

초기 피취벡터 설정을 통한 다중클래스 문제에 대한 최적 피취 추출 기법

최의선, 이철희
연세대학교 전기·컴퓨터공학과
Tel: (02)361-2779, E-mail: chulhee@yonsei.ac.kr

Optimal Feature Extraction for Multiclass Problems through Proper Choice of Initial Feature Vectors

Euisun Choi, Chulhee Lee
Dept. of Electrical and Computer Engineering, Yonsei Univ.
134 Shinchon-Dong Seodaemun-Gu Seoul, Korea

Abstract

In this paper, we propose an optimal feature extraction for multiclass problems through proper choice of initial feature vectors. Although numerous feature extraction algorithms have been proposed, those algorithms are not optimal for multiclass problems. Recently, an optimal feature extraction algorithm for multiclass problems has been proposed, which provides a better performance than the conventional feature extraction algorithms. In this paper, we improve the algorithm by choosing good initial feature vectors. As a result, the searching time is significantly reduced. The chance to be stuck in a local minimum is also reduced.

1. 서 론

피취추출(feature extraction)은 패턴분류나 패턴인식 문제에 있어서 분류기(classifier)의 성능을 향상시키고 복잡도(complexity)를 감소시켜 효율적인 데이터 처리를 가능하게 한다. 이와 같은 이유로 피취추출은 패턴분류나 패턴인식 문제에 있어서 매우 중요하게 다루어져왔던 주제이며 오랫동안 많은 연구자들에 의해 폭넓게 연구되어 왔다 [1-5].

일반적으로 패턴인식 시스템에서는 입력데이터의 효율적인 저장과 패턴 분류단(classification stage)으로의

실시간 데이터 전송 등을 위해 데이터의 크기를 줄이는 것이 바람직하는데 이 경우 원래의 데이터로부터 분류에 필요한 정보들만을 추출하게 된다. 즉 피취추출은 고차원의 데이터를 저차원의 데이터로 변환시키는 것으로 볼 수 있다 [1, 2]. 효율적인 패턴인식 시스템 구현을 위해서는 가능한 최소의 피취들을 사용해야 하고 이와 함께 분류에 필요한 정보의 손실을 최소화하여야 한다. 이러한 관점에서 전체 피취공간을 탐색하여 분류오차(classification error)를 최소화하는 순차탐색(sequential search) 및 전체탐색(global search) 알고리즘은 임의의 초기 피취벡터를 이용하여 분류오차가 최소가 되는 방향으로 초기 피취벡터를 갱신한다 [3]. 그러나 실제 알고리즘 적용 시 초기 피취벡터 선택에 따라 소요되는 계산시간이 증가하고, 또한 지역극소(local minimum)에 빠지는 문제가 발생한다.

본 논문에서는 이러한 문제점을 보완하기 위하여 기존의 피취추출 방법으로 구해진 피취를 초기 피취벡터로 사용하여 성능을 개선하는 방법을 제안한다. 제안된 피취 추출방법은 기존의 방법보다 안정된 성능을 나타낸다.

II. 연구 배경 및 이론

패턴분류 문제에서 취급하는 데이터들은 일반적으로 서로 다른 통계적 분포를 가지는 다차원 데이터이며 N 차원 유클리드 공간에서 다음과 같이 벡터로 표현할

수 있다.

$$\bar{\mathbf{x}} = [x_1, x_2, \dots, x_N]^T$$

N 차원의 데이터 $\bar{\mathbf{x}}$ 를 $M(M < N)$ 차원의 데이터 $\bar{\mathbf{y}}$ 로 변형하기 위해서 매핑함수 $W(\bar{\mathbf{x}})$ 를 생각할 수 있다. $W(\bar{\mathbf{x}})$ 는 벡터함수로서 고차원의 벡터 $\bar{\mathbf{x}}$ 를 저차원의 벡터 $\bar{\mathbf{y}}$ 로 매핑시키며 주어진 문제에 대하여 적절한 피취를 얻기 위해 모든 가능한 벡터함수들의 집합 $\{W(\bar{\mathbf{x}})\}$ 에 대하여 결정기준함수 $J\{W(\bar{\mathbf{x}})\}$ 를 최적화시킨다.

$$W(\bar{\mathbf{x}}) = \text{MAX}[J\{W(\bar{\mathbf{x}})\}] \text{ or } \text{MIN}[J\{W(\bar{\mathbf{x}})\}]$$

$$\bar{\mathbf{y}} = W(\bar{\mathbf{x}})$$

$$\bar{\mathbf{y}} = [y_1, y_2, \dots, y_M]^T$$

매핑함수 $W(\bar{\mathbf{x}})$ 가 선형일 경우 일반적으로 선형변환 행렬 Φ 를 이용하여 나타낼 수 있다 [4].

$$\bar{\mathbf{y}} = \Phi^T \bar{\mathbf{x}}$$

$$\Phi = [\phi_1, \phi_2, \dots, \phi_M] \text{ (} N \times M \text{ matrix)}$$

순차탐색 및 전체탐색 알고리즘에서는 최소의 분류에러를 가지는 피취벡터 행렬 Φ 를 구하기 위해 임의의 피취벡터 ϕ_i ($i=1, 2, \dots, M$)를 사용하여 패턴분류를 수행하고 분류오차를 계산한 다음 피취벡터를 조금씩 변화시켜 계산된 분류오차와의 차이가 가장 큰 방향으로 피취벡터를 갱신하게 된다 [3].

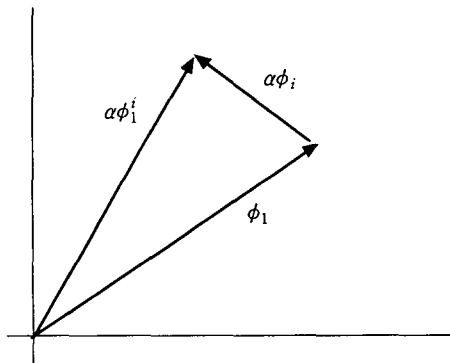


그림 1. 초기 피취벡터의 이동.

A. 순차 탐색

N 차원 유클리드 벡터공간에서 직교기저 벡터집합 $\Phi_N = \{\phi_1, \phi_2, \dots, \phi_N\}$ 을 생각한다. 여기서 ϕ_i 는 $N \times 1$ 단위 열벡터이다. 먼저 ϕ_1 을 초기 피취벡터로 가정하여 분류오차를 계산한다. 그 다음 피취벡터 ϕ_1 을 식 (1)과 같이 조금씩 이동시킨다.

$$\phi_1^i = \phi_1 + \alpha \phi_i \quad (i=2, \dots, N) \quad (1)$$

여기서 α 는 스텝 크기를 나타내는 상수이다. 그림 1은 위의 과정을 보여준다. 오차의 변화율(gradient)은 다음의 식 (2)와 같은 방법으로 계산된다.

$$r_i = \frac{\Delta \epsilon}{\alpha} = \frac{\epsilon(\phi_1^i) - \epsilon(\phi_1)}{\alpha} \quad (2)$$

여기서 $\epsilon(\phi_1^i)$ 와 $\epsilon(\phi_1)$ 는 피취벡터 ϕ_1^i 과 ϕ_1 을 패턴 분류에 사용하였을 때의 분류오차이다. 이와 같은 과정을 ϕ_i ($i=2, \dots, N$) 벡터들에 대해 반복하여 오차의 변화율 r_i ($i=2, \dots, N$)를 계산한다. 최종적으로 피취벡터 ϕ_1 은 다음의 식 (3)과 같이 갱신된다.

$$\phi_{1, \text{updated}} = \phi_1 + \beta \sum_{i=2}^N (-r_i \phi_i) \quad (3)$$

여기서 β 는 상수이다. 이와 같은 과정을 거쳐 구해진 피취벡터 $\phi_{1, \text{updated}}$ 는 다른 기저벡터들과 선형적으로 독립이 아니므로 그람-슈미트(Gram-Schmidt) 방법을 적용하여 전체 기저집합의 직교성을 유지한다 [6]. 식 (1),(2),(3)의 과정을 갱신된 피취벡터 $\phi_{1, \text{updated}}$ 을 사용하여 계산된 분류오차가 문턱치보다 작아질때까지 반복하여 최종적으로 피취벡터를 구한다. 추가로 피취벡터를 구할 경우 이미 구해진 피취벡터와 함께 최소의 분류오차를 얻을 수 있는 피취벡터를 위와 동일한 과정을 통해 구한다.

B. 전체 탐색

전체탐색 방법은 추가로 피취벡터를 구할 경우 이미 구해진 피취벡터를 사용하지 않고 임의의 새로운 초기 피취벡터 집합을 설정한다. 즉 N 차원 유클리드 벡터

공간에서 M 개의 피취벡터를 추출할 경우 임의의 초기 피취벡터 집합 $\Phi_M = \{\phi_1, \phi_2, \dots, \phi_M\}$ ($M < N$)을 가정한다. 그 다음 피취벡터들 사이의 직교성을 유지하며 각각의 피취벡터들을 나머지 벡터 $\phi_{M+1}, \phi_{M+2}, \dots, \phi_N$ 쪽으로 조금씩 움직인다. 이 경우 $(N-M)$ 개의 새로운 피취벡터 집합 $\Phi_{i,M}^j$ 이 생성된다.

$$\Phi_{i,M}^j = \{\phi_1, \phi_2, \dots, \phi_j^i, \dots, \phi_M\}$$

$$\phi_j^i = \phi_j + \alpha \phi_i \quad (i = M+1, \dots, N \quad j = 1, \dots, M)$$

새롭게 구해진 피취벡터 집합을 이용하여 분류오차를 계산하고 다음의 식 (4)와 같이 분류오차의 변화율을 계산한다.

$$r_j^i = \frac{\Delta \varepsilon}{\alpha} = \frac{\varepsilon(\Phi_{i,M}^j) - \varepsilon(\Phi_{i,M})}{\alpha} \quad (4)$$

이 과정을 $\Phi_{i,M}^j$ ($i = M+1, \dots, N$)에 대하여 반복하고 오차의 변화율 r_j^i ($i = M+1, \dots, N$)를 계산한다. 최종적으로 피취벡터 ϕ_j 는 식 (3)과 같이 $\phi_{j, updated}$ 로 갱신되며 동일한 과정을 피취벡터 ϕ_j ($j = 1, \dots, M$) 모두에 대해 적용한다. 이러한 과정을 갱신된 피취벡터 집합을 사용하여 계산한 분류오차가 문턱치보다 작을 때까지 반복하여 최종적으로 피취벡터 집합 $\Phi_{M, updated}$ 을 구한다. 전체탐색 알고리즘의 단점은 계산량이 순차탐색 알고리즘에 비해 많다는 점이다.

III. 초기 피취벡터 설정

초기 피취벡터의 설정은 순차탐색 및 전체탐색 알고리즘의 성능에 영향을 미치는 요소로서 적절한 초기 피취벡터의 설정은 탐색 과정에서 반복 횟수 및 지역 극소에 도달할 가능성을 감소시킨다. 따라서 본 논문에서는 초기 피취벡터의 선택 시, 기존의 피취추출 알고리즘으로 구한 피취벡터를 초기 피취벡터로 설정하여 성능을 개선시키는 방법을 제안한다. 표 2-4는 순차탐색 방법 적용 시 여러 가지 초기 피취벡터를 사용하여 구한 피취 벡터 중, 첫 번째 피취벡터를 사용할 때의 성능을 비교한 결과이다. 한 개의 피취벡터를 추출하는 경우는 순차탐색과 전체탐색 방법이 동일하다. 기존의 피취추출 방법으로는 canonical analysis 방법과 주성분 분석(principal component analysis)방법을 사용하였으며, 사용한 데이터는 실제 원격 탐사된 데

이터로서 5개의 클래스로 구성되어 있다 [7]. 표 1에 클래스 정보를 나타냈다. 표 2에서 알 수 있듯이 대부분의 경우 서로 다른 초기 피취벡터를 이용하여 추출된 피취벡터를 사용하여 패턴분류를 수행하는 경우 비슷한 수준의 분류정확도를 얻을 수 있다. 그러나 반복 횟수는 초기 피취벡터에 따라 큰 차이를 보이고 있다.

표 1. 클래스 정보.

| Group | Species | Date | No. of sample (train) |
|-------|---------------|----------|-----------------------|
| 1 | SPRING WHEAT | 78.05.15 | 474(300) |
| | SUMMER FALLOW | 77.06.26 | 643(300) |
| | WINTER WHEAT | 77.10.18 | 660(300) |
| | WINTER WHEAT | 77.06.26 | 677(300) |
| | WINTER WHEAT | 77.03.08 | 691(300) |
| 2 | SPRING WHEAT | 78.05.15 | 474(300) |
| | SUMMER FALLOW | 77.06.26 | 643(300) |
| | WINTER WHEAT | 77.05.03 | 657(300) |
| | WINTER WHEAT | 77.10.18 | 660(300) |
| | WINTER WHEAT | 77.06.26 | 677(300) |
| 3 | SPRING WHEAT | 78.06.02 | 515(300) |
| | SPRING WHEAT | 78.07.26 | 515(300) |
| | SUMMER FALLOW | 77.06.26 | 643(300) |
| | WINTER WHEAT | 77.05.03 | 657(300) |
| | WINTER WHEAT | 77.03.08 | 691(300) |

표 2. Group 1에 대한 순차탐색의 성능평가.

| 순차탐색 | Initial vector | Accuracy(%) | Iteration |
|-------------------|------------------------|-------------|-----------|
| sequential search | 단위행렬(I) | 73.27 | 371 |
| | canonical analysis(CA) | 72.80 | 172 |
| Group 1 | PCA | 72.53 | 231 |
| | random | 72.40 | 383 |
| CA | - | 67.9 | - |
| PCA | - | 50.2 | - |

표 3. Group 2에 대한 순차탐색의 성능평가.

| 순차탐색 | Initial vector | Accuracy(%) | Iteration |
|-------------------|------------------------|-------------|-----------|
| sequential search | 단위행렬(I) | 65.87 | 616 |
| | canonical analysis(CA) | 67.40 | 254 |
| Group 2 | PCA | 65.73 | 460 |
| | random | 36.87 | 10 |
| CA | - | 65.7 | - |
| PCA | - | 49.1 | - |

표 4. Group 3에 대한 순차탐색의 성능평가.

| 순차탐색 | Initial vector | Accuracy(%) | Iteration |
|-------------------|------------------------|-------------|-----------|
| sequential search | 단위행렬(I) | 53.40 | 109 |
| | canonical analysis(CA) | 59.33 | 264 |
| | PCA | 60.07 | 134 |
| Group 3 | random | 59.67 | 276 |
| CA | - | 47.8 | - |
| PCA | - | 44.4 | - |

즉, 표 2-4에서 볼 수 있듯이 기존의 피취추출방법으로 구해진 피취벡터를 초기 피취벡터로 사용하는 경우가 단위행렬 I와 랜덤 벡터를 사용했을 때보다 상대적으로 반복횟수가 적은 것을 볼 수 있다. 이는 순차탐색 방법 적용 시, 자유도가 가장 큰 최초의 피취벡터를 구하는 경우 계산상의 복잡도를 크게 줄일 수 있음을 의미한다. 표 3과 4에서는 각각 랜덤벡터와 단위행렬 I를 초기벡터로 사용하는 경우 지역 극소에 도달했음을 알 수 있다. 임의의 벡터를 초기벡터로 이용하는 경우, 기존 피취추출 방법으로 구해진 피취벡터를 초기벡터로 이용하는 경우 보다 지역극소에 도달할 가능성이 높은 것으로 관찰되었다. 본 논문에서 제안한 방법은 기존 피취추출 알고리즘의 성능을 향상시키는데 적용될 수 있다. 그림 2는 전체탐색 알고리즘을 이용하여 피취를 추출하는 경우 피취 수를 증가시킬 때, 기존 알고리즘인 canonical analysis방법으로 구해진 피취벡터를 초기벡터로 사용하여 최적화한 경우로 분류정확도 향상을 보여준다.

IV. 결 론

본 논문에서는 적절한 초기 피취벡터 설정을 통해 전체 피취공간을 탐색하여 직접적으로 분류오차를 최소화하는 피취 추출방법인 순차탐색 및 전체탐색 알고리즘의 성능을 개선하는 방법을 제안하였다. 기존의 알고리즘으로 구해진 피취를 초기벡터로 사용하여 피취를 추출할 경우 반복 횟수와 지역극소에 도달하는 가능성을 감소시키므로 향상된 성능을 보여 주었다. 또한 제안된 기법을 사용하여 다중 클래스 문제에 대해 기존 피취추출 알고리즘의 성능도 향상시킬 수 있음을 확인하였다.

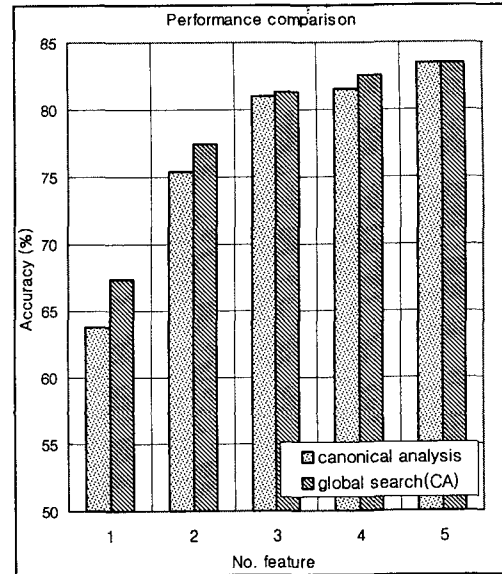


그림 2. 기존 피취추출 방법의 성능향상.

참 고 문 헌

- [1] K. Fukunaga, *Introduction to Statistical Pattern Recognition*. NewYork: Academic Press, pp.225-226, 1990.
- [2] Y. Mallet, D. Coomans, J. Kautsky and Oliver De Vel "Classification Using Adaptive Wavelets for Feature Extraction," *IEEE Trans. Computer*, vol. 19, no. 10, pp. 1058-1066, 1997.
- [3] C. Lee, E. Choi and J. Kim, "Optimal Feature Extraction for Normally Distributed Data," *SPIE AeroSense '98*, Orlando, Florida, pp. 223-232, 1998.
- [4] J. A. Richards, *Remote Sensing Digital Image Analysis*. Springer-Verlag, 1993.
- [5] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*. John Wiley & Sons, 1973.
- [6] C. G. Cullen, *Matrices and Linear Transformation*. Addison_wesley Publishing Company, 1972.
- [7] L.L. Biel and et. al., "A Crops and Soils Data Base For Scene Radiation Research," *Proc. Machine Process. of Remotely Sensed Data Symp., West Lafayette, Indiana*, 1982.