가우시안 분포의 다중클래스 데이터에 대한 최적 피쳐추출 방법

최의선, 이철희
연세대학교 전자공학과
Tel: (02)361-2779, E-mail: acuaris@feature.yonsei.ac.kr

Optimal feature extraction for normally distributed multiclass data

Euisun Choi, Chulhee Lee
Dept. of Elec. Eng., Yonsei Univ.
134 Shichondong Seodaemungu Seoul, Korea

Abstract

In this paper, we propose an optimal feature extraction method for normally distributed multiclass data. We search the whole feature space to find a set of features that give the smallest classification error for the Gaussian ML classifier. Initially, we start with an arbitrary feature vector. Assuming that the feature vector is used for classification, we compute the classification error. Then we move the feature vector slightly and compute the classification error with this vector. Finally we update the feature vector such that the classification error decreases most rapidly. This procedure is done by taking gradient. Alternatively, the initial vector can be those found by conventional feature extraction algorithms. We propose two search methods, sequential search and global search. Experiment results show that the proposed method compares favorably with the conventional feature extraction methods.

Ⅰ. 서 론

피쳐추출(feature extraction)은 여러 응용분야에서 효율적인 데이터처리를 가능하게 하고 특히 패턴분류나 패턴인식 문제에 있어서 매우 중요하게 다루어져왔던 주제이다 [1-5].

선형 피쳐추출(linear feature extraction)은 벡터공간에서 데이터의 특성을 가능한 그대로 유지하면서 고차원의 데이터를 저차원의 데이터로 선형변환(linear transformation)시키는 것으로 볼 수 있다 [1].

패턴분류(pattern classification)문제의 경우 일반적으로 원래의 피쳐보다 적은 수의 피쳐를 사용한 분류는 분류정확도(classification accuracy)측면에서 효율이 떨어진다. 그러나 분류비용(classification cost) 측면에서 볼 때 많은 수의 피쳐를 사용하는 것은 비효율적이다. 간단한 선형 분류기(linear classifier)의 경우 계산속도는 사용된 피쳐수에 비례하고, 많은 경우 자주 사용되는 가우시안 최대우도 분류기(Gaussian ML classifier)는 피쳐수의 제곱에 비례한다 [2]. 따라서 데이터를 분류하는데 기여도가 낮은 피쳐들을 제거하거나 또는 데이터를 새로운 분류좌표계로 선형변환 시키기 위한 피쳐집합을 추출하는 과정이 필요하다.

대부분의 많은 피쳐추출 알고리즘들은 두 개의 클래스에 대하여 데이터의 통계적 특성에 기초한 결정기준함수를 사용한다. 예를 들어 Fisher의 알고리즘은 가장 큰 클래스 분리도(separability)를 얻을 수 있는 결정기준함수를 사용하며[3] 최근 발표된 결정경계 피쳐추출 알고리즘은 결정경계 피쳐행렬을 결정기준함수로 사용하고 있다 [4]. 또한 다중클래스(multiclass) 문제와 관련하여 기존의 알고리즘들이 사용하는 결정기준함수들은 확장되어 사용되는데 일반적으로 두 개의 클래스를 초점으로 만들어진 피쳐추출 방법을 다중클래스 문제에 적용시킬 경우 엄밀한 의미에서 최적의 해를 기대하기 어렵다.

따라서 본 논문에서는 패턴분류 문제와 관련하여 다중클래스 데이터에 대한 피쳐의 신뢰도를 높이며 또한 기존의 방법으로 구해진 피쳐에 대해서도 성능을 향상시킬 수 있는 피쳐추출 알고리즘을 제안한다. 제안된 알고리즘은 sequential search와 global search 두 가지 방식이다.

Ⅱ. 선형피쳐추출 개요와 관련연구

패턴분류 문제에서 취급하는 데이터들은 일반적으로

서로 다른 통계적 분포를 가지며 다중 스펙트럴(multispectral) 정보를 포함하는 클래스를 형성하고 있다. 다차원 벡터공간에서 각각의 클래스데이터들은 다음과 같이 표현될 수 있다.

$$X = \Phi Y$$

여기서

$$X = \{x_1, x_2, \cdots, x_N\}^T$$
$$\Phi = \{\phi_1, \phi_2, \cdots, \phi_N\}$$
$$Y = \{y_1, y_2, \cdots, y_N\}^T$$

이고 Φ는 직교기저 행렬이며 ϕ_i는 $N\times 1$단위 기저벡터이다. Y의 원소들은 본래의 데이터를 구성하는 피춰(feature)들이다. 선형피춰추출은 효율적인 패턴분류를 위해 행렬 Φ를 적절히 변형시킴으로써 이루어진다. 예를 들어 N차원 유클리드 벡터공간의 데이터 X를 차원이 $M(M<N)$인 벡터공간의 데이터 Y_M으로 변형시킬 경우 행렬계수(rank)가 M인 직교기저행렬 Φ_M을 구한다. 즉

$$Y_M = \Phi_M^T X$$

여기서

$$\Phi_M = \{\phi_1, \phi_2, \cdots, \phi_M\}$$
$$Y_M = \{y_1, y_2, \cdots, y_M\}$$

이다.

Principal component analysis를 이용한 피춰추출은 전체 데이터들에 대한 평균과 공분산행렬을 구하고 이를 바탕으로 고유치와 고유벡터를 계산한다. 그리고 고유벡터들 중 고유치가 큰 고유벡터를 피춰벡터로 선택한다 [2]. 그러나 이 경우 클래스 각각에 대한 분산정보는 포함되지 않기 때문에 일반적으로 패턴분류에는 적합하지 않다. 이와 같은 문제에 대하여 canonical analysis방법은 클래스들간의 분산과 클래스 각각의 분산들을 동시에 고려하여 다음 식과 같은 결정기준함수의 값을 최대로 하는 벡터 d를 구하게 된다.

$$\frac{\sigma_b^2}{\sigma_w^2} = \frac{\text{between categories variance}}{\text{within categories variance}} = \frac{d_t \sum_b d}{d_t \sum_w d}$$

여기서

$$\sum_w = \sum_i P(\omega_i)\sum_i \text{ (within-scatter matrix)}$$

$$\sum_b = \sum_i P(\omega_i)(M_i - M_0)(M_i - M_0)^t$$

(between-class scatter matrix)

$$M_0 = \sum_i P(\omega_i) M_i$$

이다.

M_i, $\sum_i$, $P(\omega_i)$들은 각각 클래스 ω_i의 평균벡터와 공분산 행렬 그리고 선행(prior) 확률이다. Canonical analysis 방법은 대부분의 패턴분류 문제에 있어서 효과적이지만 클래스들간의 평균차이가 크지 않을 때에는 이 방법으로 구해진 피춰벡터는 신뢰도가 떨어진다.

III. 제안된 알고리즘

제안된 두 가지 알고리즘은 먼저 임의의 피춰벡터를 사용하여 패턴분류를 수행하고 분류오차(classification error)를 계산한 다음 피춰벡터를 조금씩 변화시켜 계산된 분류오차와의 차이가 가장 큰 방향으로 피춰벡터를 갱신하게 된다.

A. Sequential search

N차원 유클리드 벡터공간에서 직교기저 벡터집합 $\Phi_N = \{\phi_1, \phi_2, \cdots, \phi_N\}$을 생각한다. 여기서 ϕ_i는 $N\times 1$ 단위 열벡터이다. 먼저 ϕ_1을 초기 피춰벡터로 가정하여 분류오차(classification error)를 계산한다. 그 다음 피춰벡터 ϕ_1을 식 (1)과 같이 조금씩 이동시킨다.

$$\phi_1^i = \phi_1 + \alpha\phi_i \quad (i=2, \cdots, N) \tag{1}$$

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

$$r_i = \frac{\triangle\varepsilon}{\alpha} = \frac{\varepsilon(\phi_1^i) - \varepsilon(\phi_1)}{\alpha} \tag{2}$$

여기서 $\varepsilon(\phi_1^i)$와 $\varepsilon(\phi_1)$는 피춰벡터 ϕ_1^i 과 ϕ_1을 패턴분류에 사용하였을 때의 분류오차이다. 이와 같은 과정을 ϕ_i $(i=2, \cdots, N)$ 벡터들에 대해 반복하여 오차의 변화율 r_i $(i=2, \cdots, N)$를 계산한다. 최종적으로 피춰벡터 ϕ_1은 다음의 식 (3)과 같이 갱신된다.

$$\phi_{1,updated} = \phi_1 + \beta\sum_{i=2}^{N} r_i\phi_i \tag{3}$$

여기서 β는 상수이다. 이와 같은 과정을 거쳐 구해진 피춰벡터 $\phi_{1,updated}$는 다른 기저벡터들과 선형적으로 독립이 아니므로 그람-슈미트(Gram-Schumidt) 방법을 적용하여 전체 기저집합의 직교성을 유지한다 [5]. 식 (1),(2),(3)의 과정을 갱신된 피춰벡터 $\phi_{1,updated}$을 사용하여 계산된 분류오차가 더 이상 변하지 않을 때까지 반복하여 최종적으로 피춰벡터를 구한다.

식 (1)에서 피춰벡터 ϕ_1를 움직여 가는 과정에서 전체 벡터공간의 차원이 N차원이므로 $N-1$의 자유도(degree of freedom)가 존재하게 된다. 추가로 피춰벡터를 구할 경우 이미 구해진 피춰벡터와 함께 최소의 분류오차를 얻을 수 있는 피춰벡터를 위와 동일한 과

정을 통해 구한다. 하지만 이 경우엔 자유도가 하나씩 감소하게 될 것이다. 따라서 sequential search는 다음에 소개할 global search에 비해 계산량이 적다.

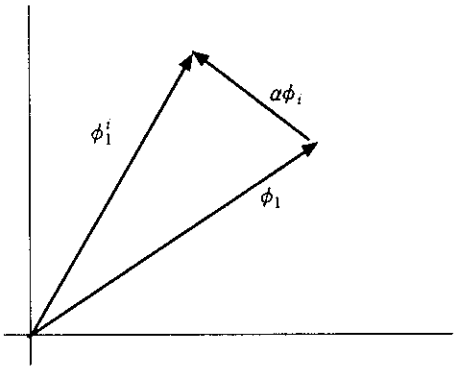

그림 1. Sequential search.

B. Global search

위의 sequential search는 비교적 간단하고 계산속도가 빠르다는 장점이 있으나 추가로 피춰벡터를 구할 경우 이미 구해진 피춰벡터를 그대로 사용하기 때문에 가장 먼저 구해진 피춰벡터가 최적의 피춰벡터가 아니라면 결과적으로 볼 때 최적의 해가 될 수 없다. 이러한 문제를 해결하기 위해 global search에서는 피춰수를 증가시킬 경우 이전에 구해진 피춰벡터를 사용하지 않고 임의의 새로운 초기피춰벡터 집합을 설정한다. 즉 N차원 유클리드 벡터공간에서 M개의 피춰벡터를 추출할 경우 먼저 임의의 초기피춰벡터 집합 $\Phi_M=\{\phi_1,\phi_2,\cdots,\phi_M\}$ $(M<N)$을 가정한다. 그 다음 피춰벡터들 사이의 직교성을 유지하며 각각의 피춰벡터들을 나머지 벡터 $\phi_{M+1},\phi_{M+2},\ldots,\phi_N$ 쪽으로 조금씩 움직이며 패턴분류를 수행한다. 이 경우 $(N-M)\times M$의 자유도가 존재하게 된다. 예를 들어 피춰벡터집합 Φ_M 의 벡터 ϕ_j를 식 (1)과 같이 움직일 때 $(N-M)$개의 새로운 피춰벡터집합 $\Phi_{j,M}^i$ $(i=M+1,\ldots,N)$이 구해진다. 즉

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

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

이다. 새롭게 구해진 피춰벡터집합을 이용하여 분류오차(classification error)를 계산하고 다음의 식 (4)와 같이 분류오차의 변화율(gradient)을 계산한다.

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

여기서 $\varepsilon(\Phi_{j,M}^i)$과 $\varepsilon(\Phi_{j,M})$는 각각 피춰벡터집합 $\Phi_{j,M}^i$과 $\Phi_{j,M}$이 사용되었을 때의 분류오차이다. 이 과정을 $\Phi_{j,M}^i$ $(i=M+1,\ldots,N)$에 대하여 반복하고 오차의 변화율 r_j^i $(i=M+1,\ldots,N)$를 계산한다. 최종적으로 피춰벡터 ϕ_j는 식 (3)과 같이 $\phi_{j,updated}$로 갱신되며 동일한 과정을 피춰벡터 ϕ_j $(j=1,\ldots,M)$ 모두에 대해 적용한다. 이 경우 sequential search와 마찬가지로 그람-슈미트(Gram-Schmidt)방법을 적용하여 전체 N차원 벡터공간에서의 직교기저집합 Φ_N의 직교성을 유지한다. 위의 과정을 갱신된 피춰벡터집합을 사용했을 때 계산되는 분류오차가 더 이상 변하지 않을 때까지 반복하여 최종적으로 피춰벡터집합 $\Phi_{M,updated}$ 을 구한다.

예를 들어 3차원 벡터공간에서 두 개의 피춰벡터를 추출한다고 가정하면 다음과 같은 피춰벡터집합 각각에 대하여 식(4)와 같이 분류오차의 변화율을 계산하여 피춰벡터 ϕ_1과 ϕ_2를 구한다.

$$\{\phi_1+\alpha\phi_3,\phi_2\}$$

$$\{\phi_1,\phi_2+\alpha\phi_3\}$$

위의 과정을 그림 2에 나타내었다.

Global search 알고리즘을 사용할 때의 장점은 여러 개의 피춰벡터를 추출하는 경우 sequential search는 이미 구해진 피춰벡터를 그대로 사용하지만 global search는 그러한 제약을 받지 않는다는 점이다. 그러나 계산량은 sequential search에 비해 많다는 단점이 있다.

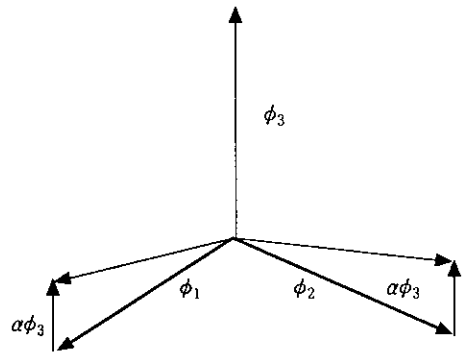

그림 2. Global search.

Ⅳ. 실험결과 및 고찰

본 논문에서 제안하는 피춰추출 알고리즘의 성능을 평가하기 위해 ML classifier를 이용하여 분류정확도(classification accuracy)를 비교하였다. 제안된 알고리즘과 성능비교를 위해 사용된 기존의 피춰추출 알고리즘은 canonical analysis방법과 principal component analysis방법이다. 실험에 사용된 데이터는 원격 탐사된 실제데이터이며[6] 5개의 클래스들로 이루어져 있

표 1. 실험에 사용한 클래스데이터.

Group	Species	Date	No. of sample
1	GRAIN SORGHUM	76. 9. 28	277
	NATIVE GRASS PAS	78. 6. 2	209
	OTHER CROPS	78. 8. 16	199
	NATIVE GRASS PAS	77. 10. 18	183
	OATS	78. 9. 21	182

다. 표 1은 클래스 정보를 보여준다.

실험을 위해 클래스의 밴드 수를 줄여 데이터의 차원을 5로 하였으며 각 클래스분포에 대한 통계 파라미터를 추정하여 샘플 수를 1000개로 발생시켰다. 제안한 알고리즘에서 α 와 β 의 값은 각각 0.1과 0.5이다.

분류정확도를 비교한 결과를 그림 3에 나타냈다. 실험결과 제안된 알고리즘이 기존의 알고리즘들보다 우수한 성능을 보이고 있음을 알 수 있다. 또한 여러 개의 피춰를 추출할 경우 global search방법이 sequential search 방법보다 더 좋은 성능을 보이고 있다. Canonical analysis방법과 principal component analysis방법으로 구해진 피춰를 초기 피춰벡터로 사용하여 실험한 결과를 그림 4.에 나타냈다. 이 경우 임의의 피춰벡터를 초기벡터로 사용했을 때보다 계산시간이 많이 줄어들었으며 성능면에서도 기존 알고리즘에 비해 향상되었다.

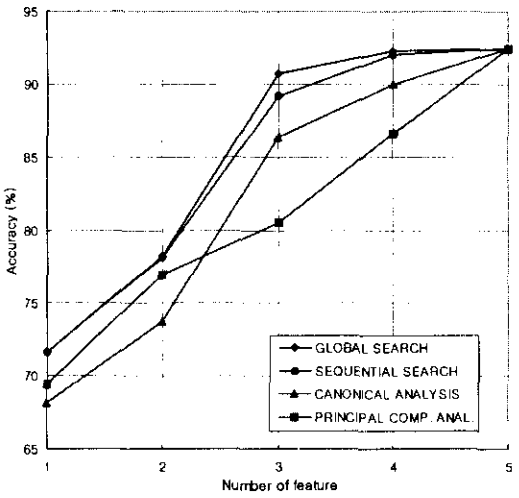

그림 3. 제안된 알고리즘과 기존알고리즘과의 성능비교.

V. 결 론

본 논문에서는 패턴분류 문제와 관련하여 임의의 초기벡터를 조금씩 변화시켜 분류오차가 크게 감소하는 방향으로 피춰를 추출하는 알고리즘을 제안하였다. 또한 기존의 알고리즘으로 구해진 피춰를 초기벡터로 사용하여 성능을 향상시킬 수 있었다. 제안된 알고리즘은 직접적으로 분류오차를 최소화시키는 방법으로 패턴분류나 패턴인식의 문제에서 최적의 해를 기대할 수 있다. 실험결과 제안된 알고리즘은 기존의 알고리즘들보다 좋은 성능을 보였으며 특히 global search방법은 1개 이상의 피춰를 추출하는 경우 sequential search방법보다 분류정확도 측면에서 우수한 성능을 보였다.

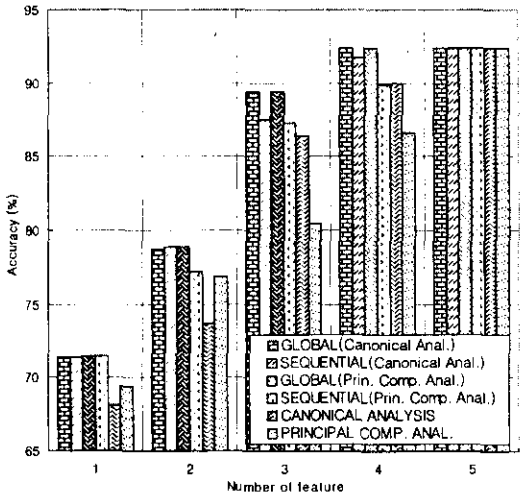

그림 4. 기존알고리즘의 피춰를 초기벡터로 사용한 경우의 성능비교.

참 고 문 헌

[1] K. Fukunaga, *Introduction to Statistical Pattern Recognition.* New York: Academic Press, pp.225-226, 1990.

[2] J. A. Richards, *Remote Sensing Digital Image Analysis.* Springer-Verlag, 1993.

[3] R.O. Duda and P.E. Hart, *Pattern Classification and Scene Analysis.* John Wiley & Sons, 1973.

[4] C. Lee and D. A. Landgrebe, "Feature extraction based on decision boundaries," *IEEE Transaction on Pattern Analysis and Machine Intelligence,* vol. 15, no. 4, pp. 388-400, 1993.

[5] C.G. Cullen, *Matrices and Linear Transformation.* Addison_wesley Publishing Company, 1972.

[6] L.L. Biel and e. al., "A Crops and Soils Data Base For Scene Radiation Research," *Proc. Machine Process. of Remotely Sensed DataSymp., West Lafayette, Indiana,* 1982.