카이제곱 통계량을 이용한 개선된 베이지안 스팸메일 필터

(An Improved Bayesian Spam Mail Filter based on Ch-square Statistics)

김진상, 최상열
(jsk@kmu.ac.kr)
(053-580-5270)
대구시 달서구 신당동 1000번지
계명대학교 정보통신대학 정보통신학부

요 약

현재까지 개발된 스팸 메일 필터는 주로 베이지안 학습을 이용한 문서분류에 바탕을 두고 있지만, 정확률 향상의 한계라는 문제점과 더불어 일반 메일을 스팸 메일로 오분류하는 치명적인 오류를 극복하지 못하는 문제점을 안고 있다. 본 논문은 카이제곱 통계량을 바탕으로 베이지안 필터의 false positive 에러를 해결하고, 더불어 정확률과 재현율 향상을 동시에 기할 수 있는 스팸 메일 필터링 방법을 기술한다. 또한 본 논문에서 사용된 방법은 사용자의 배경 지식을 기계학습 단계에서 파라미터로 반영하여 시스템의 유연성을 높이고 나아가 개인화된 시스템으로 확장시킬 수 있다는 장점도 있다.

Keywords: 스팸메일 필터, 카이제곱, 베이지안, false-positive error

Abstract

Most of the currently used spam-filters are based on a Bayesian classification technique, where some serious problems occur such as a limited precision/recall rate and the false positive error. This paper addresses a solution to the problems using a modified Bayesian classifier based on chi-square statistics. The resulting spam-filter is more accurate and flexible than traditional Bayesian spam-filters and can be a personalized one providing some parameters when the filter is learned from training data.

Keywords : Spam-filter, Bayesian Classifier, False Positive Error, Chi-square Statistics, Personalization

1. 서 론

전자우편(메일, 또는 전자 메일)은 효율적이고 편리한 통신 수단으로서 지속적으로 사용 빈도가 늘어가고 있다. 하지만 대부분의 편리한 도구들처럼 메일 역시 오남용의 사례가 증가하는 추세이다. 메일의 오남용 사례 중 하나는 '스팸 메일'이라고 하는 요청하지 않은 모르는 메일이 수많은 수신자에게 발송되는 문제이다. 스팸 메일은 보통 자동 메일 발신기를 통해 웹 페이지나 뉴스그룹 사용자들 목록을 통해 얻은 수신자들의 주소로 보내지는데, 스팸 메일의 내용은 광고성 메일에서부터 음란성 메일에 이르기 까지 매우 광범위하다. 스팸 메일의 일반적 특징은 수신자들의 대부분이 그 내용에 관심이 없다는 점이다. 하지만 음란성 스팸 메일을 청소년이나 어린이가 받는 경우에는 심각한 피해를 초래할 수 도 있다. 스팸 메일이 메일 사용자의 시간 낭비를 초래하는 문제점이나 메일 서버에 과부하를 초래하는 경제적인 피해도 매우 중요한 사안이다. 예를 들어, 유럽의 경우 스팸 메일이 모든 메일 트래픽의 70% 이상을 차지한다고 하며[1], AT&T의 분석에 의하면 스팸의 약 11%가 성인물이라고 분석 되었다[2]. 가장 큰 ISP 업체 중 하나인 UUNet은 스팸 퇴치를 위해 연간 약 12억원의 예산을 들여 6명으로 구성된 팀을 운영하고 있으며[3], 또 다른 ISP 업체인 Netcom의 경우 고객이 지불하는 금액의 약 10%가 스팸 메일 처리에 따른 비용이라고 밝히고 있다[4].

스팸 메일을 규제하기 위한 법적인 노력은 현재까지 별다른 효과를 거두지 못하고 있다. 좀 더 효과적인 방법은 도구 프로그램을 사용하여 메일 수신자가 자신에게 전달된 메일들을 검사하여 스팸 메일을 자동으로 확인하고 삭제하도록 하는 것이다. 이러한 도구 프로그램을 스팸 필터라고 하는데, 스팸 메일을 자주 보내는 사람인 스패머를 모은 블랙리스트를 이용하는 필터에서부터 메일의 내용을 분석하여 스팸인지 아닌지를 확인하는 내용기반 필터까지 다양한 종류가 있다. 블랙리스트를 이용한 필터는 스패머가 자신의 메일 주소를 변경시켜가면서 보낼 때는 효율성이 떨어지므로 내용기반 필터가 더욱 강력하다. 이러한 내용기반 스팸 필터는 메일에 들어있는 패턴들을 이용하여 스팸인지 아니면 정상적인 메일인 햄인지를 결정한다. 필터에 사용될 패턴들은 대부분 수동으로 선택해야 하며, 더 강력한 필터가 되기 위해서 사용자 자신이 스스로 패턴을 선택해야 하고 또 지속적으로 관리해 주어야 하며 때로는 전문적인 지식을 필요로 하기도 한다.

메일의 자동 분류에 관한 연구는 지금까지 다양하게 진행되어 왔다. Ruvini와 Gabriel은 Kuback-Leiber 방법을 이용하여 메일 분류기의 가능성을 연구하였으며[5], Mock은 Nearest-neighbor 알고리즘을 이용하여 아웃룩 2000의 메일을 분류할 수 있는 분류기를 실험하였다[6]. 또한 Giorgetti와 Sebastiani는 support vector machine 기법을 응용하여 일반 텍스트의 분류기를 실험하였고 [7], Manco 등은 분류기가 아닌 클러스트링 기법을 적용하여 메일을 비교사 학습하는 실험을 행하였다[8]. 이와 같은 연구는 대부분 기존의 텍스트 분류에서 사용하던 연구 방법[9,10]을 적용한 것으로, 전통적인 베이지안 분류기의 개선을 모색한 다른 연구[11,12]와 대조를 이룬다.

2002년 Graham이 발표한 논문 "A Plan for Spam"[13]은 베이지안 분류기를 이용한 스팸 필터가 가장 우수할 것이라고 주장하였으며, 실제 신경망이나 일반적인 문서 분류 기법에 사용된 필터링 기술보다 베이지안 통계를 이용한 필터가 내용기반 필터로는 가장 우수한 기술이라고 현재까지 평가되고 있다.

이 후 Robinson은 Graham이 제안한 방법이 베이지안 독립성 가정, 희소단어 처리, 단어의 확률 계산, 그리고 비대칭성 등에서 문제점이 있음을 지적하고, 그에 대한 해결책을 가진 새로운 알고리즘을 제시하였다[14, 15]. 그러나 Robinson의 알고리즘 역시 희소단어의 처리와 비대칭성을 완전하게 해결하지 못하는 문제점을 안고 있어서, 본 논문에서는 이에 대한 새로

운 해결책을 제안하고자 한다. 본 논문에서 제시하는 알고리즘은 베이지안 분류기의 확장으로서 Robinson이 제안한 알고리즘의 문제점을 개선한 기법이다.

2. 스팸 필터

스팸 필터란 수신되는 메일을 스캔하여 스팸으로 분류되면 이를 지정된 스팸 폴더로 보내는 프로그램을 의미한다. 스팸 필터는 두 종류의 오류를 일으킬 수 있는데, 첫 번째 오류는 스팸 메일을 일반 메일로 잘못 인식하는 false negative 오류이고, 두 번째 오류는 일반 메일을 스팸 메일로 잘못 인식하는 false positive 오류이다. 하지만 두 가지 오류 중 훨씬 더 치명적인 오류는 false positive 오류이며, 따라서 효율적인 필터를 설계하기 위해서는 false negative 오류를 줄이는 것보다 false positive 오류를 최소화시키는데 노력을 기울여야 한다. 현재까지 스팸 필터의 설계를 위해 사용된 규칙 생성 방법은 크게 수작업으로 생성하는 방법과 교사 학습을 이용한 자동 생성 방법으로 나눌 수 있다.

수작업 규칙 생성 방법은 초기의 필터들이 스팸을 확인하기 위해 수동으로 만든 규칙들을 사용한다. 다음은 스팸이 되기 위한 조건으로 수작업으로 생성한 규칙의 일부이다:

<제목>에 "광고" 또는 "FREE"라는 단어가 있음

<본문>에 "대출" 또는 "Loan"이라는 단어가 있음

<본문> 내용 중 대문자로 표시된 단어들이 한 문장을 구성하고 있음

<발신자>가 숫자로 시작함

이와 같은 방법으로 전형적인 필터를 만들 수 있다. 실제로 스팸 메일 중 약 60%가 제목에 "광고" 또는 "FREE"라는 단어를 포함하고 있다. 또한 이 방법을 사용하면 false positive 오류를 1% 미만으로 줄일 수 있다. 하지만 수작업 규칙 생성 방법은 크게 세 가지 심각한 결점을 안고 있다. 우선, 규칙 생성 작업이 매우 지루한 작업이고, 비경제적이며, 또한 오류를 일으킬 위험이 높다는 점이다. 두 번째 결점은 스팸 메일에 들어 있는 특정 태그나 폰트의 색상 등을 발견해 내기가 어렵고 또 그러한 특징을 규칙화하기도 어렵다는 점이다. 세 번째로는 스패머들이 그러한 일반적 규칙을 피해가기 위해 다양한 변형을 쉽게 만들어 낸다는 점이다. 예를 들어, "광고"라는 단어 대신 "광*고"라는 단어를 <제목>에 사용하여 쉽게 규칙을 피해 갈 수 있기 때문이다.

다음으로 자동 규칙 생성 방법은 베이지안 방법, 메모리 기반, SVM 등 다양한 방법들이 있으나, 본 논문에서는 베이지안 방법을 바탕으로 이를 확장 및 개선하는 기법을 사용한다. 특히 여기서는 Graham과 Robnson의 알고리즘을 분석하고, 분석 결과에서 논의된 문제점을 해결한 새로운 알고리즘을 제안하고 실험을 통해 개선의 효과를 보이는데 목적을 둔다.

2.1 Graham의 알고리즘

Graham [13]은 메일이 특정 단어를 포함하는 경우 그 메일이 스팸인지를 판단하기 위해 베이지안 통계를 사용하였다. 이를 위해 훈련 집합으로 모은 모든 메일들을 대상으로 의미 있는 단어들을 추출하고 각 단어 마다 스팸일 확률과 햄일 확률을 미리 계산하였다. 이러한 확률의 계산 과정은 다음과 같다.

메일에 나타나는 각 단어 w를 대상으로 그 단어의 출현 빈도 $g(w)$를 계산한다. 또한 스팸 메일에 나타나는 각 단어 v를 대상으로 그 단어의 출현 빈도 $b(v)$도 함께 계산한다. 햄 메일의 숫자와 스팸 메일의 숫자를 각각 $ngood$와 $nbad$라고 하면, 단어 x를 포함하는 메일이 스팸일 확률

$s(x)$는 다음과 같다.

$$s(x)=\max\left(0.01, \min\left(0.99, \frac{\min\left(1, \frac{b(x)}{nbad}\right)}{\min\left(1, \frac{g(x)}{ngood}\right)+\min\left(1, \frac{b(x)}{nbad}\right)}\right)\right) \quad (1)$$

식 (1)에서는 메일이 스팸일 확률이 항상 0.01과 0.99 사이에 있도록 하였다. 훈련 집합에 있는 모든 메일을 대상으로 각 단어의 확률을 계산하여 나이브 베이즈(naive Bayes) 정리와 결합한 결과가 식 (1)이므로, 나이브 베이지안 통계의 기본 가정인 출현 단어들 간의 상호 독립성을 적용하고 있다.

새로운 메일이 주어지면 단어들을 추출하여 중요도가 높은 상위 15개의 단어만 사용하였다. 이 때 단어 x의 중요도는 스팸일 확률 $s(x)$가 중간 값인 0.5에서 얼마나 멀리 떨어져 있는가를 나타내는 $|0.5 - s(x)|$로 측정한다. 만약 어떤 메일에서 중요도 순으로 상위 15개 단어가 $<a_1, a_2, \cdots, a_{15}>$라면 그 메일이 스팸일 확률은 식 (2)와 같다.

$$\frac{\prod_{i=1}^{15} s(a_i)}{\prod_{i=1}^{15} s(a_i)+\prod_{i=1}^{15}(1-s(a_i))} \quad (2)$$

이 때 발생하는 문제점 중 하나는 새로 주어진 메일에 처음으로 나타나는 단어가 있는 경우, 즉 훈련 메일에 나타나지 않았던 단어가 있는 경우이다. 이러한 문제점을 해결하기 위해 그는 여러 번의 실험과 시행착오를 통해 처음으로 나타나는 단어 x의 $s(x)$ 값을 0.4가 최적이라고 밝혔다. 그 이유에 대한 휴리스틱은 모든 스팸 메일들이 통상 비슷한 단어들의 집합을 형성하고 있기 때문에 처음으로 나타나는 단어는 스팸이 아닐 것이라는 가설을 바탕으로 한다. 하지만 이상의 방법은 여러 가지 문제점을 안고 있다. 다음은 Robinson이 분석한[15] 문제점의 내용이다.

1) 독립성 가정: 나이브 베이지안의 기본 가정인 단어들이 출현할 확률이 독립이라는 가정은 스팸 메일에서는 적용하기 어렵다. 왜냐하면 "대출"이라는 단어가 나타나면 "연체"라는 단어가 나타날 확률이 매우 높기 때문이다.
2) 희소 단어 처리: 훈련 과정에서 나타나지 않았던 단어를 가진 메일이 주어지면 그 단어가 스팸일 확률을 0.4로 하였다. 하지만 이와 같은 가설에는 상당한 모순점이 있다. 왜냐하면 훈련 과정에서 스팸 메일에 아주 드물게 나타났기 때문에 스팸일 확률이 매우 낮은 단어들보다 처음 나타나는 단어의 스팸일 확률이 더 높게 책정되기 때문이다.
3) 단어 확률의 계산: 식 (1)에 의한 단어의 확률 계산에는 근본적인 문제점이 있다. 왜냐하면 특정 단어를 포함하는 메일이 스팸일 확률을 계산할 때 그 단어가 훈련 집합 메일 전체에 나타나는 빈도수에 바탕을 두고 있기 때문이다. 만약 그 단어가 하나의 메일에 여러 번 나타나는 경우라면 (그래서 빈도수를 높였다면) 다른 메일이 그 단어의 확률에 영향을 미칠 가능성이 낮아지고, 이것은 진정한 의미에서 확률이라고 보기 어렵다.
4) 비대칭: 이 방법은 특정 단어가 스팸으로 분류되는데 미치는 영향과, 반대로 그 단어가 햄으로 분류되는데 미치는 영향 사이에 대칭적인 관계가 성립되지 않는다. 이유는 false positive 오류를 줄이기 위해서라고 하지만 궁극적으로는 두 가지 경우의 영향을 동일하게 유지하면서 성능을 극대화시킬 필요가 있다.

2.2 Robinson의 알고리즘

Robinson은 앞에서 본 것처럼 문제점을 지적하고 동시에 해결책도 함께 제시하였다. 그의 방법은 우선 모든 단어들의 확률을 계산한다. 이 확률들을 결합하기 위해 피셔(Fisher)의 역 카이제곱(inverse Chi-square) 검증을 적용한 후 하나의 척도 H를 구한다. 이렇게 구한 확률의 결합은 본질적으로 스팸에 큰 영향을 미치는 단어들의 확률을 1에 가까운 값으로 계산하는 것이 아니고, 햄에 큰 영향을 미치는 단어들의 확률 결합을 0에 가깝게 만들기 때문에 또 다른 척도 S를 계산하게 된다. 이 척도 S 역시 단어들의 확률을 결합하지만, 이번에는 단어들의 확률을 (1-$f(w)$)로 적용한다. 마지막으로 주어진 메일이 스팸인지 아닌지를 판단하기 위해 두 가지 척도를 결합한 제 3의 척도 I를 사용하게 된다.

2.2.1 단어의 확률 계산

훈련용 메일 집합(말뭉치)이 있다고 가정하고, 이 훈련용 메일 집합은 미리 수동으로 분류하여 햄 메일의 집합과 스팸 메일의 집합으로 나뉘어져 있다고 가정한다. 먼저 훈련용 메일 집합에서 단어들을 추출하여 각각의 단어의 출현이 스팸성 단어일 확률을 구한다. 즉, 말뭉치에 나타나는 각 단어 w에 대해 다음을 계산한다.

- $b(w)$ = w를 포함하는 스팸 메일의 수 / 전체 스팸 메일의 수
- $g(w)$ = w를 포함하는 햄 메일의 수 / 전체 햄 메일의 수 (3)
- $p(w)$ = b(w) / (b(w) + g(w))

위에서 확률 $p(w)$는 단어 w를 포함한 임의로 선택한 메일이 스팸일 확률을 나타낸다. 스팸 필터는 모든 단어 w에 대해 $p(w)$를 계산하고, 이 확률을 기초로 하여 스팸인지를 최종적으로 판단하게 된다.

2.2.2 희소 단어 처리

식 (3)처럼 단어의 확률을 계산하면 빈도수가 매우 작은 단어의 경우 문제가 발생한다. 예를 들어, 어떤 단어 w가 메일에 한 번만 나타나고 그 메일이 스팸이라면 $p(w)$=1이 된다. 하지만 이후 그 단어를 포함하는 새로운 메일이 주어질 때 마다 모두 스팸으로 처리해야 하는지는 의문이다. 이 경우는 훈련을 위한 충분한 메일이 확보되지 않았기 때문에 $p(w)$를 정확히 알 수 없다고 보아야 할 것이다.

따라서 어떤 단어 w가 메일에 한 번만 나타나고 그 메일이 스팸이라면, 다음에 그 단어 w를 포함한 메일이 주어질 때 그 메일이 스팸이란 믿음의 정도는 100%가 아닐 것이다. 그 이유는 사람이 어떤 결정을 할 때 주어진 사실뿐만 아니라 그 사실을 뒷받침하는 배경 지식도 함께 고려되기 때문이다. 경험적으로 볼 때 어떤 단어든지 스팸이나 햄 메일에 나타날 수 있고, 한 두 개 혹은 수백 개의 훈련용 메일에서 단어의 실제 확률을 구할 수 는 없을 것이다.

베이지안 방법은 계산을 통해 구한 단어의 확률과 일반적인 배경지식을 결합할 수 있다는 장점이 있다. 즉, 출현 빈도수가 아주 작은 희소 단어가 다시 나타날 때 그것을 포함하는 메일이 스팸인지 아닌지를 결정할 수 있는 믿음의 정도를 결정할 수 있다. 단어 w에 대한 믿음의 정도 $f(w)$는 다음 식 (4)로 구한다.

$$f(w) = \frac{(s*x) + (n*p(w))}{s+n} \quad (4)$$

이 때, s는 배경 지식에 대한 신뢰의 강도이고, x는 배경지식을 토대로 어떤 단어가 처음으로

스팸에 나타날 확률을 의미하며, n은 수신 메일 중 w를 포함하는 메일의 수이다. 식 (4)를 사용하여 배경지식으로부터 가상적으로 설정한 확률을 표현하는 x를 쉽게 조절할 수 있으며, 또한 그 가상적 설정에 대한 신뢰의 강도를 나타내는 s 역시 효율적으로 조절할 수 있다. 이 두 개의 값 s와 x는 성능을 최적화하기 위한 테스트 과정에서 조절될 수 있으며, 초기에는 s=1 및 x=0.5로 설정한다.

2.2.3 확률의 결합

메일이 확률의 집합으로 표현되기 때문에 각각의 확률을 결합하여 그 메일이 스팸인지 아닌지를 결정할 수 있는 하나의 통합된 기준치가 필요한데, 이를 위해 피셔의 카이제곱 분포 값을 적용하였다.

제로 가설은 "$f(w)$ 값들은 정확하고, 현재 메일 단어들은 임의로 선택한 것이며, 각 단어는 서로 독립이기 때문에 $f(w)$ 값들은 균등 분포가 아니다."이다. 그렇다면 단어 집합 전체를 위한 하나의 확률을 계산하기 위해 피셔 값을 구하는데, 이 값은 메일이 햄이면 0에 가까운 확률의 수가 많으며 동시에 이와 균형을 맞추기 위해 1에 가까운 극소수의 확률을 가지게 된다. 이와 같은 특징은 피셔 값이 최종적으로 아주 작은 (0에 가까운) 확률로 나타나게 됨을 의미한다. 따라서 제로 가설은 기각되고, 주어진 메일이 햄이란 대립 가설을 채택하게 되는 것이다. 이상과 같은 확률의 결합을 H로 표현하며, 구체적인 식은 다음 식 (5)와 같다.

$$H = C^{-1}(-2\ln\prod_{w} f(w), 2n) \tag{5}$$

여기서 $C^{-1}()$는 카이제곱 함수의 역함수이며, 카이제곱 분포의 확률변수로부터 p-값을 찾는데 사용된다.

위의 설명에서 설정한 제로 가설은 항상 거짓임을 알 수 있다. 실제 어떤 메일도 햄이나 스팸 중 어느 한 쪽에 치우치지 않고 완전히 무작위로 추출한 단어로 구성될 수는 없다. 즉, 스팸이나 햄에 관련성이 높은 단어들로 메일이 구성되어 있다고 본다. 메일에 나타난 단어들도 독립이 아니며, $f(w)$ 값 또한 균등분포라고 볼 수 없다. 하지만 스팸을 탐지하기 위해 이와 같은 제로 가설을 설정하는 데는 무리가 없다.

이와 같은 설정은 주어진 메일에 대한 확률들이 무작위가 아니라 매우 높든지 아니면 아주 낮은 방향으로 결정된다. 그 결과 제로 가설은 분명히 기각되고 두 가지 대립가설 중 하나를 선택할 수 있는 강력한 통계적 근거가 마련되는 것이다. 여기서 두 가지 대립가설이란 햄 메일이든지 아니면 스팸 메일을 뜻한다.

개개의 $f(w)$ 값들은 실제 확률들의 근사치일 뿐이다. 하지만 $f(w)$ 값이 계산되는 과정에서 0이나 1로 수렴해 갈수록 실제 확률에 가까이 감을 알 수 있다. 왜냐하면 0이나 1에 가까운 값은 단어들이 훈련 데이터에 아주 빈번히 나타나고 더불어 스팸이나 햄 중 어느 한 쪽에만 집중적으로 나타날 때 가능한 값이기 때문이다. 그럼에도 불구하고 $f(w)$가 0에 가까운 값일 때 훨씬 더 큰 영향을 미친다. 그 이유는 피셔 값이 확률의 곱에 바탕을 두고 있기 때문에 메일이 햄일 가능성을 찾는데 (즉, $f(w)$ 값이 0에 가까운 단어에) 훨씬 더 민감하게 반응하게 된다.

2.2.4 스팸/햄 지시자

지금까지의 계산은 햄 메일을 찾는데 치중해 왔다. 하지만 스팸 메일성 단어들은 $f(w)$ 값이 1에 가깝기 때문에 그들의 곱에 영향을 덜 미치게 된다. 이러한 문제점을 해결하기 위해 먼저 모든 확률의 역 확률인 (1 - $f(w)$) 값을 계산한다. 확률 $f(w)$가 w를 포함한 메일 집합에서 임의

로 선택한 하나의 메일이 스팸일 확률을 나타내기 때문에 (1 - $f(w)$)는 임의로 선택한 메일이 햄일 확률을 확률이 된다. 이 역 확률을 이용하여 또 다른 피셔 값 S를 다음과 같이 계산한다.

$$S = C^{-1}(-2\ln\prod_{w}(1-f(w)), 2n) \tag{6}$$

식 (6)의 S는 스팸성 단어들이 나타날 때 제로 가설을 기각하기 위해 0에 가까운 확률을 결합시킨다. 마지막으로 H와 S를 이용한 새로운 확률 I를 다음과 같이 정의한다.

$$I = \frac{1+H-S}{2} \tag{7}$$

식 (7)에서 구한 I는 주어진 메일이 스팸에 가까울수록 1에 가까운 값, 햄에 가까울수록 0에 가까운 값을 가지는 스팸 또는 햄을 나타내는 지시자가 된다. 물론 I가 0.5의 값을 가지는 메일은 결정이 불가능한 메일을 의미한다.

2.3 문제점 분석 및 개선책

하지만 2.2절에서 살펴 본 Robinson의 알고리즘에는 다음과 같은 문제점이 있다.

1) 확률의 계산: 식 (3)에서 정의한 단어 w가 스팸성 단어일 가능성 $b(w)$, 햄성 단어일 가능성 $g(w)$, 그리고 w를 포함한 메일이 스팸 메일일 확률 $p(w)$는 훈련 집합이 충분히 크고 동시에 w를 포함한 메일이 많을 때 유효하다. 하지만 스팸 메일에는 통상 스팸성 단어가 여러 번 반복해서 나타나는 경우가 많다. 이와 같이 메일에 나타나는 단어의 빈도수를 확률에 반영할 수 있는 방법이 없기 때문에 개선이 필요하다.
2) 분류기의 크기: Robinson의 알고리즘은 메일에 나타나는 단어를 그대로 추출하여 기계학습에 사용한다. 결과적으로 분류기의 크기가 매우 커지고 따라서 분류 시간이 많이 걸릴 수 있다. 만약 스팸 필터가 개인 클라이언트용이라면 대용량 분류기의 유지와 관리는 큰 제약이 될 수 있다.
3) 공통 단어: 이 알고리즘에는 스팸 메일 집합과 햄 메일에 공통으로 나타나는 단어들도 모두 포함하여 확률을 계산하고 있다. 하지만 일정 빈도 수 이상 공통으로 타나나는 단어들은 분류 정확률을 떨어트릴 뿐만 아니라 분류기의 크기도 증가시키기 때문에 공통 단어들은 제거하는 것이 더 효율적이다.

3. 스팸 필터의 설계와 구현

앞에서 살펴본 Robinson의 알고리즘의 문제점을 해결할 수 있는 방안은 다음과 같다. 먼저 확률을 계산할 때 스팸 및 햄 메일의 빈도수뿐만 아니라 단어의 출현 빈도수 역시 반영할 수 있는 방법을 개발한다. 다음으로 분류기의 크기를 줄이기 위해 훈련과 분류 모두 단어들에 대해 스태밍을 한다. 마지막으로 공통 단어 문제를 해결하기 위해 임계값을 정하여 스팸 메일과 햄 메일에 임계값 이상 나타나는 단어를 제거하고, 나아가 분류시에도 특정 범위를 넘어가지 못하는 메일들을 미분류 메일로 남겨두도록 한다. 이와 같은 개선책을 바탕으로 한 알고리즘은 다음과 같다.

3.1 학습 및 분류 알고리즘

훈련용 메일 집합(말뭉치)이 있다고 가정하고, 이 훈련용 메일 집합은 미리 수동으로 분류하여 햄 메일의 집합과 스팸 메일의 집합으로 나뉘어져 있다고 가정한다. 먼저 훈련용 메일 집합에서 단어들을 추출하여 각 단어가 스팸성일 확률을 구한다.

1) 말뭉치에 나타나는 각 단어 w에 대해 다음을 계산한다.

$$b(w) = \frac{w\text{를 포함하는 스팸 메일의 수}}{\text{전체 스팸 메일의 수}}$$

$$g(w) = \frac{w\text{를 포함하는 햄 메일의 수}}{\text{전체 햄 메일의 수}} \quad (7)$$

$$d(w) = \frac{b(w)}{b(w)+g(w)}$$

이 계산 과정에서 확률 $d(w)$는 식 (3)의 $p(w)$와 같다.

2) 각 단어 w에 대해 빈도수를 바탕으로 다음을 계산한다.

$$x(w) = \frac{\text{스팸 메일에 나타나는 } w\text{의 빈도수}}{w\text{를 포함하는 스팸 메일의 수}}$$

$$y(w) = \frac{\text{햄 메일에 나타나는 } w\text{의 빈도수}}{w\text{를 포함하는 햄 메일의 수}} \quad (8)$$

$$z(w) = \frac{x(w)-y(w)}{x(w)+y(w)}$$

여기서는 단어의 빈도수를 바탕으로 스팸성 정도를 계산한 것이다.

3) 단어 확률을 구하기 위한 두 확률의 결합

$$p(w) = w1^* d(w) + w2^* z(w),\ \text{단}\ w1, w2 \geq 0,\ w1 + w2 = 1 \quad (9)$$

이 때 $w1$과 $w2$ 는 $d(w)$와 $z(w)$에 각각 곱해지는 가중치이며, 실행시 사용자에 의해 주어지는 값이다.

4) 단어 확률의 믿음의 정도 계산

$$f(w) = \frac{s^* x + n^* p(w)}{s+n} \quad (10)$$

5) 세 가지 지시자 계산

$$H = C^{-1}(-2\ln\prod_w f(w), 2n)$$

$$S = C^{-1}(-2\ln\prod_w (1-f(w)), 2n) \quad (11)$$

$$I = \frac{1+H-S}{2}$$

위 4)단계와 5)단계는 Robinson의 알고리즘에서 설명한 내용과 같다.

3.2 스팸 필터의 구현

스팸 필터의 구현은 기본적으로 식 (7)에서 (11)까지의 내용을 바탕으로 하지만, 몇 가지 전처리와 구현에 따른 세부 사항들이 있다.

1) 파서: 메일을 읽어서 단어를 생성하는 부분으로서, 단어를 추출할 때 <Date>와 <To> 필드는 삭제하며, 이중 따옴표, 괄호(<,>,(,),{,}), 물음표 등도 삭제한다.
2) 스태머: 파서의 출력으로 나온 단어들은 스태머의 입력으로 들어가는데, 스태머는 주어진 단어를 기본형으로 변형시키는 역할을 한다. 예를 들어, "working", "worked" 등은 모두 기본형 "work"로 바꾸어 준다. 이렇게 스태밍된 단어들은 사전에 저장하게 되는데, 여기서는 Porter의 스태머를 사용하여 처리하였다.
3) 사전: 이것은 단어들의 해시(hash) 테이블로서, 해시 테이블의 엔트리는 [단어, 그 단어가

나타나는 스팸들, 그 단어가 나타나는 햄의 수, ...]와 같이 구성되고, 단어가 주어지면 해당 엔트리를 찾게 된다.

4) 테스터: 스팸 필터의 핵심부로서, 메일을 파싱하고 스태밍한 결과를 입력으로 받아 각 단어가 스팸성일 확률과 햄성일 확률을 계산한다. 다음으로 각 단어들의 확률을 결합한 것과 자유도 $2n$을 이용하여 카이제곱 함수의 역함수를 구한다. 마지막으로 지시자 I의 값을 구하여 0.55보다 크면 스팸으로 분류하고, 0.45보다 작으면 햄으로 처리한다. 만약 I 값이 0.55와 0.45 사이면 판정이 불가능으로 보고 미분류 메일로 처리한다.

4. 실험결과 및 성능 평가

4.1 성능 평가 척도

N_{ham}과 N_{spam}을 각각 햄 메일과 스팸 메일의 전체 개수라고 한다. 또한 $N_{Y\to Z}$는 Y 군에 속하는 메일이 필터에 의해 Z 군으로 분류된 메일의 수라고 하며, 이 때 Y와 Z는 $\{ham, spam\}$의 원소이다. 스팸 필터의 정밀도(accuracy)와 에러율(error)는 다음 식으로 정의된다.

$$Accuracy(\text{정밀도}) = \frac{N_{ham\to ham} + N_{spam\to spam}}{N_{ham} + N_{spam}}$$

$$Error = \frac{N_{ham\to spam} + N_{spam\to ham}}{N_{ham} + N_{spam}}$$

또한 스팸에 대한 재현율(SR)과 정확률(SP)은 다음 식으로 측정된다.

$$SR = \frac{N_{spam\to spam}}{N_{spam}},\ SP = \frac{N_{spam\to spam}}{N_{spam\to spam} + N_{ham\to spam}}$$

스팸 재현율은 필터가 차단한 스팸 메일의 비율을 나타내고, 스팸 정확률은 차단한 메일이 실제로 스팸인 정도를 측정한 것이다. 따라서 스팸 재현율은 필터의 효과성이고 스팸 정확률은 필터의 안전성이라고 할 수 있다. 마지막으로 오분류가 일어나는 경우 이를 false positive(FP)와 false negative(FN)로 구분하여 정의하면 다음과 같다.

$$FP = \frac{N_{ham\to spam}}{N_{ham}},\ FN = \frac{N_{spam\to ham}}{N_{spam}}$$

4.2 실험 결과 분석

실험을 위해 오픈 소스 프로젝트인 SpamAssassin 스팸 필터링 프로그램에서 사용된 훈련 및 테스트용 메일을 수집하였으며, 이 메일 집합은 햄 메일과 스팸 메일로 분류가 되어져 있다. 총 36815개의 메일을 사용하였으며, 이 중 훈련 데이터로는 7166개의 햄 메일과 18447개의 스팸 메일을 사용하였고, 테스트를 위한 메일로는 2727개의 햄 메일과 8475개의 스팸 메일을 사용하였다.

비교를 위해 먼저 Robinson의 알고리즘으로 실험하였다. 배경 정보에 따른 믿음의 강도인 s와 임의의 단어가 처음으로 스팸에 나타날 확률 x를 직접 입력하며, 이 입력 값에 따라서 메일 분류 결과가 달라짐을 알 수 있다. 표-1은 초기 값을 s=1, x=0로 시작하여 0.05의 간격으로 s값은 감소시키고, x값은 증가시켜 총 20회의 테스트를 하였다. 초기 값 x는 스팸 메일에 나타날 확률이므로 값이 0일 경우 스팸 메일은 분류가 되지 않는다. 배경 정보에 따른 믿음의 강도인 s가 0일 경우 확률 I가 미결정 값인 0.5에 가까운 값을 가져서 미분류 메일들이 많아진다. 하지만 FP 율이 0이면서 FN 율이 최소가 되는 s와 x값을 찾기 위해 s는 0.75에서 0.8사이, 그리고

x는 0.25와 0.2사이의 구간을 더 나누어 실험한 결과 s=0.793, x=0.207일 때 FP=0%와 FN=12.2596%가 되었으며, 이 때 정밀도는 84.95%, 에러율은 9.275%, 스팸 재현율은 80.779%, 스팸 정확률은 100%가 되었다.

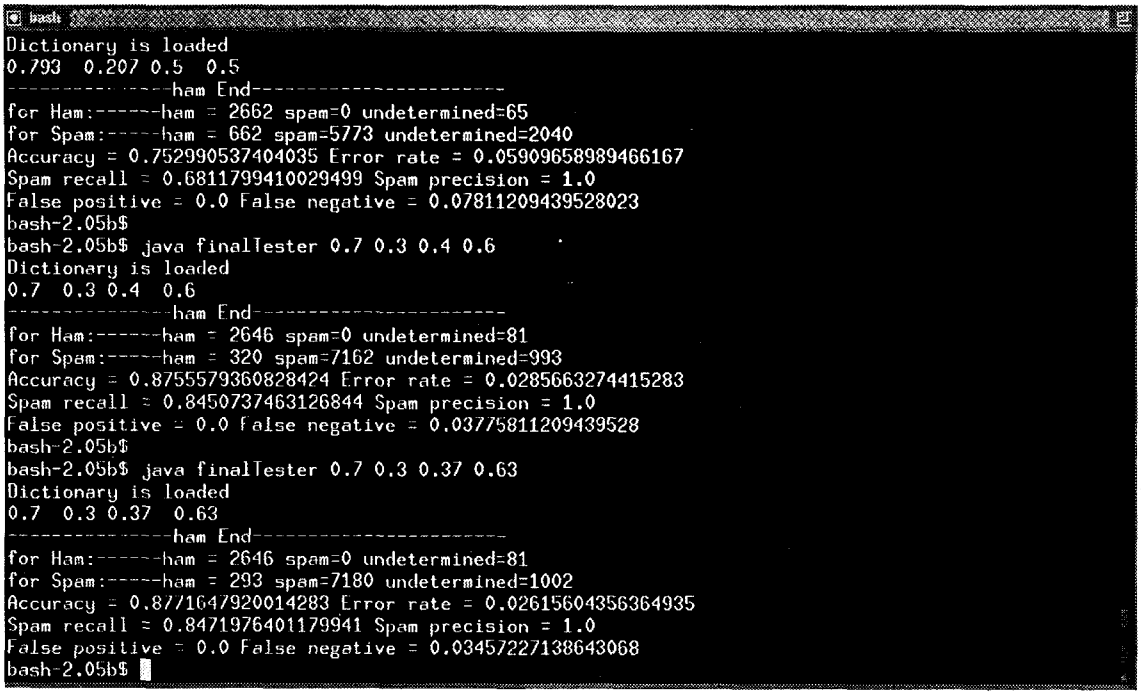
다음으로 본 논문에서 제안한 개선된 Robinson 방법을 시험하기 위해 같은 훈련용 메일 집합과 테스트용 메일 집합을 대상으로 s, x, $w1$, $w2$ 값을 유사한 방법으로 변경하면서 시험하였다. 이 경우 최적의 값은 그림-1처럼 s=0.7, x=0.3, $w1$=0.37, $w2$=0.63일 때, FP=0%와 FN=3.457%가 되었으며, 이 때 정밀도는 87.716%, 에러율은 2.616%, 스팸 재현율은 84.720%, 스팸 정확율은 100%로 나타나 훨씬 성능이 개선되었음을 알 수 있다.

s,x	햄->햄	오분류햄	미분류햄	스팸->스팸	오분류스팸	미분류스팸
1, 0	2690	0	37	0	5870	2605
0.95, 0.05	2682	0	45	3287	1938	3250
0.9, 0.1	2680	0	47	4899	1714	1862
0.85, 0.15	2676	0	51	6081	1418	976
0.8, 0.2	2674	0	53	6751	1096	628
0.75, 0.25	2670	2	55	7253	720	502
0.7, 0.3	2666	4	57	7613	351	511
0.65, 0.35	2654	4	69	7761	150	564
0.6, 0.4	2642	8	77	7885	86	504
0.55, 0.45	2632	8	87	8021	35	419
0.5, 0.5	2615	10	102	8163	8	304
0.45, 0.55	2601	10	116	8281	6	188
0.4, 0.6	2580	12	135	8322	4	149
0.35, 0.65	2558	29	140	8358	3	114
0.3, 0.7	2515	51	161	8365	2	108
0.25, 0.75	2429	122	176	8367	1	107
0.2, 0.8	2119	228	380	8382	1	92
0.15, 0.85	1333	305	1089	8396	0	79
0.1, 0.9	512	358	1857	8401	0	74
0, 1	0	0	2727	27	0	8448

[표-1] Robinson 알고리즘 테스트 결과

4.3 고찰

두 가지 알고리즘으로 실험한 결과를 나타내는 표-1과 그림-1에서 알 수 있듯이 Robinson의 알고리즘은 FP 율을 0으로 낮출 수 있다는 장점이 있지만 단어의 빈도수를 확률에 반영할 수 없기 때문에 정밀도, 에러율, 스팸 정확률, 그리고 스팸 재현율 모두에서 개선의 여지가 있음을 보인다. 따라서 본 논문에서 개선한 알고리즘은 같은 데이터 집합에 대해 표-2와 같이 오류에 해당하는 FN 율과 에러율은 각각 8.8%와 6.7% 감소효과가 있고, 정확성에 해당하는 정밀도와 재현율은 각각 2.8% 및 3.9%의 상승효과를 얻을 수 있다.



[그림-1] 개선된 알고리즘 테스트 결과

	최적의 파라미터	FP 율	FN 율	정밀도	에러율	정확율	재현율
Robinson 알고리즘	s=0.793 x=0.207	0%	12.260%	84.95%	9.275%	100%	80.779%
개선된 알고리즘	s=0.7 x=0.3 $w1$=0.37 $w2$=0.63	0%	3.457%	87.72%	2.616%	100%	84.720%
개선된 내용		동일	8.8% ↓	2.77% ↑	6.659% ↓	동일	3.941% ↑

[표-2] 개선 효과의 비교

5. 결론 및 향후 과제

현재 대부분의 메일 이용자들은 스팸 메일의 처리를 위해 많은 시간과 비용을 소비하고 있다. 효과적인 스팸 필터링을 위해 본 논문에서는 카이제곱 통계량을 이용한 통계적 학습 알고리즘인 Robinson의 알고리즘을 개선하고, 이를 이용한 스팸 필터를 설계 및 구현하고 실험을 통해 효과성을 입증하였다. 본 논문에서 제안한 시스템은 메일에서 특성 단어 추출을 위한 전처리부분과 이 단어들이 햄이나 스팸에 나타나는 빈도수를 측정하여 단어 리스트를 만드는 학습부분, 새로운 메일을 학습을 통해 만들어진 단어 사전과 비교하여 분류하는 단계로 구성된다.

학습을 위해 이미 분류되어진 메일을 이용하여 사용자가 입력하는 믿음의 정도와 주어진 단어가 처음으로 스팸에 나타날 확률 값, 그리고 단어 빈도수에 대한 믿음의 정도를 기준으로 20

회의 테스트를 하였다. 각각의 입력 값에 따른 분류 결과에서 사용자 입력 값을 조절하여 최적의 조건을 찾아내고 이 때 개선된 알고리즘의 성능 향상을 확인할 수 있었다.

하지만 실험 결과에서도 알 수 있듯이 최적의 성능을 나타내는 파라미터의 값은 Robinson의 알고리즘과 개선된 알고리즘에서 서로 상이하며, 그 이유에 대해서는 좀 더 세밀한 이론적 분석이 필요하다. 또한 본 논문은 훈련용 스팸 메일 확보의 어려움으로 인해 영문 메일에만 한정하였기 때문에, 한글 스팸 메일에 대한 추가 연구도 병행되어야 할 것이다.

6. 참고 문헌

[1] BlackSpider Technologies, A Buyers Guide to Spam Filtering, 2004.

[2] C. Lorrie and L. Brian. Spam, AT&T Labs, Technical Report 98.2.1, March 1998.

[3] Internet Week, May 4, 1998. CMP Media Inc, Manhasset, New York, 1998.

[4] New York Times, March 19, 1998.

[5] J. Ruvini and J. Gabriel. Do Users Tollerate Errors from their Assistant? Experiments with an E-mail Classifier, IUI'02, 2002.

[6] K. Mock, An Experimental Framework for Email Categorization and Management, SIGIR '01, 2001.

[7] D. Giorgetti and F. Sebastiani. Automating Survey Coding by Multiclass Text Categorization Techniques, Journal of American Society for Information Science and Technology, 54(14):1269-1277, 2003.

[8] G. Manco, E. Macciari, M. Ruffolo and A. Tagarelli. Towards an Adaptive Mail Classifier, Technical report, ISI-CNR, 2003.

[9] F. Sebastiani. Machine Learning in Automated Text Categorization, ACM Computing Survey, 34(1): 1-47, 2002.

[10] K. Williams and R. Calvo. A Framework for Text Categorization, Proc. of the 7th Australasian Document Computing Symposium, 2002.

[11] Y. Tsuruoka and J. Tsujii. Training a Naive Bayes Classifier via the EM Algorithm with a Class Distribution Constraint, 2000.

[12] J. Rennie. Tackling the Poor Assumptions of Naive Bayes Text Classifier, ICML 2003 Tutorial Notes, 2003.

[13] Paul Graham 2002. A Plan for Spam URL: http://www.paulgraham.com/spam.html

[14] Gary Robinson 2003 Spam Detection URL: http://radio.weblogs.com/0101454/categories/spam/

[15] Gary Robinson article in the Linux Journal march 2003 issue 107 http://www.linuxjournal.com/article.php?sid=6467