SVM을 이용한 LVQ3 학습의 성능개선

°김 상 운

명지대학교 컴퓨터학부
449-728 경기도 용인시 남동 산 38-2

An Improvement of LVQ3 Learning Using SVM

°Sang-Woon KIM

Div. of Computer Science & Engineering, Myongji University
San 38-2, Namdong, Yongin, Kyunggi, 449-728 Korea
Email : kimsw@mju.ac.kr

Abstract

Learning vector quantization (LVQ) is a supervised learning technique that uses class information to move the vector quantizer slightly, so as to improve the quality of the classifier decision regions. In this paper we propose a selection method of initial codebook vectors for a learning vector quantization (LVQ3) using support vector machines (SVM). The method is experimented with artificial and real design data sets and compared with conventional methods of the condensed nearest neighbor (CNN) and its modifications (mCNN). From the experiments, it is discovered that the proposed method produces higher performance than the conventional ones and then it could be used efficiently for designing nonparametric classifiers.

Keywords: Nonparametric classifiers, Nearest neighbor, Learning vector quantization, Support vector machines.

1. 서론

통계적 패턴인식의 문제에서, NN 분류기(the nearest neighbor classifier)는 결정규칙이 단순하고 분류성능 또한 우수하기 때문에 실제 응용에서 폭 넓게 이용되고 있는 분류기중의 하나로 알려져 있다[1]. 그러나 이 분류기를 성공적으로 구현하기 위해서는 유사성 계산을 위한 프로토타입의 개수를 최소화 할 수 있는 방법이나 가능한 한 빨리 최근접 이웃을 찾을 수 있는 탐색 방법을 준비하여야 한다. 프로토타입의 개수를 최소화하는 방법에는 k-means, CNN (the condensed nearest neighbor rule)[2], RNN (the reduced nearest neighbor rule)[3] 과 CNN을 수정한 mCNN (modified CNN rule)[4] 등이 있으며, 최근에는 학습벡터 양자화법 (learning vector quantization: LVQ)[5] 등이 소개되어 있다. 또한 탐색 속도를 개선하는 방법에 대해서도 Fukunaka법과 Ra법 등[6] 다양한 방법들이 제안되어 있다.

본 논문은 주어진 응용에 적합한 NN 분류기를 설계하기 위해 최고의 분류성능을 유지하면서 프로토타입의 개수를 최소화하는 방법에 대한 것으로서, 최근 우수한 성능과 폭넓은 응용으로 주목받고 있는 학습법인 SVM (support vector machines)[7]을 이용하여 초기 프로토타입을 선정한 다음, 다시 LVQ 학습을 수행하여 최적의 프로토타입을 결정하는 방법을 제안한다. 또한, 이 방법을 2 차원 2 클래스의 인공데이터 및 다차원의 벤치마크 데이터에 대하여 실험한 결과, 제안방법은 CNN과 mCNN등 기존의 방법과 비슷한 정도로 프로토타입의 벡터 수를 압축하면서 분류 성능을 향상시킬 수 있음을 확인하였다. 또한 제안한 방식의 분류기 성능이 SVM이나 LVQ3 단독의 분류기보다 우수함을 확인하였다.

2. SVM

SVM은 통계적 학습이론에 근거한 학습 방법으로서, 패턴 클래스간의 간격 (margin)을 최대화시킬 수 있는 초평면을 구하여 클래스를 분리시킨다. 이 때 기준 함수로 신경망에서 이용하는 ERM(empirical risk minimization) 대신에 SRM(structural risk minimization)함수를 이용한다. 즉, SVM은 결정함수 $d(x)=w^Tx+b$를 학습하기 위해서는 소속 클래스 $y_i=\{-1,+1\}$를 알고 있는 N개의 학습패턴 x_i , $i=1,\cdots N$이 필요한

일종의 교사학습 방법으로, 각 학습패턴 x_i가 초평면 결정함수에 미치는 영향력을 라그랑즈 승수 (Lagrange multipliers) α_i $(1 \le i \le N)$로 놓고, 이 파라미터를 구하기 위해 다음과 같은 2차 계획 문제 (quadratic programming: QP)를 풀이하는 방법을 쓴다.

$$Minimize : \frac{1}{2}\sum_{i,j=1}^{N}\alpha_i Q_{ij}\alpha_j - \sum_{i,j=1}^{N}\alpha_i, ,$$

$$Subject\ to : 0 \le \alpha_i \le C, \quad \sum_{i=1}^{N} y_i \alpha_i = 0 .$$

여기서 Q는 $N \times N$ 매트릭스로서 학습패턴 x_i 와 함수의 형태에 따라 결정되고, C 는 실험 상수로 이 값이 크면 클수록 오인식률이 높게 된다.

위의 QP문제의 해로부터, $\alpha_i \neq 0$에 해당하는 학습패턴이 결정 경계를 서포트하는 SV (support vectors)가 되며, 결정함수 $w \cdot x + b = 0$의 웨이트 벡터 w와 비례상수 b 는

$$w = \sum_{i=1}^{N_s} \alpha_i y_i x_i , \quad b = \frac{1}{2}(w^T x_p + w^T x_n)$$

로 결정된다. 여기서 N_s는 SV의 벡터 수이고, x_p 와 x_n는 각각 +1과 -1 레이블의 학습 패턴 수이며, w^T 는 w의 전치벡터 이다.

SV는 학습 패턴의 분포구조를 잘 반영하면서 초평면에 가장 가깝게 위치한다. 본 논문에서는 주어진 학습 패턴으로부터 SV를 선정하여 초기 코드북 벡터 (code book vector)를 구성한 후, 다시 LVQ3 학습을 수행하여 최적의 프로토타입을 결정하는 방법을 제안한다. 이 때 QP문제를 풀기 위해서는 공개된 소프트웨어인 SVM^{light}[8]를 이용하였고, LVQ3은 *LVQ_PAK*[9]의 프로그램을 이용한다.

3. 실험

3.1 실험 데이터

제안 방법을 평가하기 위한 실험 데이터로는, 시각적 관찰을 위해 2차원 평면에서 균일 분포로 랜덤하게 생성한 "Random"와 UCL machine learning repository (http://www.ics.uci.edu/mlearn/MLRepository.html)의 실제의 벤취마크 데이터 "Iris", "Glass", "Ionosphere"를 선택하여 이용하였다. 모든 데이터를 랜덤하게 두 세트로 나누어 한 세트는 학습 데이터로 다른 세트는 테스트 데이터로 이용하였다. 또한 모든 패턴 벡터는 -1부터 +1 사이의 실수로 정규화 하였다. 데이터별 패턴 벡터의 수, 특징 차원 및 클래스 수는 표 1과 같다.

Table 1. The benchmark data sets for experiments.

Dataset	Patterns (Train, Test)	No. of Features	No. of Classes
Random	400 (200,200)	2	2
Iris	150 (75,75)	4	3
Glass	214 (107,107)	9	6
Ionosphere	351 (176,175)	34	2

3.2 초기벡터 선정

기존의 방법인 CNN과 mCNN, 그리고 제안 방법인 SVM을 이용하여 LVQ3학습을 위한 초기 참조벡터를 선정하는 실험을 하였다. 먼저 그림 1(a)와 같은 "Random" 학습 데이터를 생성하여 CNN, mCNN, SVM법으로 프로토타입을 선정한 결과는 각각 그림 1(b), (c), (d)와 같다.

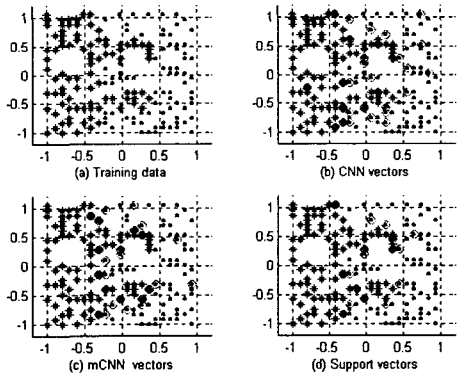

Fig. 1. A training dataset of "Random" and initial reference vectors selected with CNN, mCNN, and SVM. In the pictures, vectors of two classes are represented as "*" and ".", respectively. Then, selected initial reference vectors are indicated by extra circles. Their numbers in (b), (c), and (d) are 31, 26, and 18, respectively.

여기서, CNN 및 mCNN 방법의 경우에는 각각 문헌 [2]와 [4]의 알고리즘을 구현 하였으며, SVM방식의 경우에는 고차원 QP문제를 풀기 위해서 공개소프트웨어인 SVM^{light} [8]를 이용하였다. 실험 데이터에 대한 각 방법별 데이터 압축률 $Re(\cdot)$은 표 2와 같다. 여기서 압축은 다음 식으로 계산하였다.

$$Re(\cdot) = \frac{\text{전체벡터수} - \text{선정된벡터수}}{\text{전체벡터수}}.$$

Table 2. Data compression rates on the datasets.

Methods	Random	Iris	Glass	Ionosphere
CNN	0.85	0.72	0.80	0.72
mCNN	0.87	0.72	0.82	0.74
SVM	0.91	0.75	0.83	0.74

위의 표는 실험 데이터에 대한 각 방법별 데이터 압축률 사이에 다음과 같은 관계가 있음을 보여준다.

$$Re(CNN) \le Re(mCNN) \le Re(SVM)$$

즉, 전체적인 실험 데이터에 대하여 SVM방식의 압축률이 기존의 방법에 비하여 우수함을 알 수 있다.

표 3은 선정한 벡터를 그대로 프로토타입으로 하여 NN 분류기를 설계하였을 경우의 실험 데이터별 오인식률 이다. 여기서 SVM*는 SVM분류기, SVM**는 SV를 프로토타입으로 한 NN 분류기를 나타낸다.

Table 3. Classification error rates (%).

Methods	Random	Iris	Glass	Ionosphere
CNN	3.50	4.44	20.09	17.71
mCNN	6.00	7.56	19.00	28.57
SVM*	9.50	4.45	18.22	17.71
SVM**	13.50	12.00	28.66	17.14

위의 표 3에서 실험 데이터에 대한 각 방법별 오인식률은 다음과 같은 관계가 있음을 보여준다.

$$Err(SVM^*) \ge Err(CNN) \text{ or } Err(mCNN)$$

즉, 순수한 SVM 분류기의 오인식률은 CNN과 mCNN 방식으로 선정한 초기 벡터를 코드북으로 이용한 NN 분류기 보다 높음을 알 수 있다. 특히 SVM으로 선정한 SV를 코드북으로 이용하여 NN 분류할 경우 오인식률을 개선할 수 없음을 보여준다.

3.3 LVQ3 학습

앞 절에서 각 방법으로 주어진 샘플 패턴으로부터 선정한 벡터를 이용하여 코드북 벡터 (codebook vector)를 구성한 후 다시 LVQ3 학습을 수행하여 최적의 프로토타입을 결정하는 NN 분류기를 설계하는 실험을 하였다. LVQ3 학습 프로그램은 *LVQ_PAK*[9]의 프로그램을 이용하였다.

먼저, 시각적 관찰을 위해 그림 2(a)와 같이 랜덤하게 "Random"의 테스트 데이터를 준비한 다음 CNN, mCNN, SVM법의 코드북 벡터를 학습한 최종 프로토타입 및 이를 이용하여 설계한 NN 분류기의 오인식 패턴은 각각 그림 2(b), (c), (d)와 같다.

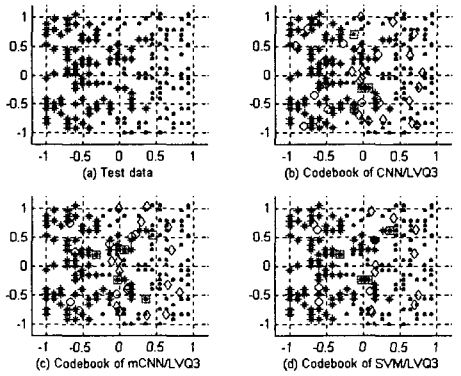

Fig. 2. A test dataset of "Random" and codebook vectors of CNN, mCNN, and SVM trained by LVQ3. In the pictures, test vectors of two classes are represented as "*" and ".", respectively. Then, trained codebook vectors of two classes are indicated by circles and diamonds, respectively, and missclassified vectors among the test data are exemplified by extra squares.

또한, 각 방법별 학습 데이터에 대한 오인식률은 다음 표 4와 같다. 여기서 Bayes는 학습 데이터로 평균 μ와 공분산 C를 구하여 구현한 베이시언 분류기의 오인식률이고, LVQ3는 랜덤하게 초기 코드북 벡터를 구성하여 학습한 NN분류기의 오인식률이다. 이 때 코드북 크기는 CNN, mCNN, SVM방식으로 선정한 벡터 수를 평균하여 구하였다. 그리고, CNN/LVQ3, mCNN/LVQ3, SVM/LVQ3는 각각 CNN, mCNN, SVM 방식으로 선정한 벡터를 LVQ3로 학습하여 프로토타입을 구한 분류기의 오인식률 이다.

표 4는 LVQ3 학습 후의 실험 데이터에 대한 오인식률은 다음과 같은 관계가 있음을 보여준다.

$$Err(SVM/LVQ3) \le Err(CNN/LVQ3) \text{ or } Err(mCNN/LVQ3)$$

즉, LVQ3 학습한 후의 분류기인 SVM/LVQ3의 오인

식률은 기존 방식으로 선정한 초기 벡터를 이용하여 학습한 분류기인 CNN/LVQ3나 mCNN/LVQ3 보다 향상되었음을 알 수 있다.

Table 4. Classification error rates (%) after LVQ3.

Methods	Random	Iris	Glass	Ionosphere
Bayes	13.50	3.33	17.11	*
LVQ3	9.00	3.11	15.42	14.29
CNN/LVQ3	2.50	4.00	17.60	12.57
mCNN/LVQ3	2.50	4.89	14.64	13.71
SVM/LVQ3	2.50	3.11	14.49	12.57

또한, SVM/LVQ3 분류기와 순수한 SVM분류기의 오인식률은

$$Err(SVM/LVQ3) \leq Err(SVM)$$

의 관계가 성립하며, 따라서 제안한 방법을 이용하면 SVM 분류기의 성능을 보다 향상시킬 수 있음을 알 수 있다.

끝으로, 학습횟수에 따른 각 방법별 분류기의 수렴 정도를 고찰하였다. "Random" 데이터에 대한 학습횟수에 대한 오인식률 변화는 그림 3과 같다.

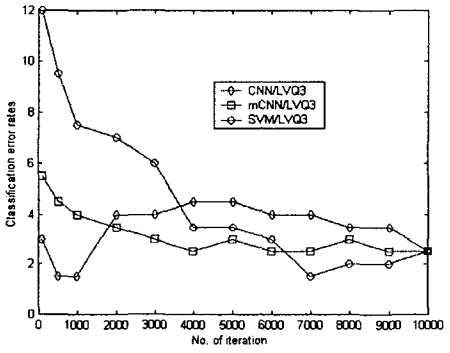

Fig. 3. The classification error rates due to the number of iterations. Usually, the number of learning steps is about 50 to 200 times the total number of codebook vectors in order to avoid the overlearn problem. This figure shows that the error rate of SVM/LVQ3 converges steadily from 100 to 10,000 iterations on the "Random" dataset.

4. 결론

본 논문에서는 SVM을 이용하여 서포트벡터 SV를 선정한 다음, 이 벡터를 초기 코드북 벡터로 이용하여 LVQ3학습을 수행하는 방법을 제안하였다. 제안 방법을 평가하기 위하여 시각적 관찰을 위한 인공 데이터와 실제의 벤취마크 데이터를 이용하였으며, 기존의 CNN 및 이의 수정본인 mCNN과 비교하였다. 실험 결과로부터, 제안 방법은 기존의 방법들과 비슷한 데이터 압축률을 유지하면서 분류성능은 향상되었음을 확인하였다. 또한 제안 방법으로 학습한 NN 분류기는 본래의 SVM 분류기보다 성능이 우수함을 확인하였다. 따라서 SVM 알고리즘이 NN 분류기를 설계하기 위한 초기 프로토타입을 효율적으로 선정 할 수 있음을 확인할 수 있었다. 이러한 실험 결과는 SVM이 추출한 서포트 벡터가 기존의 방법인 CNN이나 mCNN으로 추출한 벡터보다 학습 패턴의 분포 구조를 더 잘 반영하고 있다는 사실에 기인한 것으로 사료된다. 앞으로의 과제는 제안 방법을 데이터 마이닝이나 텍스트 분류 등과 같은 고차원 응용에 적용하는 연구이다.

참고문헌

[1] A. K. Jain, R. P. W. Duin, and J. Mao, "Statistical pattern recognition: A review," IEEE Trans. Pattern Anal. and Machine Intell., vol. PAMI-22, no. 1, pp. 4 - 37, Jan. 2000.

[2] P. E. Hart, "The condensed nearest neighbor rule," IEEE Trans. Inform. Theory, vol. IT-14, pp. 515 - 516, May 1968.

[3] G. W. Gates, "The reduced nearest neighbor rule," IEEE Trans. Inform. Theory, vol. IT-18, pp. 431 - 433, May 1972.

[4] I. Tomek, "Two modifcations of CNN," IEEE Trans. Syst., Man and Cybern., vol. SMC-6, no. 6, pp. 769 - 772, Nov. 1976.

[5] H. H. Song and S. W. Lee, "LVQ combined with simulated annealing for optimal design of large-set reference models," Neural Networks, vol. 9, no. 2, pp. 329 - 336, 1996.

[6] S.-W. Ra and J.-K Kim, "A fast mean distance-ordered partial codebook search algorithm for image vector quantization," IEEE Trans. Circuit Syst. II, vol. 40, no.2, pp. 576 - 579, 1993.

[7] http://svm.research.bell-labs.com/SVMdoc.html

[8] http://www-ai.cs.uni-dortmund.de/ SOFTWARE /SVM_LIGHT/svm_light.eng.html

[9] http://cochlea.hut.fi/research/som_lvq_pak.shtml