베이지안 부스팅학습에 의한 문서 분류

김유환 장병탁
서울대학교 컴퓨터공학부
{yhkim, btzhang}@scai.snu.ac.kr

Text Classification By Boosting Naïve Bayes

Yu-Hwan Kim Byoung-Tak Zhang
School of Computer Science and Engineering, Seoul National University

요 약

최근 들어, 여러 기계학습 알고리즘이 문서 분류와 여과에 사용되고 있다. 특히 AdaBoost 와 같은 부스팅 알고리즘은 실세계의 문서 데이터에 사용되었을 때 비교적 좋은 성능을 보이는 것으로 알려져 있다. 그러나 지금까지의 부스팅 알고리즘은 모두 단어의 존재 여부만을 가지고 판단하는 분류자를 기반으로 하고 있기 때문에 가중치 정보를 충분히 사용할 수 없다는 단점이 있다. 이 논문에서는 나이브 베이스 분류자를 약 학습자로 사용한 부스팅에 기반한 학습 알고리즘을 제안한다. 나이브 베이스를 사용한 부스팅 알고리즘은 단어의 가중치 정보를 효율적으로 사용할 수 있을 뿐 아니라, 확률적으로도 의미있는 신뢰도 (confidence ratio)를 생성할 수 있기 때문이다. TREC-7과 TREC-8의 정보 여과 트랙 (filtering track)에 대해서 실험한 결과 좋은 성능을 보여주었다.

1. 서론

정보 여과는 여러 정보 중에서 필요한 정보만을 그것을 필요로 하는 사람에게 전달해 주는 것을 의미한다. 전자 문서의 폭발적인 증가로 정보 여과는 정보 검색 분야에서 가장 중요한 응용분야 중 하나가 되었다. TREC도 필터링 트랙을 도입하여 많은 연구자들이 이에 참여하고 있다 [8].

정보 검색 시스템은 사용자의 흥미도를 프로파일의 형태로 학습하여 새로운 문서가 왔을 때 프로파일과 문서를 비교함으로써 문서가 그 사람에게 필요한지를 판단한다. 우리는 이 논문에서 프로파일을 한 번 학습하면 프로파일이 더 이상 바뀌지 않는 것으로 가정한다. 또한 문서를 사용자에게 보낼 것인가만을 판단하며 중요한 정도는 고려하지 않았다. 이것은 TREC 배치 필터링 트랙 (batch filtering track)에서 가정하고 있는 것과 일치한다.

정보 여과는 문서 분류 알고리즘과 긴밀이 연관되어 있다. 정보 여과의 성능을 향상시키기 위해서는 문서 분류 알고리즘이 필요하다. 가장 성공적인 방법 중의 하나는 AdaBoost 알고리즘이다 [1]. AdaBoost는 각각의 약 학습자 (weak learner)가 과거의 가설이 잘 못했던 부분을 집중적으로 학습한다. 이렇게 해서 최종적으로 만들어진 분류자들이에 비례한 투표 (weighted voting)을 하게 된다.

많은 연구자들이 AdaBoost를 문서 분류에 사용하였지만, 대부분 C4.5나 decision stump만을 사용하여 tf와 같은 가중치 정보를 사용하지 못하는 단점이 있었다 [4] [6]. 이 논문에서는 나이브 베이스 분류자를 약 학습자로 사용하여 부스팅을 하는 방법을 취하였다. BayesBoost라고 이름지어진 이 방법은 가중치 정보를 사용할 뿐 아니라 나이브 베이스가 확률 모델인 점을 이용하여 자연스럽게 신뢰도를 계산할 수 있다는 장점을 가지고 있다. 우리는 이 방법을 TREC-7과 TREC-8필터링 트랙에 적용하여 좋은 성능을 얻었다 [6].

2. 관련 연구

문서 여과는 두 가지 관점에서 연구되어 왔다. 첫번째는 기계 학습 관점이고, 다른 하나는 정보 검색 관점이다. 정보 검색 관점에서 본 대부분의 연구는 Rocchio 알고리즘과 긴밀하게 연관되어 있다. 많은 알고리즘이 Rocchio 알고리즘에 기반하여 설계되었는데 dynamic feedback optimization, query zoning, pivoted document normalization등이 그 대표적인 예이다. 기계 학습 관점에서는 다양한 방법이 문서 분류에 사용되었는데, 대표적인 것이 k-최근점 학습, 서보트 벡터 머쉰 (support vector machine, SVM), 나이브 베이스 분류자등이다. 최근에는 여러 개의 약 학습자를 묶어서 추정을 하고자 하는 시도가 이루어지고 있는데, 이에는 위원회 기계, 부스팅, 배깅등

이 있다. Schapire 등은 AdaBoost 와 Rocchio 를 REUTER-21578과 TREC-3에 대해서 비교하고, LF1측정 방법에 대해서 두 방법이 성능을 보임을 보였다.

3. BayesBoost를 이용한 문서 여과

3.1 AdaBoost

AdaBoost는 여러 개의 약 학습자를 생성한 후 가 학습자들이 가중치에 비례한 투표를 하는 방법으로서 Schapire등에 의해서 처음으로 제안되었다 [1]. 각각의 가설을 {-1,1}의 값을 갖는 h_t 라고 하고, 각 가설의 가중치를 α_t 라고 했을 때, 최종적인 분류자는 $\sum_{t=1}^{m}\alpha_t h_t$ 로 표현된다. 여기서 m은 가설의 개수이다. 더 향상된 방법은 신뢰도를 사용하는 것으로서 h_t가 실수 값을 가질 수 있도록 확장하고 $|h_t|$를 신뢰도, h_t의 부호를 어떤 범주에 할당할 것인가를 나타내는 것으로 해석한다 [5]. 많은 연구자들이 C4.5나 decision stump를 약 가설로 사용하여 실험을 하여 좋은 결과를 보였으나 [4], 이들 약 가설이 모두 단어의 출현 여부만을 사용하여 분류를 하기 때문에 tf와 같은 가중치 정보를 사용하지 못한다는 단점이 있다.

3.2 나이브 베이스

나이브 베이스는 문서 분류에 전통적으로 사용되어 왔던 방법이다 [3]. 여기서는 각 단어들이 서로 독립이라고 가정하였다. 이는 현실 세계의 데이터와 많은 차이가 있지만 그럼에도 불구하고 나이브 베이스는 문서 분류에서 좋은 성능을 보여 주었다.

나이브 베이스에서 훈련과정은 어떤 범주 c_k 에서 어떤 단어 w_k 가 출현할 확률($\theta_{w_k|c_k}$), 어떤 범주 c_k 가 출현할 확률(θ_{c_k})를 학습하는 것이다. 각각은 아래와 같이 학습할 수 있다.

$$\theta_{w_k|c_j}=\frac{1+\sum_{i=1}^{D}N(w_k,d_i)P(c_j|d_i)}{|V|+\sum_{s=1}^{|V|}\sum_{i=1}^{|D|}N(w_k,d_i)P(c_j|d_i)}$$

$$\theta_{c_j}=\frac{\sum_{i=1}^{|D|}P(c_j|d_i)}{|D|}$$

여기서 $N(w_k,d_i)$ 은 문서 d_i 에 단어 w_k 가 나오는 회수를 의미하고 $|V|$는 단어 집합의 크기, $|D|$는 문서의 개수를 의미한다. 이를 이용하여 새로운 문서가 들어오면 다음과 같이 분류를 할 수 있다.

$$P(c_j|d_i;\hat{\theta})=\frac{P(c_j|\hat{\theta})P(d_i|c_j;\hat{\theta})}{P(d_i|\hat{\theta})}=\frac{\hat{\theta}_{w_i|c_j}\hat{\theta}_{c_j}}{P(d_i|\hat{\theta})}$$

최종적으로 $\arg\max_j P(c_j|d_i;\hat{\theta})$ 가 그 문서가 들어갈 범주 정보를 나타낸다. 그러나 나이브 베이스는 확률 모델이기 때문에 기본적으로 많은 데이터를 가지고 있지 않으면 좋은 결과를 낼 수 없다.

3.3 BayesBoost 알고리즘

나이브 베이스는 확률 모델이기 때문에 신뢰도를 측정하는 자연스러운 방법이 될 수 있다. 또한 가중치 정보를 이용하기 때문에 Decision Stump를 이용하는 것 보다 더 좋은 성능을 낼 수 있다. 이와 같은 면을 고려해서 이 논문에서는 BayesBoost 알고리즘을 제안한다. 이 알고리즘에서 각 약 학습자는 나이브 베이스로 이루어져 있고 나이브 베이스의 출력을 이용하여

$$h_t(d_i;\hat{\theta})=\tanh\left\{\log\left(\frac{P(c_i=1|d_i;\hat{\theta})}{P(c_i=-1|d_i;\hat{\theta})}\right)\right\}$$

로 하였다. 이와 같이 하였을 때 가설의 출력은 [-1,1]이 된다. 이렇게 하면 α 를 계산할 때 수치해석적인 방법을 쓰지 않아도 되므로 계산이 간편한 이점이 있다. α 는 다음과 같이 계산된다.

$$\alpha=\frac{1}{2}\ln\left(\frac{1+r}{1-r}\right),r=\sum_i D(i)y_i h_t(x_i)$$

4. 실험 및 결과

4.1 성능 측정 방법

문서 여과에서는 여러 가지 방법이 성능 측정을 위한 수단으로서 사용된다. 그 대표적인 것이 precision/recall, break-even point 이다. 그러나 최근 TREC 에서는 linear utility를 측정을 위한 방법으로 사용할 것을 제안하였다 [2]. 그러므로 이 논문에서도 전통적인 방법 대신 linear utility를 성능 측정의 방법으로 사용하였다. Linear utility 는 다음과 같이 정의된다.

$$\text{Linear Utility} = aR_+ + bN_+ + cR_- + dN_-$$

여기서 R_+ 는 추출된 문서중에서 적합한(relevant) 문서의 개수, N_+ 는 추출된 문서중에서 적합하지 않은 문서의 개수, R_- 는 추출되지 않은 문서중에서 적합한 문서의 개수, 마지막으로 N_- 는 추출되지 않은 문서중에서 적합하지 않은 문서의 개수를 나타낸다. a, b, c, d 는 모두 상수이다. 이 실험에서는 TREC 필터링 트랙에서 사용된 LF1 을 측정 수단으로 사용하였다.

$$LF1 = 3R_+ - 2N_-$$

4.2 TREC-7과 TREC-8에 대한 실험

TREC-7필터링 트랙 데이터셋은 1988-1990년의 AP 뉴스에서 모아진 신문기사와 토픽 1-50으로 이루어져 있다. 1988년 기사를 학습 데이터로 삼고, 나머지를 테스트 데이터로 하였다. TREC-8필터링 트랙 데이터셋은 1992-1994까지의 데이터셋과 토픽 351-400으로 이루어져 있는데, 1992 년을 트레이닝 데이터로 하고, 1993-1994를 테스트 데이터로 하였다. 이것은 트랙 필터링 트랙에서 제안한 실험 환경과 정확히 일치하는 것이다. 이 실험에서는 토픽의 'description' 정보를 사용하지 않았다.

전처리로 스테밍과 불용어 제거를 한 후 추출된 단어 중에서 TREC-7에서는 5250 단어를 TREC-8에서는 4079 단어를 문서 빈도에 따라서 추출하였다. 100 개의 약 학습자를 학습한 후 이를 모아 하나의 분류자로 만들었다. 비교를

위해서 TREC-7과 TREC-8 필터링 트랙에서 가장 좋은 성능을 낸 것을 세 개씩 선별하였다. 또한 문서 분류에서 비교적 좋은 성능을 보인 SVM과도 비교를 하였다. 실험 결과는 TREC-7에 대해서는 표 1, TREC-8에 대해서는 표 2에 있다. 각 그림에서 x축은 positive example의 개수 y축은 BayesBoost와의 차이를 나타내고 있다.

5. 결론

BayesBoost는 다른 알고리즘들에 비해 비교적 좋은 성능을 보여주었다. 특히 positive example 의 수가 많을수록 다른 참가자들과 뚜렷한 성능차이를 보여 주었다. 단 pirc 에 비해서는 약간 좋지 않은 결과를 보여주었는데, 이는 TREC 데이터셋이 positive example의 수가 매우 적기 때문이다. 이러한 현상은 AP에서 더욱 두드러지게 나타난다. 데이터가 충분히 많은 경우 표 2 에서 볼 수 있듯이 SVM 이 매우 좋은 성능을 보여 주었다. 그러나 SVM 은 데이터 수가 적은 경우는 그다지 좋지 않은 결과를 보이는 경우가 많았다.

표 1. TREC -7 필터링 트랙에 대한 실험 결과. 제출된 결과 중 상위 3 개와의 비교 결과 및 SVM 과의 비교 결과.

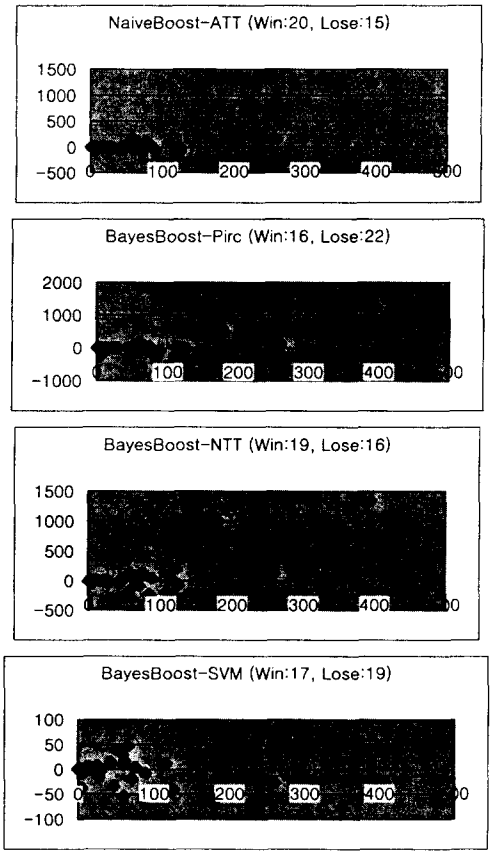
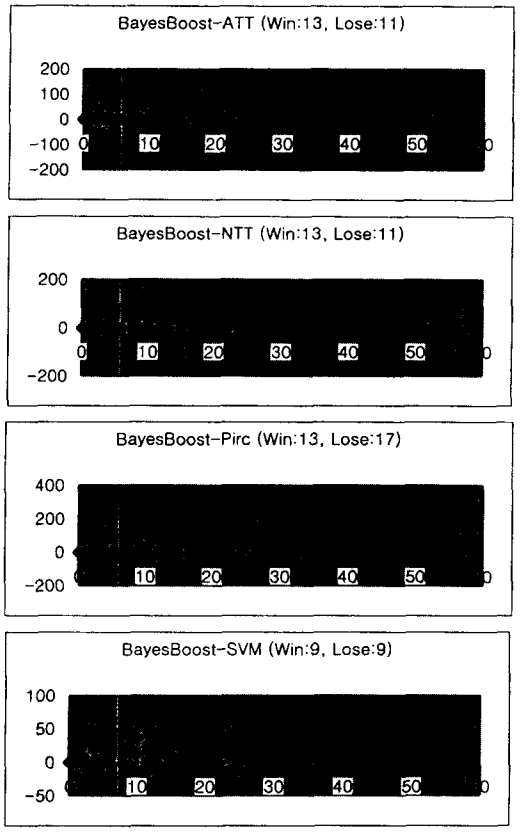


표 2. TREC-8 필터링 트랙에 대한 실험 결과. 제출된 결과 중 상위 3 개와의 비교 결과와 SVM 과의 비교 결과



감사의 글 : 본 연구는 과학 기초(98-119)에 의해서 지원되었습니다.

참고 문헌

[1] Y. Freund and R. E. Schapire, "Experiments with a new boosting algorithm ", In *Proc. 13th Int. Conf. On Machine Learning*, pp. 148-156, 1996

[2] D. Hull. "The TREC -8 filtering track : Description and analysis", In *Proc. 7th Text Retrieval Conf.* (TREC-7), pp. 33-56, 1998.

[3] T. Joachims, "A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization ", In *Proc. Int. Conf. on Machine Learning* (ICML-97), pp. 143-151, 1997.

[4] J. R. Quinlan, "bagging, boosting and C4.5 ", In *Proc. AAAI-96*, pp. 725-730, 1996.

[5] R. E. Schapire and Y. Singer, "Improved boosting algorithms using confidence -rated predictions ", *Machine Learning*, 37(3):297-336, 1999.

[6] R. E. Schapire, Y . Singer, and A. Singhal, "Boosting and Rocchio applied to text filtering", In *Proc. SIGIR-98*, pp. 251-223, 1998.

[7] D. K. Harman, "Overview of the 8th Text Retrieval Conference (TREC -8)", In *Proc. 8th Text Retrieval Conf.* (TREC-8), pp 1-10, 1999.