EM 알고리즘을 이용한 이진 분류 문서 범주화의 성능 향상†

한형동*, 고영중**, 서정연***
*서강대학교 컴퓨터학과
**동아대학교 전기전자컴퓨터공학부 컴퓨터공학전공
***서강대학교 컴퓨터학과
kamble@nlpzodiac.sogang.ac.kr* blunimph@hanmail.net** seojy@ccs.sogang.ac.kr***

Improving performance of Binary Text Classification Using the EM algorithm

Hyoungdong Han*, Youngjoong Ko**, Jungyun Seo***
*Dept. of Computer Science, Sogang University
**Computer Engineering, School of Electrical, Electronics & Computer Engineering, Dong-A University
***Dept. of Computer Science, Sogang University

요 약

문서 범주화에서 이진 분류를 다중 분류에 적용할 때, 일반적으로 One-Against-All 방법을 사용한다. 하지만, 이 One-Against-All 방법은 한가지 문제점을 가진다. 즉, positive 집합의 문서들은 사람이 직접 범주를 할당한 것이지만, negative 집합의 문서들은 사람이 직접 범주를 할당한 것이 아니기 때문에 오류 문서들이 포함될 수 있다는 것이다. 본 논문에서는 이러한 문제점을 해결하기 위해 Sliding Window 기법과 EM 알고리즘을 이진 분류 기반의 문서 범주화에 적용할 것을 제안한다. 먼저 Sliding Window 기법을 이용하여 학습 데이터로부터 오류 문서들을 추출하고 이 문서들을 EM 알고리즘을 사용해서 다시 범주를 할당함으로써 이진 분류 기반의 문서 범주화 기법의 성능을 향상시킨다.

1. 서 론

자동 문서 범주화란 문서의 내용에 기반하여 미리 정의되어 있는 범주(category)에 문서를 자동으로 할당하는 작업이다. 학습 작업을 위해 학습데이터를 구성하는 방법에는 이진 분류 구성(binary setting)과 다중 분류 구성(multi-class setting) 이 있다. 이진 분류 구성은 두 개의 범주만을 가지며, 이 두 개의 범주는 '관련성이 있는 것(relevant or positive)'과 '관련성이 없는 것(unrelevant or negative)', 즉, 범주에 속하는 문서와 범주에 속하지 않는 문서이다[1]. 일반적으로 어떠한 분류 작업들은 2개 이상의 범주를 포함한다. 2개 이상의 범주를 갖는 다중 분류 구성에 이진 분류를 적용할 때 한가지 문제점을 가진다. 다중 분류는 각 범주별로 positive 문서 집합은 존재하지만 negative 문서 집합은 존재하지 않는다. 따라서, 이러한 다중 분류의 문제를 해결하기 위해 One-Against-All 방법을 일반적으로 사용한다.

그림 1은 One-Against-All 방법을 사용하여 4가지 범주(정치, 경제, 사회, 그리고 스포츠)를 갖는 다중 분류 구성을 이진 분류 구성으로 변환한 예를 보이고 있다. 여기서, positive 집합의 문서들은 각 범주에 대해 사람이 직접 범주를 할당한 것이지만, negative 집합의 문서들은 각 범주에 대해 사람이 직접 범주를 할당한 것이 아니다. 그러므로, negative 집합은 많은 오류 문서(noise document)들이 포함될 수 있다. 이러한 오류 문서들은 이진 분류 기반의 문서 범주화(binary text classification)의 성능을 떨어뜨리는 주원인이 된다.

이 오류 문서들을 제거하기 위해서 다음과 같은 두 가지 문제를 해결해야만 한다; "오류 문서를 포함하는 경계 부분을 어떻게 찾을 것인가?", "찾은 오류 문서들을 어떻게 처리할 것인가?" 본 논문에서는 첫 번째와 두 번째 문제를 해결하기 위해 Sliding Window 기법과 EM알고리즘을 각각 사용한다.

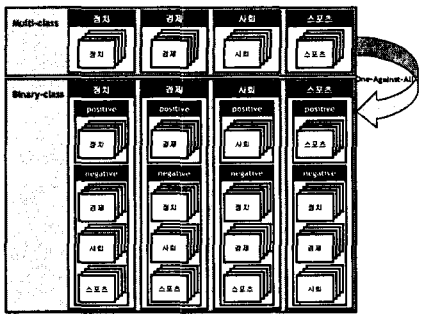

그림 1. One-Against-All 방법을 이용한 학습 데이터의 재구성

본 논문의 구성은 다음과 같다. 제2장에서는 이전의 관련 연구를 소개하고 간단히 설명한다. 제3장에서는 제안된 방법의 각 단계에 대해 자세히 설명하고, 제4장에서는 실험을 통해 나온 결과를 비교, 분석한다. 마지막으로 제5장에서는 결론과 향후 과제를 다루었다.

2. 관련 연구

Liu는 본 논문에서 다루고자 하는 문제를 해결하고자 S-EM이라고 하는 새로운 시스템을 제안하였다[2]. 이 시스템은 베이지언 확률 모

†. 본 연구는 한국과학재단(KOSEF) 목적기초연구(R01-2003-000-11588-0) 지원으로 수행되었음.

델과 EM (Expectation Maximization)알고리즘[3]에 기반한다. 이 시스템은 범주가 할당되지 않은 문서 집합으로부터 확실한 negative 문서들을 식별하기 위해 처음으로 스파이(Spy) 기술을 사용한다. 그리고, 최종 분류기를 생성해 내기 위해서 EM알고리즘을 수행한다.

Yu는 본 논문에서 다루고자 하는 문제를 해결하고자 negative집합을 범주가 할당되지 않은 문서 집합(unlabeled documents set)으로 간주한 후 이 문서 집합에 포함된 오류 문서들을 제거하고 확실한 negative 문서들(reliable negative documents)을 식별하는 PEBL이라는 시스템을 제안한다[4]. PEBL 시스템은 확실한 negative문서들을 식별하여 최종적인 분류기(classifier)를 얻기 위해 지지 벡터 기계(SVM)를 수행한다.

3. 제안한 방법

본 논문에서 제안한 방법은 크게 다음과 같은 4단계로 구성되어 있다: (1) One-Against-All, (2) 예견점수(Prediction Score) 계산, (3) Sliding Window를 사용한 혼잡도(Entropy) 계산, (4) EM 알고리즘

3.1 예견 점수(Prediction Score) 계산

3.1절과 3.2절의 목적은 많은 오류 문서를 갖는 영역의 경계 부분을 찾는 것이다. 이를 위해 먼저 One-Against-All 방법을 사용하여 posititve 문서 집합과 negative 문서 집합을 생성한다. 그 후 생성된 이진 분류 학습 데이터(binary training data)로 베이지언(NB) 분류기를 학습하고 다음 공식을 사용하여 각 문서에 대한 예견 점수를 얻어낸다.

$$\text{Prediction\_Score}(c_i \mid d_j) = \frac{P(Positive \mid d_j)}{P(Positive \mid d_j) + P(Negative \mid d_j)} \qquad (1)$$

식(1)에서의 확률 $P(Positive|d_j)$와 확률 $P(Negative|d_j)$는 일반적인 베이지언 확률 계산식으로써 계산한다[6]. 이 계산된 예견 점수에 따라 각 범주의 문서들은 점수가 높은 순으로 정렬된다.

3.2 Sliding Window를 사용한 혼잡도(Entropy) 계산

제안한 방법을 통해 하나의 경계면을 찾는 것은 positive 문서와 negative 문서가 가장 많이 섞이는 경계 구간을 찾는 것이다. 경계 구간을 찾기 위해, 본 논문에서는 먼저 Sliding 기법을 사용한다. 일정한 크기를 갖는 window들은 정렬된 예견 점수를 갖는 문서 목록에서 첫 번째 문서에서부터 마지막 문서까지 한 단계 한 단계 내려간다. 혼잡도 값은 각 window 내의 혼잡 정도(positive와 negative가 섞이는 정도)를 추정하기 위해 계산된다.

$$Entropy(S) = -p_+ \log_2 p_+ - p_- \log_2 p_- \qquad (2)$$

각 window(S)의 문서들에 대한 분포의 혼잡도는 식(2)로써 계산하고 [7], S는 positive문서들과 negative문서들이 포함된 집합을 의미한다.

본 논문에서는 가장 높은 혼잡도 값을 갖는 두 개의 window를 뽑는다. 먼저 처음으로 가장 높은 혼잡도 값을 갖는 window를 뽑고 두 번째로 마지막으로 가장 높은 혼잡도 값을 갖는 window를 뽑는다. 그 후 최대 경계값(max threshold value)은 처음으로 가장 높은 혼잡도 값을 갖는 window 속에서 가장 높은 예견 점수를 갖는 negative문서의 예견 점수를 최대 경계값으로 정하고, 최소 경계값(min threshold value)은 마지막으로 가장 높은 혼잡도 값을 갖는 window 속에서 가장 낮은 예견 점수를 갖는 positive문서의 예견 점수를 최소 경계값으로 정한다. **그림 2**의 왼쪽 그림은 최대, 최소 경계값을 어떻게 찾는지 보여주고 있다.

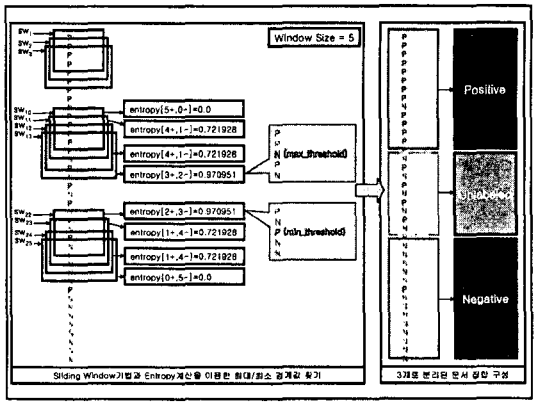

그림 2. 경계구간을 찾는 예와 3 개의 문서 집합 구성의 예

본 논문에서는 최대, 최소 경계값 사이에 있는 모든 문서들을 범주가 할당되지 않는 문서(Unlabeled document)들로 간주하며, 오류 문서(noise document)제거를 위해 이 문서들은 두 개의 범주(positive and negative) 중 하나의 범주를 다시 할당 받게 된다.

이로써 **그림 2**의 오른쪽 그림과 같이 각 범주별 3개의 문서 집합을 갖게 된다: 확실한 positive 문서들(definitely positive documents), 범주가 할당되지 않은 문서들(unlabeled documents), 확실한 negative 문서들(definitely negative documents). 본 논문에서는 이 3개의 데이터 집합을 EM 알고리즘을 적용하여 각 범주별로 범주가 할당되지 않는 문서들에게 다시 범주를 할당한다.

3.3 EM 알고리즘

본 논문에서 EM 알고리즘은 범주가 할당되지 않는 문서(Unlabeled document) 집합을 잘 정리하고 그 속에 있는 오류 문서(noise document)들을 제거하기 위해 사용된다. EM 알고리즘은 기대 단계(Expectation step)와 최대화 단계(Maximization step)의 두 단계로 구성되어 있다[3].

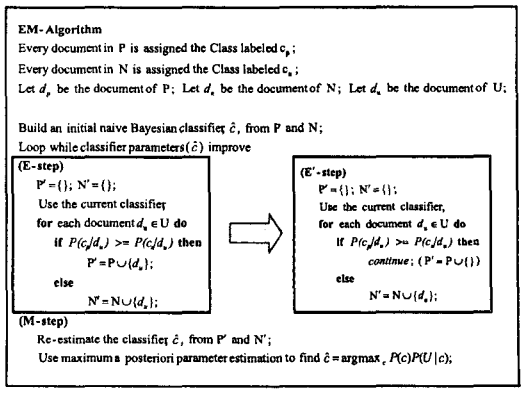

그림 3. EM 알고리즘

EM 알고리즘은 먼저 범주가 할당된 문서(Labeled document)들을 사용하여 분류기(classifier)를 학습한다. 그 후 범주가 할당되지 않는 문서들에게 범주를 할당한다(Expectation(E or E´) step). 그리고 나서 정리된 학습 데이터를 가지고 다시 분류기를 학습시킨다(Maximization(M) step). 그리고 이 과정(E(or E´)-step과 M-step)을 수렴할 때까지 반복하게 된다. 베이지언 분류기에 대해 EM알고리즘에서 사용되는 단계들은 분류기를 생성하기 위해 사용되는 것과 동일하다. **그림 3**는 본 논문에서 EM 알고리즘이 어떻게 사용되는지를 보여준다.

E´-step은 경계 부근에 있는 오류 문서들을 제거하기 위해 E-step을 변형한 것이며, 기존의 E-step과는 달리 positive 집합으로 할당되는 문서 d_u를 오류 문서로 간주하여 제거하게 된다. 결국, EM 알고리즘에 의해 새롭게 생성된 이진 학습 데이터(binary training data)를 가지고 문서 분류기를 학습할 수 있다. 본 논문의 EM알고리즘에서 알고리즘의 반복 횟수(iteration)는 모든 실험에서 두 번으로 하였다.

4. 실험 평가

4.1 실험 데이터 및 성능 평가 방법

본 논문에서는 제안한 방법을 평가하기 위해 다른 두 종류의 데이터(신문기사(Reuters21578), 웹 페이지(WebKB))를 사용한다. WebKB 데이터에 대해서는 일관성 있는 평가를 위해 five-fold cross-validation 방법을 사용하였다. Reuters21578 Distribution 1.0 데이터는 12,902개의 기사와 90개의 범주로 구성되어 있으며, 본 논문에서는 다른 연구들에서 가장 많이 사용하는 10개의 범주만을 사용하였다[8]. 그리고 Reuters 데이터의 학습데이터 중 약 20%를 검증 문서 집합(validation set)으로 사용하였다. 두 번째 데이터 집합인 WebKB 데이터는 카네기멜론(CMU)에서 제공하는 데이터로써[9], 이 데이터 집합은 여러 대학의 컴퓨터학과 홈페이지에서 수집된 웹 페이지들이다.

본 논문에서의 성능 평가 방법은 정보 검색 분야에서 일반적으로 쓰이는 이진 분류(binary classification)에 대한 평가 방법인 precision-recall BEP(breakeven point) 값을 사용하여 나타냈다[10][11].

4.2 실험 결과

이 절에서는 기본 시스템(One-Against-All방법만 사용한 시스템)과의 비교를 통해 제안된 기법과의 성능을 평가 하였다. Window 크기는 실험값에 의해 5로 하여 모든 실험들을 하였다.

4.2.1 EM 알고리즘에서의 E-step과 E´-step의 비교 실험

본 실험은 Reuter 데이터에서 추출한 검증 문서 집합(validation document set)을 사용하고 베이지언 분류기만을 사용하였다. 본 논문은 3.3절에서 새로운 기대 단계인 E´-step(E´)을 제안하였고, 제안한 방법을 검증하기 위해 기존의 E-step과 제안한 E´-step을 비교 하였다. **표 1**는 제안한 EM(E´)이 기존의 EM(E)보다 더 높은 성능을 보여줌을 알 수 있다.

표 1. EM 알고리즘에서의 E-step 과 E´-step 의 비교

	기본 시스템	기존의 EM (E-step)	제안한 EM (E´-step)
micro-avg BEP	89.75	90.52	**92.31**

4.2.2 분류기별 성능 비교 실험

본 논문에서는 베이지언(NB: Naïve Bayesian), Rocchio, 지지 벡터 기계 (SVM: Support Vector Machine) 분류기에 대한 각 분류기의 성능을 평가 하기 위한 실험을 하였다. **표 2**에서 알 수 있듯이 두 가지 다른 종류의 데이터(Reuter, WebKB)에 대해 제안한 시스템이 One-Against-All 방법만을 사용한 기본시스템보다 훨씬 월등한 성능을 보인다. 특히, 베이지언 확률 모델과 Rocchio에서는 뚜렷한 성능 차이를 확인할 수 있다.

표 2. Reuter 와 WebKB 데이터에 대한 분류기별 성능 비교

분류기 / 데이터	NB		Rocchio		SVM	
	기본 시스템	제안 시스템	기본 시스템	제안 시스템	기본 시스템	제안 시스템
Reuter	90.80	**93.86**	89.24	**91.80**	94.66	**95.52**
WebKB	85.67	**87.21**	86.52	**88.26**	92.12	**92.64**

5. 결론 및 향후 과제

본 논문에서는 One-Against-All 의 문제를 해결하기 위해 Sliding Window와 EM 알고리즘을 사용하여 새로운 이진 분류 문서 범주화 기법을 제안하였으며, 결과적으로 제안한 방법이 세가지 분류기 모두에서 상당한 성능 향상을 보임을 확인했다. 향후 과제는 다음과 같다. 먼저, 오류 문서(noise document) 제거를 위해 EM 알고리즘 대신 효과적인 다른 알고리즘을 사용하는 것에 의해 더 좋은 성능을 기대할 수 있겠다. 또한 좀더 정확한 경계면을 찾을 필요가 있다고 생각된다.

참고문헌

[1] T. Joachims, *Learning to Classify Text Using Support Vector Machines : theory and Algorithms by Thorsten Joachims*. Dept. of Computer Science, Cornell University, NY, USA, Kluwer Academic Publishers, April, 2002.

[2] B. Liu, W. S. Lee, P. S. Yu, and X. Li, Partially Supervised Classification of Text Documents. In *Proceedings of the Nineteenth International Conference on Machine Learning(ICML-2000)*, Sydney, Australia, July 8-12, 2002.

[3] A. Demster, N. M. Laird, and D. Rubin, Maximum likelihood from incomplete data via the EM algorithm. In *Journal of the Royal Statistical Society series B*, vol 39, No. 1, pp. 1-38, 1997.

[4] H. Yu, J. Han, and K. Chang, PEBL : Positive example based learning for Web page classification using SVM. In *KDD-02, 2002*.

[5] B. Zadrozny, and C. Elkan, Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers. In *Proceedings of the Eighteenth International Conference on Machine Learning*, 2001.

[6] Y. Ko , and J. Seo, Automatic Text Categorization by Unsupervised Learning. In *Proceedings of the 18th International Conference on computational Linguistics (COLING'2000)*, pp. 453-459, 2000.

[7] T. Mitchell, *Machine Learning*. New York : McGraw-Hill, 1997.

[8] K. P. Nigam, *Using Unlabeled Data to Improve Text Classification*. Doctoral dissertation, computer Science Department, Carnegie Mellon University, 2001.

[9] M. Craven, D. DiPasquo, D. Freitag, A. McCallum, T. Mitchell, K. Nigam, and S. Slattery, Learning to construct knowledge bases from the World Wide Web. *Artificial Intelligence*, 118(1-2), pp. 69-113, 2000.

[10] Y. Yang, S. Slattery, and R. Ghani. A study of approaches to hypertext categorization. In *Journal of Intelligent Information Systems*, Vol. 18, No. 2., 2002.

[11] T. Joachims, Text Categorization with Support Vector Machines: Learning with Many Relevant Features. In *ECML*, pp. 137-142, 1998.