공학석사학위논문

신경망 기반의 자연 언어 문서 검색

Natural Language Text Retrieval Based on Neural Networks

1999 년 2 월

서울대학교 대학원

컴퓨터공학과

오 장 민

요약

지구촌이 온라인으로 연결된 현대는 정보의 홍수 시대이다. 온라인상으로 쏟아지는 수많은 정보를 사용자의 관심사별로 수작업으로 분류하는 것은 매우 지루하고 힘든 일이다. 기계 학습은 이러한 문서 분류 작업을 자동화하는데 효과적인 방법이다.

문서 분류 문제는 매우 고차원의 문제 공간을 가지며, 문서 자체가 가지는 정보가 매우 빈약하게 산재되어 있다. 본 논문에서는 이런 특징의 문제 해결에 적합한 알고리즘인 SVM(Support Vector Machine)을 구현했으며, SVM 이 발견한 SV(Support Vector)들이 학습에 어떤 영향을 미치는지를 실험하기 위해서 SV 만을 가지고 다층 퍼셉트론을 학습했다. SVM 은 통계적 학습 이론에 기반해서 SRM(Structural Risk Minimization) 원리를 충실히 구현한 알고리즘이다. 또한 SVM 의 최적해를 찾는 과정은 Convex Programming 으로서 수학적으로 잘 모델링 된다.

말뭉치 Reuters-21578 문서 집합에 대한 실험에서 자주 등장하는 5 개의 범주에 대해 SVM 의 정확도는 97% 이상의 상당한 수준의 성능을 보였다. 또한 채산점(break-even point) 비교에서 SVM 의 학습 결과는 Naïve Bayesian 분류기보다 더 우수한 성능을 보였다. 또한 SV 만을 가지고 다층 퍼셉트론을 학습시킴으로써 학습 시간을 많이 줄이면서 우수한 성능을 가진 신경망을 구축할 수 있었다.

주제어: 문서 분류, 신경망, Support Vector Machine, Structural Risk Minimization, Convex Programming

차례

표 차례

그림 차례

1 장
서론

1.1 연구 배경

문서는 인간이 갖고 있는 지식을 표현하기 위한 기본적인 방법이다. 그 동안 문서의 목록을 관리하고 탐색하는 일은 도서관 직원이외의 사람에게는 관심거리 밖의 일이었다. 그러나 1950 년대 이후 컴퓨터의 발달은 커다란 변화를 가져왔다. 지구상의 네트워크로 수많은 정보들이 옛날에는 상상도 못할 정도의 양으로 밀려 들어 오고 있다. 월드 와이드 웹(WWW)이 보편화됨에 따라 정보의 온라인화는 더욱 가속화되어 문서 데이터를 조직하고 관리하는데 효과적인 문서 분류의 필요가 더욱 절실해 졌다.

정보의 양이 그리 많지 않은 시절에는 관심 있는 신문 기사는 개인이 직접 분류를 했다. 그러나 온라인으로 입수되는 수많은 정보를 개인이 분류하는 것은 엄청난 시간과 노력을 요구하게 되었다. 따라서 사람의 수작업보다는 자동으로 문서를 분류해주는 시스템을 개발하는 것이 바람직하다.

문서 분류란 어떤 문서를 미리 정의된 범주들로 구분 지어 주는 것이다 [Yang 97]. 각 문서는 하나의 범주에 속할 수 있고, 여러 개의 범주에 속할 수도, 또는 어떤 범주에도 속하지 않을 수도 있다.

본 논문에서는 문서 분류에 신경망을 적용하는 방법에 대해서 살펴보고 최근에 부각되는 SVM(Support Vector Machine)을 사용한 문서 분류 시스템을 구현하고 성능을 평가한다. 특히 SVM 에서 발견된 SV(Support Vector)를 가지고 다

층 신경망을 구축하여 기존의 문서 분류 방법과의 성능을 비교한다.

1.2 관련 연구

일반적인 분류 알고리즘 중 많이 사용되는 것들로 다음과 같은 것들이 있다.

가. **Naïve Bayesian** 분류

이 알고리즘은 확률 기반의 모델로서 자질 공간을 이루는 단어들이 서로 독립이라는 가정이 선험적인 지식으로 필요하다. 문서 Doc 에 대해서 가장 그럴 듯한 분류 결과인 클래스 값 v_{NB} 를 돌려주는 Naïve Bayesian 분류기는

$$v_{NB} = \underset{v_j \in V}{\arg\max}\, P(v_j) \prod_{i \in positions} P(a_i \mid v_j)$$

의 형태를 이룬다. 여기서 v_j 는 가능한 클래스값, a_i 는 Doc 내에서 발견된 단어를 뜻한다[Mitchell 97].

나. 결정 트리 분류

Quinlan 의 C4.5 가 가장 널리 쓰이는 결정 트리 알고리즘이다[Quinlan 93].

다. ***k*-NN(Nearest Neighbor)** 분류

예제 기반의 알고리즘으로서 주어진 문서에 대해 학습된 문서 중 가장 비슷한 순서로 순위를 매긴 후, 상위 k 개의 이웃들을 이용해 클래스를 예측한다.

1.3 논문의 구성

1 장에서는 본 논문의 연구 배경과 관련 연구에 대해서 설명하였다. 2 장에서는 문서 검색의 특징과 SVM 의 특징에 대해서 설명한다. 3 장에서는 신경망 기반의 문서 검색에 대해서 설명한다. 4 장에서는 SVM 알고리즘에 대해서 설명한다. 5 장에서는 실험과 결과에 대해서 설명하고, 6 장에서 본 연구 내용을 요약하고 결론을 기술한다.

2 장
문서 검색

2.1 문서 분류

정보 검색 연구의 대부분은 문서 검색에 집중되었다. 문서 검색 시스템은 사용자의 질의나 관심 목록에 대한 응답으로 문서를 발견해주는 시스템이다. 문서 분류는 문서 검색의 하위 개념으로서 문서 검색이 사용자의 질의에 응답하는 형식임에 비해 문서 분류는 미리 정의된 범주로 문서를 분류하는 것이다. 문서 doc_i 는 0 개 이상의 범주값을 가진다. 문서 분류 시스템을 만드는 것은 주어진

$$\{doc_i, \text{category-value}_i\}$$

쌍들의 집합을 가지고 효과적인 분류 시스템을 만드는 것이다. 기계학습의 관점에서 보면 이 학습 문제는 각 데이터에 대해서 이에 대응하는 분류값이 제공이 되므로 감독 학습(supervised learning) 문제이다.

기계 학습에서 데이터의 표현 방법은 매우 중요하다. 데이터의 표현 방법은

학습의 효율성, 에러율 등에 영향을 많이 끼친다. 문서 검색에서는 문서 표현을 주로 단어들의 벡터로서 표현한다. 벡터의 각 원소값은 해당 문서 내에서 그 단어의 빈도수를 나타낸다. 하나의 단어가 벡터의 한 원소를 나타내므로 입력 공간의 차원은 총 단어 수가 된다. 이 때 차원을 줄이는 즉, 총 단어 수를 줄이는 방법으로 스테밍(stemming) 알고리즘과 불용어 목록(stop list)이 사용된다.

스테밍 알고리즘이란 각 단어마다 활용형이 많은데 이들 활용형을 별개의 단어로 보지 않고 하나의 어근(stem)으로 보는 것이다. 예를 들어 “engineering”, “engineered”, “engineer”를 하나의 어근 “engineer”로 관리하는 것이다. 본 논문에서는 Porter 의 스테밍 알고리즘을 사용했다. Porter 의 스테밍 알고리즘은 조건/행위 집합으로 구성되어 있다. 조건들은 어간에 관한 조건, 접미사에 관한 조건, 규칙에 관한 조건등 세가지로 구분이 된다. 이 조건들을 만족할 때 해당 행위를 해서 단어를 어근 형태의 단어로 변환시켜 준다.

불용어 목록이란 쓸모 없는 단어의 목록을 미리 가지고 있어서 목록에 해당하는 단어는 전체 단어 사전에서 제외시키는 것이다. 예를 들어 “the”, “of”, “and”, “to” 등은 문서 전체에서 가장 자주 사용되는 단어 중에 하나일 것이다. 그러나 이런 단어들은 색인 용어로 가치가 없다고 인식되고 있다. 일반적으로 전형적인 영문 문서에서 10 번 이상 발생하는 단어가 전체 문서의 20~30%를 차지한다[Frakes et al. 92]. 따라서 단어 사전에 불용어 목록에 해당하는 단어를 제외시킬 경우 사전의 크기는 줄이면서 효율에는 영향을 거의 미치지 않게 된다.

따라서 스테밍 알고리즘과 불용어 목록을 활용하여 단어 사전을 구축하게 되고, 문서는 이 단어 사전을 가지고 벡터로 표현되게 된다. 그러면 벡터 표현은 자연히 <u>속성:값</u>의 형태를 가지게 되고, 값을 표현하기 위해 TF(Term Frequency)의 개념이 도입된다. 어떤 문서 doc_i 에서 단어 w_j 의 빈도수 TF 는 다

음과 같이 표현된다.

$$TF(w_i, \mathrm{doc}_i) = \text{count of } w_i \text{ occuring in document } \mathrm{doc}_i$$

그런데 문서 검색에서는 값으로 TF 만 사용하는 것 보다는 IDF(Inverse Document Frequency)라는 개념을 같이 사용하는 것이 더 효율을 높인다고 알려져 있다[Frakes et al. 92]. IDF 는 다음과 같이 표현된다.

$$IDF(w_i) = \log\left(\frac{n}{DF(w_i)}\right)$$

여기서 n 은 총 문서의 수를 나타내고, DF(Document Frequency)는 또한 다음과 같다.

$$DF(w_i) = \text{number of document where } w_i \text{ is occurring}$$

문서 검색에서는 *tfidf* 라는 용어를 사용하는데 이는 TF×IDF 의 값을 벡터 값으로 사용한다는 뜻이다. 만일 어떤 단어가 전체 문서 집합에 골고루 분포하면서 자주 나오는 단어인 경우 그 단어가 어떤 문서를 대표할 확률은 그만큼 적다고 말할 수 있다. TF 만을 사용하면 이런 특징을 제대로 표현할 수 없다. IDF 값을 보면 만일 전체 문서집합에 자주 나타나는 단어일 경우 DF 가 큰 값이 되는데 그러면 IDF 의 값은 0 에 가까워지게 된다. 따라서 *tfidf* 를 사용하는 것이 더 바람직하다.

2.2 기계 학습 관점에서 본 문서 분류의 특징

학습의 관점에서 문서 분류는 다음과 같은 특징을 지닌다.

입력 데이터 공간의 차원이 매우 높다. 스테밍과 불용어 목록을 사용치 않을 경우 사전의 크기가 20,000 이상이 되기 쉽고, 곧 입력 데이터 공간의 차원이 20,000 이상이 될 수 있다는 것을 의미한다.

문서 벡터가 빈약한 벡터(sparse vector)가 되기 쉽다. 하나의 문서는 사전에 등록된 단어들 중 일부만을 가지고 표현되는 것이 보통이다. 이 경우 문서 벡터의 원소 대부분이 0 이 된다.

대부분의 문서 분류 문제는 선형적으로 분리 가능한 문제이다.

3 장
신경망 기반의 문서 검색

3.1 신경망 분류 시스템

3.1.1 퍼셉트론

퍼셉트론은 감독자 학습의 최초의 모델로서 Rosenblatt(1958)에 의해서 제안되었다. 퍼셉트론은 가장 단순한 형태의 신경망으로서 선형 분리 가능한 분류 문제 해결에 주로 사용된다. 기본적으로 하나의 뉴론(neuron)으로 이루어지며 이 뉴론은 조절 가능한 접합 무게 벡터(synaptic weights vector)와 하나의 바이어스(bias)로 구성된다. 퍼셉트론의 모형을 그림 3-1 에 보였다.

퍼셉트론의 층입력 v 는 다음과 같이 계산된다.

$$v = \sum_{i=1}^{m} w_i x_i + b$$

여기서 w_i 는 가중치 벡터 **w** 의 원소, m 은 입력 공간의 차원, x_i 는 입력 벡터 **x** 의 원소, b 는 바이어스 이다.

이 값을 기반으로 출력은 다음과 같이 계산된다.

$$y = j\,(v) = \begin{cases} 1, & v \geq 0 \\ -1, & v < 0 \end{cases}$$

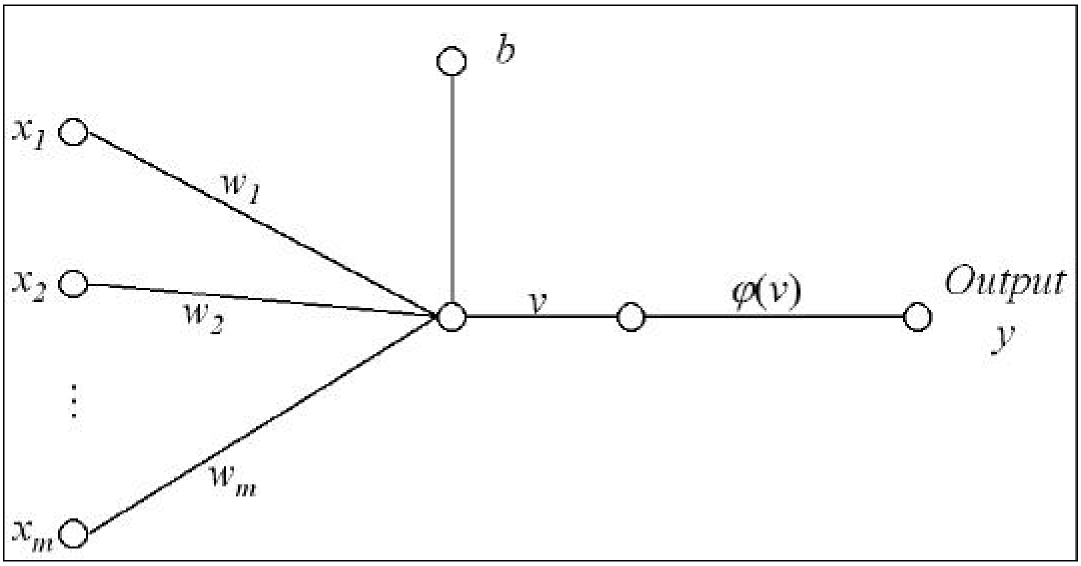

그림 3- 1 퍼셉트론의 모형

즉 퍼셉트론은 두개의 클래스를 구분 짓는 최적의 초평면[1](hyperplane)의 식 $\sum_{i=1}^{m} w_i x_i + b = 0$ 을 찾는 것이 학습의 목표이다. 무게 벡터 $\mathbf{w} = (w_1, \ldots, w_n)$ 가 초평면 식의 파라미터이므로 학습은 에러를 최소화하는 $\mathbf{w}$ 를 찾는 것이 된다.

[1] 초평면은 4.2 절에 설명되어 있다.

3.1.2 다층 퍼셉트론

퍼셉트론의 한계는 선형 분리 가능한 문제만 해결할 수 있다는 데 있다. 선형 분리 가능하지 않은 문제를 해결하기 위해 다층 퍼셉트론이 고안되었다. 다층 퍼셉트론의 모형을 그림 3-2 에 보였다. 다층 퍼셉트론이란 입력층(input layer), 출력층(output layer)외에 은닉층(hidden layer)이 하나 이상 존재하는 망(network)을 뜻한다.

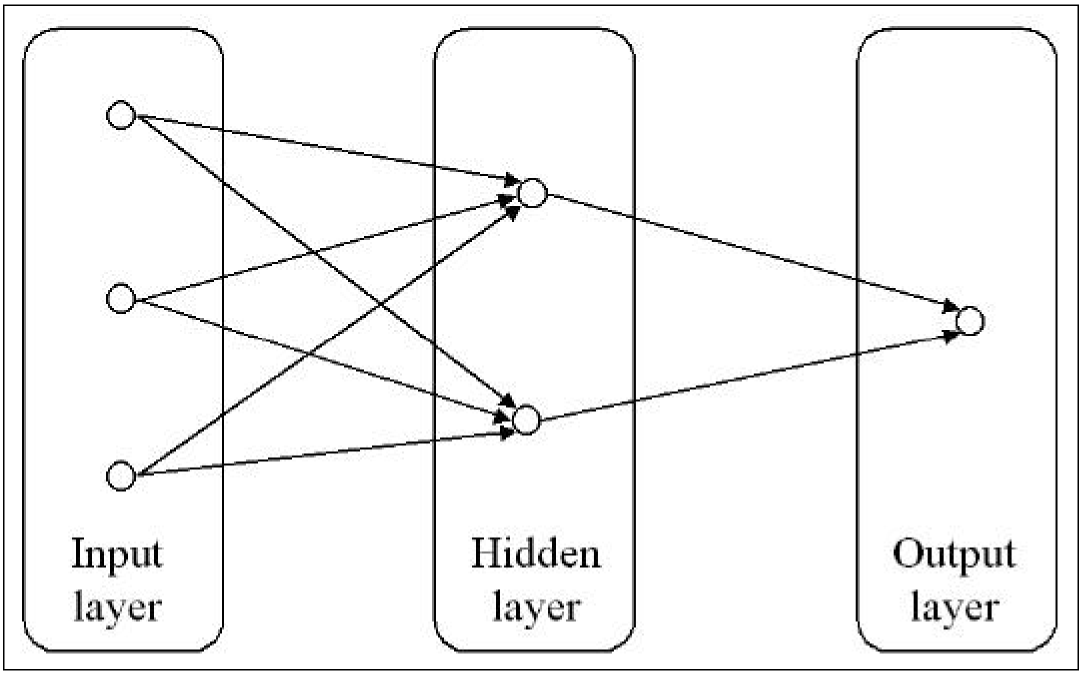

그림 3- 2 다층 신경망 모형

다층 퍼셉트론의 계산론적 특징은 다음의 것들이 있다.

1. 비선형 활성 함수(nonlinear activation function)

퍼셉트론과 달리 다층 퍼셉트론에서는 입력층을 제외한 출력층, 은닉층의 출력 함수는 비선형 함수이면서 모든 구간에서 미분 가능한 함수를 사용한다. 이중 널리 사용되는 두가지 함수는 다음과 같다.

Logistic function: $j(v) = \frac{1}{1 + \exp(-av)}$

Hyperbolic tangent function: $j(v) = a \tanh(bv)$

2. 에러 역전파 알고리즘(backpropagation algorithm)

다층 퍼셉트론의 학습 과정은 2 단계로 나뉜다. 1 단계에서는 입력층에서 출력층으로 각출력을 전달한다. 2 단계에서는 출력층으로부터 입력층으로 에러를 전달한다. 각 층에서는 전달된 에러로부터 자신의 무게 벡터를 교정한다.

3.1.3 RBF(Radial Basis Function) 망

RBF 망은 주로 하나의 은닉층으로 이루어져 있으며 출력층은 은닉층의 합으로 이루어져 있다. RBF 망의 구조를 그림 3-3 에 보였다.

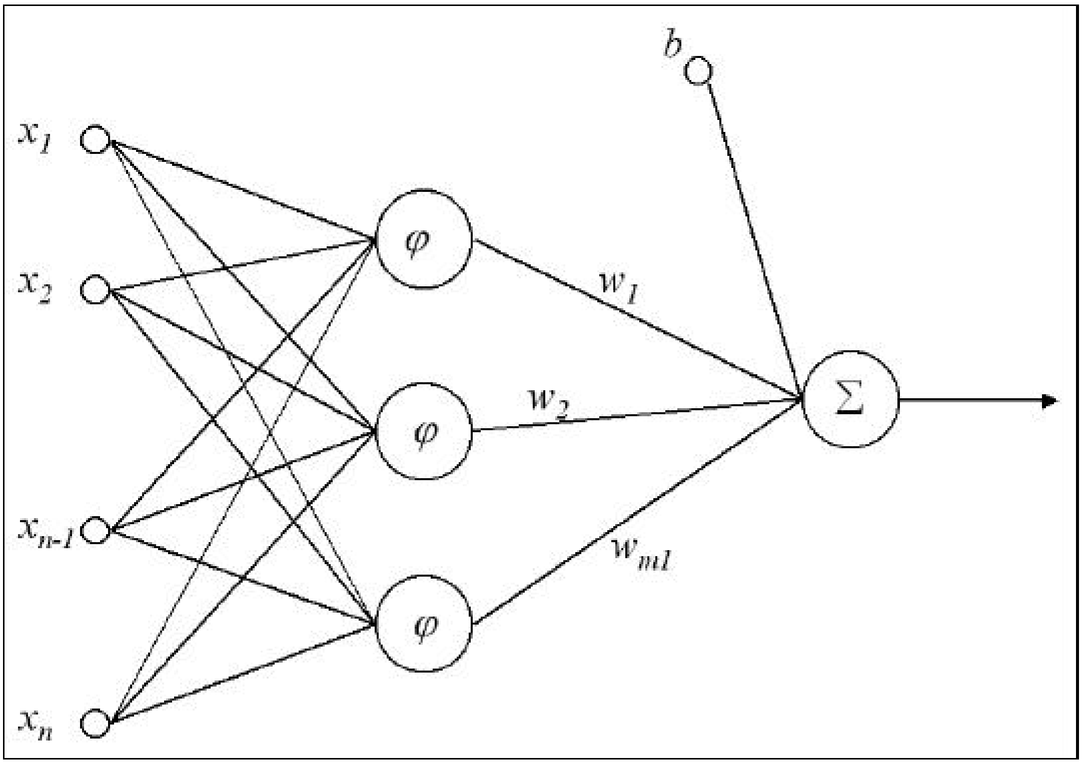

그림 3- 3 RBF 망 모형

$j(\mathbf{x})$함수를 radial-basis function 이라고 부른다. $j(\mathbf{x})$함수의 한 예인 가우스 함수를 보였다.

$$j(\mathbf{x}) = \exp\left(-\frac{\|\mathbf{x}-\mathbf{t}_i\|^2}{2s^2}\right)$$

여기서 $\mathbf{x}$는 입력 벡터, $\mathbf{t}_i$는 radial basis function의 중심점, s는 radial-basis function의 폭이다.

Radial-basis 라는 말은 위의 가우스 함수처럼 한 점($\mathbf{t}_i$)를 중심으로 방사형의 형태를 이루는 함수라는 뜻이다. 출력함수는 다음과 같다.

$$F(\mathbf{x}) = \sum_{i=1}^{m_1} w_i \varphi_i(\mathbf{x}) + b$$

3.2 Support Vector Machines (SVMs)

다층 퍼셉트론이나 RBF 망의 경우의 제약점은 에러를 최소로하는 이들 모델의 파리미터들이 전역해 (global minimum)가 아니라 국부적 해 (local minimum)만을 찾을 가능성이 크다는 것이다. 또한 4 장에서 살펴볼 통계적 학습 이론에서는 테스트 에러를 줄이기 위해서는 학습 데이터의 에러 뿐만 아니라 모델의 복잡도까지 고려해야만 한다고 주장한다. 따라서 다층 퍼셉트론의 경우 은닉 뉴론의 개수, RBF 망의 경우 radial-basis 함수의 개수를 정하는 것이 중요한 이슈가 된다. 분석적인 방법으로 최적의 모델을 구성하는 이들의 개수를 정하는 것은 시행 착오(trial-error)의 방법으로 구하는 것이 보통이다.

SVM 의 큰 특징을 세가지로 요약할 수 있다.

1. SV (support vector)만으로 이루어진 모델을 구성한다.

실제 학습에서 중요한 데이터와 그렇지 않은 데이터로 학습 데이터를 구분할 수 있다. 신경망 학습의 경우, 학습 에러가 가장 큰 순서로 학습 데이터를 배치할 수 있다면 학습의 속도를 크게 향상시킬 수 있다[haykin 98]. SVM 에서는 학습 데이터 중 SV 를 찾아내는데, 이 SV 만을 가지고 학습을 시켜도 그 성능은 같게된다.

2. 입력 데이터 공간보다 높은 자질 공간으로의 매핑

입력 데이터 공간의 차원이 큰 문제의 경우 학습이 어려워진다. 이 경우 문제를 해결하는 한가지 방법은 입력 데이터 공간을 차원이 낮은 공간으로 매핑시켜 학습하는 것이다. 이 방법의 알고리즘으로 PCA (Principal Component Analysis), SOM (Self Organized Map), VQ (Vector Quantizer)등이 있다. SVM 의 관점은 만일 입력 데이터 공간을 낮은 차원으로 매핑시키는 경우 중요한 정보의 손실이 있다는 것이다. SVM 의 해결책은 오히려 입력 데이터 공간을 더 높은 차원으로 매핑시키는 것이다.

3. 통계적 이론에 충실하다

SVM 은 4 장에서 살펴볼 구조적 에러 최소화 원리(SRM:Structural Risk Minimizaton)에 충실하다. SRM 의 아이디어는 모델의 복잡도를 증가시켜 가면서 내포 구조의 모델을 만드는데 이 모델들 중 에러의 상한값을 최저로 하는 모델을 고르는 것이다. 대부분의 학습 알고리즘의 경우 모델의 복잡도를 제어하는 것이 매우 어렵다. 하지만 SVM 에서는 모델의 복잡도를 제어하는 것이 가능하다.

다음 장에서는 SVM 알고리즘에 대해서 고찰해본다.

4 장

SVM 을 이용한 문서 분류

4.1 통계적 학습 이론

4.1.1 경험적 에러 최소화 원리

문서 분류 문제에 대한 학습 문제는 이진 패턴 분류 문제로 일반화시킬 수 있다. 예제들로부터의 학습은 다음과 같이 정리할 수 있다.

예제 집합이

$$(\mathbf{x}_1, d_1), \ldots, (\mathbf{x}_l, d_l), \qquad \mathbf{x}_i \in R^N,\ d_i \in \{-1,\ 1\}$$

처럼 주어졌을 때, 가능한 분류 함수 $F(\mathbf{x}, \mathbf{w})$ 의 집합

$$\{F(\mathbf{x}, \mathbf{w}) :\ \mathbf{w} \in \mathrm{W}\}, \qquad F(\mathbf{x}, \mathbf{w}) : R^N \to \{-1, 1\}$$

중에서 다음 식의 값을 최저값으로 만드는 $F(\mathbf{x},\mathbf{w}^*)$를 찾는다.

$$R(\mathbf{w}) = \int \left| d - F(\mathbf{x},\mathbf{w}) \right| dF_{\mathbf{x},D}(\mathbf{x},d)$$

여기서 w 는 자유 매개 변수(free parameter)의 집합을 나타내고, $F_{\mathbf{x},D}(\mathbf{x},d)$ 는 x 와 d 의 결합 확률(joint probability)을 의미한다. 이 때, $R(\mathbf{w})$를 리스크 함수(risk functional) 또는 기대 리스크(expected risk)라고 부른다.

그러나 일반적으로 결합 확률 분포 $F_{\mathbf{x},D}(\mathbf{x},d)$ 는 미지의 분포이기 때문에 $R(\mathbf{w})$의 계산은 매우 어렵게 된다. 이 문제를 해결하기 위해 l 개의 데이터에 기반한 $R(\mathbf{w})$의 근사화된 모델인 경험적 리스크(empirical risk) 모델을 사용한다.

$$R_{emp}(\mathbf{w}) = \frac{1}{l}\sum_{i=1}^{l} \left| d_i - F(\mathbf{x}_i,\mathbf{w}) \right|$$

$R_{emp}(\mathbf{w})$에는 확률 분포의 내용이 없고, 단지 주어진 학습 데이터의 정보만을 활용한다. 또한 확률 이론에 근거하면 학습 데이터의 크기가 커질수록 $R_{emp}(\mathbf{w})$의 최소값은 $R(\mathbf{w})$의 최소값에 근접하게 된다. 경험적 에러 최소화 원리에 따르면 $R_{emp}(\mathbf{w})$가 다음의 수렴 조건을 만족하면 $R_{emp}(\mathbf{w})$의 최소값의 w' 과 $R(\mathbf{w})$의 최소값 w*이 consistent 하다고 말한다.

$$P\left(\sup_{l \to \infty} \left| R(\mathbf{w}) - R_{emp}(\mathbf{w}) \right| > e \right) \to 0$$

Vapnik 과 Chervonenkis 는 수렴 정도의 경계값을 나타내기 위해 VC 차원(Vapnik-Chervonenkis dimension) 파라미터를 도입했다. VC 차원은 $F(\mathbf{x},\mathbf{w})$ 결정 함수의 표현 능력의 측정 수단으로서 $F(\mathbf{x},\mathbf{w})$에 의해 구별 가능한 최대 데이터의 수를 그 값으로 한다. Vapnik 과 Chervonenkis 에 따르면 수렴도의 경계는 다음과 같다.

(4.1) $$R(\mathbf{w}) \le R_{emp}(\mathbf{w}) + \sqrt{\frac{h\left(\ln\frac{2l}{h}+1\right) - \ln\frac{h}{4}}{l}}, \quad \forall \mathbf{w} \in W$$

위 식에서 h 는 VC 차원값을 나타낸다.

(4.1)로부터 $R(\mathbf{w})$를 최소로 하기 위해서는 $R_{emp}(\mathbf{w})$와 $\frac{h}{l}$을 동시에 줄여야 한다는 것을 알 수 있다. 일반적으로 VC 차원이 증가할수록 $R_{emp}(\mathbf{w})$는 감소하게 된다. VC 차원이 크다는 것은 학습 모델의 표현력이 크다는 것을 의미한다. 그런데 VC 차원이 크다는 것이 반드시 학습 모델의 우수성을 나타내지는 않는다. 학습 데이터의 수가 고정된 값이라면 VC 차원의 증가에 따라 $R_{emp}(\mathbf{w})$는 감소되지만 $\sqrt{\frac{h\left(\ln\frac{2l}{h}+1\right)-\ln\frac{h}{4}}{l}}$ 값이 증가하게 된다. 따라서 두항의 합을 동시에 최저로 만드는 최적의 VC 차원의 학습 모델을 얻어야 한다.

4.1.2 Structural Risk Minimization (SRM)

최적의 VC 차원 선택 문제를 해결하기 위해 Vapnik 은 구조적 에러 최소화 기법 Structural Risk Minimization (SRM)을 제안했다. 학습 데이터와 미지의 데이터를 동일 확률 분포로부터 끌어 내어 테스트를 했을 때의 학습 모델의 에러를 실제 에러(true error)라고 한다. 경험적 에러(empirical risk)와 실제 에러의 관계를 그림 4-1 에 보였다. VC 차원이 증가함에 따라서 경험적 에러는 계속 감소하게 되는 한편 신뢰 구간은 증가하게 된다. 그러나 실제 에러는 최적의 VC 차원 지점까지 감소하다가 다시 증가하게 된다. 바로 이 최소 지점이 최적의 모델에 해당하는 VC 차원이 된다.

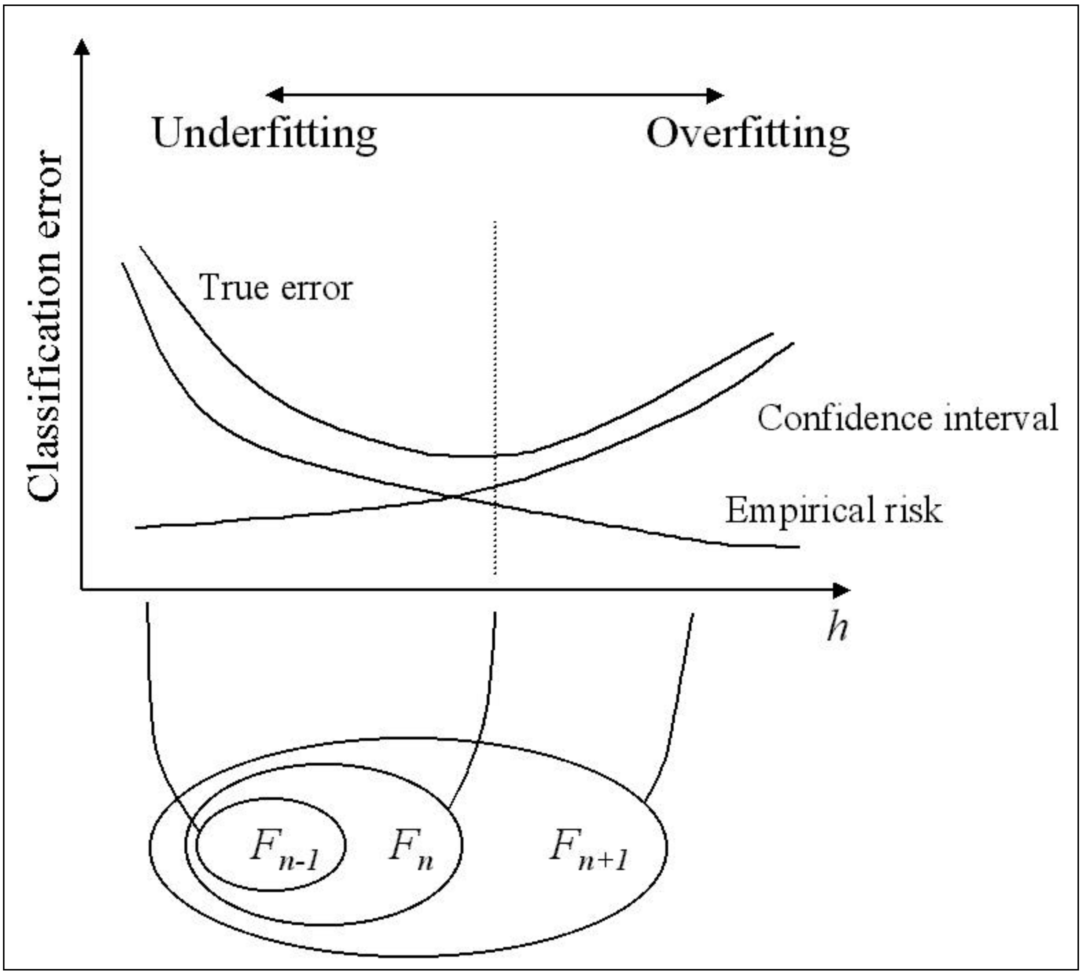

그림 4- 1 에러들간의 관계

SRM 의 기본 개념은 다음과 같다.

패턴 분류 모델

$$F_k(\mathbf{x}, \mathbf{w}); \mathbf{w} \in W_k, \qquad k = 1,2,\ldots,n$$

을 아래와 같은 포함 구조로 표현 했을 때,

$$F_1 \subset F_2 \subset \ldots \subset F_n$$

이에 대한 VC 차원은

$$h_1 \le h_2 \le \ldots \le h_n$$

와 같다. 모델 선택은 각 모델에 대한 경험적 에러를 계산한 후 (4.1) 식의 상한값을 최소로 하는 모델을 고르는 것이다.

SRM 은 수학적으로 잘 정의되어 있지만 다음과 같은 이유로 구현이 어렵다.

일반적으로 임의의 학습 모델 Fn 의 VC 차원을 계산하는 것은 어렵다. 또한 실제로 VC 차원의 계산법이 알려진 모델은 몇 안된다.

학습이 진행되는 동안에 VC 차원을 조절하는 것은 쉬운 일이 아니기 때문에 SRM 의 구현은 쉬운 일이 아니다. SVM 알고리즘에서는 VC 차원을 조절하는 일이 가능하기 때문에 VC 차원의 상한값과 학습에러를 동시에 최소로 줄일 수 있다.

4.2 SVM 알고리즘

4.2.1 선형 분리 가능한(linearly separable) 문제

SVM의 기본 원리는 선형 분리 가능한 문제에서 출발한다. 학습 데이터의 출력으로 {-1,+1}처럼 이진값만이 존재한다면 학습 데이터를

$S_+ = \{\mathbf{x}_i : (\mathbf{x}_i, d_i),\ \ d_i = +1\}$ 과

$S_- = \{\mathbf{x}_i : (\mathbf{x}_i, d_i),\ \ d_i = -1\}$ 의

두 집합으로 구분지을 수 있다. 선형 분리 가능하다는 뜻은 이렇게 두 집합으로 구분짓는 초평면[2](hyperplane)이 존재한다는 뜻이다. 그 초평면을 다음과 같이 표현할 수 있다.

(4.2) $$\mathbf{w}^T \mathbf{x} + b = 0$$

x는 입력 벡터이고, **w**는 가중치 벡터, b는 바이어스이다.

그러면 SVM을 학습한다는 것은

$$\mathbf{w}^T \mathbf{x}_i + b \geq 0,\ \ \forall \mathbf{x}_i \in S_+$$
$$\mathbf{w}^T \mathbf{x}_i + b \leq 0,\ \ \forall \mathbf{x}_i \in S_-$$

을 만족 시키는 최적의 **w**와 b를 찾는 문제가 된다. 그런데 **w**와 b에 상수를 곱하여도 초평면을 변경시키는 것은 아니다. 따라서 하나의 초평면을 표현하는 방법이 여러가지가 될 수 있기 때문에 다음과 같이 정규화를 시킬 수 있다.

(4.3) $$\begin{aligned} \mathbf{w}^T \mathbf{x}_i + b &\geq 1,\ \ \ \forall \mathbf{x}_i \in S_+ \\ \mathbf{w}^T \mathbf{x}_i + b &\leq -1,\ \ \forall \mathbf{x}_i \in S_- \end{aligned}$$

위 조건을 만족하는 초평면을 정규 초평면이라고 한다.

정규 초평면이 학습 데이터의 분류함수로 주어질 때, 정규 초평면에서 학습 데이터와의 거리 r는

---

[2] 2,3 차원의 제약 조건을 벗어나 1 차원, 또는 n 차원에 대한 평면의 정의이다.
$H = \{\mathrm{x} \in R^n : \mathrm{a} \cdot \mathrm{x} = a\}$

$$r = \frac{\left|\mathbf{w}^T \cdot \mathbf{x} + b\right|}{\|\mathbf{w}\|} \geq \frac{1}{\|\mathbf{w}\|}$$

의 관계가 있게 되고, 이 하한값의 거리에 있는 데이터는 정규 하이퍼 평면과 가장 가까운 거리에 위치하게 된다. 그러면 정규 하이퍼 평면이 분류해 내는 두 클래스간의 거리는

(4.4) $$r = 2r = \frac{2}{\|\mathbf{w}\|}$$

가 되고, 이때 r를 분류 한도(margin of separation)이라고 한다.

서포트 벡터(support vector)란 데이터로부터 평면까지의 거리가 $r = \frac{1}{\|\mathbf{w}\|}$인 점들을 말한다. 분류 한도가 크다는 것은 그만큼 학습 데이터에 대해서 잘 학습되었다는 것을 의미하고 미지의 데이터에 대한 에러의 확률을 최소로 만든다. SVM 학습의 목표는 분류 한도를 최대로 만드는 정규 하이퍼 평면을 구하는 것이다. r를 최대로 하는 것은 $\|w\|$을 최소가 되게 하는 w 를 찾는 것과 동일하다. 이 상황을 그림 4-2 에 보였다. 최대의 r를 얻어 내는데는 서포트 벡터 이상의 정보는 필요 없다는 것을 알 수 있다. 즉 학습 데이터에서 서포트 벡터를 이외의 데이터를 제외시켜도 학습의 결과에는 영향이 없다.

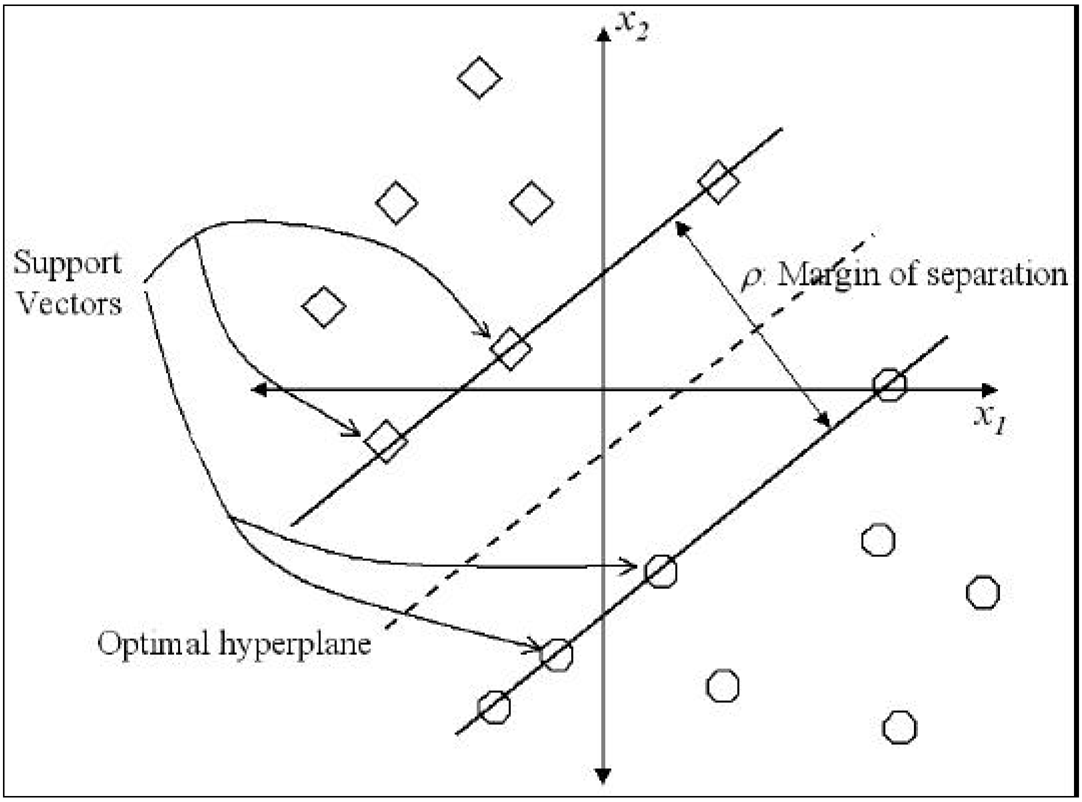

그림 4- 2 선형 분리 가능 문제와 하이퍼 평면

4.2.2 통계적 학습 모델의 관점에서 본 SVM

4.1 절의 SRM 원리를 적용시키려면 VC 차원을 다르게 하여 초평면들을 구성할 수 있어야 한다. 이렇게 구성된 초평면들 중에서 경험적 에러와 VC 차원을 동시에 최저로 만드는 초평면을 선택해내는 것이 SRM 원리이다. SVM 에서는 무게 벡터 **w** 의 유클리디언 놈 ||**w**||에 제약을 가함으로써 초평면들에 SRM 을

실현하기 위한 구조를 정의한다. Vapnik 은 다음의 정리를 보였다.

모든 데이터 $\mathbf{x}_1, \mathbf{x}_2, .., \mathbf{x}_l$ 을 포함하는 볼[3](ball)의 반지름을 R 이라고 하자. 이 때 분류 함수인 하이퍼 평면의 구조를 다음과 같이 표현할 수 있다.

$$F_k(\mathbf{w}, \mathbf{x}) = \{\mathbf{w}^{\mathrm{T}} \cdot \mathbf{x} + b : \|\mathbf{w}\|^2 \le A_k\}$$

그러면 F_k에 대한 VC 차원 h_k는 입력 데이터 공간의 차원이 m_0 일 때 다음 상한값을 갖는다.

$$h_k \le \min\{R^2 A_k^2, m_0\} + 1$$

VC 차원 h_k는 입력 데이터 공간의 차원과 독립적으로 제어할 수 있다. 즉 $\|\mathbf{w}\|$에 따라서 VC 차원을 제어할 수 있게된다. 선형 분리 가능 문제는 학습에러가 0 인 초평면을 얻어낼 수 있다. 따라서 SRM 원리에 따르면 이들 최적의 초평면들 중 VC 차원이 제일 작은 모델을 선택하면 된다.

결국, SVM 학습은 다음과 같이 정의할 수 있다.

주어진 데이터,

$$(\mathbf{x}_1, d_1), \dots, (\mathbf{x}_l, d_l), \qquad \mathbf{x}_i \in R^N, d_i \in \{-1,\ 1\}$$

에 대해서,

$$\text{(4.5)} \quad \begin{aligned} &\underset{\mathbf{w}, b}{\text{Minimize}} \quad \Phi(\mathbf{w}) = \frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} \\ &\text{subject to} \\ &\qquad d_i(\mathbf{w}^{\mathrm{T}}\mathbf{x}_i + b) \ge 1 \qquad i = 1, 2, \dots, l \end{aligned}$$

---

[3] n 차원 벡터 x 의 반지름 r 인 볼은 다음과 같이 정의된다.
$B(\mathbf{x}, r) = \{\mathbf{x}^{\mathrm{o}} \in R^n : \|\mathbf{x}^{\mathrm{o}} - \mathbf{x}\| < r\}$

4.2.3 선형 분리 가능하지 않은 문제

선형 분리 가능하지 않다는 뜻은 학습 에러가 0 이 되도록 두개의 클래스를 나누는 초평면이 존재하지 않는다는 것을 의미한다. 이 상황을 그림 4-3 에 보였다.

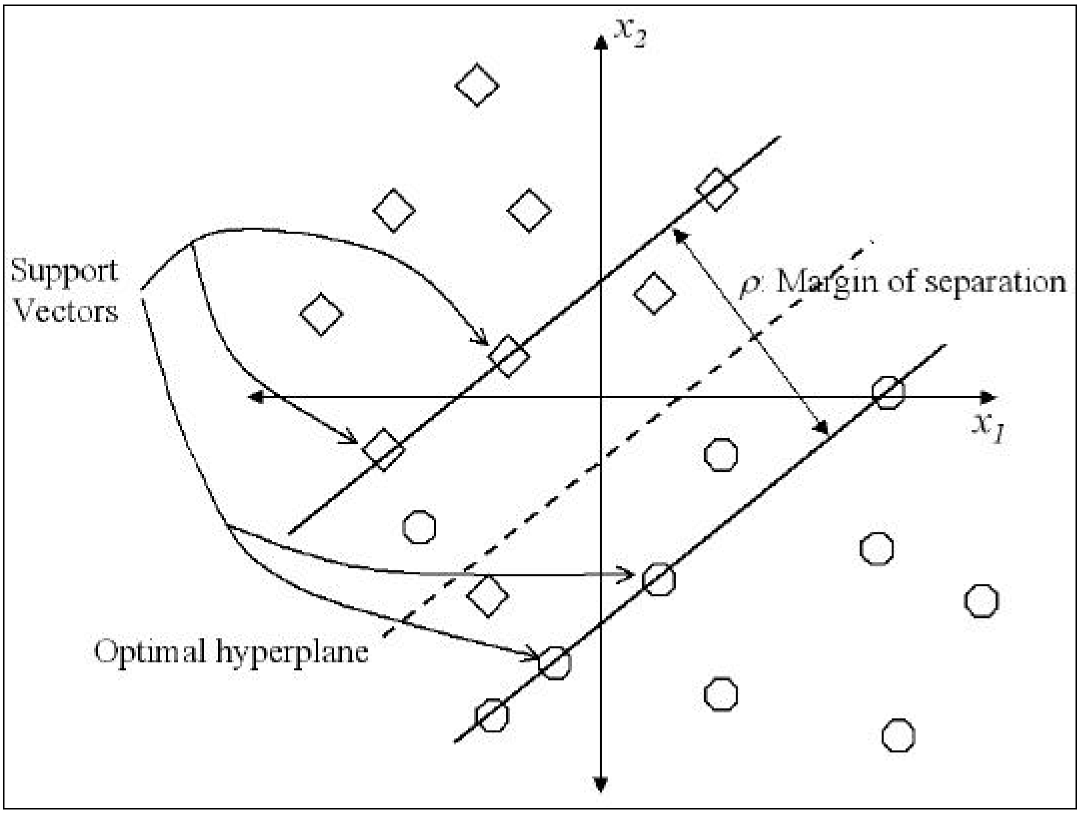

그림 4- 3 선형 분리 불가능 문제

선형 분리 불가능 문제를 해결하기 위해 $\{x_i\}_{i=1}^{l}$ 이라는 slack 변수를 두어 데이터의 초평면에 대한 편향 정도를 보상한다.

(4.6) $d_i(\mathbf{w}^{\mathrm{T}}\mathbf{x}_i + b) \geq 1 - x_i \quad i = 1,2,...,l$

ξ는 원래 제약 조건에서 어느 정도 벗어났는지에 대한 측정 수단이다. 그러면 학습 에러는

$$\Phi(\xi) = \sum_{i=1}^{l} x_i$$

과 같은 측정 수단을 사용할 수 있다.

4.2.4 비선형 자질 공간

하이퍼 평면을 가지고 입력 데이터의 분리가 불가능할 때 이 입력 공간을 분리하는 비선형 표면(nonlinear surface)을 생각할 수 있다. 그런데 이 비선형 표면의 식을 분석적으로 계산해 낸다는 것은 매우 어려운 일이다. SVM 에서는 입력 데이터 공간을 더 높은 차원의 자질 공간(feature space)으로 사상시킨다. 이런 매핑의 기본 아이디어는 다음과 같다.

낮은 차원의 비선형 함수는 높은 차원에서는 선형 함수로 근사화시킬 수 있다. 따라서 자질 공간에서의 최적의 하이퍼 평면을 구한다. 그러면 입력 데이터 공간에서는 최적의 비선형 함수를 구하는 셈이 된다. 입력 데이터 공간을 자질 공간으로 매핑 시키는 상황을 그림 4-4 에 보였다.

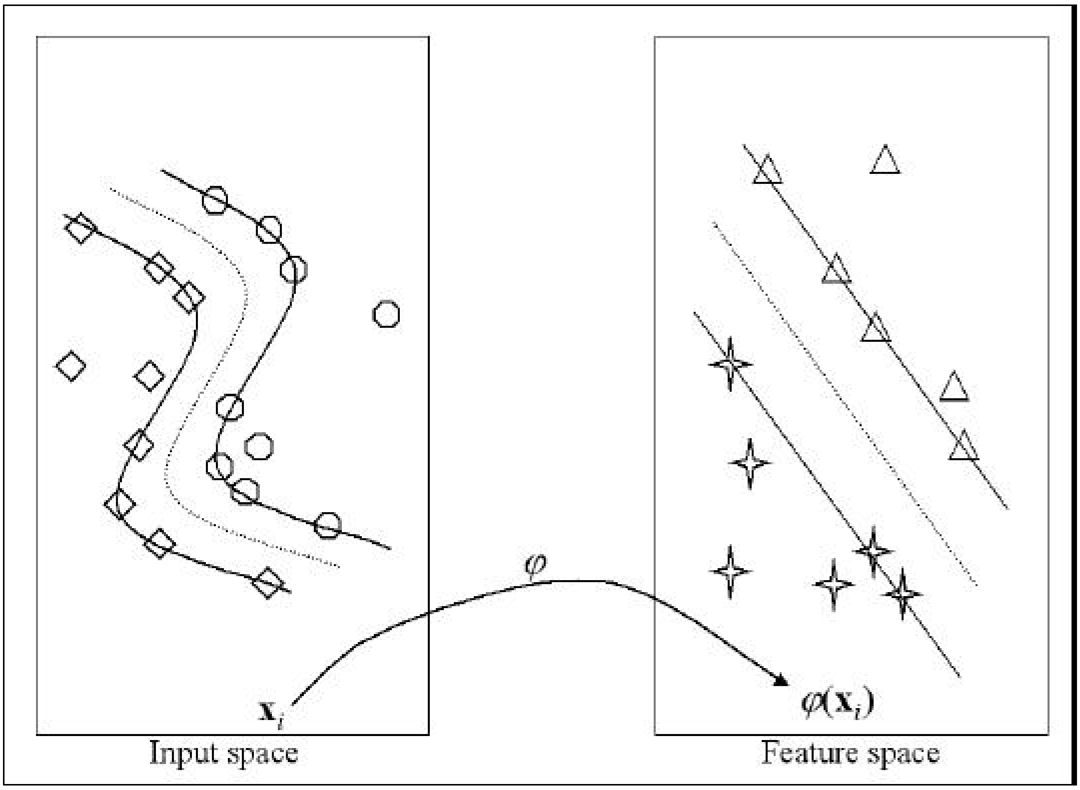

그림 4-4 비선형 공간으로의 매핑

입력 데이터 공간의 차원을 m_0 **라고 하고 자질 공간의 차원을** m_1 **이라고 하자. 그러면 입력 데이터를** m_1 **차원의 공간으로 매핑 시키는 벡터 함수** φ**를 다음과 같이 정의할 수 있다.**

$$j(\mathbf{x}) = \{ j_1(\mathbf{x}),\ldots,\ j_{m_1}(\mathbf{x})\}$$

이 때 자질 공간에서의 하이퍼 평면은 $\mathbf{w}^T j(\mathbf{x}) + b = 0$ **가 된다.**

그러면 최종적인 SVM **의 학습 문제는 다음과 같은 문제로 요약할 수 있다.**

$$\underset{\mathrm{w},b,\xi}{\text{Minimize}} \quad \Phi(\mathbf{w},\xi) = \frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} + C\sum_{i=1}^{l} x_i$$

(4.7) subject to

$$d_i(\mathbf{w}^{\mathrm{T}}\mathbf{x}_i + b) \geq 1 - x_i \qquad i = 1,2,\ldots,l$$

$$x_i \geq 0 \qquad i = 1,2,\ldots,l$$

이 때, C 는 에러 민감도로서 C 가 크면 클수록 (4.7)의 최소화 문제는 에러에 둔감한, 즉 **w** 만을 최소화하면 되는 문제가 되고, C 를 작게 설정하면 에러에 예민한 즉 에러를 최소화하는 문제가 된다.

4.2.5 SVM 학습

식 (4.7)과 같은 문제를 Convex Programming 이라 한다[Peressini et al. 88]. 다음 조건

$$f(l\mathbf{x}_1 + [1-l]\mathbf{x}_2) \leq lf(\mathbf{x}_1) + [1-l]f(\mathbf{x}_2)$$

을 만족하면 함수 $f(\mathbf{x})$를 Convex 함수라고 한다. Convex Programming 이란 제약조건도 Convex 함수이고, 목표 함수(target function)도 Convex 함수인 문제를 일컫는다. (4.7)을 직접 푸는 것은 쉽지 않기 때문에 Lagrangian Duality 를 이용해서 Dual Problem 으로 변환한다[Nash et al. 97]. 식 (4.7) 문제에 대한 Lagrangian primal function 은 다음과 같다.

(4.8) $$L(\mathbf{w},b,x,l,g) = \frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} - \sum_{i=1}^{l} l_i\left\{d_i(\mathbf{w}^T j(\mathbf{x}_i) + b) - 1 + x_i\right\} - \sum_{i=1}^{l} g_i x_i + C\sum_{i=1}^{l} x_i$$

이 때 $l_i \geq 0,\ g_i \geq 0$ 는 Lagrange Multiplier 이다.

그러면 min-max duality[4]에 의해 Dual Problem 은

$$(4.9) \quad \underset{l \geq 0, g \geq 0}{\text{maximize}} \left\{ \min_{\mathbf{w}, b, x} L(\mathbf{w}, b, x, l, g) \right\}$$

가 된다. (4.8)은 Convex Programming 이므로 $\min_{\mathbf{w},b,x} L(\mathbf{w},b,x,l,g)$ 이 최소값을 가질 조건은 다음과 같다.

$$\text{cond1}: \quad \frac{\partial L(\mathbf{w},b,x,l,g)}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^{l} l_i d_i j(\mathbf{x}_i) = 0$$

$$\text{cond2}: \quad \frac{\partial L(\mathbf{w},b,x,l,g)}{\partial b} = \sum_{i=1}^{l} l_i d_i = 0$$

$$\text{cond3}: \quad \frac{\partial L(\mathbf{w},b,x,l,g)}{\partial x} = -l_i - g_i + C = 0$$

cond1 을 (4.9)에 대입하면 다음 문제로 바뀌게된다.

$$\underset{l}{\text{maximize}} \left\{ L_*(l) = \sum_{\text{i}=1}^{l} l_i - \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} l_i l_j d_i d_j j^T(\mathbf{x}_i) j(\mathbf{x}_j) \right\}$$

$$(4.10) \qquad \sum_{i=1}^{l} l_i d_i = 0$$

$$0 \leq l_i \leq C$$

(4.10)에서 주목할 점은 SVM 알고리즘에서는 자질 공간으로의 매핑함수 자체에 대한 정보는 중요치 않고 단지 그들의 스칼라 곱인 $j^T(\mathbf{x}_i) j(\mathbf{x}_j)$ 의 크기만 알면 된다는 것이다. 그 스칼라 곱을 표현하는 함수를 커널 함수 $K(\mathbf{x}_i,\mathbf{x}_j)$라고 한다.

---

[4] $F(x_*, y) \leq F(x_*, y_*) \leq F(x, y_*)$ 를 만족하는 점 (x*,y*)가 존재하면 $\max_{y \in Y} \min_{x \in X} F(x, y) = \min_{x \in X} \max_{y \in Y} F(x, y)$ 의 조건이 성립된다.

$$K(\mathbf{x}_i,\mathbf{x}_j)=\varphi^T(\mathbf{x}_i)\varphi(\mathbf{x}_j)=\varphi^T(\mathbf{x}_j)\varphi(\mathbf{x}_i)=\sum_k a_k\varphi_k^T(\mathbf{x}_i)\varphi_k(\mathbf{x}_j)$$

SVM 에서 주로 사용하는 알려진 커널 함수로는 다음의 것들이 있다.

커널 함수	분류 시스템
$(1+\mathbf{x}^T\mathbf{y})^p$	P 차 다항식
$\exp\left(-\frac{1}{2\sigma^2}\lVert\mathbf{x}-\mathbf{x}_i\rVert^2\right)$	폭이 σ^2 인 RBF 망
$\tanh\left(b_0\mathbf{x}^T\mathbf{x}_i+b_1\right)$	2 층 퍼셉트론 망

표 4-1 알려진 커널 함수들

그러면 (4.10)의 해 λ에 대응하는 w, b 는 다음과 같이 계산이 된다.

$$\sum_{i=1}^{l}\lambda_i d_i K(\mathbf{x},\mathbf{x}_i)+b=0$$

$$\mathrm{w}=\sum_{i=1}^{Ns}\lambda_i d_i\varphi(\mathbf{x}_i),\qquad Ns=\text{Number of Support Vectors}$$

$$b=1-\mathbf{w}^T\sum_{i=1}^{Ns}\lambda_i d_i K(\mathbf{x}_i,\mathbf{x}_0)$$ 단, d_i=1, λ_0 가 $0<\lambda_0<C$ 인 $\mathbf{x}_0$

b 를 계산하는데는 Karush-Kuhn-Tucker 이론이 사용된다. KKT 이론에 따르면 안정점(saddle point)[5] ($\mathbf{w}_*, b_*, \xi_*, \lambda_*, \gamma_*$)에 대해서 다음 식이 성립한다.

$$\lambda_i[d_i(\mathbf{w}^T\varphi(\mathbf{x}_i)+b)-1+\xi_i]=0,\qquad i=1,2,\ldots,l$$
$$\gamma_i\xi_i=0,\qquad i=1,2,\ldots,l$$

만일 λ_i < C 이면 cond3 에 의해 ξ_i 는 0 이 된다. 따라서 $0<\lambda_i<C$ 인 $\mathbf{x}_i$ 를 고

[5] L 이 Lagrangian 함수일 때 $L(x^*,\lambda)\le L(x^*,\lambda^*)\le L(x,\lambda^*)$ 를 만족하는 점 (x^*,λ^*)를 안정점이라고 한다.

르면 위의 조건에서 $d_i(\mathbf{w}^T \varphi(\mathbf{x}_i)+b)-1=0$ 이 된다. 따라서 b 는 위의 식처럼 구할 수 있게 된다.

(4.10)처럼 Convex Programmming 인 경우, 2 차식으로 표현되는 문제를 Quadratic Programming 이라고 한다[Nash et al. 97]. (4.10)을 Quadratic Programming 으로 풀기 위해 정규화시키면 아래의 식을 얻는다.

$$\text{Minimize} \quad F(\Lambda) = -\Lambda 1 + \frac{1}{2}\Lambda \mathrm{H} \Lambda$$

$$\text{subject to}$$

$$\Lambda \mathrm{d} = 0$$

$$\Lambda \le C1 \qquad (4.11)$$

$$\Lambda \ge 0$$

이 때 $\mathrm{H}_{ij}=d_i d_j K(\mathbf{x}_i, \mathbf{x}_j)$ 이다. Quadratic Programming 은 매우 복잡하기 때문에 보통 상용 라이브러리를 이용한다. 본 논문에서는 LOQO[Vanderbei 97]을 사용했다.

5 장
실험

5.1 실험 데이터

본 실험에서는 문서 분류의 데이터로서 Reuters-21578 문서 집합을 사용했다. Reuters-21578 문서 집합은 Reuters newswire 에 실린 기사들의 모음이다. 이 기사 뭉치를 Reuter 와 Carnegie 그룹에서 수작업으로 범주를 인덱싱 해서 Reusters-22173 문서 집합을 공개했다. 그 후 David Lewis 등이 다듬는 작업을 해서 중복되는 595 개의 문서를 없애고 정리를 해서 Reuters-21578 문서 집합이 되었다. Reuters-21578 문서 집합은 SGML 태그를 사용해서 작성 되었다. Reuters-21578 은 다섯개의 범주 집합으로 구성되어 있고, 각 범주 집합은 여러 개의 범주로 구성된다. 표 5-1 에 범주 집합과 집합 당 범주의 수를 보인다.

범주 집합	범주 개수
EXCHANGES	39
ORGS	56
PEOPLE	267
PLACES	175
TOPICS	135

표 5-1 범주 집합

문서마다, 각 범주 집합에 대해 범주가 지정된다. 하나의 범주 집합에 여러 개의 범주가 지정될 수도 있고, 전혀 지정되지 않을 수도 있다.

범주 집합 중 TOPICS 범주 집합은 경제분야의 범주들의 모임이다. Reuters-21578 에 대한 인덱싱 작업이나 연구는 TOPICS 범주 집합에 매우 활발하므로 본 논문에서도 TOPICS 범주 집합에 대해서만 분류 실험을 행했다.

문서 분류 문제는 일반적으로 데이터를 학습 데이터와 테스트 데이터의 2 가지로 분리를 해서 학습 데이터에 대해서 문서 분류 시스템을 조직하고 테스트 데이터를 가지고 그 성능을 측정한다. 이 때 학습 데이터와 테스트 데이터 작성은 실험자마다 다를 수 있기 때문에 타 시스템과의 성능 비교를 위해 Reuters-21578 문서 집합은 학습 데이터와 테스트 데이터를 다음과 같은 3 가지 방식으로 구성할 것을 지침하고 있다.

ModLewis 분리
학습 데이터(13,625): LEWISSPLIT="TRAIN"; TOPICS="YES" or "NO" **테스트 데이터**(6,1888): LEWISSPLIT="TEST"; TOPICS="YES" or "NO" **미사용**(1,765): LEWISSPLIT="NOT-USED" or TOPICS="BYPASS"
ModApte 분리
학습 데이터(9,603): LEWISSPLIT="TRAIN"; TOPICS="YES" **테스트 데이터**(3,299): LEWISSPLIT="TEST"; TOPICS="YES"

미사용(8,676): 그외
ModHayes 분리:
학습 데이터(20856): CGISPLIT="TRAINING-SET" 테스트 데이터(722): CGISPLIT="PUBLISHED-TESTSET" 미사용(0):

표 5- 2 **Reuters-21578** 데이터 구성표

5.2 실험 데이터 구성

학습 데이터와 테스트 데이터의 구성은 ModApte 분리를 따랐다. 우선 Reuters-21578 문서 집합을 구성하는 단어들의 사전을 구축했다. 사전의 엔트리는 다음과 같은 형식으로 구성이 되었다.

단어	DF

표 5- 3 사전 엔트리

DF (Document Frequency)는 각 단어가 등장한 문서의 수를 나타낸다.

<BODY>와 </BODY> 사이에 있는 단어들이 Reuters newsline 의 기사의 본문이므로 이 본문에 대해서만 사전을 구축했다. 사전을 구축할 때는 학습 데이터와 테스트 데이터 외에 미사용 데이터까지 전부 사용했다. 본문의 모든 단어를 모두 활용할 경우 단어 수가 100,000 을 넘는 매우 방대한 사전이 만들어 졌다. 사전의 크기를 줄이기 위해 stemming 알고리즘과 불용어 목록을 써서 단어

수를 줄였고, 또한 IDF 가 4 미만인 단어는 사전에서 제외 되었다. 그렇게 해서 최종적으로 만들어진 사전의 엔트리 수는 8754 가 되었다. SVM 에서는 8754 단어 모두를 유용한 자질(feature) 정보로서 사용한다. 이 사전을 가지고 학습 데이터와 테스트 데이터 집합의 문서를 다음과 같이 벡터로 표현한다.

분류결과값	자질 0:자질값 0	…	자질 n:자질값 n

자질은 사전내 단어에 해당하고 자질값은 *tfidf* 의 값이다. 그리고 벡터를 표현할 때 정규화 시키는 것이 통상의 방법이므로 벡터는 유클리디언 놈이 1 이 되도록 정규화를 시켰다.

ModApte 의 분리 기준을 따라 학습 데이터와 테스트 데이터를 구성했는데, 이에 따르면 총 학습 데이터는 9,603 개이고 테스트 데이터는 3,299 개이다. 그런데 어떤 문서는 <BODY></BODY>가 비어 있는 문서가 있다. 즉 본문이 없는 문서인 경우이다. 본 논문에서는 본문이 없는 문서는 고려 대상에서 제외를 시켰다. 최종적으로 실험에 사용된 데이터는 총 학습 데이터가 8,762 개이고 테스트 데이터는 3,009 개였다.

TOPICS 범주 집합내 135 개 범주들 중, 학습 데이터와 테스트 데이터 내의 문서들이 많이 속하는 범주들로 순위를 매긴 후 상위 10 개의 범주들을 표에 보였다.

범주	학습	테스트
Earn	2839	1043
Acq	1611	672
Money-fx	513	143
Grain	415	125
Crude	370	156
Trade	351	102
Interest	323	97
Wheat	198	61
Ship	175	66
Corn	152	36

표 5- 4 상위 10 개 범주 구성표

본 논문에서는 10 개중 "corn", "crude", "earn", "grain", "interest"의 5 개의 범주에 대해서 분류 작업을 했다.

5.3 실험 결과

실험 결과는 두가지 평가법으로 평가 되었다.

1. 정확도(Accuracy)

$$정확도 = \frac{올바르게\ 분류된\ 문서의\ 수}{테스트\ 문서의\ 수}$$

2. 채산점(Precision/Recall break-even point)

정보 검색에서 성능 측정 수단으로 통상적으로 쓰이는 것이 recall 과 precision 이다. 다음과 같은 분할표로부터 recall 과 precision 을 이끌어 낸다.

	+1 이 정답	-1 이 정답
+1 로 분류	*a*	*b*
-1 로 분류	*c*	*d*

표 5- 5 분할표

그러면 recall = $a/(a+c)$이고 precision = $a/(a+b)$가 된다. recall 과 precision 을 조합하여 사용하는 측정 수단으로 채산점(precision/recall break-even point)의 방법이 있다.

$$\text{precision/recall break even point} = \frac{2 \times precision \times recall}{precision + recall}$$

정확도(Accuracy)는 위 분할표에 따르면 $(a+d)/(a+b+c+d)$가 된다.

5.3.1 SVM 실험 결과

C=1000 으로 고정 시킨 후 세가지 커널에 대한 SVM 실험 결과를 그림 5-1, 5-2, 5-3 에 보였다.

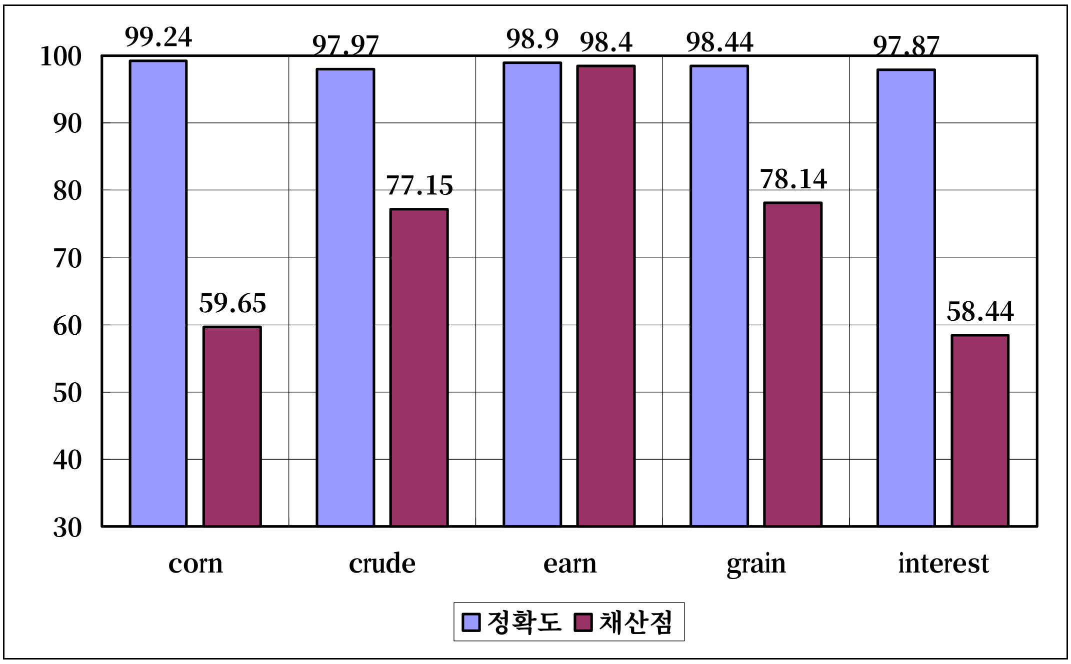

그림 5-1 차수가 5 인 다항식 커널

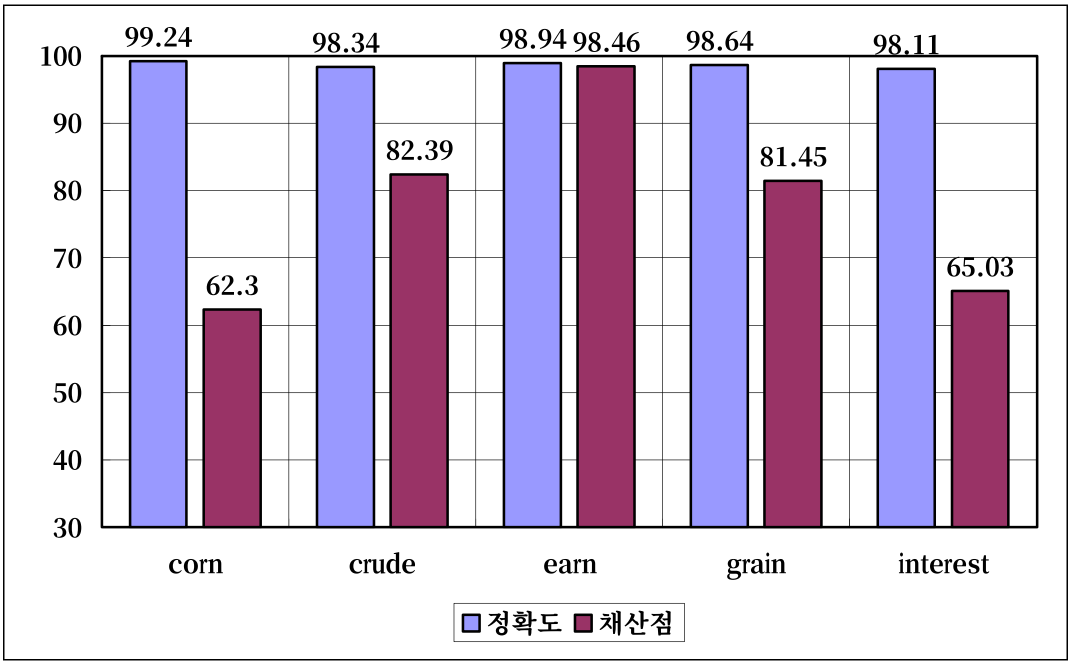

그림 5-2 s가 1 인 RBF 망

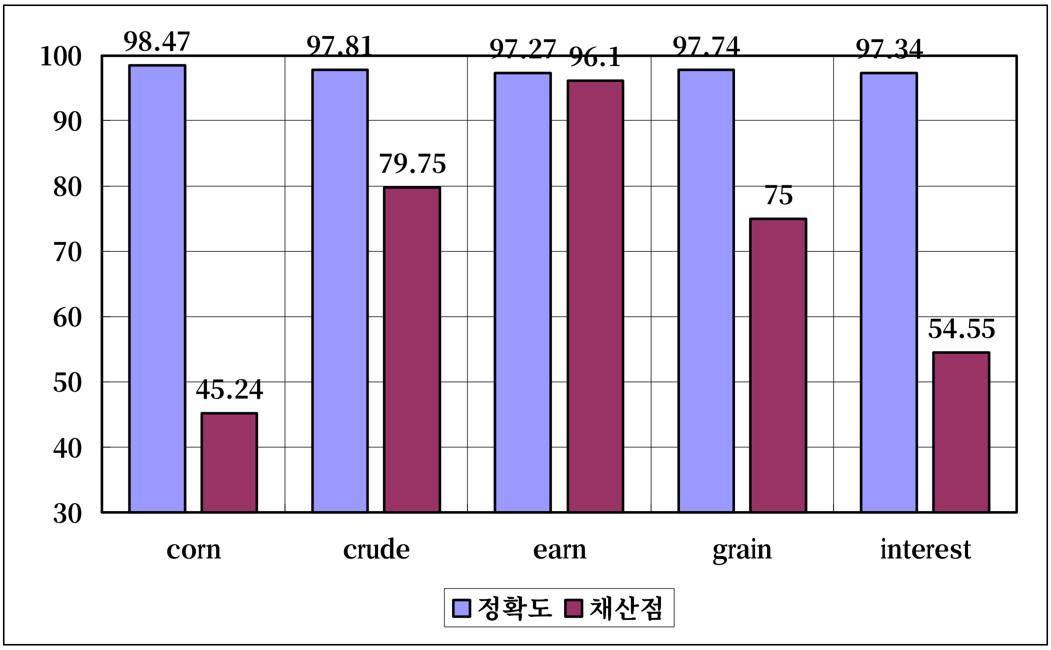

그림 5- 3 b_0=2, b_1=1 인 2 층 퍼셉트론망 커널

		범주 종류				
		corn	crude	earn	grain	interest
커널 종류	다항식	1540	2248	3297	2566	1517
	RBF	943	1491	2469	1749	1083
	퍼셉트론	124	179	419	209	286

표 5- 6 모델 당 SV 의 수

그림 5-1, 5-2, 5-3 에서 알 수 있듯이 SVM 은 정확도의 측면에서 97% 이상의 상당한 성능을 보였다. 채산점의 측면에서는 "earn" 범주를 제외하고는 그리 뛰어난 수준은 아니다. 이는 표 5-4 에서 알 수 있듯이 "corn", "interest", "crude", "grain" 범주에 속하는 데이터의 수가 극히 빈약하기 때문에 나타나는 현상이다. 반면 earn 에 대해서는 데이터의 상당 부분이 이 범주에 속하기 때문에 채산점이 높게 나온다. SV 의 수는 퍼셉트론 커널을 사용하는 경우 가장 적게 나왔다.

5.3.2 Naï ve Bayesian 분류와의 비교

Naï ve Bayesian 모델과 채산점을 비교한 그림을 5-6 에 보였다

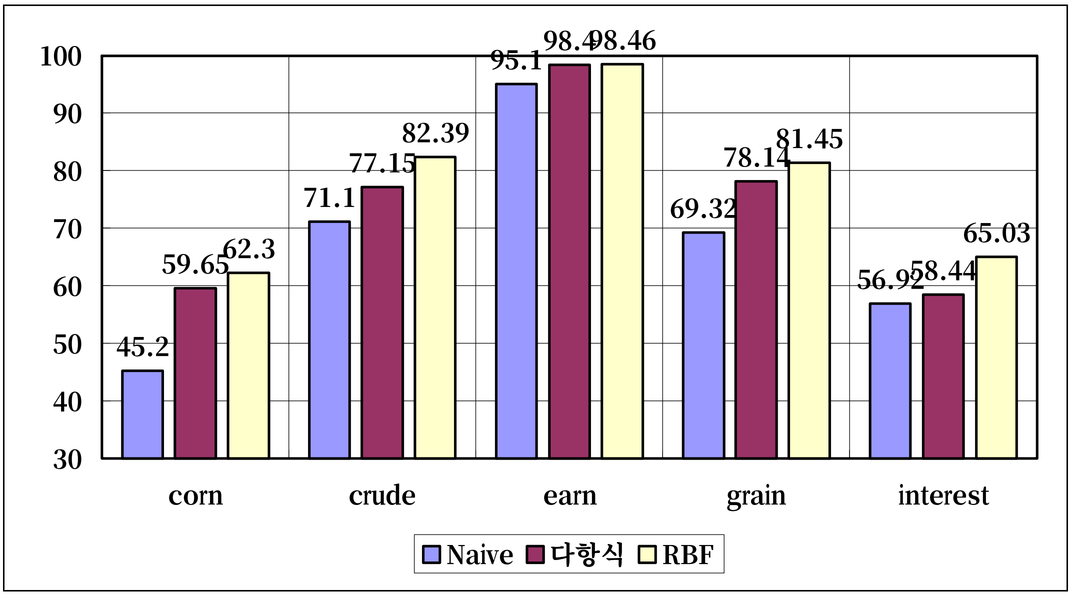

그림 5- 4 Naive Bayesian 모델과 비교

그림을 보면 SVM 의 결과가 Naï ve Bayesian 의 결과 보다 좋은 것을 알 수 있다. 다항식과 RBF 커널 함수를 사용한 SVM 의 테스트 결과가 항상 Naï ve Bayesian 의 결과보다 높은 채산점을 보인다.

5.3.3 다층 신경망으로 구현

표 5-6 의 다항식 커널과 RBF 커널에 대한 SV 들만을 가지고 다층 퍼셉트론을 학습시켰다. 전체 학습 데이터의 수가 8762 인 것에 비해 SV 에 해당하는 학습 데이터만을 가지고 학습시키는 것은 신경망 학습 속도를 매우 빠르게 진행시켰다. 학습에 사용된 다층 신경망은 하나의 은닉층으로 구성되며 이 은닉층은 2 개의 은닉 뉴런으로 구성되었다. 학습 결과를 그림 5-5, 5-6 에 보였다.

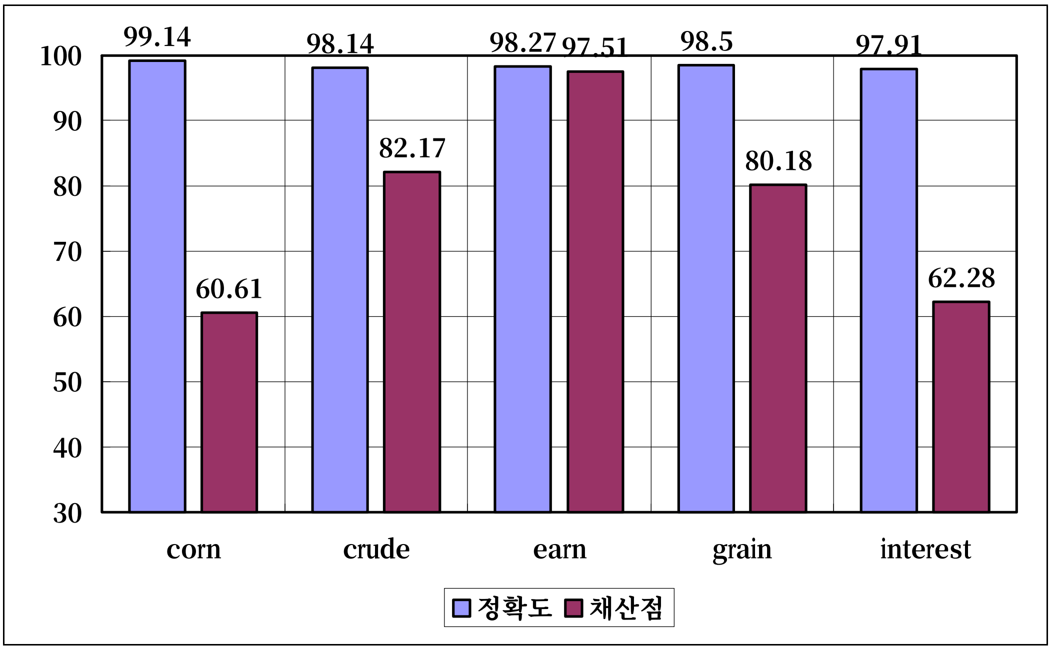

그림 5- 5 다항식 커널의 SV 에 대한 신경망 학습 결과

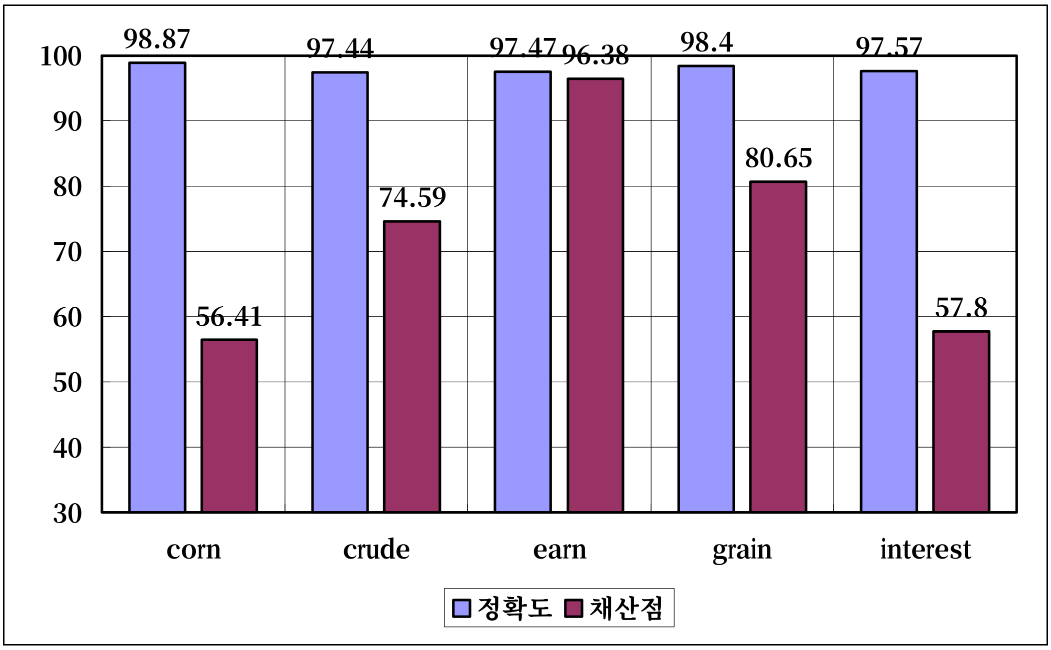

그림 5- 6 RBF 커널의 SV 에 대한 신경망 학습결과

그림 5-5, 5-6 의 결과를 그림 5-1,5-2 와 비교해 보면 성능이 거의 비슷한 것을 알 수 있다. 전체 학습 데이터를 학습에 이용해도 이와 비슷한 신경망 학습 결과를 얻을 수 있겠지만 신경망 학습의 성능에 결정적인 영향을 끼치는 중요한 데이터는 그리 많지 않다는 것을 보여 준다.

6 장
결론

본 논문은 문서를 미리 정의된 범주별로 나누는 문서 분류 시스템을 기계 학습 기법, 그 중에서도 신경망 기반의 해결법을 모색하였다. 그 중에서도 SVM 알고리즘을 도입했으며 이는 최근 연구가 매우 활발히 이루어지고 있는 분야이다. 문서 분류 문제에서 SVM 은 통계적 학습 이론을 충실히 실천한 모델이며 최적의 해를 얻는 것을 보장해 준다. 채산점 비교에서 SVM 은 Naïve Bayesian 모델보다 우수하다는 것을 알 수 있었다. SVM 학습을 통해 얻어진 SV 의 개수는 전체 학습 데이터의 개수에 비해 상대적으로 크게 적으며 이것은 학습시 학습 모델을 결정 짓는 유용한 정보는 전체 학습 데이터의 일부분에 밀집되어 있다는 것을 보여준다.

SVM 의 결과로 얻어진 SV 만을 가지고 다층 퍼셉트론을 학습시키는 것은 전체 학습 데이터를 이용하는 학습에 비해 학습 시간을 크게 감소 시킨다. 또한 SV 만으로 얻어진 다층 퍼셉트론의 성능은 전체 학습 데이터를 통한 학습에 비해 뒤떨어지지 않는다.

본 논문의 SVM 은 이진 분류만을 행하는 시스템인데 더 나아가 다중 클래스 분류가 가능한 시스템의 발전으로의 연구가 필요하다. 또한 Quadratic

6. 결론

Programmming 라이브러리들은 행렬식이 커질수록 학습 시간이 급격히 증가하고 불안정해지는 경향이 있다. SVM 에 특화된 Quadratic Optimizer 의 연구도 더 진행되어야 할 것이다.

참고 문헌

[Cherkassky et al. 98] Vladimir Cherkassky and Filip Mulier, *Learning From Data Concepts, Theory, and Methods*, John Wiley & Sons, Inc., 1997.

[Frakes et al. 92] William B. Frakes and Richard Baeze-Yates, *Information Retrieval Data Structures & Algorithms*, Prentice-Hall, Inc., 1997.

[Haykin 98] Simon Haykin, *Neural Networks*" 2nd edition, Prentice-Hall, Inc., 1997.

[Lewis 91] David D. Lewis, Evaluating Text Categorization, In *Proceedings of the Speech and Natural Language Workshop*, pp. 312-317, 1991

[Liere 97] Ray Liere and Prasad Tadepalli, Active Learning with Committees for Text Categorization, In *the Proceedings of AAAI '97*, pp.591-596, 1997.

[Mitchell 97] Tom M. Mitchell, *Machine Learning*, McGraw-Hill Companies, Inc., 1997.

[Nash et al. 97] Stephen G. Nash and Areila Sofer, *Linear and Nonlinear Programming*, McGraw-Hill Companies, Inc., 1997.

[Osuna et al. 97] Edgar E. Osuna, Robert Freund and Federico Girosi, Support Vector Machine: Training and Applications, A.I. Memo, MIT A.I. Lab., 1997.

[Peressini et al. 88] A. L. Peressini, F. E. Sullivan and J. J. Uhl, Jr., *The Mathematics of Nonlinear Programming*, Springer Verlag, New York Inc., 1997.

[Quinlan 93] J. Ross Quinlan, *C4.5: Programs for Machine Learning*, Morang Kaufmann Publishers Inc., 1993.

[Stiston 96] M. O. Stitson, J. A. E. Weston, A. Gammerman, V. Vovk and V. Vapnik, Theory of Support Vector Machines, *Technical Report CSD-TR-96-17*, Royal Holloway University of London, 1997.

[Vanderbei 97] Robert J. Vanderbei, LOQO User's Manual – Version 3.10, *Technical Report SOR-97-08*, Princeton University, 1997

[Vapnik 95] V. Vapnik, *The Nature of Statistical Learning Theory*, Springer Verlag, New York Inc., 1997.

[Yang 97] Yiming Yang, An Evaluation of Statistical Approaches to Text Categorization, *Technical Report CMU-CS-97-127*, Carnegie Mellon University, 1997.

ABSTRACT

Now that the world is connected by online network, it is an age of a flood of information. It is difficult and time-consuming to classify according to user's interests the enormous information pouring in from online network. Therefore, if the classification system can be automatically built using machine learning techniques, it will be very efficient.

The problem of classifying texts has a very higher dimension of input space, and the information that the text itself contains is sparse. In this paper, Support Vector Machine (SVM), an algorithm suitable for problems having these characteristics is implemented. In order to experiment with the effect of Support Vectors (SVs) which SVM produces, multilayer perceptron network is trained over the reduced data set using only SVs. SVM is a very strong algorithm based on Structural Risk Minimization (SRM) of the statistical learning theory. In addition, SVM's learning process which searches optimal solutions is a mathematically well modeled process, called Convex Programming.

In the experiment about 5 frequently-appeared topics of Reuters-21578 document set, it is remarkable that the resulting accuracy is higher than 97%. And, SVM shows a better break-even point than Naive bayesian classifier's. In addition, trained multilayer perceptron network using only SVs not only shows a good performance but also reduces a training time remarkably.

Keywords: Text Classification, Multilayer Perceptron Network, SVM, SRM, Convex Programming

감사의 글

2 년, 730 여일 … 인생의 긴 여정에서 제게 이만큼 큰 변화를 가져올 수 있는 순간이 과연 다시 찾아 올까? 라고 생각해봅니다. 대학원에서의 2 년의 시간은 제게 정말 소중한 시간이었습니다. 삶에 대한 시각도 성숙했으며, 학문에 대한 개념도 새롭게 인식되었습니다. 무엇보다도 미약한 저를 바로 인정하고 주님을 알게 되었습니다.

어리고 부족한 저의 대학원 생활을 아낌없는 지도와 조언, 격려를 통해 이끌어주신 김영택 교수님께 감사 드리고 연구에 대한 저의 편협한 자세를 바로 잡아주시고 새로운 학문의 길을 보여 주신 장병탁 교수님께도 감사 드립니다. 힘이 들 때 용기를 주고 인생의 선배로서 많은 것을 가르쳐준 종우형, 정호형께 감사 드리고 논문에 대해서 신경 많이 써준 성배형께 감사 드립니다. 제가 폐를 많이 끼치긴 했지만 같이 즐겁게 생활한 태진이형, 호성이형에게도 감사 드리고, 실험실의 잡다한 일 불평없이 도와준 후배 동연이, 논문에 도움준 상윤이형, 정집이에게도 감사 드립니다. 연구에 대한 열정과 후배들에 대한 따뜻함으로 푸근한 재원이형, 성동이형, 유섭이형에게도 감사 드립니다. 그리고 2 년 동안 연구실 생활의 동기이자 조언자인 친구 규백이와 같이 기쁜 마음을 나누고 싶습니다.

어느덧 6 년의 기간을 학교에서 같이 보낸 호우회 식구들과 대식, 석현, 성철, 재상, 현석, 형도와 전주에서 열심히 생활하고 있을 자환, 현희에게 고맙다는 말 전하고 싶습니다. 오빠보다 열심히 생활하는 지은이와 부족한 저를 늘 자랑스럽게 생각하시는 아버지, 어머니 사랑합니다.

마지막으로 저의 모든 성과, 기쁨을 사랑이신 주님께 감사 드립니다.