다중 클래스에 대한 피춰 추출 방법의 최적화

홍준용, 이철희

연세대학교 전자공학과

서울 서대문구 신촌동 134

Email : jyhong@feature.yonsei.ac.kr

요약

본 논문에서는 여러 개의 클래스가 정의되어 있을 경우에 피춰(feature)추출을 최적화 하는 방법을 제안한다. 제안된 알고리즘은 피춰를 하나씩 추출하며 그 과정마다 각 클래스의 가중치를 조정하여 최적의 해를 얻는 방법을 사용한다. 처음에는 각 클래스에 동일한 가중치를 주어 criterion function을 구하고 이로부터 첫 번째 피춰를 얻는다. 이 피춰에 의한 오류와 전체 피춰를 사용하였을 경우의 오류의 차이가 가장 큰 클래스에 더 많은 가중치를 주어 새로운 criterion function을 구하여 두 번째 피춰를 얻는다. 이 과정에서 오류는 Bhattacharyya distance에 의해 예측한다.

1. 서론

패턴의 분류에 필요한 계산량은 분류에 사용된 피춰의 수에 따라 증가하며, 특히 자주 사용되는 가우시안 최대 우도 분류기(Gaussian Maximum likelihood Classifier)의 경우 처리 시간은 피춰 수의 제곱에 비례하여 증가한다. 따라서 적은 수의 피춰로 분류의 정확도를 높이기 위해, 피춰추출의 방법에 대한 많은 연구들이 이루어져 왔으며 여러 가지 알고리즘이 제안되었다 [1-5]. 이런 방법들은 대개 criterion function을 정의하고 두개의 클래스에 대해 최적화 한다. 여러 개의 클래스가 정의되어 있을 경우 두개씩의 클래스에 대한 결과를 조합하여 결과를 얻는 방법을 사용할 수 있지만 여러 개의 클래스에 대한 이런 방법의 적용은 최적의 해를 구하는 방법이라 할 수 없다. 본 논문에서는 여러 개의 클래스가 정의되었을 경우에, 하나의 피춰벡터를 계산할 때마다 클래스의 가중치를 변화시켜 새로운 criterion function을 구하고 이로부터 다음의 피춰벡터를 얻는 방법을 제안한다.

2. Canonical Analysis

Canonical analysis는 한 클래스내의 분산을 작게 하고 다른 클래스와의 분산을 크게 만드는 새로운 좌표축을 찾는 방법으로 피춰벡터를 구한다 [1]. 즉,

$$\lambda = \frac{\sigma_b^2}{\sigma_w^2} = \frac{\mathbf{d}^T\mathbf{S}_b\mathbf{d}}{\mathbf{d}^T\mathbf{S}_w\mathbf{d}}$$

를 최대로 하는 벡터 **d**를 구한다. 위의 식에서 $\mathbf{S}_w$, $\mathbf{S}_b$, $\mathbf{M}_0$는 다음과 같다.

$$\mathbf{S}_w = \sum_i P(w_i)\boldsymbol{\Sigma}_i \quad \text{(within-class scatter matrix)}$$

$$\mathbf{S}_b = \sum_i P(w_i)(\mathbf{M}_i - \mathbf{M}_0)(\mathbf{M}_i - \mathbf{M}_0)^T \quad \text{(between-class scatter matrix)}$$

$$\mathbf{M}_0 = \sum_i P(w_i)\mathbf{M}_i \quad \text{(global mean)}$$

$\mathbf{M}_i$, $\boldsymbol{\Sigma}_i$, $P(w_i)$는 각각 평균 벡터(mean vector), 분산 행렬(covariance matrix), 클래스 w_i의 선행확률(priori probability)이다. 선행확률이 정해지지 않은 경우 모든 클래스에 대해 같은 값을 사용한다. 이러한 방법으로 구한 피춰들은 분류에 사용됐을 때 대부분의 경우 좋은 결과를 보이지만 여러 개의 클래스에 대해서 최적화되었다고 할 수 없다.

3. Bhattacharyya distance

Bhattacharyya distance로부터 클래스 분류에 의한 오류의 상한값과 하한값을 구할 수 있지만 그 한계값의 범위가 넓어 실제로 사용하기에 충분치 않은 경우가 많다 [2]. 최근 가우시안 최대 우도 분류기에 대해 Bhattacharyya distance를 이용하여 두 클래스간의 오류를 정확하게 예측할 수 있는 방법이 제안되었다 [6]. 정규분포에 대해 Bhattacharyya distance $\mu(\frac{1}{2})$는 다음과 같다.

$$\mu(\frac{1}{2}) = \frac{1}{8}(\mathbf{M}_2 - \mathbf{M}_1)^T[\frac{\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2}{2}]^{-1}(\mathbf{M}_2 - \mathbf{M}_1) + \frac{1}{2}\ln\frac{|(\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2)/2|}{|\boldsymbol{\Sigma}_1|^{1/2}|\boldsymbol{\Sigma}_2|^{1/2}}$$

그리고 이 거리로부터 예측할 수 있는 클래스 분류의 오류는 다음과 같다.

$$\varepsilon = 40.219 - 70.019*\mu + 63.578*\mu^2 - 32.766*\mu^3 + 8.7172*\mu^4 - 0.91875*\mu^5$$

ε, μ는 각각 클래스 분류의 오류와 Battacharyya distance이다. 위의 식을 사용하면 두 개의 클래스에 대해 가우시안 최대 우도 분류기를 사용했을 때의 오류를 1-2%의 한계 내에서 예측할 수 있다 [6].

4. 다중클래스문제에 대한 피춰추출의 최적화

Canonical analysis 많은 경우 비교적 좋은 성능을 보여주지만 여러 개의 클래스가 정의된 경우에 대해서 최적화되어 있지 않다. 여러 개의 클래스가 정의된 경우는 전체 클래스 중 두 개씩의 클래스에 대해 피춰를 추출해 이 피춰들을 조합하여 결과를 얻거나 모든 클래스를 사용하여 criterion function을 계산해서 피춰를 추출한다. 하지만 이런 방법들은 최적의 해를 구하는 방법이라 할 수 없으므로 다음과 같은 방법을 제안한다.

처음에는 각 클래스에 동일한 가중치를 주어 criterion function을 구하고 이를 사용하여 첫 번째 피춰를 얻는다. 이 피춰벡터를 사용하여 분류했을 때의 오류와 전체 피춰를 사용하여 분류했을 때의 오류의 차이가 가장 큰 클래스는 다른 피춰를 추가로 이용했을 경우 분류의 정확도가 향상될 가능성이 가장 크다고 할 수 있다. 따라서 오류의 차이에 따라 각 클래스에 가중치를 다르게 부여하고 새로운 criterion function을 구하여 이로부터 두 번째 피춰를 얻는다. 이후 같은 방법으로 피춰들을 하나씩 구해 나간다.

N차원 공간에서 정의된 클래스들에 대해 제안된 방법을 사용하여 구한 피춰들은 원래의 canonical analysis에 의해 구한 피춰들과 같다. 이렇게 얻은 피춰벡터들을 eigenvalue가 큰 순서로 정렬하여 행렬 $\boldsymbol{\Phi}$를 얻는다.

$$\boldsymbol{\Phi} = [\,\boldsymbol{\phi}_1\ \boldsymbol{\phi}_2\ \cdots\ \boldsymbol{\phi}_N]$$

이때 첫 번째 열벡터 $\boldsymbol{\phi}_1$이 eigenvalue가 가장 큰 eigenvector이므로 전체 피춰 중 분류도가 가장 높은 피춰가 된다. 이 피춰 하나만을 사용하여 분류할 때 예측되는 오류와 전체 피춰를 사용하여 분류할 때 예측되는 오류를 비교한다. 이때 두 오류의 차이가 가장 큰 클래스는 이 후에 얻을 피춰로부터 오류를 더 많이 줄일 가능성이 있다. 따라서 다음 criterion function의 계산에서 이 클래스에 더 큰 가중치를 부여한다. 예로 세 개의 클래스가 정의된 경우에 아래와 같은 상황을 가정한다.

	클래스 1	클래스 2	클래스 3
클래스 1		15%	20%
클래스 2	15%		10%
클래스 3	20%	10%	

(표 1) 한 개의 피춰를 사용하여 분류한 경우의 예상 오류

	클래스 1	클래스 2	클래스 3
클래스 1		10%	5%
클래스 2	10%		5%
클래스 3	5%	5%	

(표 2) 전체 피춰를 사용하여 분류한 경우의 예상오류

위의 표에서 전체 피춰를 사용할 경우 클래스 1과 클래스 2사이의 오류와 클래스 2와 클래스 3사이의 오류에 대해서는 5%의 향상을 예상할 수 있지만, 클래스 1과 클래스 3사이의 오류에 대해서는 15%의 향상을 예상할

수 있다. 따라서 클래스 1과 클래스 3에 더 큰 가중치를 부여한다.

두 번째 criterion function의 계산은 모든 샘플에서 첫 번째 피춰의 성분을 제거하여 구한다. 이를 위해 모든 샘플에 $\boldsymbol{\Phi}_{N-1}$을 곱하여 (N-1)차원으로 변환시킨다.

$$\mathbf{s}_{N-1} = \boldsymbol{\Phi}_{N-1}^{T} \cdot \mathbf{s}_{N}$$

$$\boldsymbol{\Phi}_{N-1} = [\ \phi_2\ \phi_3\ \cdots\ \phi_N]$$

$\boldsymbol{\Phi}_{N-1}$은 $\boldsymbol{\Phi}$에서 ϕ_1을 제거한 (N x N-1)행렬이고 $\mathbf{s}_N$과 $\mathbf{s}_{N-1}$은 N차원의 임의의 한 샘플을 나타내는 열벡터와 이를 (N-1)차원으로 바꾼 열벡터이다. 이렇게 얻은 샘플을 이용해 criterion function을 구하고 canonical analysis를 이용해 피춰벡터의 행렬 $\boldsymbol{\Psi}$를 얻는다.

$$\boldsymbol{\Psi} = [\ \phi_1\ \phi_2\ \cdots\ \phi_{N-1}]$$

이렇게 얻은 행렬의 첫 번째 열벡터 ϕ_1은 ϕ_1의 성분을 제외하였을 때 가장 분류도가 높은 피춰이므로 이것이 두 번째 피춰이다. 하지만 ϕ_1은 (N-1)차원의 벡터이므로 $\boldsymbol{\Phi}_{N-1}$을 이용해 다시 N차원의 벡터로 만들어준다.

$$\phi_1^{N} = \boldsymbol{\Phi}_{N-1} \cdot \phi_1$$

이런 방법으로 차원을 낮추면서 피춰를 하나씩 찾아 전체 피춰를 구한다.

5. 실험

실험에서는 10차원의 정규분포를 가지는 1000개의 샘플로 이루어진 클래스 5개를 사용했고 트레이닝과 테스트에 모든 샘플을 사용하였다. (그림 1)과 (그림 2)는 실험결과이다. (그림 1)의 실험에서는 두 개의 피춰를 사용했을 때 10% 이상 정확도의 향상이 있었지만 (그림 2)의 실험에서는 세 개의 피춰를 사용했을 때 2.3% 의 정확도 향상이 있었다. 대부분 제안된 방법이 canonical analysis보다 좋은 결과를 보여주었지만 클래스간의 평균의 차이에 따라 성능향상의 정도가 달라 평균의 차이에 따른 성능향상의 변화량을 관측하였다. 각 차원에서 평균의 표준편차를 구하여 이를 모두 더한 값을 평균의 차이로 정의하고 각 차원에서 두 가지 방법의 정확도의 차이를 성능향상으로 정의하여 그 관계를 분석했다.

	차원 1	차원 2	…	차원 N
클래스 1	M_{11}	M_{12}	…	M_{1N}
클래스 2	M_{21}	M_{22}	…	M_{2N}
·	·	·	…	·
·	·	·	…	·
·	·	·	…	·
클래스 K	M_{K1}	M_{K2}	…	M_{KN}
표준편차	σ_1	σ_2	…	σ_N

$$\sigma_T = \sum_{i=1}^{N} \sigma_i^2$$

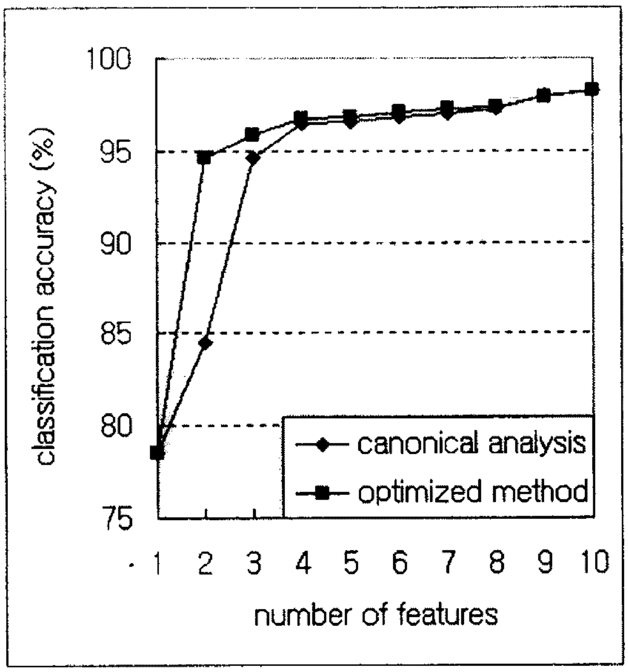
M_{kn}과 σ_n은 각각 클래스 k의 차원 n의 평균과 차원 n의 분산이다.

$$\Delta_{acc} = \sum_{n=1}^{N} (A_{op,n} - A_{cano,n})$$

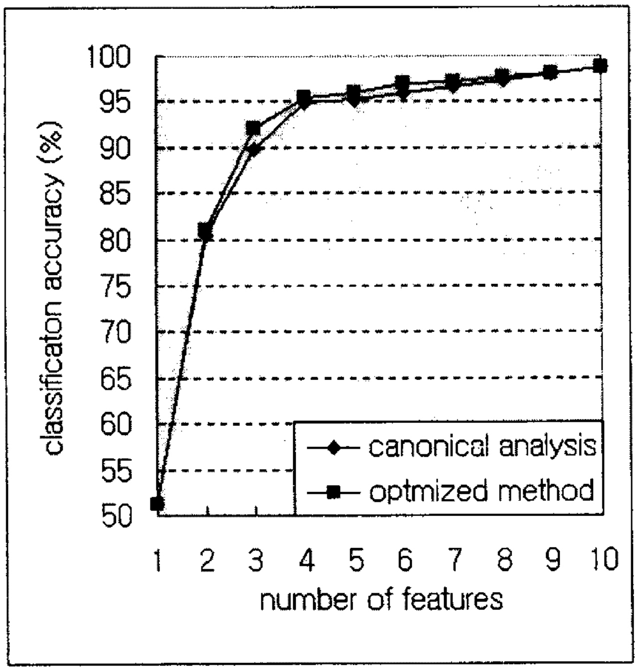
$A_{op,n}$와 $A_{cano,n}$는 각각 n번째 차원에서 제안된 방법과 canonical analysis에 의한 정확도이다. (그림 3)은 σ_T와 Δ_{acc}의 관계를 나타낸 그래프이다. 대체적으로 평균의 차이가 클수록 더 많은 성능향상이 있는 것으로 관찰되었다. (그림 1)과 (그림 2)의 실험에서 σ_T는 각각 38.704, 32.744이고 Δ_{acc}는 각각 11.4와 3.1이다. σ_T가 큰 (그림 2)쪽의 성능향상이 더 우수하였다.

6. 결론

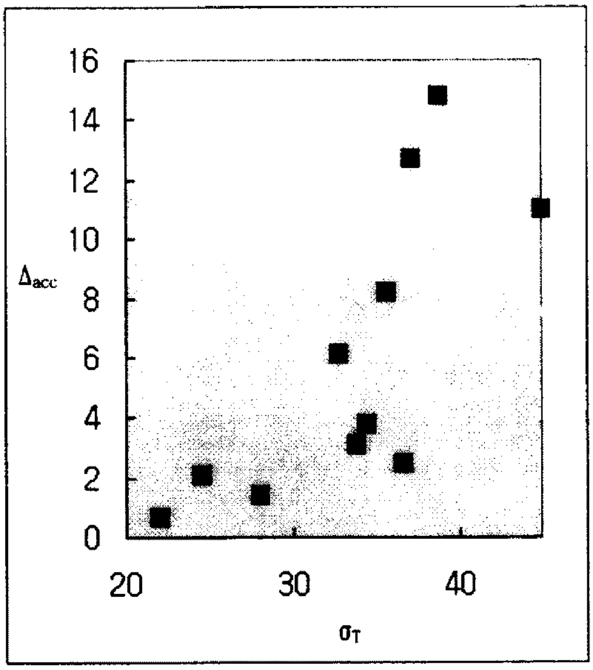
본 논문에서는 여러 개의 클래스가 정의된 경우에 클래스의 분류를 최적화하는 방법을 제안하였다. 제안된 방법은 Battacharyya distance로부터 예측한 오류를 이용하여 클래스들의 가중치를 조정하여 criterion function을 계산하고 이로부터 피춰를 구한다. 제안된 방법을 canonical analysis에 적용한 결과 기존의 방법보다 높은 정확도를 가지는 것으로 나타났다.



(그림 1) 실험결과



(그림 2) 실험결과



(그림 3) 평균의 차이와 정확도의 관계

7. References

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

[2] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, New York: Academic Press, 1990.

[3] C. Lee, D. A. Landgrebe, "Feature Extraction Based on Decision Boundaries," in Proc. *IEEE Trans. Patt. Anal. Machine. Intell.*, vol. PAMI-15, no. 4, Apr. 1993.

[4] D. H. Foley, J. H. Sammon, "An optimal set of discriminant vectors," *IEEE Trans. Comput.*, vol. C-24, no. 3, pp. 281-289, Mar. 1975.

[5] K. Fukunaga, W. L. G. Koontz, "Application of the Karhunen-Loeve expansion of feature selection and ordering," *IEEE Trans. Comput.*, vol. C-19, no. 4, pp. 311-318, Apr. 1970.

[6] C. Lee, "Error estimation of the Gaussian ML classifier," in Proc. *IEEE International Symposium on Information Theory*, Ulm, Germany, June 1997.