나이브 베이지안 분류자와 메세지 규칙을 이용한 스팸메일 필터링 시스템

조한철* 조근식*
*인하대학교 전자계산공학과
cc1232@eslab.inha.ac.kr gsjo@inha.ac.kr

Spam-mail Filtering System Using Naive Bayesian Classifier and Message Rule

Han-Cheol Cho* Geun-Sik Jo*
*Dept. of Computer Science & Engineering, Inha University

요 약

인터넷의 급속한 성장과 함께 E-Mail은 대표적인 통신수단의 하나가 되어버렸다. 편리하다는 점을 이용해서 엄청난 양의 스팸메일이 매일같이 쏟아져 오고, 그 문제의 심각성에 정보통신부에서 정보통신망 이용촉진 및 정보보호등에 관한 법률이라는 새로운 법률까지 생겨났다. 본 논문에서는 이 법률에서 요구하는 '광고'라는 문구를 걸러내는 등의 메세지 규칙을 갖는 시스템과 기존의 문서 분류에 널리 쓰이던 나이브 베이지안 분류자(Naive Bayesian Classifier)를 결합한 스팸 메일 필터링 시스템(Spam-mail Filtering System)을 제안한다. 제안된 시스템에서는 사용자가 직접 규칙을 작성할 필요없이 학습한 데이터를 갖고 자동으로 스팸메일을 분류할 수가 있다. 들어온 메일은 메세지 규칙 기반 필터가 먼저 적용되고, 메세지 규칙 기반 필터에서 분류되지 않으면 나이브 베이지안 필터에서 분류된다. 실험에서는 제안된 시스템의 성능을 평가하기 위해서 메세지 규칙을 사용한 시스템 및 나이브 베이지안 분류자 시스템과 비교 평가하였다. 또한 임계치를 변경함으로써 제안된 시스템의 성능을 높일 수 있도록 하였다.

1. 서론

인터넷이 급속하게 성장하면서 E-Mail 또한 대표적인 통신수단으로 발전하여 많은 사람들이 정보를 보내거나 받거나 하는데 이용하고 있다. 이메일은 사용하는데 있어서 비용이 거의 들지 않기 때문에 많은 개인이나 업체들이 자신들의 광고를 위해서 사용하고 있고 이로 인해서 메일 서비스를 운영하는 업체에서는 저장장치 용량의 부족 등의 문제를 겪고 있다. 단지 서버를 운영하는 입장이 아니라 메일을 주고 받는 사용자의 입장에서도 쏟아져 들어오는 원하지 않는 스팸메일을 지우는데도 매일 일정량의 시간을 투자해야 한다.

스팸메일이 심각한 사회문제가 되자 정보통신부에서 정보통신망 이용촉진 및 정보보호등에 관한 법률 시행령 및 시행규칙 개정안을 마련해서 시행하고 있다. 이 안에 따르면 모든 스팸메일들은 '광고'라는 문구를 제목란에 넣도록 되어있다. 이미 나와있는 메일 클라이언트 프로그램들이 갖고 있는 메세지 규칙 기반의 필터링 기능으로 이러한 스팸메일들을 걸러낼 수 있으나 법안을 따르지 않거나 편법을 이용한 스팸메일을 걸러내는 것은 매우 힘들다.

전자문서 분류에 대한 연구에 많이 이용되는 Naive Bayesian 분류자는 문서 내의 단어들을 대상으로 분류를 하기 때문에 법안을 따르지 않는 스팸메일도 걸러낼 수가 있다. 그러나 일정량 이상의 학습이 필요하고, 학습과 분류에 어느 정도의 시간이 걸린다는 단점이 있다.

본 논문에서는 분류시간과 정확도를 높이기 위해서 위의 두가지 방식을 결합한 방식을 제안하고 각 분류방식과 비교 평가하였다.

2. 관련 연구

2.1 베이즈 정리(Bayes theorem)

가설 공간 h와 주어진 트레이닝 데이터 D로부터 가장 좋은 가설을 구하고자 할 때 다음과 같은 베이즈 정리를 사용할 수 있다. P(h)는 h의 사전확률을 나타내고, P(h|D)는 D가 주어졌을 때 h의 사후확률이다.

$$\text{Bayes theorem: } P(h|D) = \frac{P(D|h)P(h)}{P(D)}$$

사후확률이 최대인 가설(MAP: Maximum a posteriori)을 찾기 위해서 베이즈 정리를 사용하면

$$h_{MAP} \equiv \arg\max_{h \in H} P(h|D) = \arg\max_{h \in H} \frac{P(D|h)P(h)}{P(D)}$$

$$= \arg\max_{h \in H} P(D|h)P(h)$$

마지막에서 P(D)가 생략된 것은 h에 대해서 독립적이기 때문이다.

2.2 나이브 베이지안 분류자(Naive Bayesian Classifier)

나이브 베이지안 분류자는 베이지안 학습방법 중에서 널리 쓰이는 통계적 알고리즘이다. 나이브 베이지안 분류자는 속성값들의 결합으로 이루어진 각 인스턴스 x와 특정 유한집합 V에서 어떤 값을 갖는 목적함수 f(x)가 존재하는 학습업무에 적용된다.

새로운 인스턴스를 분류하기 위한 베이지안 접근법은 인스턴스를 구성하는 속성값들 <a1, a2...,an>로부터 얻은 가장 큰 확률값을 v_{MAP}에 대입하는 방식이다.

$$v_{MAP} = \arg\max_{v_j \in V} P(v_j | a_1, a_2 \ldots a_n)$$

이 식을 베이즈 정리를 사용해서 다시 쓰면

$$v_{MAP} = \arg\max_{v_j \in V} \frac{P(a_1, a_2 \ldots a_n | v_j)P(v_j)}{P(a_1, a_2 \ldots a_n)}$$

$$= \arg\max_{v_j \in V} P(a_1, a_2 \ldots a_n | v_j)P(v_j) \text{식(1)}$$

식(1)을 이용해서 $P(v_j)$와 $P(a_1, a_2 \ldots a_n | v_j)$를 계산할 수 있다. $P(v_j)$는 트레이닝 데이터에서 v_j가 발생한 빈도를 계산하는 것으로 쉽게 구할 수 있지만 $P(a_1, a_2 \ldots a_n | v_j)$를 매우 많은 트레이닝 데이터 집합을 갖고 있지 않는 한 구하기가 쉽지 않다.

그래서 나이브 베이지안 분류자는 속성값들이 주어진 목적값에 조건부 독립적(Conditionally Independence)이라는 가정을 기반으로 한다. 실제적으로는 이 가정은 맞지 않는다. 특정 위치에서 '분류자'라는 단어가 나올 확률보다 앞의 단어가 '베이지안'인 경우에 그 다음 단어가 '분류자'가 될 확률이 더 크기 때문이다. 다행히도 실제 응용시 위치 독립적이라는 가정으로도 베이지안 분류자는 잘 동작된다[3]. a_1, a_2 ... a_n의 결합확률이 각 속성들의 확률의 곱으로 계산될 수 있다는 가정을 식(1)에 대입하면 나이브 베이지안 분류자에 사용할 공식을 구하게 된다.

$$v_{NB} = \arg\max_{v_j \in V} P(v_j) \prod_i P(a_i | v_j) \quad \text{식(2)}$$

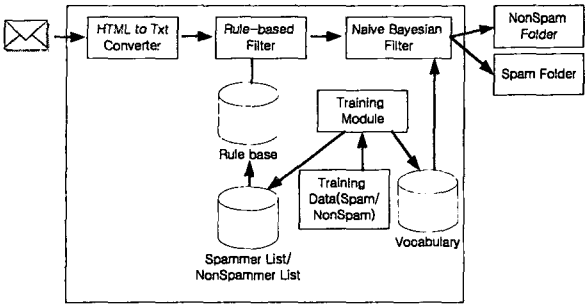
여기서 v_{NB}는 나이브 베이지안 분류자가 출력하는 목적값을 나타낸다.

3. 제안하는 스팸메일 필터링 시스템

스팸메일을 걸러내는 시스템을 만들 때 유의해야 하는 점은 잘못된 분류에 대한 비용의 차이다. 몇몇 스팸메일을 읽어야 할 메일로 구분했을 경우 사용자가 직접 삭제하는 비용만 들게 되지만, 읽어야 할 메일을 스팸메일로 구분한 경우에는 분류된 스팸메일들을 뒤져서 찾아 읽거나 스팸메일을 전부 삭제한 경우에는 메일을 아예 읽을 수 없게된다. 그러므로 잘못된 분류가 존재할 수 있음을 인정할 경우 스팸메일이 스팸이 아닌 메일로 분류될 수는 있되 그 반대의 경우는 최대한으로 일어나지 않게 시스템을 작성해야 한다.

3.1 시스템 구조

제안하는 나이브 베이지안 분류자를 이용한 규칙 기반의 스팸메일 필터링 시스템의 구조는 [그림1]과 같다.



[그림1] 시스템 구조도

수행하는 과정은 다음과 같다. 메일이 도착했을 때 먼저 그 메일이 HTML로 이루어진 문서인지 텍스트만으로 이루어진 문서인지 확인한다. HTML로 이루어졌으면 컨버터를 통해서 일반 텍스트만을 갖는 문서로 전환된다. HTML을 일반 텍스트로 바꾸는 이유는 나이브 베이지안 분류자를 학습시킬 때 그 단어의 출현빈도가 발생확률에 영향을 미치기 때문인데 메일의 내용과 상관없는 HTML태그들이 잘못된 분류로 이끌지 않도록 해야한다. 텍스트로 이루어진 메세지는 규칙기반 필터를 통과하는데 해당되는 규칙이 있으면 곧바로 해당폴더로 옮겨지고 해당 규칙이 없으면 나이브 베이지안 필터를 통과하게 된다. 나이브 베이지안 필터에서 미리 학습된 정보를 이용해서 메세지를 분류하게 된다.

3.2 메세지 규칙 기반의 필터링

이메일 필터링에서 가장 중요한 부분이 송신자와 수신자와의 신뢰도 문제이다. 신뢰할 수 있는 사람이 보낸 메일이라면 읽어야 하는 메일이 되지만 신뢰할 수 없는 사람에게서 온 메일이라면 스팸으로 간주해야 한다. 문제점은 모든 송신자의 신뢰도를 알 수는 없다는 것으로 일반 메일 클라이언트에서는 주소록을 이용하여 신뢰할 수 있는 송신자를 판별한다. 하지만 본 논문에서는 베이지안 분류자때문에 학습을 해야하므로 학습시에 스패머 리스트(spammer list)와 논스패머 리스트(Non-spammer list)를 저장하여 주소록 대신 사용하고 두 리스트에 중복되어 있는 송신자의 경우는 송신자로 필터링하지 않는다. 리스트에 없는 송신자의 경우는 송신자의 신뢰여부를 알 수 없으므로 송신자로 필터링할 수 없다.

메일의 내용을 대표하는 제목란의 경우는 '광고'라는 문구만으로 필터링으로 하는데 최근의 스팸메일들이 '광/고', '광*고' 등의 편법으로 이러한 필터링을 피하므로 '광'자와 '고'자 사이에 특수문자 한 두개가 들어간 경우도 체크를 해서 필터링을 하도록 한다.

메일의 크기로도 필터링을 하는데 일반적인 메일의 경우 HTML을 제거했을 때 본문의 내용이 그리 많지 않으므로 본문의 크기가 50KB 이상의 메일은 불법복제 프로그램 판매와 같은 메일이라고 사료됨으로 스팸메일로 간주한다.

이외에도 첨부 화일의 형태나 유무, 송신자 이메일 주소의 도메인, 본문에 포함된 특정문구, 메일을 보낸 시간, 송신자 이메일 아이디의 형태, 참조 리스트 등으로 필터링할 수 있으나 잘못 분류할 가능성이 크다고 생각되어 이 시스템에서는 사용하지 않았다.

3.3 나이브 베이지안 분류자를 이용한 필터링

나이브 베이지안 분류자를 이용한 학습 및 분류 알고리즘은 [알고리즘1]과 같다.

본 연구에서는 받은 메일이 스팸메일인가 아닌가에 촛점을 두고 있으므로 목적값의 집합인 V는 v_{Spam}과 $v_{NonSpam}$로만 이루어진다. Vocabulary는 Examples로부터 추출한 단어들이 들어가는데 추출할 때는 공백문자(space) 및 특수문자를 분리자(delimiter)로 사용한다.

Learn_Naive_Bayes(Examples, V) /* 베이지안 분류기를 학습하는 모듈 */
Examples은 목적값을 갖는 문서들의 집합
V는 가능한 목적값들의 집합
1. Examples에서 발생한 모든 단어 및 토큰들을 모은다
   Vocabulary ← Examples에서 발생한 중복없는 각 단어 및 토큰의 집합
2. $P(v_j)$와 $P(w_k | v_j)$를 계산한다
   $Docs_j$ ← Examples에서 목적값이 v_j인 문서들의 집합
   $P(v_j) \leftarrow \frac{|Docs_j|}{|Examples|}$
   $Text_j$ ← $Docs_j$의 모든 멤버들을 더해서 만든 하나의 문서
   n ← $Text_j$ 안의 단어의 수
   Vocabulary 안의 각 단어 w_k에 대해서
      n_k ← $Text_j$안에서 단어 w_k가 나온 빈도수
   $P(w_k | v_k) \leftarrow \frac{n_k + 1}{n + |Vocabulary|}$

Classify_Naive_Bayes(Doc) /* 문서를 분류하는 모듈 */
문서 Doc에 대해 평가된 목적값을 돌려준다.
a_i는 문서 Doc안의 i번째 위치에서 발견된 단어를 나타낸다
   positions ← Vocabulary에 포함된 단어들의 Doc안의 위치
   v_{NB}를 돌려준다. $v_{NB} = \arg\max_{v_j \in V} P(v_j) \prod_{i \in positions} P(a_i | v_j)$

[알고리즘1] 텍스트 학습/분류를 위한 나이브 베이즈 알고리즘

여기까지의 알고리즘만으로는 스팸일 확률이 임계치 α(α=0.5)보다 크다고 생각되면 스팸으로 간주하게 된다. 잘못된 분류의 비용을 고려한다면 α의 값을 높여서 논스팸 메일을 스팸메일로 분류하는 일을 줄여야 한다.

4. 실험 및 결과

4.1 데이터 집합

본 논문의 실험을 위해서 델파이 6.0과 ACCESS 2000을 사용해서 구현하였으며, 실험환경은 펜티엄3 800MHz, 256MB RAM의 시스템이었다.

트레이닝 및 테스트에 사용할 데이터들은 몇개월에 걸쳐서 모아온 실제 메일이고 데이터들의 구성은 [표1]과 같다. 테스트에 쓰인 모든 데이터는 트레이닝 데이터보다 나중에 온 메일이다.

	Training Data	Test Data
Spam	836	139
NonSpam	105	22
합계	941	161

[표1] 데이터셋의 구성

Cohen은 이메일을 분류할 때 효율성을 고려해서 헤더정보와 본문의 처음 100단어만을 이용했다[1]. 매우 큰 메세지들을 처리하는데 많은 비용이 들기 때문인데, 본 실험 데이터에서도 불법복제 프로그램 판매와 같이 큰 데이터들이 많이 있기 때문에 Cohen과 마찬가지로 헤더정보와 본문의 첫 100단어만을 사용했다.

4.2 실험 평가 기준

문서분류의 성능을 평가하기 위한 기준은 주로 정확도, 재현율, 오류율이 사용된다. 이 실험에서도 이 세 가지 평가방법을 사용한다. 각각

의 정의는 다음과 같다[5].

'스팸' 정확도(*precision*) = $\frac{\text{'스팸'으로 분류된 실제 '스팸' 문서수}}{\text{'스팸'으로 분류된 문서수}}$

'스팸' 재현율(*recall*) = $\frac{\text{'스팸'으로 분류된 실제 '스팸' 문서수}}{\text{전체 '스팸' 문서수}}$

에러율(*Error − rate*) = $\frac{\text{잘못 분류된 문서수}}{\text{분류된 문서수}}$

스팸 문서가 아닌 논스팸 문서에 대한 평가는 '스팸' 대신 '논스팸'을 대입하면 된다

4.3 실험 결과

각 시스템을 구현하여 분류 성능을 비교한 것이 [표2]이다. R은 규칙기반 시스템을, NB는 나이브 베이지안 분류자 시스템을, R+NB는 제안하는 시스템이다.

	R	NB	R+NB
스팸 정확도	100%	92.6%	92.7%
스팸 재현율	82%	99.3%	100%
논스팸 정확도	46.8%	91.6%	100%
논스팸 재현율	100%	50%	50%
오류율	15.5%	7.5%	6.8%
분류 시간	2.5초	324초	114초

[표2] 각 시스템의 성능 비교

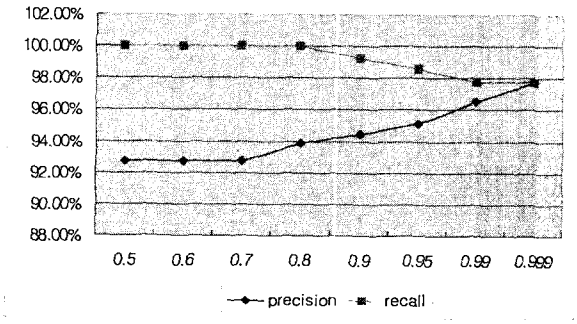
나이브 베이지안 분류자만을 사용한 시스템보다 제안하는 시스템이 모든 면에서 같거나 높은 성능을 보였다. 분류시간의 경우 단어별로 데이터베이스에 질의를 던져서 $P(w_k \mid v_j)$ 값을 구하는 방식으로 구현했기 때문에 헤더정보만을 이용하는 규칙기반 시스템에 비해서 오래 걸렸다. 제안하는 시스템은 규칙기반 시스템으로 먼저 필터링을 함으로써 나이브 베이지안 분류자만을 사용하는 시스템보다 약 3배 빠른 속도를 보였다.

논스팸 재현율의 경우 규칙기반 시스템과 비교해서 매우 저조한 결과를 보였다. 논스팸 재현율을 높이기 위해서 현재 0.5인 임계치 α의 값을 0.999까지 높여가면서 실험을 했는데 α값의 변화에 따른 성능 변화를 나타낸 것이 [그래프1], [그래프2]다.

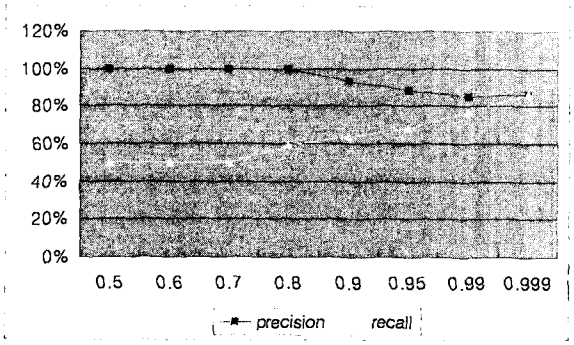
'스팸'메일에 대한 결과는 α값이 커질수록 정확도가 올라가고 재현율이 낮아졌다. '논스팸'메일에 대한 결과는 '스팸'메일에 대한 결과와 반대로 재현율이 올라가고 정확도가 낮아졌다.

두번째 실험결과는 대체적으로 만족스러웠지만 '논스팸'메일에 대한 성능에서 [2],[4],[5]보다는 조금 떨어지는 결과를 보였는데 다음과 같은 두 가지 이유로 추정된다.

첫 번째 이유는 트레이닝 데이터의 논스팸 메일의 수가 적었고 대부분의 논스팸 메일의 길이가 길지 않아서 많은 단어가 학습되지 못했다는 것이고, 두 번째 이유는 영어권의 경우 띄어쓰기를 잘 지켜주지만 본 실험에서 사용한 한국말로 된 논스팸 메일에 띄어쓰기와 맞춤법을 제대로 지키지 않은 경우가 많아서 단어를 직접 비교하는 이 실험에서 높지 않은 결과가 나온 것으로 보인다.



[그래프1] 알파값의 변화에 따른 '스팸'에 대한 정확도 및 재현율



[그래프2] 알파값의 변화에 따른 '논스팸'에 대한 정확도 및 재현율

5. 결론

본 논문에서는 스팸 메일 필터링을 위해서 나이브 베이지안 분류자를 사용한 규칙기반 시스템을 제안하고 구현하였다. 제안한 시스템의 성능을 규칙기반 시스템과 나이브 베이지안 분류자만을 사용한 시스템과 비교 평가하여 오류율, 스팸 재현율, 논스팸 정확도가 향상되었음을 보였다. 또 *임계치의 변화에* 따른 각 '스팸' 및 '논스팸'에 대한 정확도 및 재현율의 변화를 알아보았다.

향후 과제로는 '메일을', '메일이'와 같이 조사만 다른 단어들을 '메일'로 인식시키는 문서 전처리 과정을 추가하거나 [8]에서 SVM이 나이브 베이지안 방법보다 문서 분류 성능이 뛰어나다고 하므로 분류기법을 SVM으로 한다면 보다 향상된 성능을 기대할 수 있을 것이다.

6. 참고문헌

[1] W.W. Cohen. Learning Rules that Classify E-Mail. In Proc. of the AAAI Spring Symposium on Machine Learning in Information Access, Stanford, California, 1996.

[2] M. Sahami, S. Dumais, D. Heckerman, and E. Horvitz. A Bayesian Approach to Filtering Junk E-Mail. In Learning for Text Categorization - Papers from the AAAI Workshop, pp.55-62, Madison Wisconsin. AAAI Technical Report WS-98-05, 1998.

[3] P. Domingos and M. Pazzani. Beyond Independence: Conditions for the Optimality of the Simple Bayesian Classifier. In Proc. of the 13th International Conference on Machine Learning, pp. 105-112, Bari, Italy, 1996.

[4] An Evaluation of Naive Bayesian Anti-Spam Filtering, I. Androutsopoulos, J. Koutsias, K.V. Chandrinos, G. Paliouras and C.D. Spyropoulos, Workshop on Machine Learning in the New Informaion Age, ECML 2000

[5] Yanlei Diao, Hongjun Lu and Dekai Wu. A Comparative Study of Classification Based Personal E-mail Filtering. Proceedings of PAKDD-00, 4th Pacific-Asia Conference on Knowledeg Discovery and Data Mining, 2000.

[6] Mitchell, T.M. Machine Learning. Chapter 6: Bayesian Learning. McGraw-Hill, 1997.

[7] Ricardo Baeza-Yates and Berthier Ribeiro-Neto. Modern Information Retrieval. pp. 165-173. Addison-Wesley, 1999.

[8] Thorsten Joachims. Text Categorization with Support Vector Machines: Learning with Many Relevant Features. European Conference on Machine Learning, 1998.