

퍼지 k-Nearest Neighbors 와 Reconstruction Error 기반 Lazy Classifier 설계

Design of Lazy Classifier based on Fuzzy k-Nearest Neighbors and Reconstruction Error

노석범 · 안태천*

Seok-Beom Roh and Tae-Chon Ahn*

* 원광대학교 전자 및 제어 공학부

요 약

본 논문에서는 퍼지 k-NN과 reconstruction error에 기반을 둔 feature selection을 이용한 lazy 분류기 설계를 제안하였다. Reconstruction error는 locally linear reconstruction의 평가 지수이다. 새로운 입력이 주어지면, 퍼지 k-NN은 local 분류기가 유효한 로컬 영역을 정의하고, 로컬 영역 안에 포함된 데이터 패턴에 하중 값을 할당한다. 로컬 영역과 하중 값을 정의한 후에, feature space의 차원을 감소시키기 위하여 feature selection이 수행된다. Reconstruction error 관점에서 우수한 성능을 가진 여러 개의 feature들이 선택 되어 지면, 다항식의 일종인 분류기가 하중 최소화승법에 의해 결정된다. 실험 결과는 기존의 분류기인 standard neural networks, support vector machine, linear discriminant analysis, and C4.5 trees와 비교 결과를 보인다.

키워드 : Lazy Learning, Local Learning, Lazy Classifier, Locally Linear Reconstruction, Feature Selection, Fuzzy k-Nearest Neighbors

Abstract

In this paper, we proposed a new lazy classifier with fuzzy k-nearest neighbors approach and feature selection which is based on reconstruction error. Reconstruction error is the performance index for locally linear reconstruction. When a new query point is given, fuzzy k-nearest neighbors approach defines the local area where the local classifier is available and assigns the weighting values to the data patterns which are involved within the local area. After defining the local area and assigning the weighting value, the feature selection is carried out to reduce the dimension of the feature space.

When some features are selected in terms of the reconstruction error, the local classifier which is a sort of polynomial is developed using weighted least square estimation.

In addition, the experimental application covers a comparative analysis including several previously commonly encountered methods such as standard neural networks, support vector machine, linear discriminant analysis, and C4.5 trees.

Key Words : Lazy Learning, Local Learning, Lazy Classifier, Locally Linear Reconstruction, Feature Selection, Fuzzy k-Nearest Neighbors

1. 서 론

지도 학습 (supervised learning)의 기본 문제들 중의 하나인 패턴 분류 문제는 지난 수십 년 동안 다양한 접근 방법 패턴 분류 성능을 향상시키기 위해 연구되어져 왔다. 최근에는 패턴 분류를 위한 다양한 접근 방법들 중 신경망 분류기 (neural classifier)와 통계적 방법에 기반을 둔 분류기

(statistical classifier)에 대한 연구가 활발하게 진행되고 있다 [1, 2]. Neural Classifier는 크게 다층 퍼셉트론 (Multi-Layer Perceptron : MLP), Radial Basis Function (RBF) 신경 회로망, 다항식 분류기 (Polynomial Classifier) 등의 분야에서 연구되어지고 있고 [3, 4], 이와 별개로 통계적 분류기도 linear discriminant function, quadratic discriminant function, k nearest neighbors(kNN) classification, Parzen window 분류기 등의 분야에서 활발하게 연구되어지고 있다.

Statistical classifier와 neural classifier의 분류 성능의 비교가 수많은 문헌에서 제시하고 있는 다양한 실험결과들을 이용하여 수행되어 지고 있다. 일반적으로 패턴의 수가

접수일자 : 2009년 10월 22일

완료일자 : 2010년 1월 15일

감사의 글 : 이 논문은 2007년도 원광대학교 교비 지원에 의해 수행됨.

큰 경우에는, 많은 수의 데이터 패턴을 이용하여 학습이 가능한 neural classifier의 분류 정확도가 statistical classifier 보다 우수한 것으로 알려져 있다. 반면에 얻을 수 있는 데이터 패턴의 수가 적을 경우에는 statistical classifier가 더 우수한 분류 정확도를 보이는 것으로 알려져 있다 [2, 5]. Statistical classifier들 중 특히 kNN 알고리즘과 변형된 kNN 알고리즘은 패턴 분류의 많은 영역에서 우수한 성능을 보이는 것으로 알려져 있다 [6]. 일반적으로 kNN 알고리즘은 local learning 방법의 일종으로 알려져 있다. kNN 알고리즘은 학습 데이터 패턴 전체를 저장하고, 새로운 입력이 주어졌을 경우, 주어진 새로운 입력과 가장 가까운 k개의 데이터 패턴을 이용하여 새롭게 주어진 입력 패턴을 분류하게 된다.

Local learning과 달리 global learning은 전체 입력 공간에서 정의된 global classifier를 학습하기 위한 것이다. 단일 분류기를 이용하여 전체 입력 공간에서 분류 작업을 수행하고 설계자가 원하는 정도의 분류 정확도를 얻기 위해서 global learning을 적용한 분류기의 구조는 복잡해 질 수밖에 없다. 이와 달리 local learning을 적용한 분류기는 비교적 단순한 구조를 가진 분류기를 이용하여 local area에서 원하는 정도의 분류기 성능을 보장 할 수 있다는 장점을 가지고 있다.

본 논문에서는 statistical classifier이며 lazy classifier의 일종인 kNN과 linear discriminant function을 결합하여 새로운 classification 방법을 제안 하고자 한다. 제안된 새로운 분류기는 일반적인 kNN 분류기와 달리 kNN을 이용하여 간단한 구조를 가진 분류기에 정의되어지는 local area를 결정하게 된다. k개의 nearest neighbor들로 둘러싸인 local area에서 패턴 분류를 위해 적용될 분류기는 간단한 형태의 linear function을 이용하여 패턴 분류를 수행한다. 일반적으로 linear function을 이용한 패턴 분류기는 two-class 분류 문제에 적용되어 질 수 있다. 또한 제안된 분류기의 성능 개선을 위해 패턴 분류에 영향력이 큰 입력변수를 선택 할 필요가 있다. 입력 변수 선택을 위한 알고리즘으로 Locally Linear Embedding (LLE) [7]의 reconstruction 단계를 사용한 Locally Linear Reconstruction (LLR) [8]을 이용한다. 끝으로 제안된 분류기의 성능을 검증하기 위해 다양한 two-class classification 문제에 제안된 분류기를 적용한 분류 성능을 기존 패턴 분류기의 성능과 비교한다.

2. 패턴 분류를 위한 Lazy classifier인 Locally Weighted Polynomial

선형 회귀 다항식 (linear regression)은 통계적 추정 방법에 다양하게 사용되어진다. 선형 모델의 장점은 구조의 단순성과 이용의 편리함에 있다. 그러나 다른 한편으로 선형 모델은 주요한 단점을 가지고 있으며, 그 단점은 다른 모델에 비해 상대적으로 높은 바이어스를 가지고 있다는 것이다. 다시 말하면 underlying function이 affine function에 의해 정확히 근사화 되지 못하면, 선형 회귀 다항식은 좋지 못한 결과를 보인다. Local linear regression은 작은 입력 공간에 있어서 선형 관계 (linear relationship)를 이용하면 smooth function을 대체적으로 좋은 성능으로 근사화 할 수 있다는 사실을 이용한 것이다[9]. Locally weighted polynomial은 기본적인 선형 회귀 다항식을 기반으로 변형한 것이다 [10]. Locally weighted Polynomial을 적용하기

위한 각 데이터 패턴의 중요성을 결정하는 것은 적절한 데이터 패턴은 복사하고 관련 없는 데이터 패턴은 제거하는 방식으로 볼 수 있다 [11]. 더불어 locally weighted polynomial은 쿼리 패턴이 주어졌을 때 polynomial의 계수가 결정되는 방식이며, 이는 lazy learning의 일반적인 특징이다.

일반적으로 데이터 패턴 주어진 쿼리 패턴 (query pattern)과의 연관성은 아래식과 같은 데이터 패턴과 주어진 쿼리 패턴사이의 유클리디언 거리에 기반을 두어 결정된다.

$$d(\mathbf{a}, \mathbf{b}) = \sum_{l=1}^m (a_l - b_l)^2 \quad (1)$$

여기서, m은 입력 변수의 차원을 나타낸다.

제안된 알고리즘을 설명하기 위해 사용되어진 수학 기호들은 표 1에 기술된다.

표 1 기본 수학 기호

Table 1 Basic notation used in the study

(\mathbf{x}_q, y_q)	A query pattern (test pattern) and its label ($\mathbf{x}_q \in R^m, y_q \in N$)
\hat{y}_q	Predicted label of a query pattern
(\mathbf{x}_i, y_i)	Reference pattern (i.e. training pattern) and its label ($\mathbf{x}_i \in R^m, y_i \in N, i = 1, 2, \dots, n$)
W_q	Weighting matrix(a diagonal matrix with non-zero weights positioned at its diagonal)
$K(\mathbf{a}, \mathbf{b})$	Kernel function

여기서, n은 쿼리 패턴과 관련된 데이터의 수를 의미한다.

하중 함수 (weighting function) 혹은 하중 커널 (weighting kernel)은 식 (1)을 사용하여 얻어진 연관성 정보인 거리 정보를 기반으로 정의되어 질 수 있다.

일반적으로 가장 많이 사용되는 하중 함수는 아래와 같이 정의되는 가우시안 함수이다.

$$K(\mathbf{a}, \mathbf{b}) = \exp\left(-\frac{d(\mathbf{a}, \mathbf{b})}{\sigma^2}\right) \quad (2)$$

여기서 σ 는 가우시안 함수의 분산 파라미터를 의미한다.

Local model이 선형 함수로 정의 되어졌을 경우, local model의 계수를 추정하기 위하여 아래와 같은 목적함수를 정의한다.

$$J(\mathbf{x}_q) = \sum_{i=1}^n (y_i - f(\mathbf{x}_i))^2 \cdot K(\mathbf{x}_i, \mathbf{x}_q) \quad (3)$$

$$= (\mathbf{Y} - \mathbf{X}\mathbf{a}_q)^T \mathbf{W}_q (\mathbf{Y} - \mathbf{X}\mathbf{a}_q)$$

여기서, $\mathbf{W}_q = \begin{bmatrix} K(\mathbf{x}_1, \mathbf{x}_q) & 0 & \dots & 0 \\ 0 & K(\mathbf{x}_2, \mathbf{x}_q) & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \dots & 0 & K(\mathbf{x}_n, \mathbf{x}_q) \end{bmatrix}$,

$$X = \begin{bmatrix} 1 & \mathbf{x}_1 \\ 1 & \mathbf{x}_2 \\ \vdots & \vdots \\ 1 & \mathbf{x}_n \end{bmatrix} = \begin{bmatrix} 1 & x_{11} & \cdots & x_{1m} \\ 1 & x_{21} & \cdots & x_{2m} \\ \vdots & \vdots & & \vdots \\ 1 & x_{n1} & \cdots & x_{nm} \end{bmatrix}$$

$$\mathbf{a}_q = [a_{q0} \ a_{q1} \ \cdots \ a_{qm}] \in R^{(m+1)},$$

$$f(\mathbf{x}_q) = a_{q0} + \sum_{j=1}^m a_{qj} x_{qj} = [1 \ \mathbf{x}_q] \cdot \mathbf{a}_q$$

최적의 계수 파라미터 벡터 \mathbf{a}_q 를 계산하기 위해서 식(3)과 같이 주어진 목적 함수를 최소화 시키는 파라미터 벡터를 구하여야 하며, 아래와 같은 식을 이용하여 얻을 수 있다.

$$\mathbf{a}_q = (X^T W_q X)^{-1} (X^T W_q Y) \quad (4)$$

3. FKNN 기법을 이용한 local area 정의와 weighting factor 정의

앞장에서 설명한 선형 함수를 지역적 분류기(local classifier)로 적용하기 위하여, 지역 영역 (local area)을 정의해야 하며, 정의된 지역 영역에 속한 데이터 패턴 (학습 패턴)들에 하중 값을 부여하여야 한다. 일반적으로 데이터 패턴에 하중 값을 부여하기 위한 방법으로 가우시안 함수를 kernel 함수로 사용하는 경우가 많다. 그러나 본 논문에서는 가우시안 함수를 kernel 함수로 사용하지 않고, Bezdeck이 제안한 퍼지 클러스터링 알고리즘에 적용된 상대적 거리를 이용한 함수를 하중값을 부여하기 위한 하중 함수로 사용할 것이다.

3.1 Crisp k-Nearest Neighbors 기법

대표적인 non-parametric 알고리즘인 k-Nearest Neighbors (kNN) 기법은 다양한 분류 문제에 적용되어져왔다. 일반적으로, KNN 기법은 알려져 있지 않은 데이터 패턴이 주어질 경우, 주어진 데이터 패턴과 가장 가까운 위치에 있는 k개의 데이터 패턴을 선정하고, 선정된 데이터 패턴의 출력 패턴을 비교하여 새롭게 주어진 데이터 패턴의 출력 값을 추정한다. Crisp k-Nearest Neighbors 기법을 설명하기 위하여 입력 패턴 집합 X와 입력 패턴에 대응하는 출력 패턴의 집합 Y를 (5)와 (6)과 같이 정의한다. 분류 문제에서 출력 패턴의 집합은 연속적인 값이 아닌 이산적인 값을 가진다.

$$X = \{ \mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n \}, \mathbf{x}_k \in R^m \quad (5)$$

$$Y = \{ y_1, y_2, \cdots, y_n \}, y_k \in N \quad (6)$$

여기서, N은 자연수의 집합을 의미하며, n은 데이터 패턴의 수를 의미한다. 학습 데이터의 집합이 (5)와 같이 정의되었다고 가정하고, 분류해야할 query instance \mathbf{x}_q 가 주어지면, 학습 데이터 집합 X의 요소들 중에서 \mathbf{x}_q 와 가까운 거리에 있는 k개의 데이터 패턴을 찾는다.

이때 데이터 패턴들 간의 거리는 (7)과 같이 정의된 weighted Euclidean distance를 이용하여 계산한다.

$$d(\mathbf{a}, \mathbf{b}) = \sqrt{\sum_{l=1}^m \frac{(a_l - b_l)^2}{\sigma_l^2}} \quad (7)$$

여기서, σ_l^2 은 l번째 입력변수의 분산을 의미한다.

(7)에 의해 얻어진 query instance와 학습 데이터 집합의 요소들 사이의 거리 정보를 이용하여 거리가 가까운 K개의 학습 데이터들을 찾는다.

집합 L은 주어진 query instance와 거리가 가까운 데이터 패턴의 인덱스의 집합을 의미한다.

$$L = \{j: \mathbf{x}_j \text{ is one of the } k \text{ nearest instances to query point } \mathbf{q}\} \quad (8)$$

근접 위치에 있는 K개의 학습 데이터들의 출력 label을 이용하여 주어진 query instance의 출력 label을 (9)과 같이 구한다.

$$\hat{y}(\mathbf{x}_q) = \underset{c}{\operatorname{argmax}} \sum_{j=1}^k f(y(\mathbf{x}_{L(j)}), c) \quad (9)$$

$$\text{여기서, } f(y(\mathbf{x}_{L(j)}), c) = \begin{cases} 1, & \text{if } y(\mathbf{x}_{L(j)}) = c \\ 0, & \text{if } y(\mathbf{x}_{L(j)}) \neq c \end{cases}$$

3.2 Fuzzy k-Nearest Neighbors 기법

Fuzzy k-Nearest Neighbors (FkNN) 기법은 기존의 crisp k-Nearest Neighbors 기법에 퍼지 집합의 개념을 적용한 기법으로 주어진 query instance와 학습 데이터 패턴들 간의 유사성(similarity)의 정의에 기반을 두고 있다. 데이터 패턴들 간의 유사성은 (10)과 같이 정의한다.

$$w(\mathbf{x}_{L(i)}, \mathbf{q}) = \begin{cases} 1, & \text{if } \mathbf{x}_{L(i)} = \mathbf{q} \\ \frac{\left(\frac{1}{\|\mathbf{x}_{L(i)} - \mathbf{q}\|} \right)^{\frac{2}{p-1}}}{\sum_{j=1}^k \left(\frac{1}{\|\mathbf{x}_{L(j)} - \mathbf{q}\|} \right)^{\frac{2}{p-1}}}, & \text{if } \mathbf{x}_{L(i)} \neq \mathbf{q} \end{cases} \quad (10)$$

여기서, w는 데이터 $\mathbf{x}_{L(i)}$ 와 query instance \mathbf{q} 와의 유사성을 의미하며, k는 Nearest neighbor들의 수, p는 퍼지화 계수 (fuzzification coefficient)를 의미한다.

(10)를 통해 얻어진 유사성 $w(\mathbf{x}_{L(i)}, \mathbf{q})$ 는 다음 조건을 만족한다.

$$0 < w(\mathbf{x}_{L(i)}, \mathbf{q}) \leq 1 \quad (11)$$

근접 위치에 있는 K개의 학습 데이터들의 출력 label을 이용하여 FkNN 기법을 적용하여 주어진 query instance의 출력 label은 (12)과 같이 구한다.

$$\begin{aligned} \hat{y}(\mathbf{q}) &= \sum_{i=1}^k w(\mathbf{x}_{L(i)}, \mathbf{q}) \cdot y(\mathbf{x}_{L(i)}) \\ &= \sum_{i=1}^k \left(\frac{\left(\frac{1}{\|\mathbf{x}_{L(i)} - \mathbf{q}\|} \right)^{\frac{2}{p-1}} \cdot y(\mathbf{x}_{L(i)})}{\sum_{j=1}^k \left(\frac{1}{\|\mathbf{x}_{L(j)} - \mathbf{q}\|} \right)^{\frac{2}{p-1}}} \right) \\ &= \sum_{i=1}^k \left(\frac{y(\mathbf{x}_{L(i)})}{\sum_{j=1}^k \left(\frac{\|\mathbf{x}_{L(i)} - \mathbf{q}\|}{\|\mathbf{x}_{L(j)} - \mathbf{q}\|} \right)^{\frac{2}{p-1}}} \right) \end{aligned} \quad (12)$$

여기서, $\hat{y}(q)$ 는 query instance와 연관된 fuzzy k-Nearest Neighbor 기법을 이용하여 추정된 출력 label을 의미하며, $y(x)$ 는 데이터 패턴 x 와 연관된 알려진 출력 label을 의미한다.

4. Locally Linear Reconstruction (LLR)을 이용한 입력 변수 선택

패턴 분류 문제에 있어서 고차원 feature 문제가 대두되고 있다. 일반적으로 고차원 feature를 가진 패턴 분류 문제에 있어서 원하는 정도의 패턴 분류 정확성을 얻기 위해서는 분류기의 구조가 복잡해지고 학습에 어려움이 따른다. 그러므로 feature reduction이 없이는 학습의 정확도나 일반화 능력의 개선을 기대 할 수 없게 된다.

고차원 입력 공간 문제를 해결하기 위한 방법으로 다양한 feature reduction 방법이 제안되어 왔다. 개략적으로, 입력 변수들의 수는 filters, wrappers, 그리고 embedded와 같은 3가지 feature 선택방법들에 의해 감소되어진다[12-14]. Principal component analysis [15,16] 과 다른 변환 기반 feature reduction 방법들은 본 논문에서 언급되어지지 않을 것이다. 왜냐하면 변환 기반 feature reduction 방법들은 원래의 feature들의 집합에서 feature를 선택하는 것이 아니라 기존 feature 벡터를 새로운 저차원 feature 벡터로 변환하기 때문이다. 본 논문에서는 feature reduction을 위해 S.T. Roweis과 L.K. Saul 제안한 locally linear embedding[7] 방법을 변형한 LLR 방법을 사용한다[8]. Locally linear embedding 방법은 principal component analysis와 마찬가지로 변환을 기반으로 하는 방법이다. 이 방법을 기존 feature set에서 feature를 선택하는 방법으로 수정하기 위해 LLR 방법을 기반으로 하는 feature selection 방법을 이용하였다.

4.1 비선형 시스템

LLR은 locally linear embedding 방법의 reconstruction 단계를 기반으로 한다. LLR의 기본 개념은 쿼리 패턴이 가장 가까운 k개의 nearest neighbor들로 잘 표현되어 질 수 있다면 쿼리 패턴의 class label도 nearest neighbor들의 class label로 적절히 추정되어 질수 있다는 가정에 기반을 두고 있다. 다시 말하면, 입력 패턴들이 비슷하면, 출력들도 또한 비슷하다는 가정을 기반으로 LLR이 제안되었다 [8]. LLR을 수행하기 위해서 먼저 reconstruction error에 관하여 식(13)과 같이 정의 된다.

$$E(x_{qj}) = \frac{\left(x_{qj} - \sum_{i=1}^k w_i x_{ij}\right)^2}{\sigma_j^2} \quad (13)$$

여기서, x_{qj} 는 query pattern x_q 의 j번째 입력변수를 의미 하며, w_i 는 i번째 학습 패턴의 하중 값을 의미하고, x_{ij} 는 학습 패턴 x_i (다시 말하면, query pattern의 i번째 nearest neighbor를 의미한다)의 j번째 입력변수를 의미한다. σ_j 는 j번째 입력변수의 분산을 의미한다.

Reconstruction error는 query pattern의 nearest neighbor들이 각각의 하중 값을 이용하여 선형 결합 할 경우, query pattern을 얼마나 잘 모사 할 수 있는지를 나타낸다

고 말 할 수 있다. 식(13)에 이용된 하중 값 w_i 는 앞 장에서 설명한 FKNN의 유사도를 이용하여 결정되며, 식(10)을 적용하여 계산한다. 그림 1은 reconstruction error의 개략도를 나타낸다.

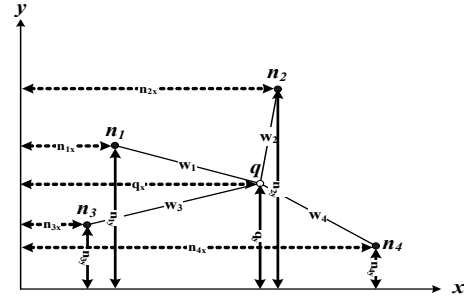


그림 1. Reconstruction Error
(q: query point, n: nearest neighbor)
Fig. 1. Reconstruction Error
(q: query point, n: nearest neighbor)

그림 1에서 reconstruction error는 아래 식(14-15)과 같이 구할 수 있다.

입력 변수 x와 관련된 reconstruction error :

$$E(x) = \frac{(q_x - (w_1 n_{1x} + w_2 n_{2x} + w_3 n_{3x} + w_4 n_{4x}))^2}{\sigma_x^2} \quad (14)$$

입력 변수 y와 관련된 reconstruction error :

$$E(y) = \frac{(q_y - (w_1 n_{1y} + w_2 n_{2y} + w_3 n_{3y} + w_4 n_{4y}))^2}{\sigma_y^2} \quad (15)$$

4.2 LLR을 이용한 입력 변수 선택

앞에 설명한 LLR 기법의 reconstruction error를 기반으로 하여 입력 변수를 선택하는 알고리즘은 다음과 같다.

- step 1. query pattern x_q 가 주어진다.
- step 2. query pattern x_q 의 k개의 nearest neighbors를 찾는다.
- step 3. 얻어진 k개의 nearest neighbor들의 하중 값을 계산하기 위해 query pattern x_q 과 nearest neighbor들 사이의 유사도를 식(10)을 이용하여 계산한다.
- step 4. step 3에서 얻은 nearest neighbor들과 각각의 하중 값을 이용하여 reconstruction error를 계산한다.
- step 5. step 4를 모든 입력변수의 경우에 적용한다.
- step 6. step 5에서 얻은 m개의 reconstruction error들 중 error값이 적은 순으로 입력 변수를 선택한다.

5. 퍼지 KNN과 LLR 기반 Local Weighted Polynomial 분류기 설계

앞에 설명한 것처럼 지역 학습 (local learning)은 비교적 단순한 구조를 가진 분류기로 특정 지역에만 해당하는 분류기를 학습시키고, 임의의 local area가 새롭게 주어지면 새로운 분류기를 학습 시키는 과정을 반복함으로써 단순한 구

조를 가진 분류기를 이용하여 전체 입력 공간에 적용할 수 있는 우수한 성능을 보이는 분류기를 설계할 수 있다는 장점을 가지고 있다.

본 논문에서는 임의의 지역 영역(local area)에서 적용하기 위한 간단한 구조를 가진 분류기로 linear function을 사용한다. 또한, 대부분의 분류 문제에서 대두되는 고차원 입력 공간 문제를 해결하기 위해 feature reduction 과정을 수행하게 되며, 이 과정을 위해 LLR의 평가 지수인 reconstruction error를 기반으로 feature를 선택한다. 퍼지 kNN 기법을 이용하여 지역 영역을 정의하고, 지역 영역 안에 존재하는 데이터 패턴들에 하중 값을 할당하고, LLR 기법의 reconstruction error를 기반으로 feature selection을 수행하고, weighted least square estimation 기법을 이용하여 지역 분류기(local classifier)인 선형 함수의 계수를 학습시킴으로써 제안한 새로운 분류기를 설계한다. 이상의 설계하는 과정을 정리하면 아래와 같다.

- step 1.** query pattern x_q 가 주어진다.
- step 2.** query pattern x_q 의 k개의 nearest neighbors를 찾는다.
- step 3.** 얻어진 k개의 nearest neighbor들의 하중 값을 계산하기 위해 query pattern x_q 과 nearest neighbor들 사이의 유사도를 식(10)을 이용하여 계산한다.
- step 4.** step 3에서 얻은 nearest neighbor들과 각각의 하중 값을 이용하여 reconstruction error를 계산한다.
- step 5.** step 4를 모든 입력변수의 경우에 적용한다.
- step 6.** step 5에서 얻은 m개의 reconstruction error들 중 error값이 적은 순으로 입력 S개의 변수를 선택한다.
- step 7.** step 6에서 선택된 S개의 변수를 가진 선형 함수의 계수를 추정하기 위하여 weighted least square estimation 기법을 적용하여 식(4)와 같이 계산한다.
- step 8.** step 7에서 추정된 계수를 가진 선형 함수에 query pattern을 대입하여 아래와 같은 렉 값을 얻는다.

$$\hat{y} = [1 \ x_{SF\{1\}} \ x_{SF\{2\}} \ \dots \ x_{SF\{S\}}] \cdot a_q \quad (16)$$

여기서, SF는 step 7에서 선택된 입력변수의 인덱스들의 집합을 의미한다.

- step 9.** step 8에서 얻은 출력 \hat{y} 값에 따라 query pattern이 속한 클래스를 아래식과 같이 결정한다.

$$y_q = \begin{cases} 1, & \hat{y} < 0 \\ 2, & \hat{y} > 0 \end{cases} \quad (17)$$

여기서, y_q 는 query pattern x_q 의 class label을 의미한다.

6. 실험 연구 및 결과 고찰

본 연구에서는 제안된 분류기의 성능을 평가하기 위하여 제안된 분류기를 모의 데이터 집합 (synthetic dataset)과 기계 학습 데이터 집합 (machine learning dataset)에 적용하여 분류기로서의 성능을 평가 및 분석한다. 선형 함수를 분류기로 채택함으로써 본 연구에서는 2-클래스 분류문제만을 다루며, 기계 학습 데이터 집합은 UCI machine learning repository로부터 획득한 5가지의 데이터 집합들을 이용하여 기존 논문에서 제안된 분류기 기법들과 비교 평가한다. 모든 데이터 집합은 7:3의 비율로 랜덤하게 학습 데이터와 테스트 데이터로 10번 분할되어 실험을 수행하게

된다. 제안된 제어기의 분류 성능 평가를 위한 성능 평가 지수는 식(18), (19)와 같이 패턴 분류기의 오분류 (misclassification rate) 비율로 한다.

$$\text{오분류율} = \frac{\sum_{i=1}^N f(y_i - \hat{y}_i)}{N} \times 100 \quad (18)$$

여기서, N은 데이터의 수를 의미한다.

$$f(y_i - \hat{y}_i) = \begin{cases} 0, & y_i = \hat{y}_i \\ 1, & y_i \neq \hat{y}_i \end{cases} \quad (19)$$

표 2. 제안된 분류기의 설계 파라미터

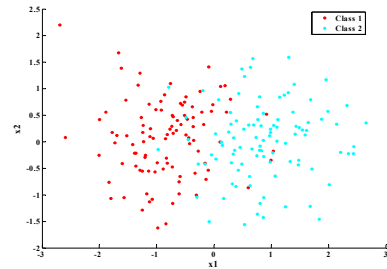
Table 2. Design parameters of the proposed classifier

Polynomial Order (O)	0(constant), 1(Linear), 2(Quadratic)
Number of neighbors	2, 3, 4, 5, 6, 7, 8, 9,10, 20, 30, 40, 50, 60, 70, 80, 90, 100
Fuzzification Coefficient (p)	In the range of 1.2~4.0 varying with step of 0.2
Shape Parameter (WF)	In the range of 0.2~2.0 varying with step of 0.2
Number of selected feature (SV)	10% ~ 70% of the whole input variables

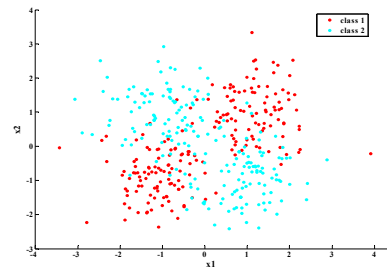
제안된 분류기를 설계할 때 필요한 파라미터들을 표 2에 나타내었다.

6.1 모의 데이터 집합

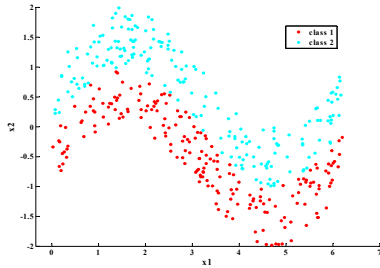
제안된 분류기의 성능 평가를 위해 그림 2와 같은 2차원 모의 데이터 집합에 제안된 분류기를 적용하여 실험을 수행하였다.



$$(a) \ m_1 = \begin{bmatrix} -1 \\ 0 \end{bmatrix}, \ m_2 = \begin{bmatrix} 1.0 \\ 0.0 \end{bmatrix}, \ \Phi = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix}$$



$$(b) \ m_1 = \begin{bmatrix} -0.5 \\ 0.5 \end{bmatrix}, \ m_2 = \begin{bmatrix} 0.5 \\ -0.5 \end{bmatrix}, \ m_3 = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix}, \\ m_4 = \begin{bmatrix} -0.5 \\ -0.5 \end{bmatrix}, \ \Phi = \begin{bmatrix} 0.2^2 & 0 \\ 0 & 0.2^2 \end{bmatrix}$$



(c) class 1: $x_2 = \sin(x_1) + U(\sigma)$,
 class 2: $x_2 = \sin(x_1) - U(\sigma)$

그림 2. 2차원 2 클래스 모의데이터 집합
 Fig. 2 Two dimensional two class synthetic datasets

그림 2-(c)에서 $U(\sigma)$ 는 균등 분포 함수를 의미하며, σ 는 균등 분포 함수의 분산을 의미한다.

$U(\sigma)$ 는 $[0, \sigma]$ 의 임의의 값이 균등한 확률분포를 가지는 함수를 의미한다. σ 는 균등 확률 분포 함수의 폭을 의미한다 할 수 있다.

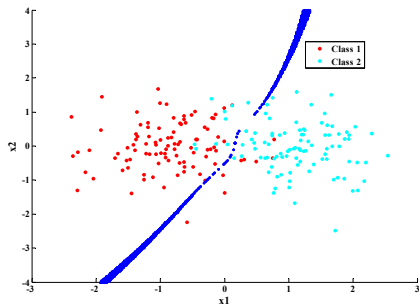
표 3은 각각의 모의 데이터 집합에 대해 제안된 분류기를 적용하여 얻은 결과들 중 가장 좋은 결과들을 보인다.

표 3. 각 모의 데이터 집합에 대한 제안된 분류기 성능
 Table 3. Classification performance of the proposed classifier for the synthetic data sets

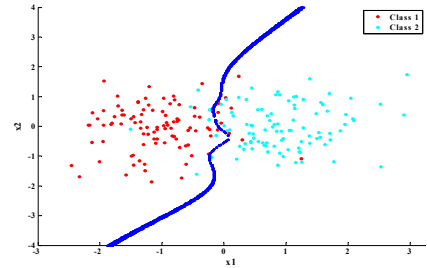
Datasets	O	k	m	Error Rate	
				Training data	Testing data
Dataset A	0	6	3.4	8.4±2.29	7.35±2.61
	1	4	3.6	6.9±1.24	6.9±1.24
	2	4	1.8	7.65±2.01	7.1±1.24
Dataset B	0	10	3.8	14.4±1.18	13.73±1.83
	1	3	2.0	13.08±2.09	13.73±2.04
	2	2	3.6	14.75±1.57	13.4±2.38
Dataset C	0	4	1.2	18.68±2.87	19.08±6.66
	1	5	2.4	5.21±0.78	4.92±2.06
	2	2	1.2	2.96±0.56	2.92±2.31

표 3에서 보인 바와 같이 대체적으로 선형 함수와 2차 다항식이 상수항으로 표현되는 분류기 보다는 분류 성능이 우수한 것으로 나타난다.

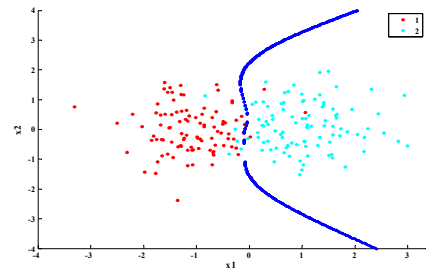
그림 3은 데이터 집합 A에 대한 제안된 분류기의 판정 경계를 보인다.



(a) Constant



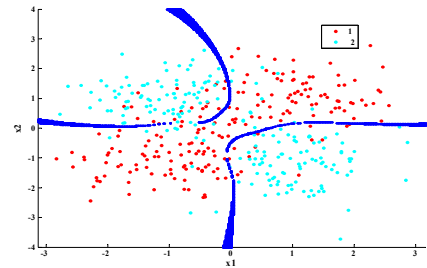
(b) Linear function



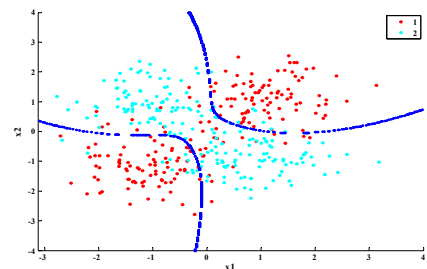
(c) quadratic polynomial

그림 3. Dataset A에 대한 분류기의 판정 경계
 Fig. 3. Boundary of classification for Dataset A

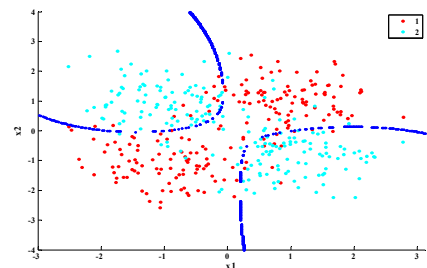
그림 4는 데이터 집합 B에 대한 제안된 분류기의 판정 경계를 보인다.



(a) Constant



(b) Linear function



(c) quadratic polynomial

그림 4. Dataset A에 대한 분류기의 판정 경계
Fig. 4. Boundary of classification for Dataset A

그림 5는 데이터 집합 C에 대한 제안된 분류기의 판정 경계를 보인다.

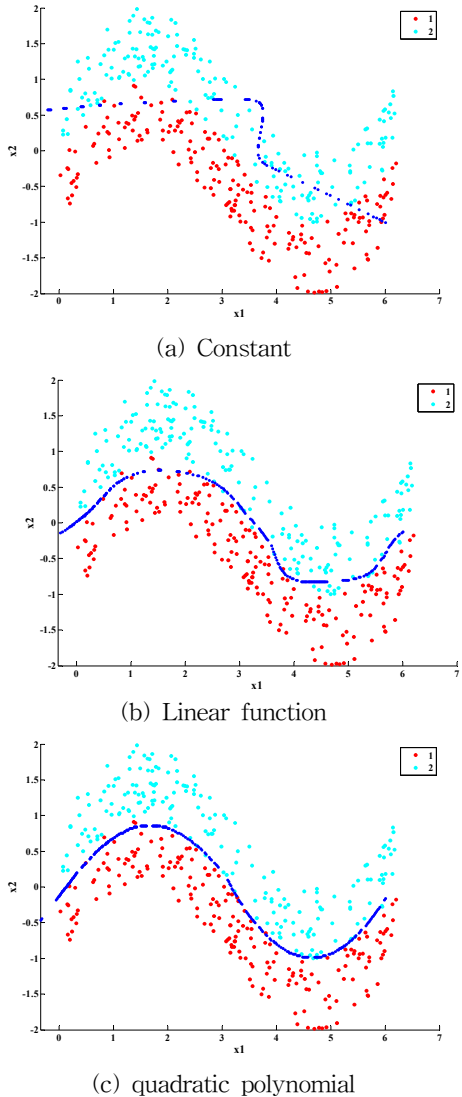


그림 5. Dataset A에 대한 분류기의 판정 경계
Fig. 5. Boundary of classification for Dataset A

6.2 기계 학습데이터

제안된 분류기의 패턴 분류 성능을 비교 평가하기 위해 기존 분류기의 benchmark data로 이용되는 기계 학습 데이터 집합에 적용하였다. 다양한 종류의 기계 학습 데이터 집합들 중에서 2 class 문제인 5개의 기계 학습 데이터 집합을 선정하여 실험을 수행 하였다. 선정된 데이터 집합에 대한 일반적인 설명을 표 4에 나열하였다.

표 4. 실험에 사용된 machine learning dataset
Table 4. Machine learning dataset used in the experiment

Dataset	Number of features	Number of patterns
Heart	13	270
Ionosphere	34	351
Pima	8	768
Sonar	60	208
WDBC	30	569

표 5는 5개의 기계 학습 데이터 집합의 경우에 제안된 분류기와 다른 잘 알려진 분류기들과의 성능을 비교한 것이다. Standard Neural Networks, 일반적인 linear discriminant analysis (LDA)와 Support vector machine (SVM) (<http://theoval.sys.uea.ac.uk/~gcc/svm/toolbox/>)은 MATLAB을 이용하여 구현된 것을 이용한 것이다.

C4.5 trees는 MATLAB classification toolbox (http://www.yom-tov.info/cgi-bin/list_uploaded_files.pl)를 이용하여 결과를 얻었다.

여기서, Standard NNs의 경우에 명시된 NN은 node의 수를 의미하며, hidden layer의 수는 1개인 뉴럴 네트워크를 사용하여 얻은 결과이다.

표 5에 나타난 바와 같이, 기존 분류기에 비하여 제안된 분류기가 Ionosphere dataset을 제외한 나머지 경우에 우수한 분류 성능을 보임을 알 수 있다.

표 5. 제안된 분류기와 다른 분류기와의 분류 성능 비교
Table 5. Comparison of the classification error of the proposed classifier and other methods

	Heart	Ionosphere	Pima	Sonar	WDBC
Proposed Classifier	12.84±2.2 7	8.76±2.32	23.70±1.7 7	12.10±4.7 0	1.93±1.10
Standard NNs[15]	21.23±4.9 7 NN = 5	18.95±3.3 7 NN=5	28.13±4.7 9 NN=15	34.19±8.7 3 NN=20	4.44±1.59 NN=5
SVM[15]	44.44±0.0	5.81±2.64	34.78±0.0	18.23±4.3 7	37.43±0.0
LDA[15]	32.84±10.08	20.19±4.2 1	34.43±1.7 7	24.52±3.9 4	3.27±1.0
C4.5trees [15]	29.63±7.2 2	8.38±1.54	32.43±3.1	36.45±7.5 7	8.25±2.26

여기서, Standard NNs의 경우에 명시된 NN은 node의 수를 의미하며, hidden layer의 수는 1개인 뉴럴 네트워크를 사용하여 얻은 결과이다.

표 5에 나타난 바와 같이, 기존 분류기에 비하여 제안된 분류기가 Ionosphere dataset을 제외한 나머지 경우에 우수한 분류 성능을 보임을 알 수 있다.

7. 결론

본 논문에서는 local learning의 장점을 살린 패턴 분류기를 설계하기 위하여 퍼지 k-Nearest Neighbors 기법과 Locally Linear Reconstruction 기법의 reconstruction error를 기반으로 하는 local 패턴 분류기를 설계하였다.

퍼지 kNN 기법을 패턴 분류에 적용하지 않고, 지역 영역

의 정의와 지역 영역 안에 존재하는 데이터 패턴들의 하중 값을 결정하는데 적용하였다. 즉, 퍼지 kNN과 같은 변형된 kNN 기법들은 높은 variance 특성을 보이며, 특히 variance가 크면 클수록 일반화 능력은 떨어지는 것으로 알려져 있는데, 이를 해결하기 위해 local learning 기법을 유지 하면서 낮은 variance 특성을 보이는 다항식 형태의 분류기 기법을 퍼지 kNN 기법에 적용하였다.

또한 패턴 분류 문제들 중에서, feature space의 차원이 높을 경우 패턴 분류기가 복잡하게 되는 현상과 패턴 분류기의 분류 성능이 떨어지는 현상이 발생하게 되는데, 이를 해결하기 위해 reconstruction error를 평가 함수로 정의하고 이 평가 함수에 따라 고차원 feature space에서 임의의 개수만큼의 feature를 선택하는 feature selection 방법도 접목하였다.

제안된 분류기의 패턴 분류 성능을 평가하기 위하여 기계 학습 데이터 집합에 적용한 결과 다른 분류기들에 비해 비교적 우수한 분류 성능을 보인다는 것을 알 수 있었다.

참 고 문 헌

[1] A.K. Jain, P. W. Duin, J. Mao, Statistical pattern recognition: a review, *IEEE trans. pattern Anal. Mach. Intell.* 22 (1) (2000) 4-37.

[2] C. L. Liu, H. Sako, Class-specific feature polynomial classifier for pattern classification and its application to hand written numeral recognition, *Pattern recognition* 39 (2006) 669-681.

[3] U. Krebel, J. Schurmann, Pattern classification techniques based on function approximation, Handbook of character recognition and document image analysis, *World Scientific*, Singapore (1997) 49-78

[4] J. Shurmann, *Pattern classification: a unified view of statistical and neural approaches*, Wiley Interscience, New York (1996).

[5] L. Holmstrom, P. Koistinen, J. Laaksonen, E. Oja, Neural and statistical classifiers-taxonomy and two case studies, *IEEE Trans. Neural Networks* 8 (1) (1997) 5-17.

[6] W. Lam, C. K. Keung, C. X. Ling, Learning good prototypes for classification using filtering and abstraction of instances, *Pattern Recognition* 35 (2002) 1491-1506

[7] S.T. Roweis, L.K. Saul, Nonlinear dimensionality reduction by locally linear embedding, *Science* 290 (5500) (2000) 2323-2326.

[8] P. S. Kang, S. Z. Cho, Locally linear reconstruction for instance-based learning, *Pattern Recognition* 41 (2008) 3507-3518.

[9] Maya R. Gupta, Member, Eric K. Garcia and Erika Chin, "Adaptive Local Linear Regression with Application to Printer Color Management," *IEEE Trans. on ImageProcessing*, 17 (6) (2008) 936-945

[10] Hiphung Leung, Yingsong Huang and Changxiu Cao, "Locally weighted regression for desulphurization intelligent decision system modeling,"

Simulation Modeling Practice and Theory, 12 (2004) 413 - 423

[11] C.C. Atkeson et al., "Locally weighted learning," *Artificial Intelligence Review* (Special Issue on Lazy Learning), 11 (1-5) (1997) 11 - 73

[12] L. Blum, P. Langley, Selection of relevant features and examples in machine learning, *Artif. Intell.* (1997) 245 -- 271.

[13] H. Liu, H. Motoda, *Feature Selection for Knowledge Discovery and Data Mining*, Kluwer Academic Publisher, Dordrecht, 1998.

[14] I. Guyon, A. Elisseeff, An introduction to variable and feature selection, *J. Mach. Learn. Res.* (2003) 1157 - 1182.

[15] B. J. Park, W. Pedrycz, S. K. Oh, Polynomial-based radial basis function neural networks (P-RBFNNs) and their application to pattern classification, *Applied Intelligence* (<http://www.springerlink.com/content/m452347279jj2943/> on-line available).

저 자 소 개



노석범(Roh Seok Beom)

1994년 : 원광대 제어계측공학과 졸업
 1996년 : 동 대학 컴퓨터 공학과 졸업(석사)
 2006년 : 동 대학 제어계측공학과 졸업(박사)
 2006년~2009년 : 원광대학교학술연구교수
 2009년~현재 : 대전대학교 Post-Doc

관심분야 : Machine Learning, Computational Intelligence, statistical learning, Pattern Recognition
 Phone : (063) 850-6344
 Fax : (063) 853-2196
 E-mail : nado@wonkwang.ac.kr



안태천(Ahn Tae Chon)

1976년 : 연세대 전기공학과 졸업
 1980년 : 동대학원 전기공학과 졸업(석사)
 1986년 : 동대학원 전기공학과 졸업(박사)
 1987년~1988년 : Uppsala 대학 방문교수
 1996년~1997년 : Georgia 공대 방문교수
 2008년~2009년 : Alberta 대학 방문교수
 1981년~현재 : 원광대학교 전자 및 제어 공학부 교수

관심분야 : System Modeling and Knowledge Discovery, Data Mining and Machine Learning, Bio-inspired Optimizations, Information Data Classifiers, Computational Intelligence
 Phone : (063) 850-6344
 Fax : (063) 853-2196
 E-mail : tcahn@wonkwang.ac.kr