

## Transductive SVM을 위한 분지-한계 알고리즘\*

박찬규\*\*

### A Branch-and-Bound Algorithm for Finding an Optimal Solution of Transductive Support Vector Machines\*

Chan-Kyoo Park\*\*

#### ■ Abstract ■

Transductive Support Vector Machine(TSVM) is one of semi-supervised learning algorithms which exploit the domain structure of the whole data by considering labeled and unlabeled data together. Although it was proposed several years ago, there has been no efficient algorithm which can handle problems with more than hundreds of training examples. In this paper, we propose an efficient branch-and-bound algorithm which can solve large-scale TSVM problems with thousands of training examples.

The proposed algorithm uses two bounding techniques: min-cut bound and reduced SVM bound. The min-cut bound is derived from a capacitated graph whose cuts represent a lower bound to the optimal objective function value of the dual problem. The reduced SVM bound is obtained by constructing the SVM problem with only labeled data. Experimental results show that the accuracy rate of TSVM can be significantly improved by learning from the optimal solution of TSVM, rather than an approximated solution.

Keyword : Transductive Support Vector Machine, Semi-supervised Learning, Branch-and-bound Algorithm, Optimal Solution

논문접수일 : 2005년 3월 21일    논문게재확정일 : 2006년 4월 18일

\* 본 논문은 한국과학재단의 해외 Post-doc. 연수지원에 의하여 연구되었음.

\*\* 동국대학교 경영학과

## 1. 서론

기계학습(machine learning)의 한 방법인 사례에 의한 학습(learning by examples)은 훈련데이터(training example)의 레이블(label)(또는 목표출력)이 알려져 있는 가에 따라 크게 지도학습(supervised learning)과 자율학습(unsupervised learning)으로 구분되어 왔다[9]. 지도학습은 모든 훈련데이터의 레이블이 알려진 상황에서 훈련데이터와 레이블 사이의 관계를 나타내는 결정함수(decision function)를 계산하고 이 결정함수는 레이블이 알려지지 않은 데이터의 레이블을 예측하는 데 사용된다. 반면, 자율학습은 훈련데이터의 레이블이 알려지지 않은 경우에 사용되는데, 주로 밀도예측(density estimation), 클러스터링(clustering) 등에 이용되어 왔다[11]. 최근에는 지도학습과 자율학습의 중간 형태를 띠는 반지도학습(semi-supervised learning)이 폭넓은 현실 문제 적용가능성으로 인해 주목을 받고 있다.

반지도학습은 레이블이 알려져 있는 소수의 훈련데이터와 레이블이 알려지지 않은 다수의 훈련데이터가 존재하는 상황을 다룬다. 예를 들어, 음성인식(speech recognition), 웹페이지 또는 하이퍼텍스트(hypertext) 등의 분류에서 훈련데이터는 손쉽게 획득할 수 있지만 훈련데이터가 어떤 그룹에 속하는 가를 알기 위해서는 전문가의 작업이 필요하기 때문에 훈련데이터의 레이블 획득에 많은 비용이 요구된다. 또한 의료진단(medical diagnosis)과 단백질체학(proteomics) 분야에서도 대량의 훈련데이터가 존재하지만 질병의 유무나 포함된 단백질의 종류 등을 나타내는 훈련데이터의 레이블을 얻기 위해서는 추가적인 시험 또는 실험을 거쳐야 하는 경우가 많다[12]. 따라서 레이블이 알려진 소수의 훈련데이터와 그렇지 않은 다수의 훈련데이터를 학습시 모두 고려하여 미지의 레이블을 예측하는 것이 반지도학습의 목적이다.

반지도학습은 '밀집된 클러스터(cluster)내의 훈련데이터는 서로 동일한 레이블을 가질 가능성이

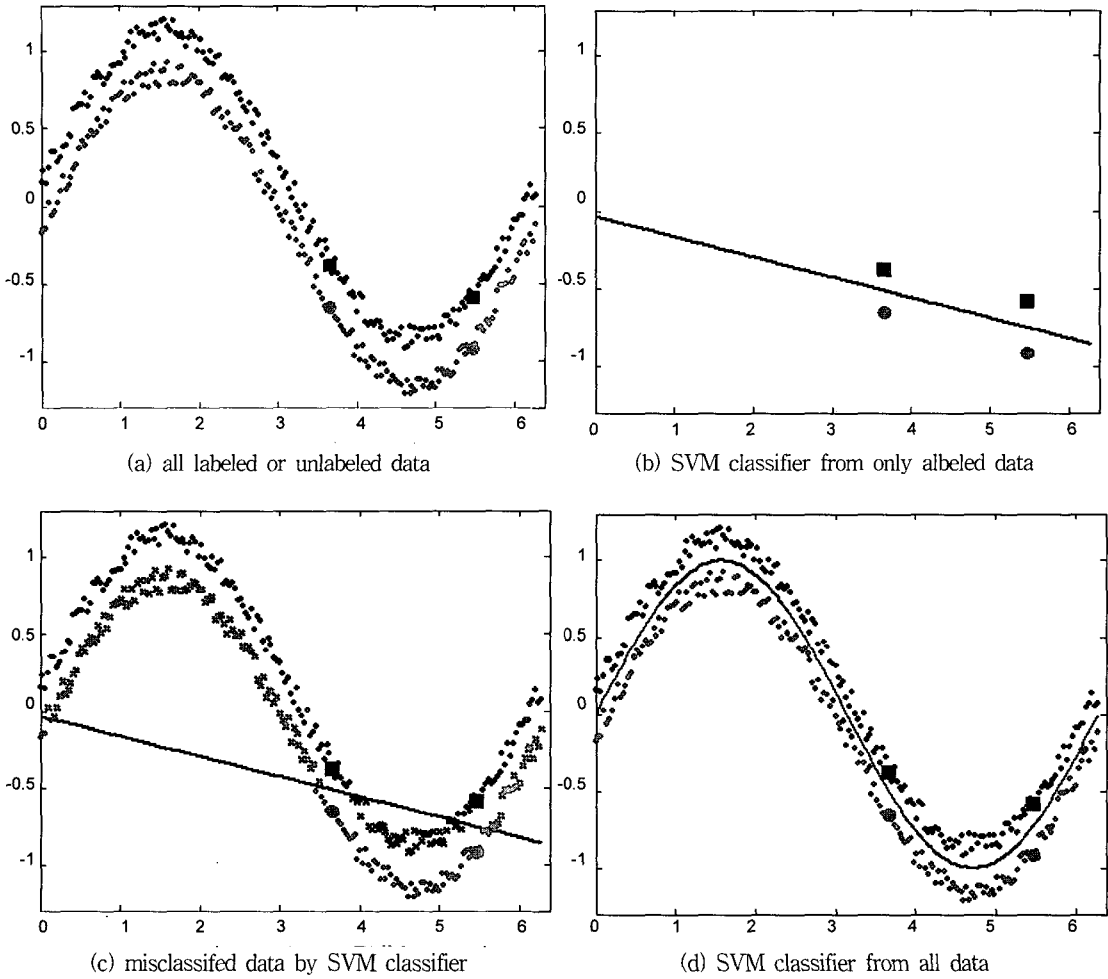
높다'라는 클러스터 가정(cluster assumption)에 기초하고 있다[5, 14]. 다시 말해 반지도학습에서는 레이블이 없는 다수의 훈련데이터로부터 도메인 구조(domain structure)에 관한 정보를 얻을 수 있으므로, 레이블이 있는 소수의 훈련데이터만을 이용하는 것보다 모든 훈련데이터를 동시에 이용함으로써 보다 정확한 결정함수(decision function)를 구할 수 있다고 가정한다. 예를 들어 [그림 1]을 보자. [그림 1](a)는  $z = \sin(x)$ 로부터 만들어진 예제데이터를 보여주고 있다.  $z = \sin(x)$  곡선의 윗부분의 점들은

$$z = \sin(x) + \epsilon + 0.07, \quad x = 0, \frac{2\pi}{180}, \frac{4\pi}{180}, \dots, 2\pi \quad (1.1)$$

식 (1.1)로부터 무작위로 생성된 훈련데이터를 나타낸다.  $\epsilon$ 는 0에서 0.15사이의 값을 갖는 일양분포(uniform distribution)에서 무작위로 생성된 값을 갖는다.  $z = \sin(x)$  곡선의 아래의 점들은

$$z = \sin(x) - \epsilon - 0.07, \quad x = 0, \frac{2\pi}{180}, \frac{4\pi}{180}, \dots, 2\pi \quad (1.2)$$

식 (1.2)로부터 무작위로 생성된 훈련데이터로,  $\epsilon$ 는 식 (1.1)과 동일하게 생성된 값이다. 식 (1.1)에 의해 생성된 점들이 '+1' 클래스(class)를 이루고, 식 (1.2)에 의해 생성된 점들은 '-1' 클래스라 하자. 반지도학습의 유용성을 예시하기 위해 [그림 1](a)에서 4개의 훈련데이터(■는 '+1' 클래스의 점, ●는 '-1' 클래스의 점을 뜻함)만 레이블을 알고 있다고 가정하고 나머지 훈련데이터는 레이블을 모른다고 가정해보자. 레이블을 알고 있는 훈련데이터만을 고려하게 되면 [그림 1](b)에 나타난 직선과 같은 결정함수를 얻게 된다. 이 결정함수를 사용하여 레이블을 모른다고 가정한 훈련데이터의 레이블을 예측해보면 [그림 1](c)와 같다. (c)에서 'x'로 표시된 점은 레이블이 틀리게 예측된 데이터를 나타내는데 그림에서 보는 바와 같이 레이블이 알려져 있는 데이터만 가지고 학습할 경우 상당히 많은 데이터가 잘못 분류되고 있다. [그림 1](d)는 레이블을 알고 있는 훈련데이터와 그렇지 않은 훈련데이터를 모두 활용하여 학습된 결정함수를 나타내고 있는데, 모



[그림 1] 지도학습과 반지도학습의 예

든 데이터를 정확하게 분류하고 있다. 이처럼 반지도학습은 학습단계에서 주어진 모든 데이터를 활용함으로써 보다 정확한 결정함수를 찾아 낼 수 있다.

지금까지 제안된 반지도학습 방법은 크게 그래프(graph) 기반 방법과 SVM를 확장한 Transductive SVM(TSVM) 방법으로 분류할 수 있다. 그래프 기반 방법은 절단(cut)이나 마코프 랜덤워크(Markov random walk) 등을 활용하여 반지도학습을 모델링한다. Blum and Chawla[3]는 각 훈련데이터간의 유사성(similarity)을 계산하여 그래프를 만든 다음 절단집합(cut set)에 속하는 호(arc)의 유사성 합이 최소가 되도록 그래프를 분할하는 최소절단(min-

cut)을 구함으로써 미지의 레이블을 예측하는 방법을 제시하였다. 그러나 일반적으로 최소절단은 그래프를 불균형하게 분할하는 단점이 있어 이를 해결하기 위한 방법으로 정규화 절단(normalized cut)이 이미지 분할(image segmentation)에 사용되어 왔고[13], 정규화 절단을 반지도학습 문제에 적용하는 연구가 수행되기도 하였다[7, 16]. 또한 Szummer and Jaakkola[14]와 Zhu and Ghahramani[17]는 마코프 랜덤워크(Markov random walk) 개념을 도입하여 반지도학습을 수행하는 방법을 제안하였다.

한편 Vapnik[15]은 지도학습에 사용되는 SVM(Support Vector Machine)을 확장하여 레이블이 있

는 훈련데이터와 그렇지 않은 훈련데이터를 모두 고려하는 반지도학습 방법으로 TSVM을 제안하였다. SVM은 이차계획법문제에 표현되어 분할방법(decomposition method)에 기반한 효과적인 해법들이 개발되어 있다[6, 10]. 반면에 TSVM은 정수제약을 포함하는 이차계획법문제에 효율적인 해법 개발이 쉽지 않아 훈련데이터의 개수가 많아짐에 따라 학습시간이 급격히 증가한다[1, 8]. 이로 인해 지금까지의 TSVM에 관한 연구는 주로 최적해가 아닌 근사 최적해로부터 결정함수를 구하는데 중점을 두고 있다. Joachims[8]은 레이블이 알려진 데이터만을 고려하여 초기 결정함수를 구한 다음 레이블이 알려지지 않은 데이터의 레이블 예측값을 변경함으로써 결정함수를 개선하는 휴리스틱 방법을 제안하였다. Bennet and Demiriz[1]은 TSVM을 0-1 혼합정수계획법 문제로 모형화하고 상용최적화 패키지인 CPLEX를 사용해 TSVM를 풀 결과를 제시하였으나, 계산시간에 의한 제약 때문에 실험에 사용된 훈련데이터의 개수는 50개 정도로 매우 작았다. 또 Bennet and Demiriz[1]은 일반적인 TSVM의 이차 목적함수가 아니라 선형목적함수를 사용하였으며, 제시된 혼합정수계획법은 특징공간이 아닌 입력공간에서만 사용될 수 있다는 한계점을 지닌다. 최근에 Bie and Cristianini[2]는 정수 제약을 완화하여 TSVM을 반정치계획법(semidefinite programming)으로 모형화하는 방법을 제안하였으나, TSVM의 변수 개수에 비해 변환된 반정치계획법의 변수 개수가 매우 많고 반정치계획법 해법의 계산복잡도도 커서 대형문제를 푸는 데에는 한계가 있다.

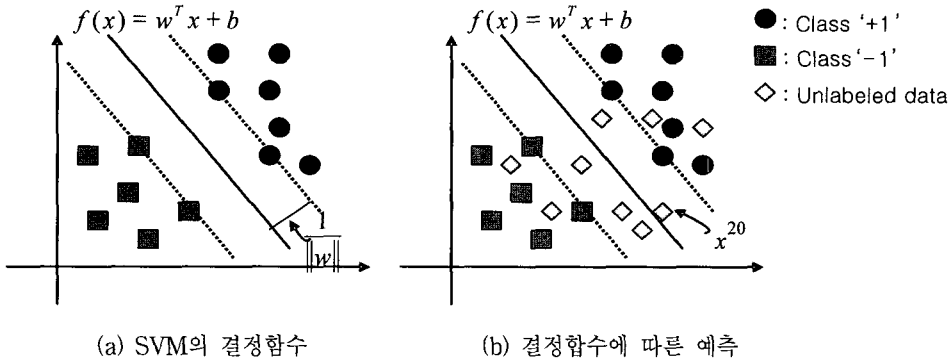
본 연구의 목적은 TSVM의 특성을 활용하여 TSVM를 풀 수 있는 분지한계 알고리즘(branch-and-bound algorithm)을 제안하는 것이다. 특징공간에서 정의되고 이차함수 형태의 목적함수를 갖는 일반적인 TSVM 문제의 최적해를 구하는 알고리즘은 아직까지 개발된 바 없다. 분지한계 알고리즘은 그래프 기반의 새로운 접근방법을 통해 부분제의 목적함수 하한을 구하여 이를 한계전략으로 사용하게 된다. 실험에 의하면 제안된 알고리즘은 기존 해

법에서도 다룰 수 없었던 1,000개 이상의 훈련데이터를 갖는 대형문제도 해결할 수 있었다는 점에서 본 연구의 가장 중요한 의의를 찾을 수 있다. 또한 근사최적해에 의한 기존의 TSVM의 정확도보다 최적해에 의한 TSVM의 정확도가 보다 높음을 보임으로써 TSVM의 유용성을 입증했다는 점에서 본 연구의 두 번째 의의를 찾을 수 있다. 본 연구에서 제시된 TSVM의 정확도는 향후 다양한 반지도학습 알고리즘의 성능을 비교하는 기준자료로 활용될 수 있을 것이다.

본 논문의 구성은 1장에서 연구 개요를 설명하고, 2장에서 SVM과 TSVM의 수리모형을 소개하고 그 의미를 설명한다. 3장에서는 제안할 분지한계 알고리즘의 한계전략(bounding strategy)을 제시하고, 4장에서 분지전략(branching strategy)을 제시한다. 5장에서는 3장과 4장에서 제시된 한계전략과 분지전략을 종합하여 분지한계 알고리즘을 제안하고, 6장에서 실험결과를 제시한다. 마지막으로, 7장에서 본 연구의 결론을 제시한다.

## 2. SVM과 TSVM

먼저 SVM에 대해 알아보자. 1장에 언급했듯이 SVM은 기계학습에 사용되는 기법이다. SVM은 다차원 상의 점들로 표현되는 훈련데이터를 두 개의 그룹으로 분할하는 초평면(hyperplane)을 구하고, 이 초평면을 미지의 데이터가 어떤 그룹에 속할지를 예측하는 결정함수(decision function)로 사용한다.  $l$ 개의 훈련데이터  $(x^1, y_1), \dots, (x^l, y_l)$ 가 있고  $x^i \in R^n$ ,  $y_i \in \{1, -1\}$ 이며 모든  $y_i$ 의 값을 알고 있다고 하자. 즉  $x^i$ 는  $i$ 번째 훈련데이터의 다차원상의 좌표이고  $y_i$ 는  $i$ 번째 훈련데이터가 '+1' 클래스(class)에 속하는지 또는 '-1' 클래스(class)에 속하는지를 나타내는 레이블(label)이다. [그림 2](a)에서 보는 것처럼 SVM은 훈련데이터를 그 레이블  $y_i$ 에 따라 두 개의 클래스로 분할하는 초평면 중에서 두 개의 클래스가 가장 멀리 떨어지도록 분할하는 초평면을 찾는다. 이때 초평면  $f(x) = w^T x + b$ 가 SVM의 결정



[그림 2] SVM의 결정함수와 예측

함수가 된다. 레이블이 알려지지 않은 데이터  $x$ 에 대해  $f(x) \geq 0$ 이면 '+1' 클래스로,  $f(x) < 0$ 이면 '-1' 클래스로 예측한다. 여기서  $(2/\|w\|)$ 를 마진(margin)이라 부르는데 초평면에 의해 분할된 두 클래스간의 최단거리를 나타낸다. SVM은 개념적으로 마진을 최대로 하는 초평면을 결정함수로 사용한다.

SVM의 결정함수  $f(x) = w^T x + b$ 의  $w$ 와  $b$ 를 구하는 문제를 수리계획법으로 모형화하면 다음 ( $SP_0$ )와 같이 이차계획법 문제로 모델링된다[15].

$$\min w^T w$$

$$(SP_0): \text{ s.t. } y_i(w^T x^i + b) \geq 1, \text{ for } i = 1, \dots, l$$

그러나 현실 문제에서는 [그림 2](a)처럼 훈련데이터를 레이블에 따라 초평면으로 완벽하게 분할하지 못하고 중첩(overlapping)되는 경우가 많다. 따라서 ( $SP_0$ )의 각 제약식을 완화한 모형인 ( $SP$ )를 사용한다. ( $SP$ )의 쌍대문제는 ( $SD$ )와 같은데, 계산상의 여러 가지 장점 때문에 실제로는 ( $SD$ )를 푼다[6]. ( $SD$ )에서  $e = (1, \dots, 1)^T \in R^l$  이고,  $Q_{i,j} = y_i y_j (x^i)^T x^j$  이다.

$$\min w^T w + C \sum_i \xi_i$$

$$(SP): \text{ s.t. } y_i(w^T x^i + b) \geq 1 - \xi_i, \forall i$$

$$\xi_i \geq 0, \forall i$$

$$\max e^T \alpha - \frac{1}{2} \alpha^T Q \alpha$$

$$(SD): \text{ s.t. } y^T \alpha = 0,$$

$$0 \leq \alpha_i \leq C, \forall i$$

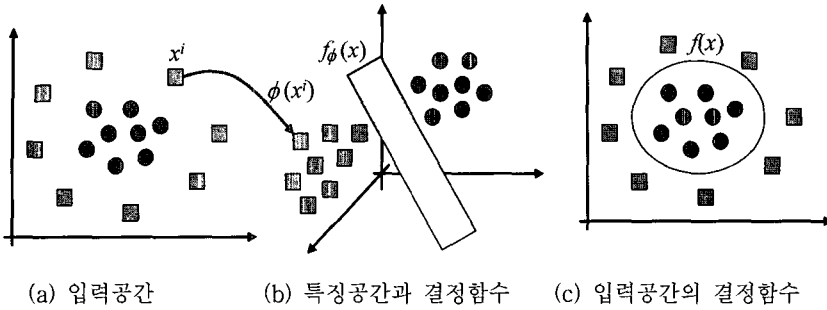
( $SP$ )의 목적함수는 두 개의 항으로 구성되어 있다. 첫 번째 항은  $w^T w = \|w\|^2$ 으로 마진의 역수이다. 결과적으로  $\|w\|^2$ 을 최소화하는 것은 마진을 최대화하는  $w$ 를 찾는 것을 의미한다. 두 번째 항의  $\sum \xi_i$ 는 구해진 결정함수가 주어진 훈련데이터에 대해 얼마만큼 오류(error)를 갖는 가를 나타내는데, 경험오류(empirical error)라 부른다. 여기서 상수  $C$ 는 마진과 경험오류간의 상대적 가중치를 나타낸다. 종합해보면 ( $SP$ )의 목적함수는 마진을 최대화하고 경험오류를 최소화하는 결정함수  $f(x)$ 를 찾는을 의미한다. 또 ( $SD$ )의 최적해를  $\alpha^*$ 라 하면 ( $SP$ )와 ( $SD$ )의 최적조건(optimality condition)으로부터  $w$ 와 결정함수  $f(x)$ 는 각각

$$w = \sum_i \alpha_i^* y_i x^i$$

$$f(x) = w^T x + b = \sum_i \alpha_i^* y_i (x^i)^T x + b$$

로 표현된다. 즉 결정함수의 법선벡터(normal vector)  $w$ 는 훈련데이터의 선형조합(linear combination)으로 표현되며 이때  $\alpha_i^* > 0$ 인  $x^i$ 를 지지벡터(support vector)라 부른다. 기하학적 관점에서 지지벡터는 두 클래스의 경계에 있는 훈련데이터라고 볼 수 있고, 결정함수는 미지의 훈련데이터가 지지벡터로부터 가능한 멀리 떨어지도록 구성된다.

SVM은 훈련데이터들을 분할하는 초평면을 결정함수로 사용하는데 [그림 2]에 나타난 바와 같이 결정함수가 단순히 훈련데이터  $x^i$ 들의 선형함수로만



[그림 3] 입력공간과 특징공간

제약된다면 SVM은 성능 면에서 많은 한계를 가지게 될 것이다. 이러한 문제점을 극복하기 위해 [그림 3](a)와 같이 입력공간(input space)에 있는 훈련데이터  $x^i$ 를 고차원의 특징공간(feature space)의 점  $\phi(x^i)$ 로 변환한 다음 [그림 3](b)와 같이 특징공간의 점들을 그 레이블에 따라 두 개의 그룹으로 분할하는 결정함수  $f_\phi(x)$ 를 구한다. 특징공간의 결정함수  $f_\phi(x)$ 을 원래의 입력공간으로 변환하면 [그림 3](c)와 같이 비선형함수 형태를 띠게 된다.

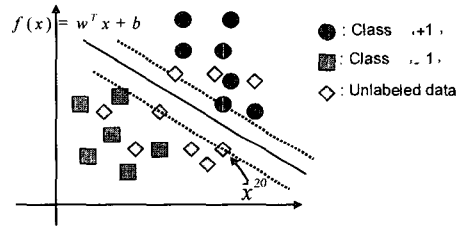
훈련데이터  $x^i$ 를 특징공간의 점으로 변환하는 과정은 함수  $\phi: x \rightarrow \phi(x^i)$ 를 통해 이루어진다. 그 결과로 특징공간에서 정의되는 SVM는 (SP)와 (SD)에서  $x^i$  대신에  $\phi(x^i)$ 를 써서 나타낼 수 있다. 이때 특징공간에서 정의되는 (SD)를 풀기 위해서는  $Q_{ij} = \phi(x^i)^T \phi(x^j)$ 를 계산해야 한다. 만약  $\phi(x^i)$ 를 명시적으로 구하여  $Q_{ij}$ 를 계산하려면  $\phi(x^i)$ 의 차원이 매우 높거나 또는 무한 차원인 경우도 있으므로 많은 계산시간이 소모될 것이다. 따라서 이러한 문제점을 해결하기 위해 함수  $\phi$ 를 직접 구할 필요 없이 커널함수(kernel function)를 통해 특징공간으로의 변환이 이루어진다. 커널함수는  $\phi(x^i)$ 와  $\phi(x^j)$ 를 명시적으로 구하지 않고도  $Q_{ij} = \phi(x^i)^T \phi(x^j)$ 를 쉽게 계산할 수 있게 해주는데, 커널함수로 가장 많이 사용되는 함수는 RBF(Radial Basis Function)커널, 다항(polynomial)커널 등이 있다. RBF 커널은

$$k(x^i, x^j) = \exp\left(-\frac{\|x^i - x^j\|^2}{\rho^2}\right)$$

이고, 다항 커널은

$$k(x^i, x^j) = ((x^i)^T x^j)^p$$

이다. RBF 커널은 훈련데이터  $x^i \in R^n$ 를 무한 차원을 갖는 특징공간의 점으로 변환하고, 다항커널은 입력데이터를  $\binom{n+d-1}{d}$  차원의 특징공간의 점으로 변환한다.



[그림 4] TSVM의 결정함수

다음으로 TSVM에 대해 알아보자. SVM은 학습할 때 레이블을 알고 있는 데이터만을 고려하지만 TSVM은 레이블이 미지인 훈련데이터까지 포함하여 학습을 수행한다. [그림 2](c)와 [그림 4]를 비교해 보자. [그림 2](c)는 레이블이 알려진 데이터만을 포함하여 학습을 수행한 결과이고 [그림 4]는 레이블이 미지인 훈련데이터까지 모두 포함하여 학습한 결과이다. [그림 4]에서 보는 바와 같이 훈련데이터의 분포를 보면  $x^{20}$ 은 클래스 '-1'에 속하는 데이터라고 판단할 수 있다. [그림 4]와 같이 TSVM의 결정함수는  $x^{20}$ 를 클래스 '-1'에 속하는 데이터로 예측하지만, [그림 2](c)와 같이 SVM은  $x^{20}$ 를 클래스 '+1'에 속하는 데이터로 잘못 예측한다. TSVM은 레이블이 미지인 데이터까지 모두 포함해서 전체 데이터를 두 개의 클래스로 분할하는 평면을 찾는

다. TSVM은 개념적으로 레이블이 알려진 데이터는 그 레이블에 따라 분류하고, 레이블이 미지인 데이터는 마진이 최대가 되도록 분류한다.

레이블을 알고 있는  $l$ 개의 훈련데이터  $(x^1, y_1), \dots, (x^l, y_l)$ 와 레이블이 미지인  $u$ 개의 훈련데이터  $(x^{l+1}, y_{l+1}), \dots, (x^{l+u}, y_{l+u})$ 가 있다고 할 때, TSVM은  $(P)$ 와 같이 나타낼 수 있다[15].

$$\begin{aligned} \min \quad & w^T w + C \sum_{i=1}^{l+u} \xi_i \\ (P) : \text{s.t.} \quad & y_i (w^T x^i + b) \geq 1 - \xi_i, \forall i \\ & \xi_i \geq 0, \forall i \\ & y_i \in \{1, -1\}, \text{ for } i=l+1, \dots, l+u \end{aligned}$$

$(SP)$ 와  $(P)$ 의 차이점은 레이블이 미지인 훈련데이터  $x^i$ 의 레이블  $y_i$ 가 변수라는 점이다. 이때  $y_i$ 는 +1 또는 -1의 값을 가져야 한다. 이로 인해  $(P)$ 는 정수제약을 갖는 이차계획법 문제가 된다.

$(P)$ 의 쌍대문제는  $(D)$ 와 같다[15].  $(D)$ 에서  $e = (1, \dots, 1)^T \in R^{l+u}$ 이고,  $Q_{ij} = y_i y_j x_i^T x_j$ 로  $Q$ 는  $(l+u) \times (l+u)$  행렬이다.

$$\begin{aligned} \min_{y_{l+1}, \dots, y_{l+u}} \max_{\alpha} \quad & e^T \alpha - \frac{1}{2} \alpha^T Q \alpha \\ (D) : \text{s.t.} \quad & y^T \alpha = 0, \\ & 0 \leq \alpha_i \leq C, \forall i \end{aligned}$$

$(D)$ 와  $(SD)$ 는 편의상 동일한 기호  $e$ 와  $Q$ 를 사용하고 있으나 차원이 다름에 주의하자. 혼란을 피하기 위해 이후의 논의에서  $e$ 와  $Q$ 는  $(D)$ 에서 사용된 차원을 갖는다. 또한 TSVM도 SVM와 동일하게  $x^i$ 를 특징공간의 점으로 변화하게 된다. 따라서 이후의 논의에서  $x^i$ 는 모두 특징공간으로 변환된 점  $\phi(x^i)$ 를 나타내기로 하고  $(P)$ 와  $(D)$ 도 특징공간에서 정의되는 TSVM 문제를 지칭하기로 하자.

### 3. 한계전략

분지한계 알고리즘은 미지의 레이블인  $y_i (i=l+1, \dots, l+u)$ 을 +1 또는 -1로 설정해 나가면서  $(P)$ 의

목적함수값이 최소가 되는  $y_i$ 와  $w$ 를 구한다. 따라서 분지한계 알고리즘에 의해 생성되는 부분제에서 미지의 레이블 중 일부는 값이 +1 또는 -1로 고정되고 나머지 미지의 레이블은 고정되지 않은 채 있게 된다. 이러한 상황에서 부분제의 최적목적함수값의 범위를 알면 불필요한 분지를 줄일 수 있어 결과적으로 생성되는 부분제(subproblem) 개수를 줄일 수 있다. 본 절에서는 그래프를 사용하여  $(P)$ 의 최적목적함수값의 하한을 구하는 방법과 축소된 SVM를 사용한 한계전략 등 두 가지 한계전략을 새롭게 제시한다.

#### 3.1 최소절단을 이용한 한계전략

분지한계 알고리즘에서 각 부분제는 레이블이 정해진 데이터와 레이블이 미지인 데이터의 인덱스를 통해 유일하게 식별할 수 있다. 분지한계 알고리즘의 임의의 부분제에서 레이블이 정해져 있는 훈련데이터의 인덱스 집합과 레이블이 정해지지 않은 훈련데이터의 인덱스 집합을 각각  $L$ 과  $U$ 라 나타내자.  $L$ 에서 레이블이 '+1' 또는 '-1'인 데이터의 인덱스 집합을 각각  $L^+$ 과  $L^-$ 로 나타내자. 물론  $U = \{1, \dots, l+u\} - (L^+ \cup L^-)$ 이다. 분지한계 알고리즘의 부분제  $(P(L^+, L^-))$ 를 수식으로 나타내면 다음과 같다.

$$\begin{aligned} \min \quad & f_p(w, \xi) = \frac{1}{2} w^T w + C \sum_{i=1}^{l+u} \xi_i \\ (P(L^+, L^-)) : \text{s.t.} \quad & y_i (w^T x^i + b) \geq 1 - \xi_i, \forall i \\ & \xi_i \geq 0, \forall i \\ & y_i \in \{1, -1\}, \text{ for } i \in U \end{aligned}$$

훈련데이터의 전체 인덱스 집합  $\{1, 2, \dots, l+u\}$ 을 두 개의 부분집합  $S^+$ 와  $S^-$ 로 분할(partition)한 것을  $\sigma = (S^+, S^-)$ 로 나타내자. 즉  $\sigma$ 은 전체 훈련데이터를  $S^+ \cup S^- = \{1, 2, \dots, l+u\}$ ,  $S^+ \cap S^- = \emptyset$ 을 만족하는 인덱스 집합  $S^+$ 와  $S^-$ 로 분할하는 것을 말한다.  $\sigma = (S^+, S^-)$ 에서  $i \in S^+$ 이면 미지의 레이블  $y_i$ 가 +1로 설정됨을 의미하고  $i \in S^-$ 이면  $y_i$ 의 레이블이 -1로 설정됨을 의미한다. 레이블이 알려져 있는

$y_i$ 에 대해  $y_i = +1$ 이면  $i \in S^+$ ,  $y_i = -1$ 이면  $i \in S^-$ 인 분할  $\sigma = (S^+, S^-)$ 을 유효분할(valid partition)이라 하자. 본 연구에서는 유효분할만을 고려하므로 앞으로의 논의에서 분할이라 하면 모두 유효분할을 의미하기로 한다.

부분제  $(P(L^+, L^-))$ 의 쌍대문제  $(D(L^+, L^-))$ 를 유효분할을 써서 표현하면 다음과 같이 쓸 수 있다. 쌍대문제  $(D(L^+, L^-))$ 의 모든 유효분할의 집합을  $\Omega$ 라 표시하자.

$$\begin{aligned} \min_{\sigma = (S^+, S^-) \in \Omega} \max_{\alpha} f_{D\sigma}(\alpha) \\ = \sum_{i=1}^{l+u} \alpha_i - \frac{1}{2} \left\| \sum_{i \in S^+} \alpha_i x^i - \sum_{i \in S^-} \alpha_i x^i \right\|^2 \\ (D(L^+, L^-)) : \text{s.t.} \sum_{i \in S^+} \alpha_i = \sum_{i \in S^-} \alpha_i \\ 0 \leq \alpha_i \leq C, \forall i \end{aligned}$$

부분제  $(P(L^+, L^-))$ 에서 정의되는 임의의 유효분할  $\sigma = (S^+, S^-)$ 는  $L^+ \subset S^+$ 이고  $L^- \subset S^-$ 이다. 쌍대문제  $(D(L^+, L^-))$ 를 살펴보면 임의의 유효분할  $\sigma = (S^+, S^-)$ 에 대해 다음과 같이  $(D(\sigma))$ 를 풀게 된다.

$$\begin{aligned} \max_{\alpha} f_{D\sigma}(\alpha) \\ = \sum_{i=1}^{l+u} \alpha_i - \frac{1}{2} \left\| \sum_{i \in S^+} \alpha_i x^i - \sum_{i \in S^-} \alpha_i x^i \right\|^2 \\ (D(\sigma)) : \text{s.t.} \sum_{i \in S^+} \alpha_i = \sum_{i \in S^-} \alpha_i \\ 0 \leq \alpha_i \leq C, \forall i \end{aligned}$$

여기서  $f_{D\sigma}(\alpha)$ 의 최적함수값을  $f_{D\sigma}^*$ 라 하면 다음식이 성립한다. 단,  $f_D^*$ 는  $(D(L^+, L^-))$ 의 최적목적함수값이다.

$$\min_{\sigma = (S^+, S^-) \in \Omega} f_{D\sigma}^* = f_D^*$$

즉  $\sigma \in \Omega$ 인 임의의 유효분할  $\sigma$ 에 대해  $f_{D\sigma}^*$ 를 구하여 그 중에 최소값이  $(D(L^+, L^-))$ 의 최적목적함수값이 된다. 그러나  $(D(L^+, L^-))$ 에서 정의되는 유효분할의 개수는  $2^{(l+u) - |L^+ \cup L^-|}$ 로 현실적으로 모든 유효분할을 열거할 수가 없다. 또한 각 유효분할에서 정의되는  $(D(\sigma))$ 의 최적해를 구하는 것도 많은 계산

을 수반한다. 따라서  $(D(L^+, L^-))$ 를 푸는 것보다는  $(D(L^+, L^-))$ 를 이용하여 분지한계알고리즘의 한계를 계산하는 방법을 알아보자.

$(D(L^+, L^-))$ 를 이용하여 한계를 계산하는 방법은 그래프를 통해 이루어진다. 그래프는 임의의 유효분할  $\sigma$ 에 대해  $(D(\sigma))$ 의 가능해와 그 가능해의 목적함수 근사값을 쉽게 계산할 수 있게 해준다.  $(D(\sigma))$ 의 어떤 가능해의 목적함수값을  $\bar{f}_{D\sigma}$ 하면  $(D(\sigma))$ 는 최대화문제이므로  $\bar{f}_{D\sigma} \leq f_{D\sigma}^*$ 가 성립한다. 따라서 모든 유효분할에 대해서도

$$\min_{\sigma = (S^+, S^-) \in \Omega} \bar{f}_{D\sigma} \leq \min_{\sigma = (S^+, S^-) \in \Omega} f_{D\sigma}^* = f_D^* \quad (3.1)$$

가 성립한다. 본 연구에서 제시하는 그래프를 이용한 한계 계산 방법에서는 임의의 유효분할은 그래프에서 점집합을 두 개로 분할하는 것과 대응되고, 유효분할  $\sigma$ 에 의해 정의되는  $(D(\sigma))$ 의 하한은 그래프의 절단(cut)에 대응된다. 따라서 그래프의 최소절단(min-cut)을 계산함으로써  $(D(L^+, L^-))$ 의 하한을 계산할 수 있다. 먼저  $(D(L^+, L^-))$ 로부터 그래프를 생성하는 방법을 설명하고 이후에 그래프의 최소절단이  $(D(L^+, L^-))$ 의 하한이 됨을 보인다.

특징공간의 두 훈련데이터  $\phi(x^i)$ 와  $\phi(x^j)$ 간의 거리를  $d_{ij}$ 로 나타내면,  $d_{ij}$ 는

$$\begin{aligned} d_{ij} &= \|\phi(x^i) - \phi(x^j)\| \\ &= \sqrt{k(x^i, x^i) + k(x^j, x^j) - 2k(x^i, x^j)} \end{aligned}$$

로 정의된다. 본 연구에서는  $i \neq j$ 이면  $d_{ij} > 0$ 라 가정한다.  $(D(L^+, L^-))$ 로부터 다음 (C1)~(C4)를 만족하도록 생성된 무방향 그래프(undirected graph)를  $H = (N, E_H)$ 라 정의한다. (C2)에서  $KNN(\phi(x))$ 는  $x$ 에 최인접한  $K$ 개의 훈련데이터 집합을 나타낸다.

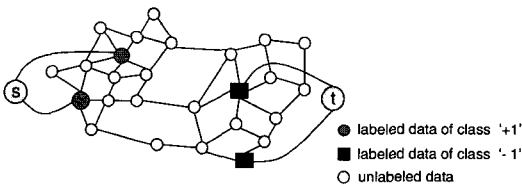
#### 그래프 생성 조건

- (C1)  $N = \{1, 2, \dots, l+u\} \cup \{s, t\}$ ,
- (C2)  $(i, j) \in E_H$ , if  $\phi(x^i) \in KNN(\phi(x^j))$  or  $\phi(x^j) \in KNN(\phi(x^i))$ ,
- (C3)  $(s, i) \in E_H$ , if  $i \in L^+$ ,
- (C4)  $(i, t) \in E_H$ , if  $i \in L^-$ .



(C1)은 훈련데이터  $x^i$ 는  $H$ 의 점(node)  $i$ 에 대응되고 그 외에 임시점(dummy node)인  $s$ 와  $t$ 가  $H$ 에 포함되어 있음을 말한다. (C2)는 훈련데이터  $\phi(x^i)$ 에 대해  $\phi(x^j)$ 가  $KNM(\phi(x^i))$ 에 속하면 호(arc)  $(i, j)$ 가  $E_H$ 에 있음을 말한다. (C3)는 레이블이 +1인 데이터와  $s$  사이가 호에 의해 연결됨을 의미하고, 마지막으로 (C4)는 레이블이 -1인 모든 데이터와  $t$  간에 호가 있음을 나타낸다. (C3)과 (C4)에 의해 생성되는 호를 임시호(dummy arc)라 하며, (C2)에 의해 생성되는 호를 정규호(normal arc)라 부르기로 한다. (C3)와 (C4)에 의해 알 수 있듯이  $H$ 는  $L^+$ 와  $L^-$ 에 따라 달라지는데  $H(L^+, L^-)$  대신에 편의상  $H$ 로 나타내고 있음에 유의하자. 이후  $H$ 로부터 생성되는 모든 부분그래프도  $H$ 처럼  $L^+$ 와  $L^-$ 와의 종속관계를 생략하고 간단히 표현하기로 한다. 또 (C2)의  $KNM(\cdot)$ 에 포함되는 최인접 데이터의 개수  $K$ 를 조정함으로써  $H$ 는 항상 연결(connected)되게 할 수 있으므로 앞으로의 논의에서  $H$ 는 연결되었다고 가정한다.

예로  $H$ 가 어떻게 생성되는지 알아보자. [그림 5]에서 ●는 '+1' 레이블을 갖는 데이터, ■는 '-1' 레이블을 갖는 데이터를 나타낸다. '+1' 레이블을 데이터가 2개, '-1' 레이블을 갖는 데이터가 2개 있으며, 나머지 ○로 표시된 데이터는 모두 레이블이 미지인 데이터를 의미한다. 임시점  $s$ 와  $t$ 를 추가하고, 각 점별로 인접한  $K$ 개의 훈련데이터를 찾아 서로 연결한 다음 마지막으로 '+1' 레이블을 갖는 점과  $s$  간의 호를 생성하고 '-1' 레이블을 갖는 점과  $t$  간의 호를 생성하면 그래프  $H$ 가 만들어진다.



[그림 5] 그래프  $H$  구성 예

다음으로 그래프  $H$ 로부터 부분그래프(subgraph)  $G$ 를 생성하는데, 그래프  $G$ 는  $H$ 와 달리 각 호

$(i, j)$ 의 용량이 정의된다. 그래프  $G$ 의 점은  $H$ 와 동일하며  $G$ 의 호는 다음 조건 (C5)와 (C6)를 만족하도록  $H$ 에서 추출한다. 또한,  $G$ 의 각 호의 용량  $c_{ij}$ 를 다음 (C7)과 (C8)에 따라 정의한다. 용량  $c_{ij}$ 는 비음(non-negative)이어야 하는데 이를 만족하도록 (C7)의  $r_{ij}$ 를 설정하게 된다. 여기서  $Adj_G(i)$ 는 그래프  $G$ 에서 점  $i$ 에 인접한 점집합을 나타낸다.

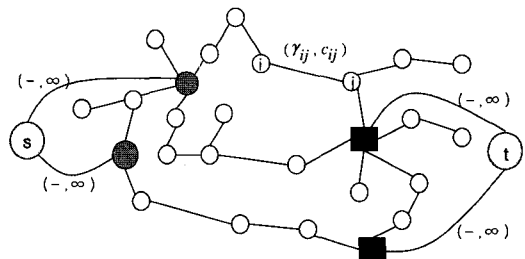
**그래프 생성 조건**

- (C5)  $G$ 는 연결(connected)되어야 한다.
- (C6)  $s$ 로부터  $t$ 까지 가는 경로(path)의 개수가 최대  $p$ 개 이하여야 한다.
- (C7) 모든 정규호  $(i, j)$ 에 대해, 아래 제약식을 만족하는 비음(nonnegative)인 값  $\gamma_{ij}$ 가 정의된다.

$$\sum_{j \in Adj_G(i) - s, t} \gamma_{ij} \leq C, \forall i = 1, \dots, l + u$$

- (C8) 정규호  $(i, j)$ 의 용량  $c_{ij}$ 는  $c_{ij} = 2\gamma_{ij} - (p/2)d_{ij}^2\gamma_{ij}^2$ 로 정의한다.
- (C9) 모든 임시호의 용량은  $\infty$ 로 둔다.

예로 [그림 5]의  $H$ 로부터  $p=3$ 인 그래프  $G$ 를 [그림 6]과 같이 구성할 수 있다. [그림 6]에서 정규호  $(i, j)$ 에 대해서는  $\gamma_{ij}$ 와 그에 따른 용량  $c_{ij}$ 가 정의된다. 또한 모든 임시호에 대해  $\gamma_{ij}$ 는 정의되지 않지만 용량  $c_{ij}$ 는  $\infty$ 로 정의된다. 추후 다시 설명되지만  $r_{ij}$ 는  $(D(L^+, L^-))$ 의 가능해를 찾는데 사용되고,  $c_{ij}$ 는 가능해의 목적함수값을 근사적으로 계산하는데 사용된다.



[그림 6] 그래프  $G$  구성 예

이제 그래프  $G$ 로부터  $(P(L^+, L^-))$ 의 최적목적함

수값의 하한을 구하는 방법을 제시한다.  $G$ 의 점집합  $N$ 을  $(N^+, N^-)$ 로 분할하고  $S^+ = N^+ - s$ ,  $S^- = N^- - t$ 라 두자. 만약  $(N^+, N^-)$ 이 다음 조건 (i), (ii), (iii)를 만족하면  $\sigma = (S^+, S^-)$ 은 유효분할이 된다.

- (i)  $s \in N^+$  이고  $t \in N^-$ ,
- (ii)  $i \in N^+$ , if  $i \in L^+$ ,
- (iii)  $i \in N^-$ , if  $i \in L^-$ .

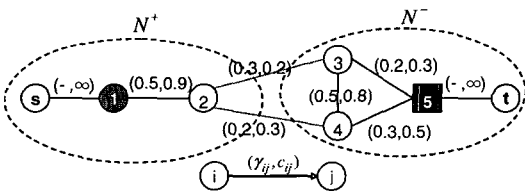
이 때 분할  $(N^+, N^-)$ 로부터 유도되는 그래프  $G$ 의 호집합  $[N^+, N^-] = \{(i, j) | i \in N^+, j \in N^-\}$ 을 유효절단집합(valid cut set)이라고 부르자. 또한 유효절단집합  $[N^+, N^-]$ 에 대응하는 유효분할은 항상  $\sigma = (S^+, S^-) = (N^+ - \{s\}, N^- - \{t\})$ 로 나타내기로 한다.

다음으로 유효절단집합  $[N^+, N^-]$ 과 이에 대응하는 유효분할  $\sigma = (S^+, S^-)$ 가 주어질 때  $(D(\sigma))$ 의 가능해를 구하는 방법을 제시한다. 벡터  $\alpha(\sigma) \in R^{l+u}$ 를 다음과 같이 정의하자. 단, 식 (3.2)에서  $i=1, 2, \dots, l+u$ 이다.

$$\alpha_i(\sigma) = \begin{cases} \sum_{j | (i,j) \in [N^+, N^-]} \gamma_{ij}, & \text{if } \{j | (i,j) \in [N^+, N^-]\} \neq \emptyset \\ 0, & \text{if } \{j | (i,j) \in [N^+, N^-]\} = \emptyset \end{cases} \quad (3.2)$$

$\alpha(\sigma)$ 를 유효절단집합  $[N^+, N^-]$ 의 절단해(cut solution)라 부르기로 하자.

예를 들어, [그림 7]과 같이  $L^+ = \{1\}$ ,  $L^- = \{5\}$ 이고  $S^+ = \{1, 2\}$ ,  $S^- = \{3, 4, 5\}$ 라고 하자. 유효분할  $\sigma = (S^+, S^-)$ 에 대응되는 유효절단집합  $[N^+, N^-]$ 는  $[N^+, N^-] = \{(2, 3), (2, 4)\}$ 이다. 절단해  $\alpha(\sigma)$ 는  $\alpha_1(\sigma) = 0$ ,  $\alpha_2(\sigma) = 0.5$ ,  $\alpha_3(\sigma) = 0.3$ ,  $\alpha_4(\sigma) = 0.2$ ,  $\alpha_5(\sigma) = 0$ 이다.



[그림 7] 절단해 예

식 (3.2)에 의해 정의된 절단해가  $(D(\sigma))$ 의 가능해가 됨을 다음 보조정리를 통해 알 수 있다.

**[보조정리 1]**

부문제  $(D(L^+, L^-))$ 의 임의의 유효분할을  $\sigma = (S^+, S^-)$ 라 하고 이에 대응하는  $G$ 의 유효절단집합을  $[N^+, N^-]$ 라 하자. 그래프  $G$ 에서 식 (3.2)와 같이  $\alpha(\sigma)$ 를 정의하면  $\alpha(\sigma)$ 는  $(D(\sigma))$ 의 가능해이다.

(증명) 첫 번째  $\alpha(\sigma)$ 가  $(D(\sigma))$ 의 등호제약식(equality constraint)을 만족함을 보인다. 식 (3.2)로부터

$$\sum_{i \in S^+} \alpha_i(\sigma) = \sum_{(i,j) \in [N^+, N^-]} \gamma_{ij} = \sum_{i \in S^-} \alpha_i(\sigma)$$

이다. 따라서  $\alpha(\sigma)$ 는  $(D(\sigma))$ 의 등호 제약을 만족한다. 두 번째로  $\alpha(\sigma)$ 가  $(D(\sigma))$ 의 범위제약(range constraint)을 만족함을 보인다.  $G$ 는 정의에 의해 그래프 생성조건 (C7)을 만족하므로 임의의 점  $i$  ( $i=1, \dots, l+u$ )에서  $\sum_{j \in Adj_{\alpha(i)} - \{s, t\}} \gamma_{ij} \leq C_i$ 가 성립하므로  $0 \leq \alpha_i(\sigma) \leq C_i$ 이다. 따라서  $\alpha(\sigma)$ 는  $(D(\sigma))$ 의 가능해이다. □

[보조정리 1]에 의해 부문제  $(D(L^+, L^-))$ 의 유효분할  $\sigma = (S^+, S^-)$ 에 대해 절단해  $\alpha(\sigma)$ 를 구할 수 있고  $\alpha(\sigma)$ 는  $(D(\sigma))$ 의 가능해이다. 그러나  $\alpha(\sigma)$ 의 목적함수값  $f_{D, \sigma}(\alpha(\sigma))$ 를 계산하는 것은 간단치 않다. 왜냐하면  $f_{D, \sigma}(\alpha(\sigma))$ 를 계산하는데 상당한 시간이 요구되고, 더군다나 미지의 레이블이 증가할수록 가능한 유효분할의 개수도 지수적으로 증가하기 때문이다. 따라서  $f_{D, \sigma}(\alpha(\sigma))$ 를 정확하게 계산하는 대신 근사값을 사용하여 하한을 구하는 방법이 보다 효과적일 수 있다. [보조정리 2]는  $G$ 의 절단(cut)이 절단해의 쌍대목적함수값의 하한이 됨을 보인다.  $G$ 의 임의의 절단집합  $[N^+, N^-]$ 에 대해  $[\widehat{N}^+, \widehat{N}^-] \subset [N^+, N^-]$ 을 만족하는 또 다른 절단집합  $[\widehat{N}^+, \widehat{N}^-]$ 이 존재하지 않으면 절단집합  $[N^+, N^-]$ 를 최소절단집합이라 하자. 즉 최소절단집합  $[N^+, N^-]$ 란  $[N^+, N^-]$ 의 어떠한 부분집합도  $G$ 의 절단집합이 되지 않음을 의미한다.

**[보조정리 2]**

부문제  $(D(L^+, L^-))$ 로부터 정의되는  $G$ 의 절단집합

$[N^+, N^-]$ 에 대응하는 유효분할  $\sigma = (S^+, S^-)$ 라 하자. 만약  $[N^+, N^-]$ 가 최소절단집합이면

$$\sum_{(i,j) \in [N^+, N^-]} c_{ij} \leq f_{D,\sigma}(\alpha(\sigma))$$

이다.

**(증명)** 그래프 생성조건 (C5)와 (C9)에 의해 절단 집합  $[N^+, N^-]$ 에는 임시호가 포함되지 않는다. 또  $d_{ij}^2 = \|\phi(x^i) - \phi(x^j)\|^2$ 와 식 (3.2)에 의해 다음 식이 유도된다.

$$\begin{aligned} & \sum_{(i,j) \in [N^+, N^-]} c_{ij} \\ &= \sum_{(i,j) \in [N^+, N^-]} (2\gamma_{ij} - \frac{p}{2}\gamma_{ij}^2 d_{ij}^2) \\ &= \sum_i \sum_{j|(i,j) \in [N^+, N^-]} \gamma_{ij} - \frac{p}{2} \sum_{(i,j) \in [N^+, N^-]} \|\gamma_{ij}\phi(x^i) - \gamma_{ij}\phi(x^j)\|^2 \\ &\leq \sum_i \alpha_i(\sigma) - \frac{1}{2} \left\| \sum_{(i,j) \in [N^+, N^-]} (\gamma_{ij}\phi(x^i) - \gamma_{ij}\phi(x^j)) \right\|^2 \end{aligned}$$

$G$ 는 그래프 생성조건 (C6)에 의해 최대  $p$ 개의 경로를 가지므로, 임의의 최소절단집합에 속하는 호의 개수는 최대  $p$ 개다. 따라서 식 (3.2)와 Cauchy-Schwarz 부등식에 의해 다음 식이 성립한다.

$$\begin{aligned} & \sum_{(i,j) \in [N^+, N^-]} c_{ij} \\ &\leq \sum_i \alpha_i(\sigma) - \frac{1}{2} \left\| \sum_{(i,j) \in [N^+, N^-]} (\gamma_{ij}\phi(x^i) - \gamma_{ij}\phi(x^j)) \right\|^2 \\ &= \sum_i \alpha_i(\sigma) - \frac{1}{2} \left\| \sum_{i \in S^+} \alpha_i(\sigma)\phi(x^i) - \sum_{i \in S^-} \alpha_i(\sigma)\phi(x^i) \right\|^2 \\ &= f_{D,\sigma}(\alpha(\sigma)). \quad \square \end{aligned}$$

[보조정리 2]를 모든 유효절단에 적용함으로써 [정리 3]과 같이 부문제  $(P(L^+, L^-))$ 의 하한을 구할 수 있다.

**[정리 3]**

부문제  $(D(L^+, L^-))$ 로부터 그래프 생성조건 (C1)~(C9)에 의해 정의되는 그래프를  $G$ 라 하자. 부문제

$(D(L^+, L^-))$ 의 임의의 유효분할  $\sigma = (S^+, S^-)$ 에 대응하는 그래프  $G$ 의 절단집합을  $[N^+, N^-]$ 라 하면  $G$ 의 최소절단(min-cut)은 부문제  $(P(L^+, L^-))$ 의 최적목적함수값의 하한이다.

**(증명)** [보조정리 2]로부터 다음 식이 성립한다.

$$\min_{\sigma = (S^+, S^-) \in \Omega} \sum_{(i,j) \in [N^+, N^-]} c_{ij} \leq \min_{\sigma \in \Omega} f_{D,\sigma}(\alpha(\sigma))$$

쌍대정리에 의해  $(D(L^+, L^-))$ 의 최적목적함수값과  $(P(L^+, L^-))$ 의 최적목적함수값은 같으므로  $(D(L^+, L^-))$ 의 최적목적함수값을  $f_D^*$ 라 두면,

$$\min_{\sigma = (S^+, S^-) \in \Omega} \sum_{(i,j) \in [N^+, N^-]} c_{ij} \leq \min_{\sigma \in \Omega} f_{D,\sigma}(\alpha(\sigma)) = f_D^*$$

가 성립한다.  $\square$

**3.2 축소된 SVM를 이용한 한계전략**

다음으로 부문제  $(P(L^+, L^-))$ 에서 레이블이 고정된 훈련데이터만을 이용하여  $(P(L^+, L^-))$ 의 최적목적함수값의 하한을 구할 수 있다. 부문제  $(P(L^+, L^-))$ 에서 레이블이 알려진 데이터로만 다음과 같이  $(SP(L^+, L^-))$ 를 만들 수 있다.  $(SP(L^+, L^-))$ 에서  $L = L^+ \cup L^-$ 이다.

$$\min f_{SP}(w_L, \xi_L) = \frac{1}{2} w_L^T w_L + C \sum_{i \in L} \xi_i$$

$$(SP(L^+, L^-)) : \text{s.t. } y_i(w^T x^i + b) \geq 1 - \xi_i, \forall i \in L$$

$$\xi_i \geq 0, \forall i \in L$$

$(SP(L^+, L^-))$ 를 풀어 [정리 4]와 같이 하한을 구할 수 있다.

**[정리 4]**

$(SP(L^+, L^-))$ 의 최적목적함수값을  $f_{SP}^*$ 라 하면  $f_{SP}^*$ 는 부문제  $(P(L^+, L^-))$ 의 최적목적함수값의 하한이다.

**(증명)**  $U = \{1, \dots, l+u\} - (L^+ \cup L^-)$ 라 두고  $(P(L^+, L^-))$ 의 임의의 가능해를  $(w, b, \xi, y_U)$ 라 하면  $(w, b, \xi, y_U)$ 의 요소 중  $L$ 에 속하는 요소들만으로 만들어지는  $(w_L, b, \xi_L)$ 는  $(SP(L^+, L^-))$ 의 가능해이다. 따라서

$(SP(L^+, L^-))$ 의 최적목적함수값은 항상 부문제  $(P(L^+, L^-))$ 의 최적목적함수값보다 작거나 같다.  $\square$

## 4. 분지전략

분지는 레이블이 미지인 데이터 중에서 그 레이블 값을 각각 '+1' 또는 '-1'로 고정할 데이터를 선정하는 것을 말한다. 본 연구에서는 우선적으로 두 클래스에 경계에 있을 것으로 추정되는 훈련데이터를 분지 데이터로 선정하였다. 왜냐하면 경계가 아닌 클래스 내부에 있는 데이터는 한계전략에 의해 계산되는 하한에 영향을 미치지 못해 불필요한 분지로 이어질 가능성이 높기 때문이다.

두 클래스의 경계에 있는 데이터를 정확하게 찾기는 쉽지 않으므로 본 연구에서는 최단거리를 사용한 휴리스틱 방법을 통해 분지할 데이터를 찾는다. 부문제  $(D(L^+, L^-))$ 의 그래프  $G$ 에서  $s$ 로부터 점  $i$ 까지의 최단거리를  $d^+(i)$ 로 나타내고, 점  $i$ 로부터  $t$ 까지의 최단거리를  $d^-(i)$ 로 나타내자. 다음 식 (4.1)을 만족하는 미지의 레이블  $y_i$ 를 분지할 레이블로 선택한다.

$$i^* = \min_{i \in U} |d^+(i) - d^-(i)| \quad (4.1)$$

단,  $U = \{1, \dots, l+u\} - (L^+ \cup L^-)$ 이다.

식 (4.1)은 그래프  $G$ 에서  $s$ 로부터의 최단거리와  $t$ 로부터의 최단거리간의 차이가 가장 작은 훈련데이터  $x^i$ 를 찾아서  $y_i$ 를 분지할 레이블로 선정함을 의미한다. 이는  $d^+(i)$ 와  $d^-(i)$ 간의 차이가 작을수록  $x^i$ 가 두 클래스의 경계에 있을 가능성이 높다고 보기 때문이다.

## 5. 분지한계 알고리즘

훈련데이터  $(x^1, y_1), \dots, (x^l, y_l), (x^{l+1}, y_{l+1}), \dots, (x^{l+u}, y_{l+u})$ 가 주어지고 이 중 처음  $l$ 개의 훈련데이터  $L_0 = \{1, 2, \dots, l\}$ 의 레이블은 알려져 있고 나머지 데이

터의 레이블은 미지라고 하자.  $L_0$ 에서 레이블이 '+1'인 데이터와 '-1'인 데이터 집합을 각각  $L_0^+, L_0^-$ 라 하자. TSVM를 푸는 분지한계 알고리즘은 다음과 같다.

### 단계 1. 초기화

$$P \leftarrow (P(L_0^+, L_0^-)), f_p^* \leftarrow \infty$$

### 단계 2. 부문제 선정

부문제 목록  $P$ 에서 부문제를 선정한다(선정된 부문제를  $(P(L^+, L^-))$ 라 하자).

### 단계 3. 한계

$(P(L^+, L^-))$ 의 쌍대문제  $(D(L^+, L^-))$ 로부터 그래프 생성조건 (C1)~(C9)에 의해 그래프  $G$ 를 생성하고, 그래프  $G$ 의 최소절단집합  $[N^+, N^-]$ 와 최소절단에 의한 하한  $f_{CUR}$ 을 구한다.

만약  $f_{CUR} \geq f_p^*$ 이면, 부문제  $(P(L^+, L^-))$ 를 제거하고 단계 2로 간다.

만약  $f_{CUR} < f_p^*$ 이면, 축소된 SVM를 이용한 하한  $f_{SP}$ 를 구한다.

만약  $f_{SP} \geq f_p^*$ 이면, 부문제  $(P(L^+, L^-))$ 를 제거하고 단계 2로 간다.

만약  $f_{SP} < f_p^*$ 이면,  $\sigma = (N^+ - \{s\}, N^- - \{t\})$ 로 두고  $(D(\sigma))$ 의 최적목적함수값  $f_{SVM}^*$ 를 구한다.

만약  $f_{SVM}^* < f_p^*$ 이면,  $f_p^* \leftarrow f_{SVM}^*$ 로 두고  $(D(\sigma))$ 의 최적해를 보관한다.

단계 4로 간다.

### 단계 4. 분지

만약  $L^+ \cup L^- = \{1, 2, \dots, l+u\}$ 이면, 단계 2로 간다.

식 (4.1)에 의해 분지할 훈련데이터의 인덱스  $i^*$ 를 선정한다.

부문제를 생성하여  $P = P \cup (P(L^+ \cup \{i^*\}, L^-))$ ,

$(P(L^+, L^- \cup \{i^*\}))$ 로 둔다.

단계 2로 간다.

제시된 분지한계 알고리즘을 구현하기 위해서는 일반적인 분지한계 알고리즘에서와 같이 여러 가지 세부 사항들을 결정해야 하는데, 본 연구에서 사용

한 방법들은 다음과 같다.

### 5.1 부문제 선정

단계 2의 부문제를 선정할 때 계산된 하한이 가장 작은 부문제를 선택하였다. 이는 최적해에 근접한 해를 가질 가능성이 높은 부문제를 우선적으로 풀어  $f_p$ 를 낮춤으로써, 불필요한 부문제 생성을 줄이고 궁극적으로 알고리즘을 빠른 시간 내에 종료할 수 있기 때문이다.

### 5.2 그래프 $G$ 에 포함될 호 선정

그래프  $G$ 의 정의에 따라 호  $(i, j)$ 의 용량  $c_{ij}$ 는 다음 식과 같이 할당된다.

$$c_{ij} = 2\gamma_{ij} - (p/2)d_{ij}^2\gamma_{ij}^2 \quad (5.1)$$

$$= -\frac{p}{2}(d_{ij}\gamma_{ij} - \frac{2}{pd_{ij}})^2 + \frac{2}{pd_{ij}^2}$$

식 (5.1)에서  $d_{ij}$ 는 훈련데이터  $x^i$ 와  $x^j$ 간의 거리로  $d_{ij}$ 가 작을수록  $c_{ij}$ 의 최대값은 증가한다. 따라서 가능한 한  $d_{ij}$ 가 작은 호들로  $G$ 를 구성하는 것이 좋다. 또한 식 (5.1)에서  $c_{ij}$ 는  $p$ 가 증가할수록 감소하게 된다. 그러나  $p$ 를 작게 하면  $G$ 의 최소절단집합  $[N^+, N^-]$ 에 속하는 호의 개수가 작아져 결과적으로 최소절단  $\sum_{(i,j) \in [N^+, N^-]} c_{ij}$ 을 감소시킬 수도 있다. 따라서  $s$ 로부터  $t$ 까지 경로의 개수가 적절하게 설정되어야  $G$ 의 최소절단으로부터 유도되는 하한의 정확도를 높일 수 있다.

그래프  $G$ 는 크게 세 단계를 거쳐 구성하였다.

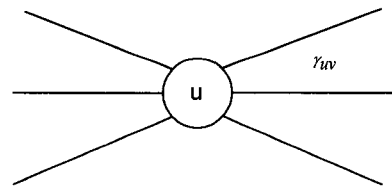
첫 번째 단계에서는 먼저 그래프  $H$ 를 구성하여  $H$ 의 정규호  $(i, j)$ 의 길이는  $d_{ij}$ 로 두고 임시호의 길이는 0으로 둔다.

두 번째 단계에서는  $s$ 로부터  $t$ 까지의 최단경로를 찾아 최단경로에 속하는 호를  $G$ 에 포함시킨다. 이때  $G$ 가 그래프 생성조건 (C6)를 만족하도록 서로 겹치지 않는 최단경로를  $p$ 개 찾아서 최단경로에 포함된 호만을 뽑아 그래프  $G$ 를 구성한다.

세 번째 단계에서는 두 번째 단계에 의해 구성된 그래프에 모든 점이 포함되어 있지 않을 때 수행된다. 즉, 두 번째 단계에서 구한 어느 경로에도 포함되지 않는 점이 있는 경우 그래프  $G$ 가 이러한 점들을 포함할 수 있도록 호를 추가한다. 이는 그래프  $G$ 가 연결성(connectedness)을 만족하도록 호를 추가하는 과정이다. 포함되지 않는 각 점에 대해  $s$ 로부터 최단거리와  $t$ 로부터의 최단거리를 구하여 보다 짧은 경로를 추가한다. 즉 포함되지 않는 점  $i$ 에 대해 점  $s$ 로부터 최단거리가 점  $t$ 로부터의 최단거리보다 작은 경우에는 점  $s$ 로부터 점  $i$ 까지 가는 최단경로가, 반대의 경우는 점  $i$ 로부터 점  $t$ 까지의 경로가 그래프  $G$ 에 포함되도록 호를 추가한다.

### 5.3 그래프 $G$ 에서 $\gamma_{ij}$ 결정

호  $(i, j)$ 의  $\gamma_{ij}$ 을 어떻게 할당하느냐에 따라 호의 용량  $c_{ij}$ 가 달라지고, 궁극적으로 최소절단의 의해 구한 하한의 정확도도 달라진다. 따라서 가능한 한  $c_{ij}$  값이 크게  $\gamma_{ij}$ 를 결정하는 것이 좋다. [그림 8]과 같이 점  $u$ 에 연결된 호의  $\gamma_{ij}$ 를 결정하는 방법을 설명한다.



[그림 8] 호의  $\gamma_{ij}$  결정

점  $u$ 에 연결된 호의 용량합이 최대가 되도록  $\gamma_{ij}$ 를 설정하는 문제를 수리식으로 모형화하면 다음  $(W_u)$ 와 같다.

$$\max_{i \in Adj(u)} (2\gamma_{ui} - \frac{p}{2}d_{ui}^2\gamma_{ui}^2)$$

$$(W_u) : \text{s.t. } \sum_{i \in Adj(u)} \gamma_{ui} \leq C_u,$$

$$\gamma_{ui} \geq 0, \forall i \in Adj(u)$$

$(W_u)$ 의 첫 번째 제약식은  $u$ 에 인접한 호의  $\gamma_{ui}$  합이  $C_u$ 보다 작음을 나타내는데, 이는 식 (3.2)에 의해 유도되는 절단해가 가능해가 되도록 절단해의 각 요소가  $C$ 를 넘지 않게 하는 조건이다.  $C_u$ 는  $C$ 보다 작거나 같은 양(positive)의 값으로서  $C$ 에서 점  $u$ 의 호 중에 이미 설정된 호의  $\gamma_{ui}$ 의 합을 뺀 값이다.  $(W_u)$ 는 목적함수와 제약식이 모두 볼록(convex)인 볼록계획법(convex programming)이다.  $(W_u)$ 의 라그랑지안(Lagrangian) 함수는 식 (5.2)와 같다.

$$L(\gamma, \lambda, \mu) = \sum_{i \in Adj_G(u)} (2\gamma_{ui} - \frac{p}{2} d_{ui}^2 \gamma_{ui}^2) - \lambda (\sum_{i \in Adj_G(u)} \gamma_{ui} - C_u) + \sum_{i \in Adj_G(u)} \gamma_{ui} \mu_i \quad (5.2)$$

$(W_u)$ 의 최적해  $(\gamma^*, \lambda^*, \mu^*)$ 는 다음 네 가지 조건을 모두 만족해야 한다.

$$\frac{\partial L}{\partial \gamma_{ui}} = 2 - p d_{ui}^2 \gamma_{ui}^* - \lambda^* + \mu_i^* = 0, \forall i \in Adj_G(u) \quad (5.3)$$

$$\sum_{i \in Adj_G(u)} \gamma_{ui}^* \leq C, \text{ and } \gamma_{ui}^* \geq 0, \forall i \in Adj_G(u) \quad (5.4)$$

$$\lambda^* (\sum_{i \in Adj_G(u)} \gamma_{ui}^* - C_u) = 0, \forall \lambda \geq 0 \quad (5.5)$$

$$\gamma_{ui} \mu_i^* = 0, \mu_i^* \geq 0, \forall i \in Adj_G(u) \quad (5.6)$$

$\Delta = \sum_{i \in Adj_G(u)} 2/(p d_{ui}^2)$ 라 두면 식 (5.3)~(5.6)을 만족하는  $(W_u)$ 의 최적해는 다음과 같이 구할 수 있다.

$$\text{만약 } \Delta \leq C_u \text{ 이면, } \gamma_{ui}^* = 2/(p d_{ui}^2), \\ \lambda^* = 0, \mu_i^* = 0 \quad (5.7)$$

$$\text{만약 } \Delta > C_u \text{ 이면, } \gamma_{ui}^* = (2 - \lambda^*) / (p d_{ui}^2), \\ \lambda^* = 2(\Delta - C_u) / \Delta, \mu_i^* = 0 \quad (5.8)$$

식 (5.7)과 (5.8)에 의해 항상  $r_{ui} \leq (2/p d_{ui}^2)$ 가 성립하므로 그래프 생성조건 (C8)로부터 유도되는 정규호의 용량은 항상  $c_{ui} \geq 0$ 이다. 한편 모든 점에 대해  $(W_u)$ 를 동시에 풀 수는 없다. 따라서 본 연구에서는 가장 큰 차수(degree)를 갖는 호부터 차례로  $(W_u)$ 를 풀어 각 호의  $\gamma_{ij}$ 를 설정하였다.

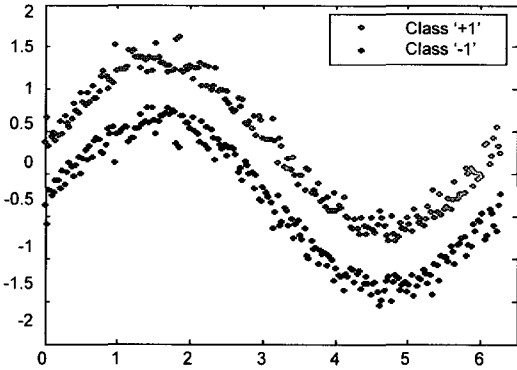
## 6. 실험결과

3~5장에서 제시된 알고리즘을 구현하여 간단한 예제 데이터와 실제 데이터에 적용해 보았다. 제안된 알고리즘은 C로 구현하였으며, 그래프  $G$ 를 구성하기 위해 최단경로 프로그램으로 NETDIK<sup>1)</sup>을 사용하였고, 최소절단을 구하는 프로그램으로는 PREFLO<sup>1)</sup>를 사용하였다. 또한, 축소된 SVM 문제의 최적해를 구하는 프로그램으로는 LIBSVM[4]을 사용하였다.

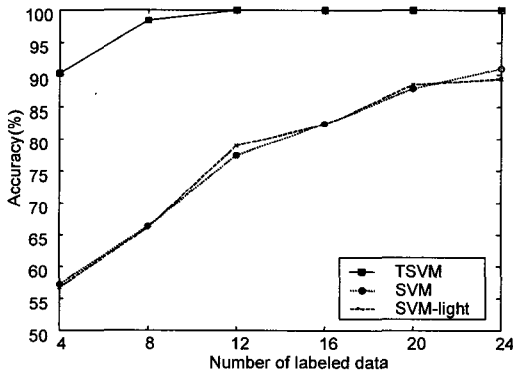
먼저 간단한 예제 데이터를 사용하여 제안된 알고리즘과 기존의 방법들을 비교하였다. [그림 9]와 같이  $y = \sin(x)$  곡선을 따라 두 개의 클래스를 생성하였다. 클래스 '+1'과 '-1'에 속하는 데이터를 각각 201개씩 총 402개의 데이터를 생성하였다. 비교 대상으로 SVM 방법과 SVM-light[8]를 사용하였다. SVM-light는 극부탐색(local search)을 통해 TSVM의 근사최적해를 찾는 프로그램이다. [그림 10]에서 'SVM'은 레이블이 알려진 데이터만 사용하여 SVM으로 학습한 후 미지의 레이블을 예측한 결과이다. 전체 402개 데이터 중 각각 4, 8, 12, 16, 20, 24개의 데이터만 레이블이 알려진 것으로 설정하여 SVM으로 학습한 후 나머지 데이터의 레이블을 예측하였다. 이때 각 설정마다 레이블을 줄 데이터를 무작위로 30번씩 추출하여 반복 실험한 후 정확도의 평균값을 구했다. 세 가지 방법 모두 RBF 커널함수를 사용하였고, RBF 커널의  $\rho^2$  값은 1로 두었으며,  $(P)$ 의  $C$  값은 100,000으로 설정하였다. 또 그래프  $G$ 를 생성할 때  $K=20$ ,  $p=20$ 으로 설정하였다. [그림 10]에서 볼 수 있듯이 SVM과 SVM-light의 정확도는 거의 동일한 것으로 나타났다. 이는 근사해를 사용하는 SVM-light는 레이블이 알려진 데이터가 많아지더라도 학습의 정확도가 크게 개선되지 않음을 의미한다. 반면에 본 연구에서 제안된 알고리즘으로 구현된 TSVM은 소수 데이터의 레이블만 알고 있더라도 나머지 미지의 레이블을

1) <http://www.orlab.org>

거의 정확하게 예측함을 볼 수 있다.



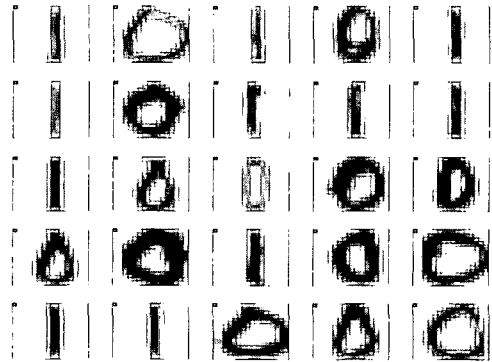
[그림 9] 예제 데이터



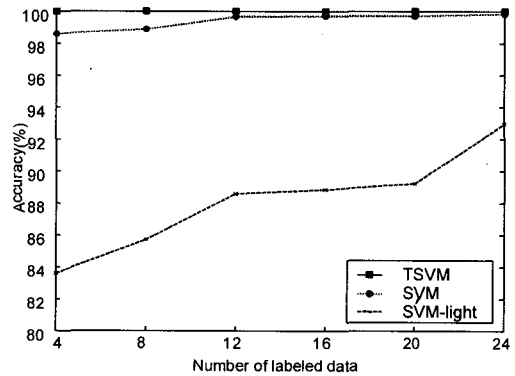
[그림 10] 예제 데이터 적용 결과 비교

다음으로 기계학습 분야의 실험에서 널리 사용되는 USPS 데이터에 적용해 보았다. USPS 데이터는 여러 사람들이 쓴 0에서 9까지의 숫자를 비트맵 이미지로 표현한 데이터로 총 7,291개로 구성되어 있다. 각 데이터는 16×16 비트맵 이미지를 나타내는 256개의 속성으로 구성된다. 전체 데이터에서 숫자 0과 1을 나타내는 데이터를 뽑아 총 2,199개 데이터로 실험하였다. 실험 대상이 되는 데이터의 일부를 그림으로 나타내면 [그림 11]과 같다. TSMV, SVM, SVM-light의 정확도를 비교한 결과는 [그림 12]과 같다. [그림 12]은 4, 8, 12, 16, 20, 24개 데이터의 레이블을 알려 주고 나머지 데이터의 레이블은 미지로 둔 경우에 정확도 변화를 보여준다. [그림 12]의

정확도는 동일한 설정에 대해 15번 무작위로 실험한 결과를 평균한 값이다. 세 가지 방법 모두 RBF 커널을 사용하였고, RBF 커널의  $\beta$  값은 100으로 두었으며,  $(P)$ 의  $C$  값은 100,000으로 설정하였다. 또 그래프  $G$ 를 생성할 때  $K=20$ ,  $p=20$ 으로 설정하였다. TSMV는 4개의 레이블만 주어진 경우에도 100% 정확도를 보여주고 있다. SVM 방법도 높은 정확도를 보여주고 있는데 이는 USPS 데이터가 SVM의 결정함수에 의해 분할될(separable) 수 있는 데이터이기 때문으로 분석된다. 반면 극부탐색에 의해 TSMV의 근사해를 찾는 SVM-light의 정확도는 매우 낮음을 알 수 있다.



[그림 11] USPS '0'과 '1' 데이터



[그림 12] USPS 데이터 실험결과(정확도)

마지막으로 본 연구에서 제안한 두 가지 한계전략의 유용성을 분석해 본다. USPS 데이터에 적용한 결과를 <표 1>에 정리하였다. <표 1>에서 첫

〈표 1〉 USPS 데이터 실험결과(계산시간)

#Data	#Labeled	#Subproblem	Time	#FathomedByCut	#FathomedByRSVM	#SVM
2199	4	17,167	3,592.6	6,890(40.1%)	1,378(8.0%)	153
	8	12,193	2,551.4	4,661(38.2%)	1,140(9.3%)	146
	12	9,550	2,017.6	3,466(36.3%)	1,376(14.4%)	117
	16	10,626	2,205.5	4,003(37.7%)	1,088(10.2%)	110
	20	9,881	2,062.2	3,620(36.6%)	1,082(11.0%)	118
	24	9,726	2,044.3	3,542(36.4%)	1,060(10.9%)	129

번째 열은 훈련데이터의 개수를 말하고 두 번째 열은 레이블이 고정된 데이터의 개수를 말한다. 동일한 훈련데이터에서 레이블을 고정할 데이터와 그 개수만 변경하여 실험하였다. 첫 번째 열에서 두 번째 열을 뺀 값이 레이블이 미지인 데이터의 개수가 되고 동시에 정수 제약을 갖는 변수의 개수이기도 하다. 세 번째 열은 4장에 제시된 분지한계알고리즘에 의해 생성된 부문제 개수이고 네 번째 열은 계산시간을 초단위로 표시한 것이다. 다섯 번째와 여섯 번째 열은 최소절단과 축소된 SVM의 하한에 의해 제거된 부문제 개수를 각각 나타낸다. 마지막 열은 분지한계알고리즘에서 축소되지 않은 SVM 문제를 푼 회수를 말한다. <표 1>의 각 행에 나타난 모든 수치는 레이블을 고정할 데이터를 무작위로 추출하여 실험하는 과정을 15회 반복하여 나온 수치를 평균한 값이다. 정수변수의 개수가 2,172~2,195개임에도 분지한계알고리즘에 의해 생성된 부문제 개수는 20,000개 미만으로 비교적 작았다. 이는 부문제 중 상당수가 최소절단 또는 축소된 SVM으로부터 구한 하한에 의해 제거되었기 때문으로 볼 수 있다. 이는 최소절단과 축소된 SVM에 의한 하한이 상당히 효과적임을 보여준다. 또한  $f_{SVM}$ 를 구하기 위해 모든 훈련데이터를 포함하는 SVM를 푼 회수도 전체 부문제의 약 1% 정도로 매우 작아 제시된 분지한계알고리즘이 효율적임을 확인할 수 있었다.

## 7. 결론 및 추후 과제

본 연구에서는 반지도학습 방법으로 Vapnik에

의해 제안되었던 Transductive SVM을 효과적으로 풀 수 있는 분지한계 알고리즘을 제안하였다. 본 연구는 TSVM 문제의 특성을 이용하는 한계전략(bounding strategy)을 제시함으로써 기존의 범용 혼합정수계획법 프로그램으로 해결할 수 없었던 대형문제를 다룰 수 있음을 보였다. 또한 제시한 분지한계 알고리즘으로 TSVM을 푼 결과와 근사적인 TSVM 해법으로 푼 결과를 비교해보면 정확도면에서 큰 차이가 나고 있음을 확인할 수 있었다.

본 연구에서 제시한 방법이 실제 문제에 널리 사용되기 위해서는 알고리즘을 보다 효율적으로 개선할 필요가 있다. 첫 번째로 보다 정교한 한계전략을 고안할 필요가 있다. 이를 위해 제시된 최소절단에 의한 하한을 개선할 수 있는 방법에 관한 연구가 필요하며, 선형계획법 등 대형문제에 적합한 모형으로 완화하는 한계전략도 고려할 수 있을 것이다. 두 번째로 효과적인 구현이 될 수 있도록 여러 가지 파라미터나 구현방법에 대한 실험이 이루어질 필요가 있다. 예를 들어, 효과적인 분지방법, 부문제의 선정,  $G$ 의 호의 용량 설정 방법 등 실제 구현시 문제가 되는 사항들에 대해 광범위한 실험과 분석이 요구된다.

## 참고 문헌

- [1] Bennett K.P. and A. Demiriz, "Semi-supervised support vector machines," *Advances in Neural Information Processing Systems*, Vol.10(1998), MIT Press.
- [2] Bie, T. De and N. Christianini, "Convex Me-



- thods for Transduction,” *Advances in Neural Information Processing Systems*, Vol. 16(2004), MIT Press.
- [3] Blum, A. and S. Chawla, “Learning from Labeled and Unlabeled Data using Graph Mincuts,” *Proceedings of the 18th International Conference on Machine Learning (ICML)*, 2001.
- [4] Chang, C.-C. and C.-J. Lin, *LIBSVM : A Library for Support Vector Machines*, <http://www.csie.ntu.edu.tw/~cjlin/libsvm>, 2001.
- [5] Chapelle, O., J. Weston, and B. Schölkopf, “Cluster Kernels for Semi-supervised Learning,” *Advances in Neural Information Processing Systems*, Vol.15(2003), MIT Press.
- [6] Cristianini, J. Shawe-Taylor, *An Introduction to Support Vector Machines and other Kernel-based Learning Methods*, Cambridge University Press, 2000.
- [7] Joachims, T., “Transductive Learning via Spectral Graph Partitioning,” *Proceedings of the 20th International Conference on Machine Learning(ICML)*, 2003.
- [8] Joachims, T., “Transductive Inference for Text Classification using Support Vector Machines,” *Proceedings of the 20th International Conference on Machine Learning (ICML)*, 1999.
- [9] Nilsson, N.J., *Introduction to Machine Learning*, <http://robotics.stanford.edu/people/nilsson/mlbook.html>
- [10] Platt, J., “Fast Training of Support Vector Machines using Sequential Minimal Optimization,” In B. Schölkopf, C.J.C. Burges, and A.J. Smola, editors, *Advances in Kernel Methods-Support Vector Learning*, pp. 185-208.
- [11] Schölkopf, B. and A.J. Smola, *Learning with Kernels : Support Vector Machines, Regularization, Optimization, and Beyond*, MIT Press, 2002.
- [12] Seeger, M., *Learning with Labeled and Unlabeled Data*, Technical Report, Institute for Adaptive and Neural Computation, University of Edinburgh, 2001.
- [13] Shi, J. and J. Malik, “Normalized Cuts and Image Segmentation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.22(2000), pp.888-905.
- [14] Szummer, M. and T. Jaakkola, “Partially Labeled Classification with Markov Random Walks,” *Advances in Neural Information Processing Systems*, Vol.14(2002), MIT Press.
- [15] Vapnik, V., *Statistical Learning Theory*, Wiley, 1998.
- [16] Yu, S.X. and J. Shi, “Grouping with Bias,” *Advances in Neural Information Processing Systems*, Vol.14(2001), MIT Press.
- [17] Zhu, X., Z. Ghahramani, and J. Lafferty, “Semi-supervised Learning using Gaussian Fields and Harmonic Functions,” *Proceedings of the 20th International Conference on Machine Learning(ICML)*, 2003.