The Korean Communications
in Statistics Vol. 10 No. 1, 2003
pp. 61-77

Web Log Analysis Using Support Vector Regression[1)]

Sung-Hae Jun[2)] Min-Taik Lim[3)] Hongseok Jorn[4)] Jinsoo Hwang[5)]
SeongYong Choi[6)] Jeeyun Kim[7)] Kyung-Whan Oh[8)]

Abstract

Due to the wide expansion of the internet, people can freely get information what they want with lesser efforts. However without adequate forms or rules to follow, it is getting more and more difficult to get necessary information. Because of seemingly chaotic status of the current web environment, it is sometimes called "Dizzy web". The user should wander from page to page to get necessary information. Therefore we need to construct system which properly recommends appropriate information for general user. The representative research field for this system is called Recommendation System(RS). The collaborative recommendation system is one of the RS. It was known to perform better than the other systems. When we perform the web user modeling or other web-mining tasks, the continuous feedback data is very important and frequently used. In this paper, we propose a collaborative recommendation system which can deal with the continuous feedback data and tried to construct the web page prediction system. We use a sojourn time of a user as continuous feedback data and combine the traditional model-based algorithm framework with the Support Vector Regression technique. In our experiments, we show the accuracy of our system and the computing time of page prediction compared with Pearson's correlation algorithm.

Keywords: SVR; Web Log Data; Collaborative Recommendation System.

---

1) This research was supported by Brain Tech program sponsored by Korea Ministry of Science and Technology.
2) Senior Lecturer, Dept. of Statistics, Chongju University
E-mail : shjun@ailab.sogang.ac.kr
3) Researcher, CDMA Lab., LG Electronics
4) Professor, Dept. of Statistics, Inha University
5) Associate Professor, Dept. of Statistics, Inha University
6) Graduate Student, Dept. of Computer Science Engineering, Inha University
7) Graduate Student, Dept. of Statistics, Inha University
8) Professor, Dept. of Computer Science, Sogang University

1. 서론

인터넷으로부터 필요한 정보를 얻기 위하여 무의미한 탐색을 반복하는 경우가 자주 나타나고 있다[7]. 유용한 정보와 무의미한 자료가 섞여 있는 복잡한 Web으로부터 사용자와 관련 있는 정보를 추천해 주는 방법에 대한 연구가 많이 진행되고 있다[3][17][19][20]. 특히 협동 추천시스템(collaborative filtering)에 대한 연구가 활발히 진행되고 있다. 이 추천 시스템의 구현 기법들 중에서 기존의 메모리 기반 알고리즘은 수행 시간에 대한 비용 부담이 크고 모델 기반 알고리즘은 연속성 데이터에 대한 처리가 어렵거나 불가능하다는 문제가 있다[12]. 본 논문에서는 특히 웹 사용자 모델에서 효과적인 연속성 피드백 데이터를 이용한 사용자 모델링(user modeling) 방법을 제안하고 이를 통해 각 사용자에 따라 선호되는 웹 페이지를 예측할 수 있는 시스템을 소개하였다. 논문에 사용된 연속성 데이터는 사용자의 웹 페이지 방문 시간(duration time)이고 이를 분석하기 위해서 기존의 모델 기반 알고리즘에 Support Vector Regression (SVR) 기법을 결합하는 알고리즘을 설계하였다. 실험에서는 제안 모델의 정확성과 예측 능력에 대하여 기존의 Pearson 알고리즘과 비교하였다. 논문에서 제안하는 방법이 매우 적은 시간비용을 요구하면서도 유의한 수준의 결과를 얻을 수 있음이 확인되었다. 인터넷 사용자들이 그들이 원하는 정보를 보는 시간이 전체 소비 시간 중 42%에 지나지 않고 전체 인터넷 사이트의 51%는 웹 페이지에서 제시하는 내용을 쉽게 알 수 없으며 90%이상이 적절치 못한 구조를 가지고 있다는 연구결과가 발표되었다[10]. 따라서 비효율적인 인터넷 상에서 사용자가 경제적으로 정보를 수집할 수 있는 방법이 필요하게 되었다[9]. Personalized Web은 이와 같은 문제를 해결하려는 연구 분야 중 하나로서 각종 정보로부터 사용자의 성향을 파악하고 이를 기반으로 웹 사이트를 적응, 변화시키며 서비스를 제공하는 방법이다. 즉, 이에 대한 연구는 해당 사이트로부터 효과적으로 사용자에게 적절한 정보를 제공하고자 하는 것이 일차 목표이고 사용자에게 특정 정보만을 추려서 제공함으로써 시스템의 부하를 줄이고 성능 향상을 추구하는 것이 두 번째 목표이다. 궁극적인 목표는 웹 사이트에서 사용자의 여러 반응을 분석하여 최적의 추천 시스템을 구축하는 것이다. 인터넷 환경에서 개별 사용자에 대하여 차별화된 웹 서비스를 제공해 주는 웹 개인화(web personalization)를 위한 연구 중에서 사용자 모델링(user modeling)은 사이트를 찾아온 사용자가 어떤 부류에 속하고 이용 패턴 및 전반적인 성향은 어떤지를 구체화하여 이를 시스템에서 이용할 수 있는 형태로 모델링하는 연구이다[21][23]. 추천 시스템의 사용자 모델링에서 중요한 요소 중 하나는 사용자로부터 얻어지는 피드백이다. 주어진 컨텐트에 대한 사용자의 반응으로부터 사용자의 성향을 파악하고 사용자에 맞는 상품, 정보, 페이지를 제공한다. 일반적으로 피드백은 명시적 피드백(explicit feedback)과 암시적 피드백(implicit feedback)으로 구분되며 명시적 피드백은 컨텐트, 상품 등에 대해 사용자로부터 직접 얻어지는 정보를 의미하고 암시적 피드백은 마우스의 움직임, 페이지에 머문 시간, 페이지간의 이동 등과 같이 사용자의 행동으로부터 간접적으로 관찰될 수 있는 정보를 말한다. 현재 구현되고 있는 대부분의 추천시스템은 명시적 피드백 중 사용자로부터의 등급평가 정보만을 이용하고 있으며, 이 경우 전체 사용자로부터 반응을 얻기가 어렵기 때문에 데이터의 희소성 문제를 유발한다. 등급평가 정보와 같은 이산적인 데이터가 아닌 많은 경우의 피드백들이 나타내는 연속성 데이터를 처리할 수 있는 방법이 현재 거의 없는 실정이다. 본 논문은 추천시스템에 적용될 수 있는 사용자 모델링을 구현에 있어서 어느 웹 사이트에서나 손쉽게 얻어질 수 있는 로그 데이터를 기반

으로 사용자로부터 연속성 피드백을 이용하여 최근 빠른 학습 속도와 비교적 높은 정확성으로 패턴인식 분야에서 많은 연구가 되고 있는 SVR을 사용한 모델링 방법을 제시하였고 객관적인 데이터를 통하여 그 성능을 실험하고 검증하였다. 2절에서는 제안하는 연속성 피드백 데이터 기반의 웹 페이지 예측 모형의 구축에 대한 이론적 배경에 대해 알아보고 비선형 회귀 모형을 통한 웹 페이지 예측 시스템에 대한 제안은 3절에서 소개하였다. 4절에서는 구현 및 실험결과를 알아보고, 마지막 5절에서는 결론 및 향후 연구에 대해서 논의하였다.

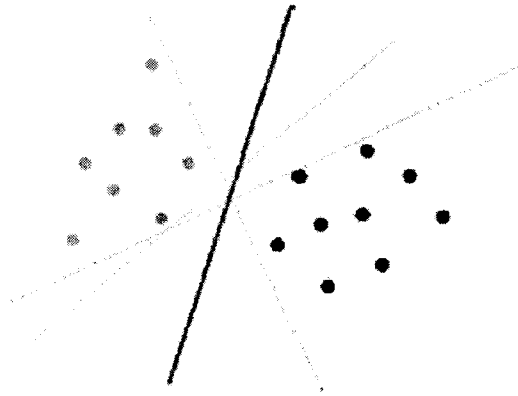
2. Support Vector Regression

2.1 SVM(Support Vector Machine) 구조

분류(classification) 문제에 있어서 Vapnik은 주어진 데이터들을 이분법적으로 나눌 수 있는 이상적인 선형평면을 구하는 방법을 제시하였다[16]. 입력 데이터, $(y_1, x_1), \ldots, (y_l, x_l)$에 대한 평면 방정식이 주어졌을 때 분류를 위한 함수식은 다음과 같이 비선형의 구조를 갖는다[1].

$$f(x, \alpha) = sign(\sum_{sv} y_i \alpha_i K(x, x_i) + b) \tag{2.1}$$

여기서 $K(\cdot)$는 커널(kernel) 함수이고 a와 b는 구해야 하는 모형의 모수(parameter)이다. 이 식을 만족하는 x들 중에서 최적 평면과 가장 가까운 것들을 Support Vector(SV)라고 한다. 이러한 SV를 이용하여 식 (2.1)의 부호에 의해 주어진 자료에 대한 분류가 이루어진다. 그림 1은 실제 문제 공간에서 이 평면과 방정식이 어떻게 표현되고 적용될 수 있는지 보여준다.



<그림 1> 최적 평면(Optimal Hyperplane)의 도식화 표현

위 그림에서 두 개의 집단을 가장 잘 분류하는 최적 평면은 중앙의 굵은 직선으로 표시되어 있다. 물론 두 집단을 분류하는 평면은 무수히 많이 존재하지만 최적 평면은 두 집단으로부터 동시

에 가장 멀리 떨어져 있는 평면을 표시하며 이런 성질을 만족하는 평면은 새로운 개체에 대한 분류도 다른 평면식에 비해 정확히 이루어지게 된다. 따라서 그림 1에서 중앙의 굵은 직선을 구해내는 것이 SVM의 최종 목표이다. 이러한 최적 평면은 각 개체들간의 폭(margin)을 최대로 하고 분류기로서의 몇 가지 조건들을 만족한다. 최적 평면은 다음의 식을 만족해야 한다. SVM의 최적 평면은 식 (2.2)을 만족해야 한다.

$$y_i\left(\sum_{i=1}^{l} y_i\alpha_i(x_i x)+b\right)\geq 1 \tag{2.2}$$

그리고 한 개체 x와 평면과의 거리는 다음과 같이 구한다.

$$R=\sum_{i=1}^{l}\alpha_i^{\sigma}\ ,\ \ \alpha_i\geq 0 \tag{2.3}$$

위 식에서 σ는 충분히 작은 값이다. 따라서 최적 평면은 위의 식 (2.2)의 조건을 만족하고 식 (2.3)을 최소화 하는 α와 b를 갖는다.

2.2 Kernel 함수

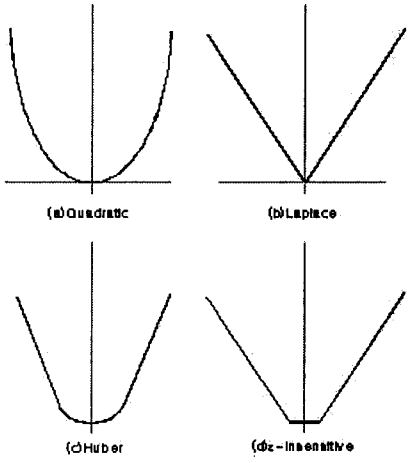
SVM은 최적 평면 방정식의 해를 찾는다는 특징과 함께 커널 함수를 이용하여 주어진 데이터를 다른 dot product 공간으로 표현하는 특징을 포함하고 있다. 입력 공간(Input Space)은 비선형 맵핑 함수 $\Phi: R^N \rightarrow F$ 를 사용하여 특징 공간(Feature space)로 전사(mapping)된다. 즉, 식 (2.4)와 같이 맵핑 함수들의 dot product로 나타낼 수 있다.

$$k(x,y)=(\Phi(x)\cdot\Phi(y)) \tag{2.4}$$

이 함수들을 통하여 문제를 선형분리가 가능하도록 유도한 후 앞서 기술하였던 최적 평면방정식을 도출함으로써 SVM은 비선형 분류기의 기능을 수행하게 된다.

2.3 Support Vector Regression(SVR)

SVM은 손실함수(Loss function)를 최적평면 방정식에서 사용하여 회귀 문제에 적용할 수 있다. 손실함수는 기대 값과 측정 값에 오차가 있을 경우, 오차를 어떻게 구하고자 하는 함수식에 반영시킬 것인가를 결정해 주는 함수이다. 일반적으로 손실 함수에는 4개의 대표적인 형태가 있다[16].



<그림 2> 4 Loss functions

그림 2의 (a)는 전통적인 least square error방식에 대응되는 2차(quadratic) 손실 함수이고, (b)는 (a)의 quadratic 함수보다 다소 덜 민감한 손실 함수이다. Huber는 (a)와(b)를 결합한 형태의 (c)의 robust 손실함수를 제안하였다[6]. 이 함수는 주어진 데이터의 분포가 알려지지 않았을 때 좋은 성능을 보인다. 그러나 위의 세가지 손실 함수는 희소데이터(sparse data)에 대해서는 적당치 않다. 반면 (d)에서 제시되는 ε -insensitive 손실 함수는 희소데이터 분포를 지닌 데이터들에 대해서 비교적 우수한 성능을 보인다. 본 논문의 데이터는 비교적 높은 희소성을 가지므로 (d)의 손실 함수를 이용하였다[15].

데이터 $D=\{(x_1, y_1), \ldots, (x_l, y_l)\}$, $x \in R^N$, $y \in R$ 가 다음의 직선식

$$f(x_i) = w \cdot x_i + b \tag{2.5}$$

로 ε -근사(approximating)하는 최선의 회귀 함수는 SVM 선형 분류기와 유사하게 다음 문제로 표현할 수 있다.

$$\begin{aligned} &\min imize \quad \frac{1}{2}\| w \|^2 \\ &subject\ to \quad y_i - w \cdot x_i - b \le \varepsilon \ , \quad w \cdot x_i + b - y_i \le \varepsilon \end{aligned} \tag{2.6}$$

이를 다시 구간 변수(slack variable) ξ 를 고려한 최적화 문제로 다음과 같이 나타낼 수 있다 [16].

$$\begin{aligned} &\textit{minimize} \quad \frac{1}{2}\| w \|^2 + C \sum_{i=1}^{l} (\xi_i^+ + \xi_i^-) \\ &\textit{subject to} \quad y_i - w \cdot x_i - b \le \varepsilon + \xi^+ \ , \ w \cdot x_i + b - y_i \le \varepsilon + \xi^- \ , \xi^+ \ , \ \xi^- \ge 0 \end{aligned} \tag{2.7}$$

위 식에서 C는 모형의 복잡성과 평활도에 대한 정도를 서로 보정해 주는 역할을 담당하는 계수이다. C값이 너무 크면 학습 자료에 지나치게 편향되는 경향이 있고, 반대로 이 값이 너무 작으면 단지 마진(margin)을 최대화하는 관점에서만 최적화 해주는 경향이 있다. 식 (2.7)에 ε -insensitive 손실함수인 식 (2.8)을 적용한다.

$$L_\varepsilon(y) = \begin{cases} 0 & \text{for} \ | f(x) - y | < \varepsilon \\ | f(x) - y | - \varepsilon & \textit{otherwise} \end{cases} \tag{2.8}$$

따라서 SVR의 해를 구하기 위하여 라그랑지(Lagrange) 함수를 사용하면 다음과 같은 식을 만들 수 있다.

$$\begin{aligned} &\max_{\alpha, \alpha^*} W(\alpha, \alpha^*) \\ &= \max_{\alpha, \alpha^*} \left(-\frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) x_i \cdot x_j + \sum_{i=1}^{l} \alpha_i (y_i - \varepsilon) - \alpha_i^* (y_i + \varepsilon) \right) \\ &\textit{subject to} \quad 0 \le \alpha_i, \alpha_i^* \le C \ , \ (i = 1, \ldots, l) \ , \ \sum_{i=1}^{l} (\alpha_i - \alpha_i^*) = 0 \end{aligned}$$

$$\begin{aligned} &\max_{\alpha, \alpha^*} W(\alpha, \alpha^*) \\ &= \max_{\alpha, \alpha^*} \left(-\frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) x_i \cdot x_j + \sum_{i=1}^{l} \alpha_i (y_i - \varepsilon) - \alpha_i^* (y_i + \varepsilon) \right) \\ &\textit{subject to} \quad 0 \le \alpha_i, \alpha_i^* \le C \ , \ (i = 1, \ldots, l) \ , \ \sum_{i=1}^{l} (\alpha_i - \alpha_i^*) = 0 \end{aligned} \tag{2.9}$$

따라서 식 (2.6)의 해는 식 (2.9)식의 과정을 거쳐 식 (2.10)과 같이 구해진다.

$$w = \sum_{i=1}^{l} (\alpha_i - \alpha_i^*) x_i \ , \ b = -\frac{1}{2} (w \cdot (x_r + x_s)) \tag{2.10}$$

이와같은 유도는 ε-insensitive 손실함수 뿐 아니라, 나머지 손실함수에도 적용할 수 있다. 웹 로그 데이터를 분석하는 본 논문에서는 회귀모델을 구축하는 데 있어서 데이터가 희소성을 보이므로 위의 ε -insensitive 함수를 사용하였다. SVR에서의 목표는 모든 학습 데이터에서 실제로 관측된 목표값으로부터 기껏해야 ε 만큼의 편이(deviation)를 갖는 함수를 찾는 것이다. 동시에 이러한 함수는 가능한한 편평한(flat) 구조를 갖도록 한다[13]. 위에서는 선형 회귀모형에 대한 해를 제시하였으나 2.2 절에서 언급한 커널함수(kernel function)를 이용하면 비선형 회귀모형(nonlinear

regression model)의 해도 구할 수 있다[8]. 다음절에서는 본 논문에서 제시하고 있는 웹 페이지 예측 모형이 적용되는 웹 로그 분석과 제안 기법과 비교되는 기존 방법에 대한 설명을 한다.

3. 웹 로그 분석

3.1 웹 마이닝(Web Mining)

Web은 그 양이 방대할 뿐만 아니라 웹 페이지들 간에 수많은 고리(link)들로 이루어진 구조를 가지므로 매순간 동적인 접근과 고리의 생성 소멸이 반복되고 있으며 이와 관련하여 부가적으로 발생되는 데이터 또한 방대한 양의 정보가 되어 저장되고 있다. Web mining은 이와 같이 많은 양의 내용(contents)정보와 연결 정보 들로부터 의미 있는 정보를 추출하고 새로운 정보를 창출하여 웹 페이지의 재설계, 특정 사용자에게 부합하는 정보의 제공 등의 일련의 과정들을 말한다. 정보의 거대성과 역동성(dynamics), 복잡성(complexity), 산만한 구조(diversity of communities), 그리고 희박한 연관성(lack of relevance and usefulness) 등 웹이 가지는 특성으로 인하여 웹 마이닝에서는 기존의 마이닝 작업에서 사용되었던 기법뿐만 아니라 새로운 방법들에 대한 연구가 많이 수행되고 있다. 이러한 웹 마이닝에 대한 연구는 마이닝을 수행하는 대상에 따라 일반적으로 크게 3가지, 내용 마이닝(content mining), 구조 마이닝(structural mining), 사용 마이닝(usage mining)으로 나눈다. 본 논문에서는 웹 사용 마이닝에 대한 적용 방법에 대한 연구를 수행한다.

3.2 웹 사용 마이닝(Web Usage Mining)

웹 사용 마이닝은 사용자들이 웹에 접근하면서 발생하는 직, 간접적 피드백이나 웹 로그를 대상으로 마이닝 작업을 수행하여 웹을 이용하는 사용자의 행동패턴 혹은 접근하는 웹 페이지의 경로들에 대한 유용한 정보를 얻어내는 과정이다. 웹 사용 마이닝은 일반적으로 다음의 과정들을 거쳐 수행되지만, 일부과정이 생략되거나 중복되는 과정을 거칠 수도 있다. 우선 가공되지 않은 거대한 웹 로그(large raw web log)로부터 유용한 정보들만을 추출하기 위하여 정제(cleaning), 압축(condensing), 변형(transforming)의 전처리(preprocessing) 작업이 이루어진다. 다음으로 가치가 있다고 판단되는 URL, 시간, 접속 주소, 웹 페이지 내용정보 등과 같은 feature들에 대하여 웹 로그 DB의 구축 혹은 다차원 OLAP 분석 등이 수행될 수 있다. 마지막으로 위의 과정들을 거친 웹 로그 레코드를 이용하여 마이닝을 수행하고 이로부터, 연관 관계, 순차적 패턴, 웹 접근 경향 등을 찾을 수 있다. 이와 같은 단계를 거쳐 생성된 결과는 시스템 측면에서 시스템의 성능분석, 웹 캐싱을 통한 시스템 구조 향상, 네트워크 트래픽의 완화 등에 이용될 수 있고 사용자측면에서는 적응형 웹 사이트의 구축[22]과 같이 사용자의 기호에 맞는 웹 페이지 추천, 사용자별로 특화 된 웹 사이트 제공 등에 적용될 수 있다.

3.3 협동 추천시스템(Collaborative Recommendation System)

협동 추천은 기본적으로 사용자들의 아이템에 대한 평가 정보를 기반으로 하여 특정 사용자의 특

정 아이템에 대한 유용성(utility) 혹은 선호도(preference)를 예측해 내는 것을 목적으로 하고 있다. 협동 추천(collaborative recommendation)과 협동 필터링(colloaborative filtering)은 그 목적에 있어서 전자는 사용자의 성향에 가장 부합하는 아이템을 제시하는 반면 후자는 사용자의 성향에 적합치 않은 아이템을 제거해 내고 나머지를 추천한다는 점에서 약간의 차이를 보이지만 대개의 경우 같은 용어로 혼용하여 사용되고 있다. 본 논문에서는 앞으로 이를 협동 추천(collaborative recommendation)으로 표현하도록 하겠다. 협동 추천에는 대표적으로 두 가지의 접근 방법이 있는데 주어진 전체 데이터베이스를 이용하여 직접 예측을 해내는 메모리 기반(memory-based) 접근 방식과 주어진 데이터베이스를 이용하여 학습 모델을 구축하고 이를 바탕으로 선호도를 근사추측하는 모델 기반(model based) 접근 방식이다. 이러한 협동 추천 시스템은 명시적 피드백에 기반하는지 암시적 피드백에 바탕을 두는지에 따라 구분되어지기도 한다. 명시적 피드백(explicit feedback)은 사용자가 이미 경험한 아이템에 의하여 직접 단위화 되어진 점수를 부여하는 방법으로서 이에 기반한 대표적인 시스템은 사용자의 정보를 바탕으로 영화를 추천해주는 GroupLens의 MovieLens 시스템이 있다[5]. 이에 반해 암시적 피드백(implicit feedback)은 사용자의 반응을 간접적으로 획득하여 이를 이용하는 방식으로서, 사용자의 브라우징 데이터, 구매 내역, 아이템에 대한 접근의 각종 패턴등이 이에 속한다. 이를 이용한 시스템 중에는 협동 추천 시스템의 경우는 아니지만 암시적 피드백을 얻기 위하여 시스템을 따로 구성하여 사용자 모델링을 수행하기도 하였다[15]. 이런 협동 추천시스템이 공통적으로 지니게 되는 결함 중 한 가지는 결측 자료(missing data) 문제이다. 우리는 대부분의 협동 추천 시스템에서 모든 아이템에 대하여 모든 사용자들의 평가를 기대하기는 어려우며 이것은 매우 현실적인 문제이다. 이 결측 자료에 대한 문제는 암시적 피드백보다 명시적 피드백 기반의 시스템에 있어 더욱 취약한 문제가 된다. 명시적 피드백을 이용하는 시스템의 추가되는 대표적인 문제점은 사용자에 의해 부여되는 점수를 이용하므로 시스템 자체도 이산데이터(discrete data)를 이용할 수밖에 없다는 점이며 연속성을 지닌 데이터(continuous data)를 다룰 수 있는 방법을 제시하지 못한다는 단점을 지니고 있다. 그러나 현실적으로 많은 응용프로그램에서 사용자에 의해 정확히 부여되는 이산데이터를 얻는 것은 매우 제한되어 있으며 오히려 시스템에 의해 주어지는 연속성 데이터를 사용해야 하는 경우가 많다. 사용자의 편의 측면에서 보았을 때에도 명시적 피드백을 얻는 시스템은 궁극적으로 바람직하지는 못하다고 할 수 있겠다. 다음에서는 위에서 언급한 메모리 기반 시스템의 알고리즘들과 모델 기반 시스템의 알고리즘에 대하여 설명한다.

3.3.1 메모리 기반 알고리즘(Memory based algorithm)

메모리 기반 알고리즘은 사용자들이 미리 평가, 부여한 선호도 점수(rating)의 데이터 베이스를 이용하여 특정 사용자의 특정 아이템에 대한 선호도를 직접 구해내는 방법이다[4]. 우선 다음은 이런 방식의 알고리즘들에서 기본적으로 고려하는 수식들이다.

$$\overline{v_i} = \frac{1}{I_i} \sum_{j \in I_i} v_{ij} \tag{3.1}$$

: 사용자 i의 평균 vote

$$P_{aj} = \overline{v_a} + k \sum_{i=1}^{n} w(a,i) \cdot (v_{ij} - \overline{v_i}) \tag{3.2}$$

: 아이템 j에 대한 현재 사용자에 대한 예측된 vote

위 식에서 $w(a,i)$는 현재 사용자 a 와 기존에 점수를 부여한 사용자 i 와의 연관도를 나타내는 가중치이다. 이 가중치를 부여 방식의 차이가 메모리 기반 알고리즘을 분류하는 기준이 된다. 본 논문에서는 대표적인 Pearson의 상관계수를 이용하는 알고리즘과 제안하는 SVR을 비교하였다. Pearson의 상관계수는 GroupLens 프로젝트에서 기반으로 사용되는 방법으로 다음 식과 같이 정의된다[11].

$$w(a,i) = \frac{\sum_j (v_{aj} - \overline{v_a})(v_{ij} - \overline{v_i})}{\sqrt{\sum_j (v_{aj} - \overline{v_a})^2} \sqrt{\sum_j (v_{ij} - \overline{v_i})^2}} \tag{3.3}$$

: j는 사용자 a와 i가 함께 vote한 아이템

즉, 사용자 a 와 i 의 유사도는 각 아이템과 사용자의 평균 rating의 편차들을 모두 합한 값을 평균 편차로 나누어 준 것으로 해석한다.

3.3.2 모델 기반 알고리즘(Model based algorithm)

확률적 관점에서 보면, 협동 추천이라는 작업은 기존의 부여된 점수 혹은 반응정도를 바탕으로 해당 사용자의 새로운 아이템에 관한 반응도를 예측해내는 작업으로 볼 수 있다[5]. 반응도(voting)를 양의 정수 0부터 m까지의 값을 갖는다고 보았을 때 해당 사용자 a 의 새로운 아이템 j 에 대한 반응도의 예측값을 다음 식에 의해 구할 수 있다.

$$P_{a,j} = E(v_{a,j}) = \sum_{k=0}^{m} \Pr(v_{a,j} = 1 \mid v_{a,k},\ k \in I_a) \tag{3.4}$$

이러한 확률구조에 기반하여 적절한 확률모델을 구축하고 이에 맞추어 예측값을 얻어내는 방법이 모델 기반 협동 추천 방법이다[14].

4. SVR을 이용한 웹 페이지 예측 모형

4.1 웹 페이지 예측 시스템 절차

본 논문의 전체 시스템은 5단계의 절차로 이루어진다. 로그 파일은 로그를 생성해 낸 서버의 종류와 웹 사이트의 성격 그리고 사이트 제공 주체에 따라 다양한 형태를 보인다. 최초의 로그 데이

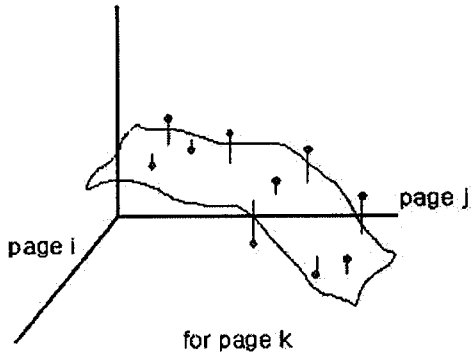
터는 전처리(preprocessing)과정을 통해 모델 구축에 필요한 데이터로 표현된다[11]. 정제된 로그 정보는 사용자가 방문한 페이지와 머문 시간을 하나의 인스턴스로 이용하고 적절한 수의 인스턴스를 추출하여 학습데이터로 사용하는 SVR 모델을 구축한다. 특히 이 모델은 모든 웹 페이지들에 대하여 개별적으로 작성된다. 다음으로 테스트 데이터의 각 페이지를 SVR 모델에 적용시켜 각 페이지에 대한 사용자의 선호도를 비교하여 모델에 대한 타당성을 조사한다. 테스트 데이터는 정제된 로그 파일에서 학습에 사용되지 않은 데이터로서 단순 임의 추출에 의해 구성된다. 마지막은 위 과정들을 통해 얻어진 각 페이지들에 대한 예측 값에 각 사용자의 평균 관심도와 각 페이지의 평균 관심도를 고려하여 선호도를 계산하여 우선순위가 높은 페이지를 추천하는 단계이다. 전체 시스템에 대한 평가 방법으로서 실제 주어진 선호도와 예측 선호도의 오차정도를 평균제곱오차(mean squared error: MSE)로 측정하였고 우선순위가 높은 페이지와 낮은 페이지를 추출하여 실제로 해당 페이지에 대하여 어떠한 반응을 보였는지를 측정함으로서 본 시스템의 성능을 평가하였다. 로그 파일의 전처리는 두 가지 가정을 전제하여 구축하였다. 첫째, 사용자는 자신이 관심이 높은 페이지에 대하여 더 많은 시간 동안 페이지에 머문다. 둘째, 페이지에 머문 시간과 페이지에 대한 선호도는 비례한다. 이러한 가정 하에 실제 전처리 과정에서는 로그의 많은 항목 가운데 사용자의 쿠키(cookie) ID와 요청페이지(request page), 요청날짜(request date), 요청시간(request time)을 선택하였고 이를 이용하여 다음과 같은 정보를 얻어내었다. 실제로 본 논문에서 사용한 로그 데이터는 CGI 서버를 사용하여 요청된 URL을 직접 페이지로 사용한다는 것이 무의미하여 각 페이지를 내용과 사이트의 구조에 맞추어 사전에 웹 서버에서 분류해 놓은 항목을 사용하였다. 그림 3은 전처리 과정을 마친 정제 데이터의 개념적 모습이다.

user i	page 1	page 2	page 3	···	page n-1	page n
		browsing duration for each page				

<그림 3> 정제된 로그(Refined Log) 데이터

4.2 SVR 학습 모형과 웹 페이지 선호도 예측

SVR 모형은 각 페이지에 대하여 구축하며, 해당 페이지를 제외한 나머지 페이지들에 대한 선호도를 축으로 하는 회귀 모형으로 작성되었다. 그림 4는 이를 개념적으로 도식화 한 것이다.



<그림 4> 웹 페이지 예측 모형

그림 4에서 각 점은 사용자 한 명을 나타내며 평면에 있는 축들은 이미 사용자가 지나온 페이지 브라우징 시간, 세로축은 해당 페이지에 대한 브라우징 시간이다. 따라서 그림은 해당 페이지를 제외한 다른 페이지의 브라우징 시간에 따라 해당 사용자의 해당 페이지에 관한 브라우징 시간을 예측하는 과정을 나타내고 있다. 그림에서의 평면은 학습된 회귀함수를 나타내며, 사용자를 나타내는 각 점은 학습 시에 사용된 인스턴스가 된다. 물론 테스트 데이터를 통한 예측 단계에서는 이 각 점의 세로축 값이 해당 페이지에 대한 선호도를 예측하는 근거이다. 이 때 대상으로 하고 있는 로그 데이터는 희소성의 특징을 갖는다. 이는 방문한 페이지 수보다는 방문하지 않은 페이지의 수가 압도적으로 많음을 의미하며, 따라서 모델을 구축하는 단계에서 안정된 성능의 보장을 위하여 방문하지 않은 페이지에 대한 반응(선호)시간을 다음과 같이 주었다.

$$x_{uk} = \frac{\mu_u + \mu_k}{2} \tag{4.1}$$

: k는 사용자 u에 의해 방문되어지지 않은 페이지

사용자 u가 웹 페이지에 평균 머문 시간(μ_u)과 페이지 k를 방문한 사용자들의 k에 대한 평균 브라우징 시간(μ_k)의 산술평균을 기본적으로 부여하였다. 테스트 데이터를 생성하여 각 페이지별로 구축된 SVR 모형에 입력하면 해당 페이지의 브라우징 시간에 대한 예측 값이 출력되게 된다. 이 예측값은 본 시스템에서 두 가지에 이용된다. 하나는 테스트 데이터의 실제 해당페이지 브라우징 시간과 비교를 위한 데이터로서 이용되고 다른 하나는 이를 다시 선호도로 변환하여 실제 사용자에게 추천하는 페이지를 선택하는 데 사용된다. 예측된 브라우징 시간을 선호도로 변환하기 위해서 페이지에 대한 특성과 각 사용자에 대한 특성이 식 (4.2)에 의해 반영되었다.

$$PREF_{uk} = \frac{\mu_u + \mu_k}{2} + \frac{(P_{uk} - \mu_u)}{\sigma_u} \cdot \frac{(P_{uk} - \mu_k)}{\sigma_k} \tag{4.2}$$

P_{uk} : 사용자를 위한 페이지의 예측된 브라우징 시간

위 식에서 σ_u, σ_k는 각각 사용자 u의 전체 아이템들에 의한 브리우징 시간의 표준 편차와 아이템 k에 대한 전체 사용자의 브리우징 시간의 표준편차이다.

5. 실험 및 결과

5.1 Data 및 전처리

본 실험에서 사용한 데이터는 KDD Cup 2000(Knowledge Discovery & Data miningi Cup)에서 문제로 주어졌던 로그로써 인터넷 쇼핑몰 Gazelle.com 의 2개월 간의 클릭 스트림 만을 모아 놓은 1.2GB의 텍스트 데이터이다[18]. 해당 쇼핑몰은 Leg-care 혹은 Leg-wear 제품을 전문적으로 판매하는 업체로서, 본 데이터는 이러한 인터넷 쇼핑몰의 로그라는 특성으로 인하여 비교적 방대한 양의 정보를 담고 있다. 구체적으로 한 개의 로그 정보는 217개의 attribute로 구성되어 있다. 구체적인 전처리는 유효 사용자 추출, 특성 attribute 데이터 추출, 시간 데이터 변환의 3가지 과정을 거쳤다. 우선 사용자의 구분은 Cookie ID를 이용하여 부여하였다. Cookie 정보는 서버가 클라이언트에 부여하는 고유한 정보로서 일반적으로 로그 분석을 할 때는 IP 주소를 부여할 수도 있으나, 본 논문에서 사용한 데이터는 공개되어진 데이터로서 개인의 신상에 관한 정보가 이미 생략되어져 있으므로 서버에 의하여 부여된 Cookie ID를 이용해 사용자를 구분하였다. 최근 많은 연구들에서 IP 주소보다 Cookie ID를 이용하는 경우가 많은 것도 사실이다. 추출된 Cookie ID와 접근한 page 정보를 이용하여 클릭 스트림의 길이가 10 미만인 사용자는 사용자의 패턴을 파악하는 것이 무의미하다고 판단하여 원래 데이터로부터 제외시킴으로써 대상이 되는 사용자 수를 한정시켰다. 웹 페이지의 URL은 CGI 서버에 의하여 생성되어 복잡한 구조와 형태를 띠고 있었다. 따라서 페이지를 나타내는데 적절한 정보라 보기 힘들어 각 페이지가 포함하는 쇼핑몰의 상품 종류와 웹의 구조에 따라 분류한 269개의 ASSORTMENT 정보를 웹 페이지로 대치하였다. 페이지 방문시간의 계산은 페이지 요청 처리(page request processing)시간부터 다음 클릭 스트림이 발생할 때까지의 시간 간격으로 계산하였고, 만약 다음 클릭 스트림이 해당 사용자가 아닐 경우에는 그 때까지 클릭 스트림에 대하여 계산된 방문 시간의 평균을 부여하였다. 한편 session이 다를 경우에는 같은 cookie ID를 지닌 사용자일지라도 별도의 사용자로 가정하고 시간을 계산하였다. 한편 상한 임계값은 1000sec으로 설정하였다. 이상의 과정을 통하여 정제되어진 데이터는 340000여 건의 클릭 스트림으로서 각각 Cookie ID, Assortment_ID, visit duration 컬럼들로 이루어져 있으며 표 1과 같이 정리된다.

표 1. Data set의 특성

Column	범위(개수)
cookie ID	1-13109(13109명)
assortment ID	0-268(269개)
visit duration	0-1000(연속)

위와 같이 정제 되어진 데이터를 사용하는데 있어, 모델 학습의 원활함과 타 실험과의 비교를 위하여 전체 데이터를 0부터 5사이의 수치로 Scaling을 수행하였다. 이는 이산 데이터를 다루는 많은 메모리 기반 혹은 모델 기반 협동 추천시스템들에서 사용하는 MOVIE 데이터가 0부터 5 까지의 이산적인 피드백을 바탕으로 구성하므로, 연구간의 비교를 위하여 위와 같이 설정하였다. 위의 정제된 데이터로부터 학습 데이터와 테스트 데이터를 추출하였다. 이 과정에서 6개의 특정 페이지의 경우 페이지에 접근한 사용자의 수가 300 이하가 됨으로써, 이 8개의 페이지와, 메인 화면을 나타내는 0번 ASSORTMENT 페이지의 경우 13000여 사용자 데이터 중 12500개 이상의 데이터에서 이용하여 모델링의 필요성이 낮아 이를 포함한 총 7개의 페이지가 제외된 260개 페이지에 대하여 데이터를 추출하였다. 생성시킨 인스턴스의 수는 각 페이지 별로 생성 가능한 데이터의 2/3를 무작위로 추출하여 학습데이터로 사용하고 나머지 1/3를 생성하여 학습과 성능 확인을 반복하여 3-fold validation을 수행하였다. 데이터가 1000개 이하일 경우에는 전체 생성 가능한 인스턴스 수의 80%를 학습데이터로, 20%를 테스트 데이터로 생성하여 5-fold validation을 수행하여 충분한 모델 형성이 가능하도록 유도하였다.

5.2 SVR 기반 협동추천시스템 모델구현

SVR 모형을 실제 구현함에 있어서 커널 함수는 RBF kernel을 이용하였고 실험의 반복을 통한 경험적 결과로 다음과 같이 parameter를 결정하였다.

표 2. SVR의 주요 parameter

	Gamma (kernel)	Epsilon (loss-function)	Cost (regression)
value	1/269	0.1	1

실험의 결과의 측정 메트릭은 두 가지를 사용하였다. 실험 1에서 사용된 메트릭은 SVR 모델의 유효성을 평가하기 위하여 SVR의 출력결과를 주어진 값과 비교하여 그 모델의 일반적인 정확성을 살피기 위해 일반적으로 널리 사용되는 MSE를 이용하였다. 실험 2에서 사용된 메트릭은 모델을 이용한 추천시스템의 성능을 측정하기 위하여 Ranking Rate를 사용하였다. Ranking Rate는 사용자의 만족 여부를 나타내는 척도로서 사용자 a의 페이지 k에 대한 만족 정도는 다음 식으로 정의한다.

$$R_{ak} = \frac{P_{ak}}{r_a} \cdot \frac{P_{ak}}{r_k} \tag{5.1}$$

식 (5.1)의 값이 1 이 넘는 경우에는 만족함을 1 미만일 경우는 그렇지 못함을 의미한다. 실험에서는 이를 이용하여 각 학습 데이터에 포함되지 않은 모든 페이지에 대하여 예측을 실시하고 Ranking rate를 계산하여 선호하는 페이지의 순위를 부여하게 된다.

5.3 실험 1 : 모델의 정확도와 시간비용

실험 1은 구성한 SVR기반의 연속성 데이터를 이용하는 웹 사용자 모델 자체의 성능을 MSE를 통해 Pearson의 상관계수 기법과 비교하여 보았다.

표 3. 모델의 정확성 비교

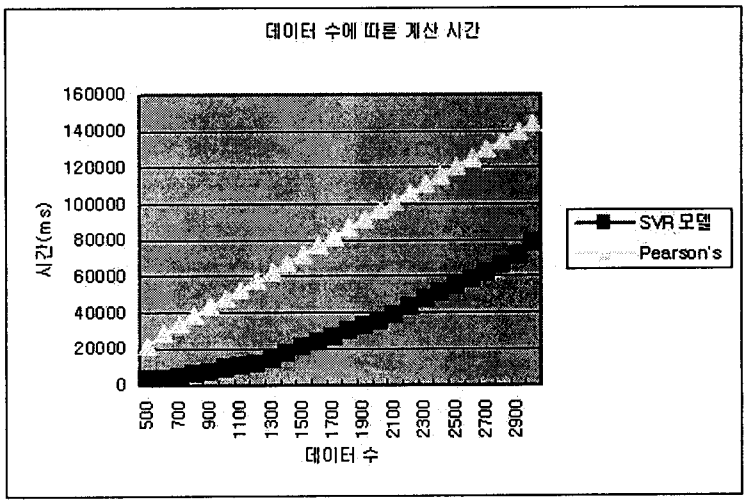
	SVR	Pearson
MSE(전체)	1.75	1.37
MSE(상위 50%)	1.19	1.01

위 표는 전체 261개 페이지 모델에 대하여 페이지 별로 각각 생성된 테스트 데이터에 대한 결과들의 평균 MSE값이다. 표에서 보여주듯이 전반적인 정확도에 있어 Pearson 알고리즘이 성능이 다소 높음을 볼 수 있다. 그러나 그 수행시간에 있어 Pearson 알고리즘은 SVR방식에 비해 매우 느림을 실험 수행 내내 경험적으로 체감할 수 있었으며, 다음 그림 5는 데이터 수에 따른 예측시간을 나타낸 것이다.

표 4. 데이터의 크기에 따른 계산 시간

데이터 수	SVR 모델	Pearson 모델	데이터수	SVR 모델	Pearson 모델
500	2941	21873	1800	29395	86617
600	3404	28872	1900	32143	91429
700	4111	33684	2000	35170	96241
800	5573	38496	2100	38640	101053
900	6921	43309	2200	42715	105865
1000	8890	48121	2300	47266	110677
1100	9910	52933	2400	50640	115489
1200	10986	57745	2500	54391	120302
1300	13984	62557	2600	57982	125114
1400	16830	67369	2700	61258	129926
1500	20631	72181	2800	65308	134738
1600	23775	76993	2900	70989	139550
1700	26166	81805	3000	77967	144362

표 4를 시각적으로 표현하면 그림 5와 같이 나타낼 수 있다.



<그림 5> 데이터 크기에 따른 계산 시간

그림에서 보듯이 SVR은 Pearson의 상관계수 기법에 비해서 탁월한 시간 감소효과를 보인다. 데이터의 크기가 1000 이하일 경우에는 정확도가 매우 떨어져 의미가 없긴 하지만 데이터 크기에 따른 두 기법간의 계산 시간을 측정하기 위하여 이 부분까지 포함하여 제시하였다. 특히 SVR 모델이 제시하는 계산 시간은 학습 시간까지 포함한 것이다. 실제 학습된 모델을 이용하여 선호도를 예측할 때는 실시간 내(평균 0.1sec이내)에 동작하였다. 한편 Pearson's 알고리즘의 시간은 모든 $w(a,u)$를 계산해 놓은 상태에서 특정페이지에 대한 해당사용자의 선호도 예측치를 구하는 시간만을 고려한 수치이다. 따라서 SVR 기반의 웹 사용자 모델링 방식은 학습시간을 제외할 경우 실시간 예측이 가능하였다.

5.4 실험 2 :웹 페이지 추천 시스템의 성능

실험 2는 본 논문이 제시하는 SVR 기반 사용자 모델링을 이용한 추천시스템의 성능을 평가하는 실험이다. 20 페이지 이상을 방문한 사용자 150명에 대하여 10개의 페이지는 이미 방문한 페이지로 설정하고 나머지 10개의 페이지에 대하여 예측한 선호도를 이용하여 Raking rate를 구하고 이것의 순서대로 예측 페이지의 30%인 3개 씩의 HIGH preference item과 LOW preference item을 선택한 후, 실제 데이터에서 보이는 선호도와 비교하였다. 그 결과는 다음 표와 같다.

표 5. Ranking rate를 이용한 웹 페이지 예측 시스템의 정확도

	SVR	Pearson
Pr(HIGH / HIGH)	0.31	0.35
Pr(LOW / LOW)	0.29	0.31
Pr(HIGH / LOW)	0.18	0.16
Pr(LOW / HIGH)	0.15	0.13

SVR은 이와 같이 Pearson 알고리즘과 비교 하였을 때 성능의 차이를 크게 보이지 않으면서도 빠른 시간에 학습과 예측을 할 수 있었고 학습시간을 제외한다면 실시간 예측이 가능하였다.

6. 결 론

본 논문은 SVR 기법을 이용하여 연속성 피드백을 이용하는 웹 페이지 사용자모델링 방법을 제안하였고 이를 웹 페이지 추천시스템에 적용하였다. SVR 기반의 사용자 모델링 방법의 가장 큰 장점은 기존의 모델기반 방식 협동추천시스템에서 제한되었던 연속성 피드백 정보를 다룰 수 있다는 것이다. 현재 대부분 추천시스템에서 사용되어졌던 이산성(discrete) 데이터는 사용자로부터 별도의 추가적 행위에 의해서만 획득할 수 있다는 점에서, 일반적인 사용자 모델을 구축함에 있어 한계를 보일 수 밖에 없었다. 특히 웹의 확장이 지속되고 그 체계가 복잡해지면서 웹 마이닝과 웹 개인화가 강조되었고, 이에 따라 웹 로그를 분석할 수 있는 기법들이 요구되는 현실을 감안한다면, 대부분이 연속성 데이터로 이루어진 웹 로그를 이용해 사용자 모델을 구축할 수 있는 본 논문의 SVR기반의 웹 사용자 모델링 기법은 매우 유용하다고 할 수 있겠다. 실험을 통하여 제안한 모델링 방법의 성능을 측정한 결과, 매우 간단한 피드백 정보인 방문시간만을 기초하여 모델링 하였음에도 불구하고, 모델링의 정확도에 있어 유의할만한 수준을 보여주었고, 기존의 Pearson 방식에 비하면 정확도는 다소 떨어지나 시간비용에 있어 성능이 매우 우수함을 알 수 있었다. 또한 웹 페이지 예측시스템으로 구현함에 있어서도, 그 예측력에 있어 마찬가지로 유의할 만한 성능을 보여 주었다. 그러나 높은 성능을 보장할 수 있는 시스템을 구축하기 위해서는, 우선 사용자의 선호도를 보다 잘 표현할 수 있는 데이터를 선택할 수 있는 연구가 필요하다고 하겠다. 또한 본질적으로 협동추천시스템이 안고 있는 결측 자료를 해결, 혹은 완화 할 수 있는 방안들이 결합되었을 때 본 모델도 더 나은 성능을 기대할 수 있을 것이다. 본 논문에서 제안한 웹 페이지 사용자 모델링 기법은 논문에서 사용된 웹 페이지 예측 시스템 뿐 아니라 개인화 된 웹 사이트의 구현, 인터넷 상거래에서의 추천 시스템, 사용자에 맞추어 편의를 제공하는 인터페이스 에이전트의 개발, 개인에게 특화된 정보검색 시스템 등 다양한 시스템 구축에 유용하게 사용될 수 있을 것이다.

참 고 문 헌

[1] Basu, C. et al., (1988), Recommendation as classification : Using Social and Content-based Information in Recommendation, Proceedings of the Workshop on Recommendation system. AAAI Press.

[2] Card, S et al., (2001), Information Scent as a Driver of Web Behavior Graphs: Results of a Protocol Analysis Method for Web Usability, KDDM01.

[3] Chakrabarti, S., (2000), Data mining for hypertext: A tutorial survey, SIGKDD Explorations 1, pp. 1-11.

[4] Fisher, D. et al., (2000), SWAMI: A Framework for Collaborative Filtering Algorithm Development and Evaluation, SIGIR 2000.

[5] Han, J and Kamber, M. (2001), Data Mining: Concepts and Techniques, Morgan Kaufmann Publishers, pp. 435-436.

[6] Huber, P., (1985), Projection pursuit, Ann. Stat. 13, pp. 435-475.

[7] Jon M., (1997), Authoritative Sources in a Hyperlinked Environment, Journal of the ACM.

[8] Kuhn, H. W. and Tucker, A. W. (1951), Nonlinear programming, In Proc. 2nd Berkeley symposium on Mathematiccal Statistics and Probabilistics, pp. 481-492.

[9] Lewis, D. (1991), Evaluating text caterorization, Proceedings of Speech and Natural Language Workshop.

[10] Recardo, B. Y. et. al., (1999), Modern Information Retrieval, ACM Press, pp. 6-8.

[11] Resnick, P et al., (1994), GroupLens : An Open Architecture for collaborative filtering of Netnews, Proceedings of ACM 1994 Conference on Computer Supported Cooperative Work.

[12] Salton and McGill (1983), Introduction to Modern Information Retrieval., McGraw-Hill.

[13] Smola, A. et al., (1996), Regression Estimation with Support Vector Learning Machines, Technical Report, ARPA and GNSF.

[14] Steady, W. et al., (1995), Recommending and Evaluating Choices in a Virtual Community of Use, In Proceedings of the CHI-95 Conference.

[15] Vapnik, V. N. et al., (1995), Support vector networks, Machine Learning 20, pp. 273-297.

[16] Vapnik, V. N. (1998), Statistical Learning Theory, Wiley, pp.445-448.

[17] Nature지, 제400호, (1999), pp. 107-109,

[18] www.ecn.purdue.edu/KDDCUP/, (2002년 2월 접속).

[19] educorner.com/courses/ia, (2002년 4월 접속).

[20] guir.cs.berkeley.edu/projects/swami, (2002년 3월 접속).

[21] www.csie.ntu.edu.tw/~cjlin/libsvm, (2002년 4월 접속).

[22] www.ittc.ukans.edu/obiwan, (2002년 3월 접속).

[23] www.i-biznet.com, (2002년 3월 접속).

[2002년 10월 접수, 2003년 3월 채택]