새로운 점진적 인스턴스 기반 학습기법

한진철, 윤충화
명지대학교 컴퓨터공학과
e-mail : jchan0415@mju.ac.kr, yoonch@mju.ac.kr

A New Incremental Instance-Based Learning Algorithm

Jin-chul Han, Chung-hwa Yoon
Dept. of Computer Engineering, Myongji University

요 약

메모리 기반 추론 기법에서 기억공간의 효율적 사용과 분류 시간을 줄이기 위한 다양한 방법들이 연구되고 있으며, NGE(Nested Generalized Exemplar) 이론을 예로 들 수 있다. 본 논문에서는 학습 패턴 집합으로부터 대표패턴을 생성하는 RPA(Recursive Partition Averaging) 기법과 점진적으로 대표패턴을 추출하는 IRPA(Incremental RPA) 기법을 제안한다.

1. 서론

메모리 기반 추론 기법은 단순히 학습패턴 전체를 단순히 메모리에 저장한 다음, 테스트 패턴과의 거리를 계산하여 분류하므로 거리기반 학습 (Distance Based Learning) 이라고도 한다[1, 2].

메모리 기반 추론 기법의 대표적인 기계학습 방법인 k-NN (k-Nearest Neighbors) 기법은 학습패턴과 테스트 패턴의 거리를 계산하여 가장 가까운 k 개의 학습패턴을 선택하고, 가장 많은 학습패턴이 소속된 클래스로 테스트 패턴을 분류한다[2, 3]. k-NN 기법은 성능 면에서 만족할 만한 결과를 보여주지만, 학습패턴을 모두 메모리에 저장하기 때문에 다른 기계학습 방법에 비하여 많은 메모리 공간을 필요로 하며, 학습패턴 개수가 증가할 수록 분류에 필요한 시간도 많이 소요된다는 단점을 갖는다[4,5]. 이러한 메모리 기반 추론 기법의 문제점을 해결하기 위한 연구가 활발히 진행되고 있으며, 대표적인 연구로 NGE(Nested Generalized Exemplar)[6,7] 이론이 있다.

본 논문에서는 IBL (Instance-Based Learning) 기법을 기반으로 한 새로운 학습 방법인 RPA (Recursive Partition Averaging) 기법과, 점진적으로 대표패턴을 추출하는 IRPA (Incremental RPA) 기법을 제안하며, UCI Machine Learning Repository 에서 데이터를 발췌하여 제안한 기법과 k-NN 기법, EACH 시스템의 분류 성능과 메모리 사용 효율을 실험적으로 검증하였다.

2. 관련 연구

2.1 k-NN 기법

k-NN 기법은 메모리 기반 추론 기법을 사용한 최초의 분류기로 Lazy Learning Algorithm 이라고도 하는데, 그 이유는 단순히 학습패턴 전체를 메모리에 저장한 다음 테스트 패턴을 분류할 때 모든 계산이 수행되기 때문이다[8]. k-NN 기법의 알고리즘은 다음 <표 1>과 같다.

<표 1> k-NN 기법

① 학습패턴들을 메모리에 저장한다.
② 테스트 패턴과 학습패턴들과의 거리를 수식 (1)을 이용하여 계산한다.
③ 테스트 패턴과 근접한 k개의 학습패턴을 선정한다.
④ 이 k개중에서 가장 많은 수의 학습패턴을 포함하는 클래스로 테스트 패턴을 분류한다.

$$D = \sqrt{\sum_{i=1}^{n} (E_i - Q_i)^2} \qquad (1)$$

E_i와 Q_i는 학습 패턴과 테스트 패턴의 i 번째 특징 값이며, n은 패턴의 특징 개수이다.

이때, k 값은 분류기의 성능을 최적화하기 위하여 Leave-one-out cross-validation 기법을 사용하여 사전에 결정한다[2, 3].

k-NN 기법의 단계 ④에서, 테스트 패턴과 가까운 학습패턴에 큰 가중치-거리의 역($1/D$)-를 부여하는 방법을 WeightVote k-NN 기법이라고 하며, 클래스별로 가중치의 합을 구하고, 합이 가장 큰 클래스로 테스트 패턴을 분류한다[3].

2.2 EACH 시스템

NGE (Nested Generalized Exemplar) 이론에 기반한 학습 기법인 EACH 시스템은 학습패턴을 그대로 저장하지 않고, 인접한 학습패턴들을 초월평면(Hyperrectangle)의 형태로 저장하며, 그 결과 k-NN 기법보다 적은 메모리를 사용한다[6, 7, 9]. 다음의 <표 2>는 EACH 시스템의 알고리즘을 설명한다.

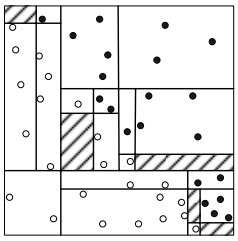
<표 2> EACH 시스템

① 무작위로 몇 개의 학습패턴을 시드 (seed)로 선택하여 예제(Exemplar)로 저장한다.
② 학습패턴을 선택하고, 가장 거리가 가까운 예제를 검색한다.
③ 학습패턴의 클래스와 가장 가까운 예제의 클래스가 동일하면, 학습패턴을 이용해서 예제를 확장하고 가중치를 수정한 후, 단계 ⑥을 수행한다.
④ 클래스가 다른 경우, 가중치를 수정하고 두 번째로 가까운 예제를 선택한다.
⑤ 학습패턴의 클래스와 두 번째로 가까운 예제의 클래스가 동일하면, 예제를 확장하고 가중치를 수정하며, 다를 경우, 학습패턴을 새로운 예제로 저장한다.
⑥ 학습패턴 집합이 공집합이 될 때까지 단계 ②-⑤를 반복한다.

EACH 시스템의 학습이 끝나면, 학습패턴들은 예제의 집합으로 표현되며, 예제는 점 또는 초월평면의 형태를 취하게 된다. 테스트 패턴은 가장 가까운 예제의 클래스로 분류한다. 예제가 점일 경우에는 점과의 거리를 계산하며, 초월평면일 경우에는 가까운 면과의 거리를 계산한다.

3. RPA(Recursive Partition Averaging) 기법

본 논문에서 제안하는 RPA 기법은 전체 학습패턴 공간을 (그림 1)과 같이 재귀적으로 분할하면서 대표패턴을 생성하며, 대표패턴은 인스턴스 평균(Instance Averaging)법을 이용하여 계산한다[10,11].



(그림 1) RPA 기법의 학습패턴 공간 분할

(그림 1)은 패턴공간이 RPA 기법에 의해 재귀적으로 분할된 예제이며, 총 17 개의 대표패턴이 생성되고, 빗금친 분할영역은 학습패턴이 존재하지 않으므로 대표패턴이 생성되지 않는다. 또한, RPA 기법은 특징간의 영향력을 평준화하기 위하여 학습 개시 이전에 모든 특징 값들을 0 과 1 사이의 값으로 정규화하며, 테스트 패턴에 대한 분류 정확도를 높이기 위하여 특징 가중치 값을 사용한다.

3.1 패턴공간의 분할과 대표패턴 생성

RPA 기법에서 분할이 필요한 경우, 모든 특징에 대한 분할점을 결정해야 하며, 특징의 분할점을 선택하기 위하여 특징 값을 오름차순으로 정렬하고 특징 값이 변화하는 위치를 경계값으로 선정한다. 예를 들어, 70 과 72 사이의 경계값은 두 특징 값의 평균인 71 이 된다.

구한 경계 값들 중, 결정트리 알고리즘에서 특징의 분할점을 선정할 때 사용하는 *IG* (Information Gain) 값을 이용하여 가장 변별력이 좋은 경계 값을 분할점으로 선택한다[12]. *IG* 값은 수식 (2), (3)을 이용하여 계산한다.

$$I = -\sum_{i=1}^{C} p_i \log_2 p_i \qquad (2)$$

p_i 는 학습패턴 집합에서 클래스 i 에 소속되는 패턴의 비율이며, C 는 클래스의 개수를 의미한다.

$$IG(f) = I - \sum_{i=1}^{2} P_i I_i \qquad (3)$$

I 는 분할 이전의 정보량이며, P_i 는 분할 이전의 학습패턴 중에서 분할된 영역에 포함된 학습패턴의 비율이다. I_i 는 특정 경계 값 f 를 기준으로 분할했을 때 분할된 각 공간의 정보량을 의미하며, 수식 (2)를 이용하여 계산한다.

여기에서 I 값이 크다는 사실은 올바르게 분류하기 위하여 많은 양의 정보가 필요하다는 것을 의미하며, *IG* 값은 분할 이전의 정보량과 경계 값을 기준으로 분할했을 경우의 정보량의 차이를 의미한다. 즉, *IG* 값은 분할 이후의 정보량이 작아질 경우에 큰 값을 갖게 되며, 결국 *IG* 값이 큰 경계 값을 분할점으로 선택할 때 효율적인 분할이 가능하다.

다음 <표 3>은 RPA 기법의 알고리즘을 보여준다.

<표 3> RPA 기법

초기화 단계
① 전체 패턴 집합을 정규화한다.
② 패턴 집합을 학습패턴과 테스트 패턴 집합으로 분리한다.
③ 전체 학습패턴 집합을 포함하는 영역을 초기 분할 영역으로 정의한 후, 다음의 학습 알고리즘을 적용한다.

학습 알고리즘
① 현재 분할영역에 포함된 모든 학습패턴의 클래스를 검사한다.
② 만약 모든 학습패턴의 클래스가 동일하면, 인스턴스 평균법으로 대표패턴을 추출하고 종료한다.
③ 만약 클래스가 다른 학습패턴이 존재하면, 현재 분할영역의 특징 별로 새로운 경계 값을 구하고, 이 중에서 가장 효율적인 경계 값을 분할점으로 선정한다.
④ 단계 ③에서 선정된 분할점을 이용하여 새로운 분할영역들을 구성한다.
⑤ 단계 ④에서 구성된 하나 이상의 학습패턴을 포함하는 모든 분할영역에 대해서 위의 학습 알고리즘을 재귀 호출한다.

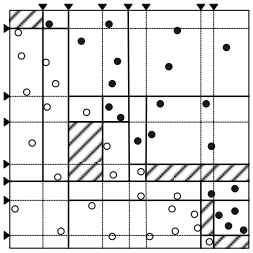
본 논문에서는 학습개시 이전에 전체 패턴 집합을 정규화하기 때문에 초기 분할영역을 구성하는 모든

특징의 하한 값과 상한 값은 0 과 1 이다. 인스턴스 평균법은 여러 개의 학습패턴의 특징 값들을 평균하여 하나의 대표패턴으로 대치하는 방법을 의미한다.

3.2 특징 가중치의 계산

본 논문에서는 정확한 분류를 위하여 패턴간의 거리를 계산할 때 특징 가중치를 이용하며, 특징 가중치는 학습이 완료되면, 분할된 패턴공간의 학습패턴 분포를 이용하여 계산한다.

우선, 특징 가중치를 구하기 위하여 각 특징의 분할영역의 개수를 검사한다. (그림 2)은 RPA 기법으로 학습했을 경우의 분할된 패턴 공간을 보여주고 있으며, 이때 실제 분할된 분할영역의 경계선에서 가상의 분할선을 연장하여 각 특징에 대한 분할 개수를 결정한다.



(그림 2) 특징가중치 계산을 위한 분할

(그림 2)에서 굵은 실선으로 표시된 부분은 RPA 기법에 의해 분할된 영역을 나타내는 것이며, 가는 점선으로 표시된 부분은 특징 가중치 계산을 위하여 패턴공간을 가상으로 분할한 선을 나타낸다. 이 경우 가로 특징 축 8 개, 세로 특징 축 8 개로 분할 된 것을 알 수 있다. 특징 가중치 값은 수식 (4)를 이용하여 계산한다.

$$W_i = I - \sum_{i=1}^{N} P_i I_i \qquad (4)$$

P_i는 분할 이전의 학습패턴 중 분할된 영역에 포함된 학습패턴의 비율이다. I 는 분할 이전의 정보량, I_i는 각 분할점들을 기준으로 분할했을 때 각 공간의 정보량이며, 이들은 수식 (2)를 이용하여 계산한다. 또한, N은 특징 f의 최종 분할영역 개수이다.

4. IRPA (Incremental RPA) 기법

RPA 기법은 패턴공간을 재귀적으로 분할하여 대표패턴을 생성한다. 하지만, 과도한 분할이 발생할 경우, 생성되는 대표패턴이 많아질 수 있다. 본 논문에서는 불필요한 대표패턴의 생성을 방지하여 메모리 사용 효율을 높이고 분류 시간을 단축시키기 위해서 점진적으로 대표패턴을 추출하는 IRPA 기법을 제안한다.

<표 4> IRPA 기법

① RPA 기법을 수행하여 대표패턴을 생성한다.
② 가장 많은 학습패턴을 이용하여 생성된 대표패턴을 선택하고, 이때 사용된 학습패턴을 학습패턴 집합으로부터 제거한다.
③ 더 이상 학습할 패턴이 없을 때까지, ①-② 단계를 반복 수행한다.

<표 4>는 IRPA 기법의 알고리즘을 보여주고 있다. IRPA 기법은 대표패턴을 추출하기 위하여 RPA 기법을 여러 번 수행하며, 분류를 위한 특징 가중치 값은 최초 분할된 패턴공간을 이용하여 계산한다.

5. 분류 알고리즘

본 논문에서 제안하는 RPA, IRPA 기법은 테스트 패턴을 분류하기 위하여 대표패턴들과 수식 (5)로 거리계산을 하며, 가장 가까운 대표패턴의 클래스를 출력으로 결정한다. 따라서 k-NN 기법과는 달리 사전에 최적의 k 값을 구할 필요가 없다.

$$D = \sqrt{\sum_{i=1}^{n} W_i (E_i - Q_i)^2} \qquad (5)$$

W_i는 i번째 특징의 가중치 값이며, E_i와 Q_i는 대표패턴과 테스트 패턴의 i번째 특징 값이다. n 은 패턴의 특징 개수를 의미한다.

6. 실험 및 분석

본 논문에서 제안한 RPA, IRPA 기법의 성능을 k-NN 기법, WeightVote k-NN 기법 그리고 EACH 시스템과 비교 검증하였으며, 실험 방법은 stratified 10-fold cross-validation 기법을 사용하였다. 실험데이터는 기계 학습의 벤치마크 자료로 사용되는 UCI Machine Learning Repository 에서 Breast-Cancer-Wisconsin, Glass, Ionosphere, Iris, New-Thyroid, Sonar, Wine 데이터를 발췌하여 사용하였으며, 이들 데이터는 모든 특징이 실수 값으로 구성되어 있다.

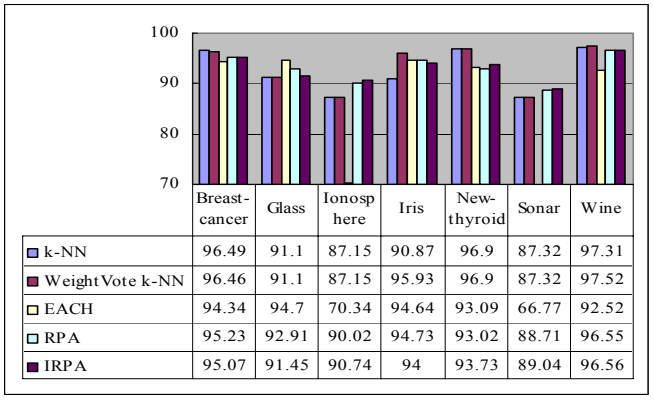
6.1 분류 성능

분류 성능 실험에서 k-NN 기법은 Leave-one-out cross-validation 기법으로 계산한 최적의 k 값을 사용하였으며[10], EACH 시스템은 시드 개수 5, 가중치 변화량 0.2 로 설정하여 실험하였다.

다음의 (그림 3)은 분류 성능을 보여주고 있다. 본 논문에서 제안한 RPA, IRPA 기법은 k-NN 기법, EACH 시스템과 비교하여 유사한 성능 또는 향상된 분류 성능을 보여주고 있으며, Ionosphere, Sonar 의 경우에 EACH 시스템보다 높은 분류 성능을 보여주고 있다. EACH 시스템이 Ionosphere, Sonar 에서 저조한 성능을 보여주고 있는 것은 초기 시드 (seed)의 영향으로 볼 수 있으며[9], 본 논문에서 제안한 기법이 EACH 시스템보다 모든 데이터 셋에서 안정적인 성능을 보여준다.

<표 5> 분류 성능에 대한 표준편차

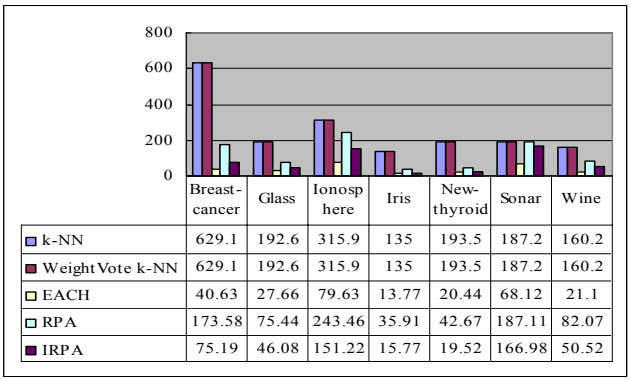
	Breast-cancer	Glass	Ionos-phere	Iris	New-thyroid	Sonar	Wine
k-NN	2.24	5.37	5.08	7.16	3.66	7.29	3.57
Weight-Vote k-NN	2.26	5.37	5.08	4.21	3.66	7.29	3.4
EACH	2.89	3.77	6.62	5.14	4.16	4.77	4.98
RPA	2.2	4.94	4.65	4.45	5.03	6.64	4.32
IRPA	2.5	5.33	4.73	5.85	5.24	6.61	4.25



	Breast-cancer	Glass	Ionosphere	Iris	New-thyroid	Sonar	Wine
k-NN	96.49	91.1	87.15	90.87	96.9	87.32	97.31
Weight Vote k-NN	96.46	91.1	87.15	95.93	96.9	87.32	97.52
EACH	94.34	94.7	70.34	94.64	93.09	66.77	92.52
RPA	95.23	92.91	90.02	94.73	93.02	88.71	96.55
IRPA	95.07	91.45	90.74	94	93.73	89.04	96.56

(그림 3) 분류 성능

6.2 메모리 사용량



	Breast-cancer	Glass	Ionosphere	Iris	New-thyroid	Sonar	Wine
k-NN	629.1	192.6	315.9	135	193.5	187.2	160.2
Weight Vote k-NN	629.1	192.6	315.9	135	193.5	187.2	160.2
EACH	40.63	27.66	79.63	13.77	20.44	68.12	21.1
RPA	173.58	75.44	243.46	35.91	42.67	187.11	82.07
IRPA	75.19	46.08	151.22	15.77	19.52	166.98	50.52

(그림 4) 메모리 사용량

(그림 4)는 메모리에 저장되는 패턴의 개수를 보여주고 있다. RPA 기법은 k-NN 기법보다 적은 개수의 패턴으로 구성되며, IRPA 기법의 경우에는 k-NN 기법보다 Breast-cancer 90%, Glass 77%, Ionosphere 53%, Iris 89%, New-Thyroid 91%, Sonar 12%, Wine 69% 정도 줄어드는 것을 볼 수 있다. 따라서 k-NN 기법보다 빠른 분류가 가능하다.

EACH 시스템에 비하여 RPA, IRPA 기법의 대표패턴 개수가 전반적으로 많이 생성되며, Ionosphere, Sonar 의 경우에는 두 배 이상 많은 것을 볼 수 있다. 이는 모든 특징에 대해서 분할이 발생하기 때문으로 사료된다.

7. 결론

본 논문에서 제안하고 있는 RPA, IRPA 기법은 k-NN 기법보다 적은 개수의 예제를 이용하여 유사한 성능을 보여주고 있으며, 데이터셋 마다 최적의 k 값을 구할 필요가 없다는 장점을 가진다. 또한, IRPA 기법은 학습패턴 집합에서 점진적으로 대표패턴을 추출하는 방법을 이용하여, 보다 적은 패턴 개수로 k-NN, RPA 기법과 유사하거나 향상된 분류 성능을 보장한다. 한편, EACH 시스템은 초기 시드의 선택에 따라 분류 성능의 편차가 심하지만, 본 논문에서 제안한 기법은 모든 데이터 셋에서 안정적인 분류 성능을 보장한다.

8. 향후 연구

Sonar, Ionosphere 와 같이 특징의 개수가 많은 경우에는 과도한 분할이 발생하여 생성되는 대표패턴이 많아지게 된다. 그러므로 향후 연구로 이러한 데이터 셋에서 불필요한 분할을 방지할 수 있는 방법과 학습에 필요한 특징의 수를 줄일 수 있는 방법에 대해서 연구할 예정이다.

참고문헌

[1] T. Dietterich, " A Study of Distance-Based Machine Learning Algorithms", Ph. D. Thesis, computer Science Dept., Oregon State University, 1995.

[2] D. Wettschereck and T. Dietterich, " Locally Adaptive Nearest Neighbor Algorithms", Advances in Neural Information Processing Systems 6, pp. 184-191, Morgan Kaufmann, San Mateo, CA. 1994.

[3] D. Wettschereck, " Weighted k-NN versus Majority k-NN A Recommendation". German National Research Center for Information Technology, 1995.

[4] D. Aha, " A Study of Instance-Based Algorithms for Supervised Learning Tasks: Mathematical, Empirical, and Psychological Evaluations", Ph. D. Thesis, Information and Computer Science Dept., University of California, Irvine, 1990.

[5] D. Aha, " Instance-Based Learning Algorithms, Machine Learning", Vol. 6, No. 1, pp. 37-66, 1991.

[6] D. Wettschereck and T. Dietterich, " An Experimental Comparison of the Nearest-Neighbor and Nearest-Hyperrectangle Algorithms", Machine Learning, Vol. 19, No. 1, pp. 1-25, 1995.

[7] S. Salzberg, " A Nearest hyperrectangle learning method, Machine Learning, No. 1, pp. 251-276, 1991.

[8] D. Wettschereck, et al., " A Review and Empirical Evaluation of Feature Weighting Methods for a Class of Lazy Learning Algorithms", Artificial Intelligence Review Journal, 1996.

[9] 심범식, 정태선, 윤충화, " 최근접 초월평면 학습법에서 시드 개수의 영향에 대한 분석", 한국정보처리학회 ' 98 춘계학술대회, 1998.

[10] 정태선, 이형일, 윤충화, " 고정 분할 평균 알고리즘을 사용하는 향상된 메모리 기반 추론", 한국정보처리학회논문지, Vol.6, No.6, pp. 1563-1570, 1999.

[11] 이형일, 정태선, 윤충화, 강경식, " 재귀 분할 평균법을 이용한 새로운 메모리기반 추론 알고리즘", 한국정보처리학회논문지, Vol.6, No.7, pp.1849-1857, 1999.

[12] Ian H. Witten, Eibe Frank, " Data Mining", Morgan Kaufmann , pp.89~94, 1999,