반복적 부스팅 학습을 이용한 문서 여과
(Text Filtering using Iterative Boosting Algorithms)

한 상 윤 † 장 병 탁 ††
(Sang-Youn Hahn) (Byoung-Tak Zang)

요 약 문서 여과 문제 (text filtering)는 어떤 문서가 특정한 주제에 속하는지의 여부를 판별하는 문제이다. 인터넷과 웹이 널리 퍼지고 이메일로 전송되는 문서의 양이 폭발적으로 증가함에 따라 문서 여과의 중요성도 따라서 증가하고 있는 추세이다. 이 논문에서는 새로운 학습 방법인 에이다부스트 학습 방법을 문서 여과 문제에 적용하여 기존의 방법들보다 우수한 분류 결과를 나타내는 문서 여과 시스템을 생성하고자 한다. 에이다 부스트는 간단한 가설의 집합을 생성하고 묶는 기법인데, 이 때 각각의 가설들은 문서가 특정 단어를 포함하고 있는지 검사하여 이에 따라 문서의 적합성을 판별한다. 먼저 최종 여과 시스템을 구성하는 각 가설의 출력이 1 또는 -1이 되는 이진 가설을 사용하는 기존의 에이다부스트 알고리즘에서 출발하여 좀 더 최근에 제안된 확신 정도 (실수값)를 출력하는 가설을 이용하는 에이다부스트 알고리즘을 적용함으로써 오류 감소 속도와 최종 오류율을 개선하고자 하였다. 또 각 데이타에 대한 초기 가중치를 연속 포아송 분포에 따라 임의로 부여하여 여러 번의 부스팅을 수행한 후 그 결과를 결합하는 방법을 사용함으로써 적은 학습 데이타로 인해 발생하는 과도학습의 문제를 완화하고자 하였다. 실험 데이타로는 TREC-8 필터링 트랙 데이타셋을 사용하였다. 이 데이타셋은 1992년도부터 1994년도 사이의 파이낸셜 타임스 기사로 이루어져 있다. 실험 결과, 실수값을 출력하는 가설을 사용했을 때 이진값을 갖는 가설을 사용했을 때 보다 좋은 결과를 보였고 임의 가중치를 사용하여 여러번 부스팅을 하는 방법이 더욱 향상된 성능을 나타내었다. 다른 TREC 참가자들과의 비교결과도 제시한다.

키워드 : 문서 여과, 에이다부스트, 반복적 부스팅 학습, TREC.

Abstract *Text filtering is a task of deciding whether a document has relevance to a specified topic.* As Internet and Web becomes wide-spread and the number of documents delivered by e-mail explosively grows the importance of text filtering increases as well. The aim of this paper is to improve the accuracy of text filtering systems by using machine learning techniques. We apply AdaBoost algorithms to the filtering task. An AdaBoost algorithm generates and combines a series of simple hypotheses. Each of the hypotheses decides the relevance of a document to a topic on the basis of whether or not the document includes a certain word. We begin with an existing AdaBoost algorithm which uses weak hypotheses with their output of 1 or -1. Then we extend the algorithm to use weak hypotheses with real-valued outputs which was proposed recently to improve error reduction rates and final filtering performance. Next, we attempt to achieve further improvement in the AdaBoost's performance by first setting weights randomly according to the continuous Poisson distribution, executing AdaBoost, repeating these steps several times, and then combining all the hypotheses learned. This has the effect of mitigating the overfitting problem which may occur when learning from a small number of data. Experiments have been performed on the real document collections used in TREC-8, a well-established text retrieval contest. This dataset includes Financial Times articles from 1992 to 1994. The experimental results show that AdaBoost with real-valued hypotheses outperforms AdaBoost with binary-valued hypotheses, and that AdaBoost iterated with

† 비 회 원 : University of Washingten 전산학
syhahn@u.washingten.edu
†† 종신회원 : 서울대학교 컴퓨터공학부 교수
btzhang@cse.snu.ac.kr

문 접 수 : 2000년 3월 24일
심사완료 : 2002년 1월 17일

random weights further improves filtering accuracy. Comparison results of all the participants of the TREC-8 filtering task are also provided.

Key words : Text Filtering, AdaBoost, Iterative Boosting Algorithms, TREC.

1. 서 론

문서 여과 문제는 어떤 문서가 특정한 사용자의 관심 분야에 관련이 있는지를 결정하는 이진분류 문제이다. 문서 여과를 위한 학습 시스템에서는 문서와 그 문서에 대한 특정 사용자의 관심 여부 (1 또는 -1)로 이루어진 순서쌍들의 집합을 학습 데이타로 하여 규칙을 찾아내고 이를 새로운 문서에 적용하여 그 문서가 사용자의 관심사에 속하는지를 판별하는 시스템을 생성한다. 학습된 문서 여과 시스템의 성능을 평가하는 척도에 대한 많은 연구가 진행되어 왔는데 특히 1992년 처음 개최되어 매년 열리고 있는 TREC(Text REtrieval Conference)에서 데이타의 수집과 적절한 평가 척도의 개발, 보급에 노력하여 현재 여과 시스템들에 대한 어느 정도 객관적인 비교가 가능하게 되었다[1].

문서 여과 문제에 대한 연구는 정보 검색 분야에서 뿐 만 아니라 기계 학습 분야에서도 중요한 응용 문제로 다루어지고 있다. 이들 중 최근 기계 학습 분야에서 많은 관심을 받고 있는 부스팅(boosting) 학습 방법을 문서 여과 문제에 적용한 연구에서는 매우 단순한 규칙을 결합하여 이해와 구현이 쉬우면서도 탁월한 성능을 나타내는 학습 방법을 제시하고 있다[9]. 부스팅 학습 방법은 재표본 추출에 의존하는 배깅 (bagging)과 함께 위원회 결정 모델 (decision-committee model)의 일종이다. 에이다부스트 학습 방법은 분류가 어려운 데이타들을 보다 잘 분류할 수 있도록 데이타 분포를 변경하고 변경된 분포로부터 가설을 학습하여 결합하는 방법이다. 많은 응용 분야에서 우수한 성능을 보이는 것으로 알려져 있으나 학습 데이타의 크기가 작은 경우 과다학습(overfitting) 현상이 발생할 수 있고, 학습 속도가 느리다는 문제를 갖는다.

이 논문에서는 기존의 문서 여과 문제에 적용한 에이다부스트 학습 알고리즘을 개선하여 이를 표준 데이타 집합에 적용함으로써 분류 시스템의 성능을 향상하고 이를 객관적 평가 척도에 따라 다른 분류 방법들과 비교하고자 한다. 이를 위해 먼저 2절에서 위원회 결정 알고리즘들과 관련된 이전의 연구를 살펴보고 3절에서는 문서 여과 문제에 대한 정의와 문서 여과 시스템의 성능 평가 방법에 대해 살펴보기로 한다. 4절에서는 이 논문에서 사용하는 학습 방법을 자세히 설명하고, 5절에서는 실험에 사용한 데이타에 대한 설명과 실험 방법 및 실험 결과를 상술하며, 마지막 절에서는 지금까지의 결과를 요약하고 향후 과제에 대해 토론하고자 한다.

2. 부스팅 및 배깅 학습 방법

위원회 결정 모델은 약 학습 (weak learning) 모델에 기초하고 있다. 약 학습 모델은 Valiant의 PAC (Probably Approximately Correct) 학습 모델에 비해 오류 상한 조건을 완화하여 임의로 추측한 경우보다 약간 좋은 정확도(이진 분류의 경우 1/2)를 갖는 가설을 여러개 학습한 후 이를 결합함으로써 임의의 작은 오류 상한을 갖도록 학습할 수 있게 한다[6]. Schapire는 1990년 새로운 학습 알고리즘을 제안하여 이를 입증하였는데 이것이 최초의 부스팅 알고리즘이다[11]. 이 방법은 여과 방식의 부스팅 방법으로 다른 부스팅 방법에 비해 구현시 메모리를 적게 사용하는 이점이 있으나 학습에 많은 데이타가 필요하다는 단점을 갖는다. 에이다부스트 학습 알고리즘은 이를 개선한 새로운 부스팅 학습 방법으로[5] 이 방법에서는 학습된 가설의 오류율에 따라 원래의 데이타로부터 재표본 추출을 수행하여 새로운 분포를 갖는 데이타 집합을 생성한 후 이를 사용하여 다시 새로운 가설을 학습하고 학습된 가설들로 위원회를 구성한다. 데이타 집합에 대한 재표본 추출은 가중치의 조절을 통해 시뮬레이트 되기도 하는데 초기 데이타 분포는 균일(uniform) 분포, 즉 m이 학습 데이타의 개수라면 각 데이타에 대한 가중치가 $\frac{1}{m}$이라고 가정한다. 이러한 분포를 사용하여 문서를 재표본 추출 한 후 생성된 문서에서 다시 가설을 학습하고 가설의 오류에 따라 가설이 옳게 예측한 데이타의 가중치는 줄이고 잘못 예측한 데이타의 가중치는 증가시킴으로써 분류하기 어려운 데이타에 집중하도록 한다.

$$\alpha = \frac{1}{2}\ln\left(\frac{1-\varepsilon}{\varepsilon}\right), \varepsilon = \text{error rate}$$

일 때, 최종적으로 모든 학습된 가설들간의 투표는 가중치 α를 곱하여 합해지는데 이것은 적은 오류를 갖는 가설의 예측이 전체 시스템의 결정에 보다 중요한 영향을 미치게 하는 역할을 한다. 에이다부스트는 실제에 있어 이론적 오류 상한 값 보다 적은 학습 오류를 나타내며 일반화 오류 역시 학습 오류와 함께 감소하는 경향

을 보인다. 이러한 경향에 대한 설명으로 Schapire는 에이다부스트 알고리즘이 최종 분류기의 경계 폭(margin)을 최대화한다는 점을 들고 있다 [8]. 분류 시스템의 오류는 바이어스와 분산의 합으로 나누어 생각할 수 있는데, 바이어스를 주로 하여 [11] 바이어스와 분산을 모두 줄여주는 것으로 알려져 있다[4][8].

약 학습 모델에 기반한 다른 학습 방법으로 배깅(bagging)이 있는데, 이는 임의로 학습 데이타 집합에 대한 재표본 추출을 수행하여 생성된 데이타 집합으로부터 각 가설을 학습하는 방법이다. 즉 배깅에서는 m 개의 데이타를 갖는 집합으로부터 임의 복원 추출을 통해 새로운 분포의 데이타 집합을 만들고 이 데이타 집합을 사용하여 새로운 가설을 학습한 후 이와 같은 과정을 반복하여 생성된 가설들을 결합함으로써 예측을 위한 위원회를 구성한다[3]. 가설을 학습하는 시스템은 데이타의 분포에 민감한 불안정한 (instable) 방법이 좋다[3]. 주로 분산을 줄이는데 [4] 그 감소 정도가 에이다부스트를 능가하고 있다[2]. 또 에이다부스트와 배깅의 장점을 결합하여 분류 성능을 향상하려는 연구도 발표되었는데, 멀티부스팅 (MultiBoosting)이라고 하는 이 방법에서는 에이다부스트 방법을 사용하여 소위원회를 구성하고 각 소위원회마다 재표본 추출을 하여 새로운 데이타 분포에 대한 학습을 수행한 후 학습된 가설들간의 투표를 통해 분류를 수행한다 [11].

3. 문서 여과 문제

문서 여과 시스템은 주어진 문서의 집합과 각 문서에 대한 주제와의 관련성 여부를 표시하는 값(1 또는 -1)으로부터 규칙을 발견하여 새로운 문서에 대해 주제와의 관련성 여부를 판단해 주는 시스템이다. 문서 여과 시스템의 입력으로는 문서를 기술하는 속성들의 벡터인 X와 그 문서가 사용자의 관심 분야에 속하는지 관련성 여부를 나타내는 y의 순서쌍 집합이 제공되고 시스템은 문서와 관련성 여부 사이의 규칙을 자동적으로 학습하여 새로운 문서에 대한 관련성 여부를 결정하는 규칙을 생성하는 시스템이다. 이 시스템의 목표는 생성된 규칙의 분류 오류 (ε)가 최소가 되도록 하는 것이다. 문서 x_i와 이 문서와 사용자의 관심 분야와의 관련성 여부를 나타내는 $y_i \in \{-1, 1\}$가 주어지면

$$\varepsilon = \frac{1}{m} \sum_{i:C(x_i) \neq y_i} 1$$

문서 여과 시스템의 성능을 평가하는 방법으로는 오류(error), 회수율(recall), 정확도(precision), 채산점(break-even points), F1척도, 유틸리티척도(linear utility) 등이 있다 [7]. 이들 중 이 논문에서는 오류와 TREC에서 사용하는 LF1 유틸리티척도를 사용한다. 오류는 최종 분류 규칙들간의 위에서 제시한 ε값을 비교하는 것으로, 주제와 관련된 문서가 적은 경우 모든 문서를 -1로 예측하면 작은 오류를 얻게 되는 문제를 갖는다. LF1 유틸리티 척도는

$$\text{LF1} = 3R_+ - 2N_+$$

로 정의되는데 여기서 R+는 관련된 것으로 옳게 예측한 문서 수이고 N+는 관련 있다고 잘못 예측한 문서 수이다. 이 척도에서는 여과 시스템이 관련 있다고 예측한 문서가 실제로 관련성을 갖을 경우 큰 점수를 주고 이 예측이 틀린 경우에는 이보다 작은 벌점을 주고 있다. 여과 시스템은 LF1이 최대가 되게 하는 규칙을 생성함으로써 분류 오류를 최소화할 수 있으며 모든 문서에 대해 관련이 없다고 예측함으로써 작은 오류를 나타내는 시스템보다 관련성 있는 문서를 정확히 예측하는 규칙을 선호하게 한다.

4. 연구 방법

이 논문에서 사용하고 있는 에이다부스트는 Schapire가 문서 여과 문제에 적용한 알고리즘에 기초하고 있다[9]. 이 알고리즘에서는 어떤 단어 (또는 어구)가 문서에 포함되어 있는지의 여부를 가설 (규칙)로 사용한다. 예를 들어 어떤 가설은 "금감원"이라는 단어를 이용하여 문서에 "금감원"이 출현하였을 경우 적합하다고 판정하고 그렇지 않을 경우 부적합하다고 판단한다. 마찬가지로 다른 가설들은 다른 단어들을 이용하여 문서를 분류할 수 있다. 이렇게 하였을 때 비록 각각의 가설들의 정확도는 좋지 않지만, 이러한 가설들을 여러 개 선택하여 체계적인 방법으로 결합함으로써 하나의 높은 정확도를 갖는 가설을 생성할 수 있다. 이 때 학습된 단순한 가설을 약 가설이라 하고 여러 개의 약 가설을 결합한 것을 최종 가설이라고 한다. 이 알고리즘에서 약가설은 1 또는 -1의 이진값을 출력하고 각 가설에 대한 가중치는

$$\alpha = \frac{1}{2} \ln\left(\frac{1-\varepsilon}{\varepsilon}\right)$$

로 설정한다. 에이다 부스트의 일반적인 형태는 표 1에 나온 바와 같다.

이 방법은 다른 학습 방법들에 비해 좋은 성능을 보이지만 분류 오류를 충분히 감소시키는데 많은 시간이 걸리는 문제가 있다. 따라서 이 논문에서는 실수 값을 출력하는 가설을 이용한 에이다부스트 알고리즘을 적용하여 오류 감소 속도를 향상하고자 하였다.

Schapire와 Singer는 학습된 약 가설이 실수 값을 가질 수 있도록 에이다부스트 알고리즘을 확장하고 있다 [10]. 즉 가설 $h:X\rightarrow R$가 확신 정도를 나타내는 실수값을 출력하여 그 부호가 양수이면 1, 음수이면 -1을, $|h|$는 이 예측에 대한 확신 정도를 나타내도록 한다. Schapire et al.은 문서 x_i와 주제와의 관련성 여부를 나타내는 y_i가 주어진 경우, 학습 오류가 Z에 의해서 바운드 되어 있음을 밝혔다.

$$\frac{1}{m}\sum_i |i:H(x_i)\neq y_i| \leq \prod_{t=1}^{T} Z_t$$

$$|\pi| = 1 \text{ if } \pi = true \text{ ,otherwise } 0$$

그러므로,

$$Z_s = \sum_i D_t(i)\exp(-\alpha_t y_i h_t(x_i)) \quad (1)$$

값을 최소화하도록 가설을 선택함으로써 오류를 줄일 수 있다 [10]. Z_t, $H(\cdot)$, 그리고, $D_t(\cdot)$에 대해서는 표 1을 참조하면 된다. 이를 위해 먼저 α를 h_t에 포함시켜 h_t가 스케일링이 가능하다고 본다. 또한 어떤 단어가 포함된 문서의 집합을 X_1, 포함되어 있지 않은 문서들의 집합을 X_0으로 표시하고,

$$x\in X_j, j\in\{0,1\}$$

이에 대해

$$c_j = h(x) \text{ for } x\in X_j$$

로 정의한다. 이 때 c_j의 부호는 문서에 단어가 포함되어 있는지의 여부에 따라 주제와의 관련성 여부를 가리킨다. 즉 어떤 단어가 포함되어 있는 문서 중 주제와 관련된 문서의 수가 많으면 c_1의 값이 양수가 되고, 주제와 관련 없는 문서의 수가 많으면 음수가 된다. 마찬가지로 어떤 단어가 포함되어 있지 않은 문서 중 주제와 관련된 문서의 수가 많으면 c_0의 값이 양수가 된다. $|c_j|$는 이렇게 판단할 때의 확신 정도를 나타내게 된다. $|c_j|$를 계산하기 위하여 먼저 어떤 단어가 문서에 포함되어 있는가의 여부, 그 문서가 주제와 관련이 있는가의 여부에 따라 결정되는 문서 집합 크기의 전체 문서에 대한 비율을 나타내는

$$w_b^j = \sum_{i:x_i\in X_j \wedge y_i = b} D(i)$$

를 정의한자. 여기서 b는 문서가 주제와 관련성을 갖는지 여부를 나타내는 값으로 +또는 −로 표시한다. 이 때 식 (1)에서 α는 1이므로 무시하고 앞의 식을 넣으면 다음과 같이 표시된다.

$$Z = \sum_j \sum_{i:x_i\in X_j} D(i)\exp(-y_i c_j) \quad (2)$$

$$= \sum_j \left(w_+^j e^{-c_j} + w_-^j e^{c_j}\right) \quad (3)$$

이 때 이 식을 최소화하는 c_j는

표 1 실수값을 출력하는 가설을 이용한 에이다부스트

입력: 학습데이타 $\langle (x_1,y_1),\cdots,(x_m,y_m)\rangle$
$x_i\in X, y_i\in\{-1,1\}$
반복 횟수: T

데이타에 대한 가중치를 초기화한다.

$$D_1(i) = \frac{1}{m}$$

For s=1 to T
1. D_s를 사용하여 약 가설 $h_s: X\rightarrow R$를 학습
2. $\alpha_s\in R$를 선택
3. 다음과 같이 데이타 분포를 갱신

$$D_{s+1}(i) = \frac{D_s(i)}{Z_s}\exp(-\alpha_s y_i h_s(x_i))$$

여기서 Z_s는 데이타 분포를 정규화하기 위한 값이다.
다음과 같은 최종 가설을 출력한다.

$$H(x) = \text{sign}\left(\sum_{s=1}^{T}\alpha_s h_s(x)\right)$$

$$c_j = \frac{1}{2}\ln\left(\frac{w_+^j}{w_-^j}\right)$$

가 되고, Z_s는

$$Z_s = 2\sum_j \sqrt{(w_+^j w_-^j)}$$

가 된다. 여기서 Z_s를 최소화하는 분할이 되도록 단어를 선택하여 새로운 weak learner를 찾을 수 있다. 이 때 c_j는 각 분할에 대한 신뢰정도를 나타낸다. 일반화 오류를 줄이기 위하여 Z_s와 c_j의 계산에 Smoothing 방법을 적용하였다 [10]. 이 때 Z_s와 c_j는 각각 다음과 같이 표시된다.

$$c_j = \frac{1}{2}\ln\left(\frac{w_+^j+\varepsilon}{w_-^j+\varepsilon}\right)$$

$$Z_s \leq 2\sum_j \sqrt{(w_+^j w_-^j)} + \sqrt{2N\varepsilon}$$

확신도를 가지는 에이다부스트 알고리즘을 사용할 경우 기존의 방법과 거의 같은 계산량을 가지면서도 빠른 수렴속도와 개선된 예측 성능을 나타낸다. 특히 학습 데이타의 수가 많은 경우 학습에 필요한 시간을 많이 줄일 수 있으며 분류 성능도 상당히 좋아지는 것을 볼 수 있다. 그러나 학습 데이타의 수가 적은 경우에는 과도학습 문제가 발생하여 일반화 성능면에서 문제가 생긴다. 이러한 문제를 완화하기 위해 표 2의 방법을 에이다부스트 알고리즘에 추가하였다. 이 알고리즘에서는 데이타 분포의 초기화를 달리하여 에이다부스트를 여러번 수행한다. 최초의 데이타 분포 $D_1(i)$는 에이다부스트와 같이 균일하게 $\frac{1}{m}$ 값을 부여한다. 정해진 라운드 수만큼 에이다부스트를 수행한 후 다시 D를 초기화할 때 연속

표 2 반복적 에이다부스트

입력: 에이다부스트 알고리즘
에이다부스트의 라운드 수: T
에이다부스트 수행 횟수 : U
1. T 라운드만큼 에이다부스트를 수행하여 약 가설들을 생성한다.
2. 데이타 분포 D를 연속 포아송 분포에 따라 임의로 초기화한 후 합이 1이 되도록 정규화한다.
3. 1.,2의 과정을 U번 반복한다.
4. 학습된 모든 가설들간의 가중치가 부여된 투표를 실시하는 최종 가설을 출력한다.

포아송 분포를 사용하는데 이러한 분포를 갖는 가중치는 다음과 같이 생성한다.

$$poisson() = -\log\left(\frac{random(1,\cdots,999)}{1000}\right)$$

여기서 $random(\min,\cdots,\max)$는 $\min$과 $\max$사이의 임의의 값을 반환하는 함수이다. 이 알고리즘에서 에이다부스트의 수행 라운드 수와 몇번 데이타 분포를 초기화하여 에이다부스트를 수행할 것인가의 문제는 실험을 통해 결정하였다.

이 알고리즘은 2절에서 언급한 멀티부스트 알고리즘의 가중치 재 초기화 방법을 단순화한 것으로 TREC과 같이 학습 데이타의 수가 작은 데이타 집합에 대해 에이다부스트의 과도학습 문제를 완화하면서 일반화 오류를 줄이기 위해 도입한 것이다. 위와 같이 데이타 분포를 임의로 여러 번 초기화하여 부스팅을 수행함으로써 데이타 분포에 따른 분산을 줄이게 되어 일반화 오류가 줄어들 수 있다.

5. 실험 및 평가

제안된 부스팅 학습 기법의 성능을 평가하기 위해 이 논문에서는 두 가지 실험을 수행하였다. 먼저 기존의 에이다부스트 알고리즘과 실수값을 출력하는 가설을 이용한 에이다부스트 방법의 오류를 비교하는 실험을 Reuters 데이타에 대해 수행하였다. 데이타는 10개의 주제에 대한 관련성 여부가 표시된 문서들의 집합으로 8574개의 항을 갖는 8762개의 학습 데이타, 3009개의 검증 데이타를 사용하였다. 그림 1에서는 주제 acq에 대한 AdaBoost-tf (기존의 알고리즘)와 real-AdaBoost (실수 가설 알고리즘)의 학습오류와 검증 오류를 보여주고 있다. AdaBoost-tf의 경우 10,000개, real-AdaBoost의 경우는 1,000개의 약 가설을 학습하였는데, 최종 오류에 있어서는 real-AdaBoost가 학습 오류와 검증 오류모두 작은 값을 보이고 있다. 또 검증 오류의 수렴까

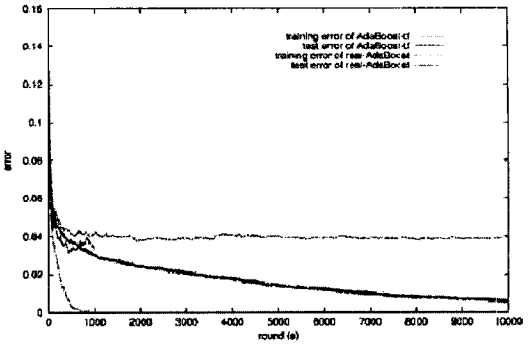

그림 1 AdaBoost-tf와 real-AdaBoost의 학습 데이타와 검증 데이타에 대한 오류, 여기서 AdaBoost-tf는 기존의 문서 여과를 위한 에이다부스트 알고리즘이고, real-AdaBoost는 실수값을 출력하는 가설을 이용한 에이다부스트 알고리즘이다.

표 3 real-AdaBoost와 i-AdaBoost의 LF1 측정치

주제	real-AdaBoost		i-AdaBoost	
	라운드 수	LF1	라운드 수	LF1
351	25	63	62	12
352	21	20	25	63
353	9	13	21	20
354	16	0	68	14
355	9	1	16	0
356	3	34	9	1
357	1	0	3	34
358	1	0	1	0
359	2	1	1	0
360	1	0	2	1
...				
385	1	0	104	22
386	18	0	18	0
387	1	0	1	0
388	1	0	1	0
389	69	50	63	50
390	1	0	1	0
391	3	83	3	83
392	2	9	2	9
393	1	0	1	0
394	1	0	1	0
395	16	3	270	16
396	1	0	1	0
397	2	0	2	0
398	1	0	154	7
399	1	0	1	0
400	9	-10	9	-10
total		374		454

지 AdaBoost-tf는 4,000여회 이상, real-AdaBoost의 경우 430 여 회의 학습이 진행되는 것으로 보아 약 10 배 정도의 학습 시간 단축 효과를 얻고 있다. 이러한 성능을 나타내는 것은 real-AdaBoost가 약 가설의 예측에 대한 확신 정도를 출력하고 이를 데이타분포 갱신과 다음 번 가설학습에 사용함으로써 AdaBoost-tf보다 정확한 오류조절이 가능하기 때문이다. 또, real-Ada Boost에서는 문서의 단어가 포함되어 있지 않을 경우도 고려하기 때문에 빠른 학습속도를 보이는 것이다.

두 번째 실험에서는 TREC-8 문서 여과 과제의 규정에 입각하여 real-AdaBoost와 i-AdaBoost (반복적 에이다부스트)을 사용한 실험을 수행하였다. 데이타로는 TREC-8에서 사용한 351~400의 50개 주제에 속하는 1992(학습데이타)년, 1993~1994(검증데이타)년 Financial Times 기사 모음을 사용하였다. 그리고, "the", "this"와 같은 불용어(stop-list)를 제거하고, stemming을 수행하였으며, 문서 빈도에 따라 3060개의 단어 추출 등의 전처리를 하였다. 이 데이타는 관련성여부, 특히 관련 있는 것으로 표시된 문서가 매우 적다는 특성을 갖는다.

이 실험에서 real-AdaBoost는 각 주제에 대해 100개의 가설을 학습시킨 후, 검증 데이타에 대해 얻어진 최고의 LF1값들을 모두 합하여 최종 결과로 삼았다. i-AdaBoost는 서로 다른 데이타 분포에 대해 6회의 에이다부스트를 각각 50라운드씩 수행한 후 최고의 LF1 값들을 합해서 총점을 계산하였다.

표 3의 일부 주제에 대한 최고 LF1값과 이 값을 얻은 라운드 수를 표시하였다. i-AdaBoost가 real-AdaBoost에 비해 높은 LF1의 총점 수를 얻게 하는 것은 에이다부스트의 기본적 성능을 유지하면서 서로 다른 데이타 분포에 대해 에이다부스트를 여러 번 수행함으로써 과도학습 상태에서 탈출하여 향상된 LF1값을 얻을 수 있게 되기 때문이다. 그러나 모든 주제에 대해 이러한 효과가 나타나는 것은 아니고, 관련 있는 문서가 비교적 많은 일부 주제에서만 나타난다. 다음으로 표 4에서는 TREC-8 여과 과제에 참가한 7개 팀 중 상

표 4 TREC-8 참가자들과 i-AdaBoost의 LF1 비교

주제	i-AdaBoost	Plt8f1	Plt8f2	CL99bfl1	pirc9BF1	Scai8Ft	Maximal
351	12	22	25	20	25	0	51
352	63	82	115	12	130	149	570
353	20	32	42	4	28	0	87
354	14	-9	-9	-2	-7	0	147
355	0	0	0	0	0	0	3
356	1	7	7	-5	4	-2	45
357	34	47	41	34	59	3	183
358	0	0	0	0	0	0	6
359	0	-5	-5	0	0	-8	90
...							
383	84	97	113	-2	72	-2	201
384	0	0	0	0	0	0	9
385	22	9	14	24	28	0	81
386	0	0	0	0	-2	0	18
387	0	-10	-10	-6	-1	0	27
388	0	0	0	-3	0	0	42
389	50	145	113	122	76	218	387
390	0	-4	-4	0	0	-4	204
391	83	36	-18	57	-35	-138	381
392	9	-32	-32	-5	2	4	96
393	0	7	7	-2	1	0	15
394	0	0	0	-8	0	0	15
395	16	-8	-1	-6	-50	-9	186
396	0	1	1	-6	4	0	15
397	0	-2	-2	0	0	0	15
398	7	9	9	-1	0	0	30
399	0	0	0	4	3	0	27
400	-10	-13	-8	3	-9	0	48
total	454	363	357	212	295	208	3828

위 5팀과 I- AdaBoost의 각 주제에 대한 LF1 값 및 총점수를 보여주고 있다. 여기서 LF1의 총점에 있어 i-AdaBoost가 가장 높은 점수를 나타내고 있다. 이것은 검증 데이타에대한 최고 LF1값을 선택했기 때문에 음수값을 얻게 될 가능성이 적고 다양한 데이타 분포로부터의 학습을 통해 일반화 오류를 줄일 수 있기 때문이다. 한편 352번 주제와 같이 비교적 많은 관련 있는 문서를 포함한 주제에 대해서는 최고의 성능을 나타내지 못하고 있는데 그 이유는 i-AdaBoost가 많은 문서에 대해 학습을 잘 수행하지 못하기 때문이 아니라 이 정도 수의 적은 문서에 대해서는 학습 데이타에 대한 과다학습 현상이 발생하여 낮은 일반화 오류를 얻기 힘들기 때문이다.

6. 결론 및 토의 사항

이 논문에서는 비교적 새로운 기계 학습 방법인 에이다부스트 학습 방법을 사용하여 문서 검색 대회인 TREC의 문서 여과 문제에 적합한 문서 여과 시스템을 생성하고자 하였다. 먼저, 문서 여과 문제에 적용한 에이다부스트 알고리즘을 구현하고 이의 학습 시간이 오래 걸리는 문제를 개선하기 위해 이진 값을 출력하는 가설이 아닌 실수값을 출력하는 가설을 이용하여 학습 시간을 단축하였다. 또 여러개의 데이타 분포에 대해 에이다부스트를 반복적으로 수행하는 알고리즘을 적용하여 TREC 데이타에 대해 일반화 성능을 개선하였다. real-Adaboost가 Adaboost-tf보다 열 배 가까운 속도 향상을 보였고, i-Adaboost는 평가척도로 사용한 LF1에 대해 다른 학습방법들 보다 우수한 성능을 나타내었다. 이번 연구와 관련하여 앞으로 더욱 많은 연구가 필요한 몇 가지 사항들을 살펴보면 다음과 같다.

1. 에이다부스트를 이용하여 학습을 진행함에 따라 발생하는 과다학습 문제와 관련하여 어떻게 일반화 오류를 최소화하는 시점에서 학습을 중단할 것인가의 문제를 해결할 필요가 있다.
2. TREC 데이타의 항 수가 문서 여과 시스템의 성능에 미치는 영향에 대해 자세히 알아볼 필요가 있다.
3. i-AdaBoost 알고리즘의 적용에 있어 한번 에이다부스트를 수행하는 라운드 수와, 몇 번의 에이다부스트를 수행할 것인가의 문제를 보다 체계적으로 결정하는 방법이 필요하다.

지금까지 살펴본 에이다부스트를 사용한 문서 여과 시스템의 경우 최종 결정에 사용되는 각 가설이 어떤 단어가 문서에 포함되어 있는가의 여부이므로 이 시스템의 사용자가 시스템의 결정규칙을 쉽게 이해할 수 있고 각 가설이 판단 기준이 되는 단어와 이에 관련된 확신정도, 이 두개의 값만을 저장하기 때문에 의사결정트리나 인공신경망에 비해 비교적 적은 메모리를 사용하여 실제 구현에 많은 문제를 초래하지 않는다. 이와 같이 구현과 이해가 용이하고 우수한 성능을 나타내는 학습 방법에 대한 보다 많은 연구를 통해 일반 사용자들이 보다 쉽게 문서를 검색할 수 있는 방법들이 개발될 수 있을 것이다.

참 고 문 헌

[1] D. Hull, "The TREC-7 filtering track: Description and analysis," *Proceedings of the 7th Text Retrieval Conference* (TREC-7), pp. 33-56, 1998.

[2] E. Bauer and R. Kohavi, "An empirical comparison of voting classification algorithms: bagging, boosting, and variants," *Machine Learning*, Vol. 36, No. 1, pp. 105-139, 1999.

[3] L. Breiman, "Bagging predictors," *Machine Learning*, Vol. 24, No. 2, pp. 123-140, 1996.

[4] L. Breiman, "Bias, variance, and arcing classifiers," *Technical Report 460, Berkeley, CA: University of California: Department of Statistics*, 1996.

[5] Y. Freund, R. E. Schapire, "Experiments with a new boosting algorithm," *Proceedings of the Thirteenth International Conference on Machine Learning*, pp. 148-156, 1996.

[6] S. Haykin, *Neural Networks*, Prentice-Hall, 1999.

[7] G. Salton, M. J. McGill, "Introduction to modern information retrieval," McGraw-Hill, 1983.

[8] R. E. Schapire, Y. Freund, P. Bartlett, W. S. Lee, "Boosting the margin: A new explanation for the effectiveness of voting methods," *The Annals of Statistics*, Vol. 26, No. 5, pp. 1651-1686, 1998.

[9] R. E. Schapire, Y. Singer, and A. Singhal, "Boosting and Rocchio applied to text filtering," *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Query and Profile Modification*, pp. 215-223, 1998.

[10] R. E. Schapire, Y. Singer, "Improved boosting algorithms using confidence-rated predictions," *Machine Learning*, Vol. 37, No. 3, pp. 297-336, 1999.

[11] R. E. Schapire, "The strength of weak learnability," *Machine Learning*, Vol. 5, No. 2, pp. 197-227, 1990.

[12] G. I. Webb, "MultiBoosting: A technique for combining Boosting and Wagging," *Machine Learning*, to appear, 2000.

한 상 윤

1998년 연세대 영문과 학사. 전산학 부전공. 2000년 서울대 컴퓨터공학부 석사. 2001년 ~ 현재 University of Washington, 전산학 박사과정. 관심분야는 기계학습, 자연언어처리, 부스팅 알고리즘

장 병 탁

1986년 서울대 컴퓨터공학과 학사. 1988년 서울대 컴퓨터공학과 석사. 1992년 독일 Bonm 대학교 컴퓨터과학과 박사. 1997 ~ 현재 서울대학교 컴퓨터공학부 조교수, 부교수. 2001 ~ 현재 서울대학교 바이오정보기술연구센터(CBIT) 센터장.