ROC curves," Proceedings of the 23th International Conference on Machine Learning(ICML-2006), pp. 233-240, 2006.

저 자 소 개

황 두 성(정회원)

1985년 충남대학교 계산통계학과 학사
1990년 충남대학교 계산통계학과 석사
2003년 Wayne State University, Computer Science 박사
1990년～1991년 국토개발연구원 연구원
1991년～1998년 전자통신연구소 선임연구원
2003년~ 현재 단국대학교 컴퓨터과학과 부교수
<관심분야 : 데이터 마이닝, 기계학습, 병렬처리, 바이오인포매틱스>