Support Vector Machine에 대한 커널 함수의 성능 분석

Performance Analysis of Kernel Function for Support Vector Machine

*심우성 , **성세영 , ***정차근
Woo-Sung Sim , Se-Young Sung , Cha-Keon Cheng

Abstract - SVM(Support Vector Machine) is a classification method which is recently watched in mechanical learning system. Vapnik, Osuna, Platt etc. had suggested methodology in order to solve needed QP(Quadratic Programming) to realize SVM so that have extended application field. SVM find hyperplane which classify into 2 class by converting from input space converter vector to characteristic space vector using Kernel Function. This is very systematic and theoretical more than neural network which is experiential study method. Although SVM has superior generalization characteristic, it depends on Kernel Function. There are three category in the Kernel Function as Polynomial Kernel, RBF(Radial Basis Function) Kernel, Sigmoid Kernel. This paper has analyzed performance of SVM against kernel using virtual data.

Key Words :SVM분류기, Kernel Function, Polynomial, RBF, Sigmoid

1. 서 론

일반적으로 패턴인식의 문제에 사용된 분류기는 신경망에 기초하였으며, 이는 훈련 패턴에 대한 경험론적 방법에 근거하고 있다. 이러한 신경망에 기초한 분류기들은 훈련 패턴의 입력 순서, 입력 패턴 수, 학습 시간 등에 의존적이라는 문제점을 내포하고 있다. 즉, 경험적 방법을 사용하는 패턴 분류기가 최적 분류를 보장하는지에 대한 여부는 증명 불가능한 일이다. 따라서 최적 패턴 분류를 보장하는 새로운 분류기에 대한 요구가 높아졌고 이에 대한 해결책 중의 하나로 구조적 방법론에 의거한 새로운 분류기가 제안되었다[1]. Vapnik와 Bell 연구소의 연구팀이 1995년에 제안한 SVM(Support Vector Machine)은 이론적으로 VC(Vapnik-Chervonenkis)-차원을 최소화 하는 2-클래스 분류기를 만들어 낸다. VC-차원이 작아질수록 기대되는 일반화 에러의 확률이 더 적어지므로 VC-차원을 최소화 하는 것은 일반화 성능 향상을 의미한다[2].

SVM은 뛰어난 일반화 성능에도 불구하고 2-클래스 분류기라는 제약이 있으며 사용되는 Kernel에 높은 의존성을 가진다[3]. 즉 Kernel Function의 Parameter에 따라 분류기의 일반화 성능이 변화한다[4]. 본 논문에서는 Random한 입력 데이터를 이용하여 Kernel Function에 따라 SVM Classification의 성능 변화를 측정 및 분석하였다.

저자 소개
* 심 우 성 : 호서대학교 공과대학 시스템제어공학과
** 성 세 영 : 호서대학교 공과대학 메카트로닉스공학과
*** 정 차 근 : 호서대학교 공과대학 시스템제어공학과

2. SVM 관련 연구

2.1절 Support Vector Machine

SVM(Support Vector Machine)은 특징 공간에서 2-클래스의 데이터를 양분하는 초평면(Hyperplane)을 찾는 기법으로 초평면 근처의 데이터 포인트와 초평면과의 수직거리를 최대화 하는 수식을 사용하여 초평면을 찾는다. 여기에서 초평면 근처의 데이터 포인터를 Support Vector라 한다[5]. 그림 1.에서 볼 수 있듯이 2개의 Support Vector가 있고 2-클래스를 나누는 초평면을 구한다.

각각의 데이터를 $X=(x_i , x_j)$, 데이터에 대한 클래스 레이블을 $t_i = \{+1 , -1\}$ 라고 하면 초평면을 결정하는 함수는 식(1)과 같다.

$$d(x) = w^T x + b = 0 \tag{1}$$

$$\begin{cases} Subject.to. & t_i(w^T X + b) - 1 \geq 0 \\ Min. & \left\{ \dfrac{1}{2\|w\|} \right\} \end{cases} \tag{2}$$

식(2)을 최대화 문제로 바꾸기 위해 라그랑지안 함수를 활용하여 식(3)를 얻을 수 있다.

$$L(w,b,\alpha) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{N} \alpha_i (t_i(w^T x_i + b) - 1) \tag{3}$$

여기서 α값은 라그랑제 승수(Lagrange multiplier)이다. 식(3)에 KKT(Karush-Kuhn-Tucker)조건을 이용하여 조건부 최적화 하는 식(4)를 얻을 수 있다.

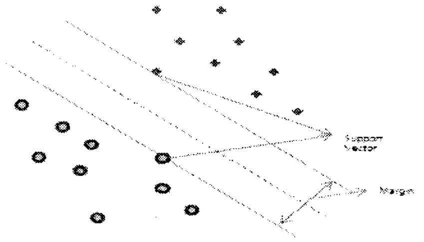

그림 1. Support Vector Machine의 선형 분리 초평면

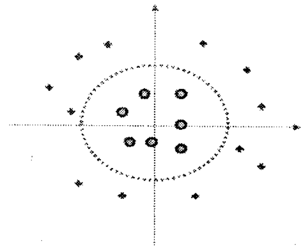

그림 2. 입력 공간에서 비선형적 분류 2차원 데이터

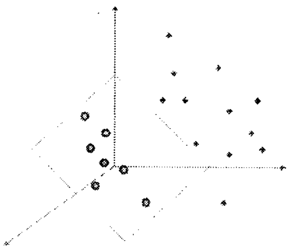

그림 3. 3차원 공간으로 Mapping 된 2차원 데이터

$$\begin{cases} Subject.to. & \sum_{i=1}^{N}\alpha_i t_i = 0 \\ Subject.to. & \alpha_i \ge 0 \\ Max. & \left\{\tilde{L}(\alpha)=\sum_{i=1}^{N}\alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j t_i t_j x_i{}^T x_j\right\} \end{cases} \quad (4)$$

지금까지 다룬 SVM은 Linear-SVM 으로써 실제 세계에서 발생하는 분류 문제에서 선형 분류기로 높은 성능을 얻는다는 것은 가능성이 낮다. 때문에 Nonlinear-SVM을 적용한다. Nonlinear-SVM은 명확하게 선형분리가 되지 않기 때문에 식(5)와 같이 α값의 범위에 C값이 추가되고 Kernel Function인 $K(x_i, x_i)$이 추가된다.

$$\begin{cases} Subject.to. & \sum_{i=1}^{N}\alpha_i t_i = 0i \\ Subject.to. & 0 \le \alpha_i \le C \\ Min. & \left\{\tilde{L}(\alpha)=\sum_{i=1}^{N}\alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j t_i t_j K(x_i, x_j)\right\} \end{cases} \quad (5)$$

여기에서 파라미터 C는 오분류와 성능간의 trade-off를 나타내는 비용 변수로 'trade-off 파라미터'라고 정의한다[6].

2.2절 Kernel Function

SVM에서 그림 2와 같이 입력 공간에서 데이터가 선형적으로 분류 되지 못할 때, 그림3과 같이 Kernel을 사용하여 입력공간의 데이터를 고차원의 특징 공간으로 Mapping 시킴으로 선형적으로 분류를 가능하게 할 수 있다[7].

표 1. Kernel Function 의 종류

Kernel Function	Type of Classifier
Polynomial	$K(x, y) = (x \cdot y)^p$
RBF	$K(x, y) = e^{-\parallel x-y \parallel^2/2\sigma^2}$
Sigmoid	$K(x, y) = \tanh(\beta_0 x \cdot y + \beta_1)$

그림 2에서 중앙의 원으로 표시된 데이터와 주변의 +로 표시된 데이터 2-클래스로 나뉘어있을 때 선형 분류가 되지 못하므로 Kernel을 사용하여 3차원 데이터로 변환 시킨다. 일반 적으로 Kernel은 식(6)과 같이 표현하며, 표(1)과 같이 Kernel Function으로 나눌 수 있다.

$$K(x, y) = \Phi(x) \cdot \Phi(y) \quad (6)$$

본 논문에서는 비선형적 2차원 데이터를 가지고 Kernel Function을 사용했을 때 각각의 오차율을 분석하였다.

3. 실 험

3.1 실험 설계

실험에는 랜덤함수를 돌려서 만든 150개의 데이터를 사용하였다. 이 데이터는 50개를 첫 번째 클래스 150개의 데이터가 두 번째 클래스로써 2-클래스로 구성되어 있다. 본 연구에서는 K.F(Kernel Function)을 사용하여 C 값에 따른 오분류 데이터와 SV(Support Vector)의 수를 측정하였다.

3.2 실험 결과

3.2.1. Polynomial

Polynomial Function 일 경우 그림 4와 같이 C값에 따라 SV수의 변화를 나타내었고 그림 5는 C값에 따라 오분류된 데이터수를 나타내었다. 그림 4에서 볼 수 있듯이 C값이 15 미만일 때 SV의 수는 28개 로 일정하다가 C값이 15이상일 때 SV의 수는 급격하게 상승하였다. 그림 5에서는 C값이 15 미만일 때 오분류된 데이터 수는 26개로 일정하다가 C값이 15이상일 때부터 오분류된 데이터 수는 급격하게 감소하다가 C값이 16일 때 오분류된 데이터 수는 16개로 가장 적게 나왔다. 즉 Polynomial Function을 사용할 때에는 C값을 15이상으로 해야 오분류 데이터수를 줄일 수 있다.

3.2.2. RBF(Radial Basis Function)

RBF를 사용했을 경우 그림 6과 같이 C값이 2와 16일 경우 SV의 수가 82개로 최대가 되었고 C값이 5와 7 사이일 경우 76개로 최소가 되었다. 그림 7은 C값이 10미만일 때와 14 이상일 때 오분류된 데이터가 50개로 일정하였고 C값이 10과 14사이일 때 오분류된 데이터수가 100이 되었다. 즉 RBF를 사용했을 때 C값이 2일 때 가장 최적화 시킬 수 있다.

3.3.3. Sigmoid

Sigmod를 사용했을 경우 그림 8과 같이 C값이 1일 때 SV의 수는 52개이고 C값이 4일 때 34개로 감소하였다. C값이 5에서 14사일 때 SV의 수는 36개로 일정하다가 C값이 14에서 15사이에 SV의 수는 40개로 증가하였고 C값이 15이상일 될

때 SV의 수는 급격하게 상승하여 C값이 16일 때 SV의 수는 120개가 되었다. 그림 9는 C값에 따른 오분류 데이터 수로써 C값이 3과 15사이일 때 오분류 데이터가 50개로 가장 적었고 C 값이 3미만일 때와 15이상일 때 오분류 데이터 수는 급격하게 상승하여 최대 100개까지 되었다. 즉 Sigmod를 사용했을 때 C값을 15로 해야 가장 최적화 시킬 수 있다.

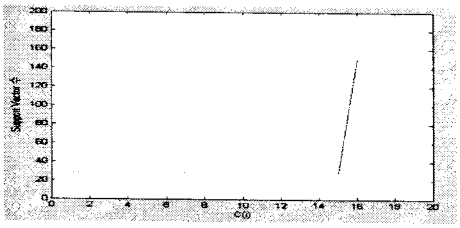
그림 4. Polynomial K.F의 C값에 따른 SV 수의 변화

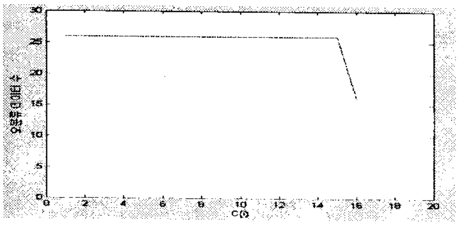
그림 5. Polynomial K.F의 C값에 따른 오분류 데이터 수의 변화

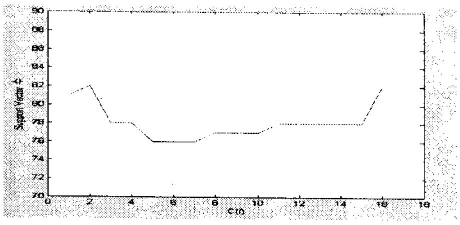
그림 6. RBF K.F의 C값에 따른 SV 수의 변화

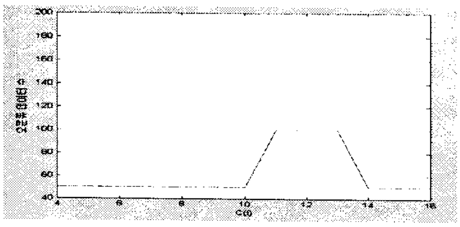
그림 7. RBF K.F의 C값에 따른 오분류 데이터 수의 변화

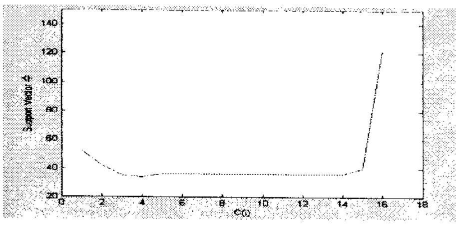
그림 8. Sigmoid K.F의 C값에 따른 SV 수의 변화

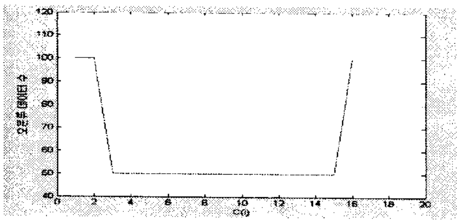
그림 7. Sigmoid K.F의 C값에 따른 오분류 데이터 수의 변화

4. 결 론

본 논문에서는 C값의 변화에 따라 데이터에 발생할 수 있는 오분류 데이터수와 Support Vector의 수에 변화를 측정하고 K.F(Kernel Function)을 사용하여 각각의 K.F에 대한 결과를 측정하고 분석하였다. C값의 변화는 α값의 범위 변화를 말한다. 그 결과 Nonlinear 데이터일 경우 오분류 데이터 수는 다른 Kernel 보다 적으나 C 값이 커야하므로 데이터 처리 시간이 늘어나고 RBF 일 경우에는 C 값에 따라 Support Vector의 변화가 가장 심하게 나왔다. Sigmoid일 때에는 Support Vector의 수가 Polynomial과는 반대가 되었다. 또한 RBF를 사용할 경우 σ값을 결정해야하고 Sigmoid를 사용할 경우 β_0, β_1값을 결정해야 하므로 각각의 파라미터 값에 따라 또 다른 오분류 데이터가 나오게 된다. Polynomial K.F는 RBF K.F과 Sigmoid K.F에 비해 처리 속도가 약20%빠르나 입력 데이터의 클래스수가 많아질 경우 RBF K.F가 분류기 성능은 약7%정도 높다. 이와 같은 실험 결과로 인해 2-클래스 입력 데이터일 때에는 C값에 따라 변화폭이 적고 결정해야하는 파라미터가 적은 Polynomial을 사용하는 것이 가장 적절하다. 본 논문으로부터 SVM 성능 개선을 위한 후속 연구에 대해 기초 자료로서 중요한 역할을 할 것으로 기대된다.

참 고 문 헌

[1] C. J. C. Burges, "Simplified Support Vector Decision Rules", IEEE Intl Conf. on Machine Learning, pp. 71-77, 1996

[2] Simon Haykin , " Neural Networks a Comprehensive Fundation 2nd ed", Prentice Hall, pp. 318-348, 1999.

[3] K. Crammer, J. Keshet, and Y. Singer, "Kernel Design Using Boosting", IEEE Conf. on Advances in Neural Information Processing Systems(NIPS), 2002

[4] J. Juang, X. Shao, and H. Wechsler, "Face Pose Discrimination using Support Vector Machines", Proc. of 14th ICPR, pp. 154-156, 1998

[5] C. Burges, "A tutorial on support vector machines for pattern recognition", Data Mining and Know-ledge Discovery 2, Vol. 2, No. 2, pp. 121-167, 1998.

[6] A. Ben-Hur, D. Horn, HT.Siegelmann, V. Vapnik, "A Support Vector Method for Clustering", Neural Information Processing Systems, pp. 367-373, 2000.

[7] N. Crammer, A. Elisseeff, J. Shawe-Taylor, J. Kanla, "On kernel target alignment", IEEE Conf. on Advances in Neural Information Processing Systems (NIPS), pp. 367-374, 2002.