지역적 k값을 사용한 k-Nearest Neighbor Classifier

이상훈*, 오경환
서강대학교 컴퓨터학과
sadclan@ailab.sogang.ac.kr, kwoh@ccs.sogang.ac.kr

k-Nearest Neighbor Classifier using Local Values of k

Sanghoon Lee, Kyung-whan Oh
Dept. of Computer Science, Sogang Univ.

요 약

본 논문에서는 k-Nearest Neighbor(k-NN) 알고리즘을 최적화하기 위해 지역적으로 다른 k(고려할 neighbor의 개수)를 사용하는 새로운 방법을 제안한다. 인스턴스 공간(instance space)에서 노이즈(noise)의 분포가 지역적(local)으로 다를 경우, 각 지점에서 고려해야 할 최적의 이웃 인스턴스(neighbor)의 수는 해당 지점에서의 국부적인 노이즈 분포에 따라 다르다. 그러나 기존의 방법은 전체 인스턴스 공간에 대해 동일한 k를 사용하기 때문에 이러한 인스턴스 공간의 지역적인 특성을 고려하지 못한다. 따라서 본 논문에서는 지역적으로 분포가 다른 노이즈 문제를 해결하기 위해 인스턴스 공간을 여러 개의 부분으로 나누고, 각 부분에 최적화된 k의 값을 사용하여 kNN을 수행하는 새로운 방법인 Local-k Nearest Neighbor 알고리즘(LkNN Algorithm)을 제안한다. LkNN을 통해 생성된 k의 집합은 인스턴스 공간의 각 부분을 대표하는 값으로, 해당 지역의 인스턴스가 고려해야 할 이웃(neighbor)의 수를 결정지어준다. 제안한 알고리즘에 적합한 데이터의 도메인(domain)과 그것의 향상된 성능은 UCI ML Data Repository 데이터를 사용한 실험을 통해 검증하였다.

1 서 론

Nearest Neighbor 분류 방법(NN Classification method)[1]은 인스턴스 기반의 학습(Instance-based Learning) 방법 중 하나로, 분류 모델(classification model)을 생성하는 방법이 간단하고 예측 정확도가 높아 전통적으로 많은 분야에서 사용되어온 분류 방법이다[2]. NN 분류 방법에서는 학습(training) 시에 단순히 학습 집합(training set)의 인스턴스(instance)를 기억하고, 이후 예측하고자 하는 인스턴스가 입력되었을 때 기억된 학습 집합의 인스턴스 중 가장 유사한 인스턴스의 목적속성(class)을 참조하여 분류 작업을 수행한다. 따라서 분류 모델이 생성되는 시점은 예측하고자 하는 인스턴스가 입력되는 시점이다. 이러한 이유로 NN 방법은 lazy-learning 방식이라고도 불린다[3]. NN 분류 방법은 인스턴스 공간(instance space)의 지역적인 정보를 통해 분류 모델을 생성하기 때문에 특히 복잡한 개념(concept)을 학습할 때 좋은 성능을 보이는 것으로 알려졌다[4].

NN 분류 방법은 학습 집합의 인스턴스 중 예측하고자 하는 인스턴스와 가장 유사한 한 개의 인스턴스 만을 고려하기 때문에 학습 집합에 노이즈(noise)가 섞여 있을 경우, 그 예측 정확도가 떨어지게 된다. 따라서 이러한 경우 노이즈의 영향력을 감소시키기 위해 한 개 이상의 이웃(neighbor) 인스턴스를 고려할 수 있는데, 이렇게 가까운 k개의 인스턴스를 고려하는 방법이 k-Nearest Neighbor 분류 방법(kNN Classification method)이다. kNN에서 고려해야할 이웃 인스턴스의 수인 k는 인스턴스 공간에 분포하는 노이즈의 정도에 따라 결정되는 값으로, 일반적으로 여러 개의 k값을 통해 실험적으로 (즉, k-fold Cross-Validation이나 Bootstrap 등과 같은 검정 방법을 통해) 결정된다. 그런데 이 때, 기존의 kNN 방법은 전체 인스턴스 공간에 대해 동일한 k를 사용하기 때문에, 인스턴스 공간에 국부적(local)으로 노이즈 분포가 다른 부분이 포함되어 있을 경우 공간의 각 부분에 대해 최적화된 k값을 구하기가 힘들다. 즉, 전체 공간에 대해 평균적으로 최적화된 단일한 k값은 평균적인 정도의 노이즈 분포를 갖는 지역에 대해서는 최적 값을 갖지만, 노이즈의 분포가 평균과 다른 지역에 대해서는 해당 부분에서 필요로 하는 최적 값보다 작거나 또는 큰 값을 갖게 된다. 따라서 전체 인스턴스 공간에 대해 동일한 k를 사용한다면, 노이즈의 분포가 지역적으로 다른 부분에 대한 에러율(error rate)의 증가는 불가피하다.

본 논문에서는 이러한 문제점을 해결하기 위해 인스턴스 공간의 지역적인 노이즈 분포 차이를 고려할 수 있는 새로운 알고리즘을 제안하였다. 제안한 방법에서는 먼저 학습 집합을 통해 정의되는 인스턴스 공간을 여러 개의 부분으로 나누고, 각 부분에 대한 최적의 k값을 계산한다. 이후 예측하고자하는 인스턴스가 입력되었을 때, 해당 인스턴스가 공간의 어떤 부분에 속하는지 찾고, 그 부분에 정의된 최적의 k 만큼의 이웃을 고려하여 분류 작업을 수행하게 된다.

2 k-NN 알고리즘의 최적화

지금까지 NN 또는 k-NN 알고리즘을 최적화하기 위한 연구는 많이 이루어져 왔지만, 그것들은 대부분 유사도 측도(distance metric)를 개선하기 위한 것, 또는 학습 집합의 인스턴스를 선별(노이즈를 제거)하기 위한 것[5] 등에 국한되어 왔다. 비교적 최근의 연구로는 앙상블(ensemble) 방법 중 하나인 ECOC 기법[6]과 feature subset을 결합한 방법[7][8]을 통해 kNN 알고리즘의 정확도(accuracy)를 향상시키고 편차(variance)를 줄일 수 있다는 연구 결과가 나와 있다. 하지만, 인스턴스 공간에서 지역적으로 다른 값의 k를 사용하려는 시도는 지금까지 이루어지지 않았다.

이상적으로, 분류하고자 하는 인스턴스가 n개라면, kNN 알고리즘에서 최적의 k값은 n개의 인스턴스 중에서 오분류(misclassification)의 비율이 가장 적도록 n개의 인스턴스 각각이 자신에 적합한 k값을 갖는 경우이다. 실제로 전체 인스턴스 공간에서 단일한 k를 사용하는 경우, 최적의 k값은 개별 인스턴스에 적합한 n개의 k값을 평균한 정도의 값을 취하게 된다. 따라서 인스턴스 공

* 본 연구는 과학 기술부 주관 뇌신경 정보학 사업에 의해 지원 되었음.

간의 각 지역에 따른 적합한 k값을 찾을 수 있다면, 인스턴스 기반 분류 방법의 최대 장점인 지역성(locality)을 보존하면서 동시에 노이즈에 강인한(robust) 알고리즘의 설계가 가능할 것이다.

현실적으로 고려해야할 문제는, 첫째로 인스턴스 공간의 각 지점에서 최적의 k값을 찾는 방법, 그리고 둘째로 인스턴스 공간에 정의되어야 할 k의 개수(즉, 공간을 어떻게 나눌 것인가)를 결정짓는 문제이다. 인스턴스 공간의 어떤 지점에서 최적의 k값을 찾는 문제는 Cross-Validation(CV) 또는 다른 검정 방법을 통해 해당 지점에 적합한 k의 값을 추정함으로써 결정이 가능할 것이다. 그런데 이 때 주의해야할 점은 over-fitting의 문제 -즉, 학습 집합에 대해 지나치게 최적화된 결과가 검정 집합(test set)에 대한 일반성을 갖지 못하는 문제- 이다. 따라서 지역적인 k값을 결정할 때는 over-fitting을 막기 위한 주의가 필요하다. 한편으로 인스턴스 공간을 나누는 문제, 즉 인스턴스 공간에서 사용되어야 할 k의 개수를 결정짓는 문제는 좀 더 복잡한 양상을 보인다. 예를 들어, 인스턴스를 구성하는 벡터(vector)가 n차원이고, 연속형 속성(numeric attribute)을 갖는다고 하자. 이 때 각 속성(attribute)을 m개의 구간으로 나눈다고 하면, 전체 인스턴스 공간은 m^n개의 지역으로 나뉜다. 따라서 인스턴스를 구성하는 벡터의 차원(dimension)이 높고, 각 속성의 구간을 조밀하게 나눌 경우, 계산해야할 k의 개수는 기하급수적으로 증가하게 된다. 그러므로 실제 사용가능(feasible)한 알고리즘을 설계하기 위해서는 이러한 문제점이 해결되어야 할 것이다. 따라서 다음 절에서는, 지금까지 살펴본 개념과 현실적인 문제점들을 바탕으로, 지역적으로 다른 최적의 k값을 갖는 kNN 알고리즘(LkNN)을 설계하기 위한 -초석이 될 수 있는- 한 방법을 제시한다.

3 Local-k Nearest Neighbor(LkNN) 방법

앞서 2절에서 설명한 개념들을 종합해보면, 크게 인스턴스 공간을 나누는 것, 그리고 해당 공간에서의 고려해야할 이웃의 수인 k의 최적 값을 찾는 것의 두 가지로 정리될 수 있다. 이 두 가지 방법이 구체적으로 어떤 방식을 취하는가와 상관없이 Local-k Nearest Neighbor (LkNN) 알고리즘은 다음과 같은 일반적인 형태로 정의될 수 있다. 그림. 1에 간략화된 LkNN알고리즘이 기술되어 있다.

```
LkNN Algorithm

LkNN(trainingSet, testSet){
        find_optimum_local_k(trainingSet);
        for each instance in testSet{
                k=local_k(instance);
                kNN(trainingSet, instance, k);
        }
}
find_optimum_local_k(trainingSet){
        local_area=divide_instance_space(trainingSet);
        for each area in local_area{
                find optimum value of k;
        }
}
```

그림. 1 일반화된 LkNN 알고리즘

알고리즘을 보면 먼저, 학습 집합을 통해 인스턴스 공간을 지역적인 k값이 계산될 부분으로 나누고, 각 부분에 최적화된 k값을 구한다. 다음으로 예측할 인스턴스가 입력되었을 때, 이미 나누어진 인스턴스 공간의 어떤 부분에 해당 인스턴스가 속하는지 찾고, 그 지역의 k값을 할당받는다. 그리고 이 k 값을 통해 일반적인 kNN알고리즘을 수행한다. LkNN 알고리즘의 성능은 인스턴스 공간을 얼마나 적절하게 나누고, 각 공간이 갖는 k값을 over-fitting 없이 얼마나 잘 최적화시키느냐에 따라 달라진다. 따라서 다음으로 이 두 가지 목적을 달성하기 위한 세부적인 내용을 살펴본다.

3.1 인스턴스 공간에 정의되어야 할 k의 개수

2절에서 논의한 것과 같이 인스턴스 공간을 일정한 크기의 부분으로 나누는 것은 인스턴스 벡터의 차원이 클 경우, 계산 비용(computational cost)의 측면에서 실현 불가능하다. 따라서 본 논문에서는 간단한 휴리스틱(heuristic)을 통해 지역을 구분하는 방법을 제안한다. 제안한 방법에서는 먼저 다음의 명제를 가정한다.

가정.1 어떤 두 인스턴스가 인스턴스 공간에서 가깝게 위치한다면, 두 인스턴스에 최적화된 kNN의 k값은 동일하다.

이러한 가정을 바탕으로, 인스턴스 공간은 학습 집합의 인스턴스가 존재하는 점을 중심으로 나누어지게 된다. 즉, 학습 집합의 인스턴스 수가 N_{train}이라고 한다면, 인스턴스 공간은 N_{train}개의 부분으로 나누어지게 된다. 그리고 학습 집합의 각 인스턴스는 자신에게 최적화된 k의 값을 갖게 된다. 이후에 예측하고자 하는 인스턴스가 입력되었을 때, 적합한 k값을 찾는 방법은 단순히 학습 집합의 인스턴스 중 가장 가까운 인스턴스를 찾는 것이다. 그리고 찾은 인스턴스가 갖는 k값을 해당 지역에 최적화된 k값으로 간주하고, 그 값을 입력된 인스턴스의 분류를 위해 사용한다. 이에 대한 예가 그림 2에 자세히 나타나있다. 그림에서 점은 학습 집합의 인스턴스이고, 점선은 각 인스턴스가 담당하는 공간의 경계를 나타낸다. 예측할 인스턴스가 어떤 한 공간에 들어간다면, 그 인스턴스는 해당 지역의 k값(즉, 학습 집합의 인스턴스 중, 가장 가까운 인스턴스가 갖는 k값)을 할당받게 된다.

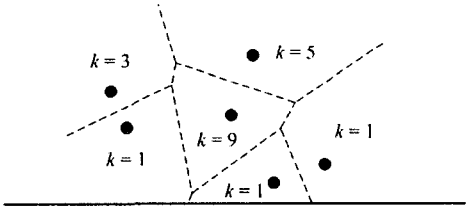

그림. 2 2차원 상에서 나누어진 인스턴스 공간의 예

3.2 각 공간에 최적화된 k값의 결정

3.1절에서 제안한 방법을 통해 인스턴스 공간이 나누어졌다면, 남은 문제는 각각의 공간에 최적화된 k의 값을 찾는 것이다. 본 논문에서는 인스턴스 공간의 각 부분에 적합한 k값을 찾기 위해 Leave-One-Out Cross-Validation(LOOCV)[2]을 사용하였다. 학습 집합의 각 인스턴스에 대해 k의 값을 1부터 임의의 k_{max} 까지 변화시키면서, kNN의 분류 확률을 가장 높이는 k값을 선택하였다. 즉, k의 값을 변화시키며 LOOCV를 수행한 후, 각 인스턴스가 고려한 k개의 이웃 중, 자기 자신의 목적속성(class)과 동일한 목적속성을 갖는 인스턴스의 비율을 최대로 하는 k값을 해당 인스턴스의 최적 값으로 결정하였다. 이때 k_{max}는 최대로 고려할 이웃의 수로, 노이즈의 정도와 계산 비용 등을 고려해 설정되는 값이다. 인스턴스가 정확히 분류될 확률 $P_{correct}$는 다음과 같이 정의된다.

$$P_{correct} = \frac{N_{correct}}{k}$$

이 때, $N_{correct}$는 가장 가까운 k개의 이웃 중 자신의 목적 속성과 동일한 목적 속성을 갖는 인스턴스의 수이고, k는 고려한 이웃의

수이다.

만약 고려한 모든 k에 대해(즉, 1부터 k_{max}까지) $P_{correct}$가 오분류를 일으킬 정도로 작은 값을 갖는다면, 예를 들어 가장 큰 $P_{correct}$값을 산출하는 k를 사용하더라도 k개의 이웃 중 자신의 목적속성과 다른 목적속성을 갖는 인스턴스의 비율이 더 높을 경우, 결정된 k값은 해당 지역에 최적화된 k값이라고 판단하기 힘들다. 다시 말해, 해당 지역의 인스턴스 분포만으로는 최적의 k를 결정하고, 분류를 하기 위한 정보가 부족하다. 따라서 이러한 경우 k값이 부적절한 정보에 의해 over-fitting되는 것을 막기 위해, $P_{correct}$값이 특정한 값 δ 보다 작다면, 해당 인스턴스의 k값은 일반적인 kNN을 최적화하는 k값으로 설정한다. 이때 δ 값은 목적속성의 개수에 역수를 취한 상수 값을 사용하거나, 또는 자신과 다른 목적속성을 갖는 인스턴스의 비율과 비교하여 오분류를 일으키지 않을 정도의 비율을 동적으로 적용할 수 있다.

4 실험 및 결과

제안한 방법의 성능을 평가하기 위해 UCI ML Data Repository[9]의 데이터를 사용하여 간단한 실험을 수행하였다. 실험에 사용된 데이터의 종류와 특성은 표 1과 같다.

표. 1 데이터의 특성

Dataset	Instances	Features	Class	Major Class
breast-cancer-w	699	10	2	66%
glass	214	10	7	36%
heart-disease	294	14	2	65%
ionosphere	351	35	2	64%
iris	150	5	3	33%
sonar	208	61	2	53%

실험은 10-folds Cross-Validation을 사용하여 정확도를 측정하였고, 총 10번 시행한 결과를 평균하였다. 이때 각 시행에서는 객관적인 비교를 위해 동일한 fold로 kNN과 LkNN의 정확도를 측정하였다. LkNN에서 최대로 고려할 이웃의 수인 k_{max}는 11로 설정하였고 δ값은 각 데이터의 목적속성 수에 역수를 취한 값을 사용하였다.

표 2에 10-folds CV을 10번 반복해 얻은 결과와 그 표준 편차를 나타내었다. 표에서 kNN의 결과는 1부터 11까지의 k중 CV 정확도를 가장 높게 하는 k의 결과를 나타내었고, 그 때의 값을 k에 표시하였다. Average k는 LkNN에서 사용한 모든 공간의 k 값들을 평균한 값이고, 마찬가지로 표준 편차를 함께 나타내었다.

표. 2 10-folds CV 결과

Dataset	kNN Accuracy	k	LkNN Accuracy	Average k
breast-cancer-w	96.79±0.16%	5	96.80±0.13%	4.29±3.36
glass	72.50±0.97%	1	74.25±1.25%	5.14±3.72
heart-disease	67.75±1.31%	4	68.24±1.09%	3.96±2.98
ionosphere	86.25±0.70%	1	86.64±0.69%	7.95±4.11
iris	96.88±0.66%	7	97.01±0.48%	9.49±2.67
sonar	82.91±0.70%	2	84.49±0.67%	5.03±3.13

결과를 보면, 전반적으로 LkNN을 사용했을 때의 정확도가 최적의 k를 사용한 kNN의 정확도보다 향상된 것을 볼 수 있다. 이러한 사실은 앞서 2절에서 기술된 인스턴스 공간의 특성 -즉, 인스턴스 공간의 각 부분에서 요구되는 최적의 k값이 각 지역의 노이즈 분포에 따라 다르다는 것- 을 반영한 결과로 해석할 수 있다. 그런데 이때, LkNN에서 사용된 k의 평균값과 표준편차를 주목해야 하는데, 이 값을 통해 LkNN에서 결정된 k의 값들이 얼마나 학습 집합에 over-fitting되어 있는가를 판단할 수 있기 때문이다. breast-cancer, heart-disease, iris, sonar 데이터의 경우 LkNN의 평균 k값과 kNN의 최적 k값과의 차이가 표준편차의 범위 안에 있는 반면에, glass와 ionosphere 데이터의 경우 그 차이가 표준 편차의 범위를 넘어선다. 특히 ionosphere의 경우 이 차이가 현저하게 발생한 것을 볼 수 있다. 이러한 사실을 통해, ionosphere 데이터의 경우, 결정된 각 공간의 k값이 학습 집합에 over-fitting 되었다는 것을 추정할 수 있다.

5 결론 및 향후 과제

본 논문에서는 인스턴스 공간의 각 부분에서 다른 수의 이웃을 고려하는 kNN 방법인 Local-k Nearest Neighbor (LkNN) 알고리즘을 제안하였다. 제안한 방법은 인스턴스 공간의 각 부분에 따라 노이즈의 분포가 다를 경우, 단일한 k값을 사용하는 kNN 알고리즘이 손실할 수 있는 정보를 보존함으로써 분류 정확도를 향상시킬 수 있었다. UCI 데이터를 사용한 실험을 통해 제안한 방법의 성능을 검증하였고, 이 때 사용된 LkNN의 평균 k값과 표준편차를 통해 over-fitting의 문제를 살펴보았다.

2절에서 기술했듯이 LkNN 알고리즘의 성능은 노이즈의 분포에 맞게 인스턴스 공간을 적절히 나누는 것, 그리고 각 공간에서 적절한 k의 값을 over-fitting없이 최적화 시키는 정도에 따라 달라진다. 따라서 3.1절과 3.2절에서 제시한 간단한 휴리스틱 보다 더 정교한 방법을 통해 LkNN의 성능을 향상시킬 수 있을 것이다. 가능한 방법으로는 공간을 군집(clustering)하거나, 계층적(hierarchical)으로 구성하여 k값을 얻는 방법을 생각해 볼 수 있다. 이밖에 다른 여러 가지 기법들을 통해 각 공간의 최적 k값을 결정할 수 있다면, LkNN 알고리즘의 성능은 극대화 될 수 있을 것이다.

참 고 문 헌

[1] Johns, M. V. An empirical Bayes approach to non-parametric two-way classification. In Solomon, H., editor, *Studies in item analysis and prediction.* Palo Alto, CA: Stanford University Press. 1961.

[2] Ian H. Witten, Eibe Frank, "Data Mining", Morgan Kaufmann Publishers, 2000.

[3] Tom M. Mitchell, "Machine Learning", McGraw-Hill, 1997

[4] J. Han and M. Kamber, "Data Mining Concepts and Techniques", Morgan Kaufmann Publishers, 2001

[5] D. W. Aha, Tolerating noisy, irrelevant, and novel attributes in instance-based learning algorithms. *International Journal of Man-Machine Studies* 36(2), page(s): 267-287, 1992.

[6] T. G. Dietterich and G. Bakiri, Error-correcting output codes: A general method for improving multi-class inductive learning programs. *In Proc. of 9th AAAI-91*, page(s): 572-577, AAAI Press, 1991.

[7] F. Ricci and D. W. Aha, Error-Correcting Output Codes for Local Learners, *In Proc. of the 10th European Conference on Machine Learning*, page(s): 280-291, 1998.

[8] S. D. Bay, Nearest neighbor Classification from Multiple Feature Subsets, *Intelligent Data Analysis*, vol.3, page(s): 191-209, 1999.

[9] C. J. Merz & P. M. Murphy, UCI repository of machine databases. University of California, Irvine, http://www.ics.uci.edu/~mlearn./