자동 생성 메일계정 인식을 통한 스팸 필터링
(Spam-Filtering by Identifying Automatically Generated Email Accounts)

이 상 호 †
(Sangho Lee)

요 약 본 논문에서는 기존의 스팸 메일 필터링 시스템의 성능을 향상시키기 위한 새로운 필터링 방법을 설명한다. 대부분의 스팸 필터링 시스템은 메일의 제목이나 혹은 그 문서 안에서 발견되는 단어들의 분포를 조사하여 이루어진다. 한편, 최근의 스팸 발송자들은 메일 서비스 업체가 제공하는 웹메일 계정을 이용하여 스팸을 발송하기 시작하였다. 이렇게 웹메일을 통해 발송되는 스팸 메일의 특징을 보면, 그 메일 계정이 자동으로 생성되기 때문에 일반 사용자의 메일 계정과 많은 차이를 보인다. 본 연구에서는 이러한 점에 착안하여, 발송자의 메일 계정이 자동 생성된 메일 계정인지를 예측하고 이를 통해 스팸을 필터링하고자 한다. 메일 계정을 분류하기 위해서는 패턴 인식 문제에서 사용되어 온 결정 트리를 이용하였으며, 메일 서비스 업체로부터 수집된 약 215 만개의 메일 계정에 대해 실험하였다. 실험 결과, 96.3%의 정확률을 나타내었으며, 기존 시스템과 연동하여 새로운 형태의 스팸을 필터링할 수 있었다.

키워드 : 패턴 인식, 스팸 필터링, 결정 트리, 이메일

Abstract In this paper, we describe a novel method of spam-filtering to improve the performance of conventional spam-filtering systems. Conventional systems filter emails by investigating words distribution in email headers or bodies. Nowadays, spammers begin making email accounts in web-based email service sites and sending emails as if they are not spams. Investigating the email accounts of those spams, we notice that there is a large difference between the automatically generated accounts and ordinaries. Based on that difference, incoming emails are classified into spam/non-spam classes. To classify emails from only account strings, we used decision trees, which have been generally used for conventional pattern classification problems. We collected about 2.15 million account strings from email service sites, and our account checker resulted in the accuracy of 96.3%. The previous filter system with the checker yielded the improved filtering performance.

Key words : pattern recognition, spam-filtering, decision tree, e-mail

1. 서 론

스팸 메일이란 내용이 상업적인가 비상업적인가에는 무관하게 사용자가 요청하지 않은 정보를 사용자의 의지와 무관하게 대량으로 전달하는 전자우편을 말한다. 단연, 광고성 정보가 많다. 스팸성 광고를 발송하는 발송자는 대량으로 메일을 발송해야 광고 효과를 볼 수 있기 때문에, 일반 메일 서버와 다르게 매우 많은 메일을 발송한다. 이러한 스팸 메일은 그 메일을 받는 사람들에게 정신적인 스트레스를 줄 뿐만 아니라, 공공의 성격을 지닌 네트워크 자원을 아무런 비용의 지불없이 사용하므로 사회 전반에 걸쳐 상당한 피해를 준다고 볼 수 있다.

스팸 메일이 이렇게 사회문제가 되자 많은 웹 메일 서비스 업체들은 스팸 메일을 필터링하기에 이르렀으며, 동시에 스팸 메일을 필터링하는 방법이 연구자들에 의해 고안되었다. 임의의 메일을 스팸인지 아닌지 구별하는 문제는 메일을 입력으로 하는 패턴 인식 문제로 볼 수 있다. 그러므로 대부분의 기존 방법들은 메일의 제목과 내용에 있는 단어들을 관측 대상으로 보고, 이로부터 특징 벡터를 추출한 후 패턴 분류기를 이용한다. 일반적으로 특징 값들로는 메일 내 단어를 이용하고 패턴 분류기로는 나이브 베이지안 분류기[1-3], 신경회로망 분류기[4], SVM(Support vector Machine) 분류기[5,6] 등이 현재까지 사용되었다. 이 방법들은 단어의 발생 분

† 정 회 원 : 한국산업기술대학교 게임공학과 교수
sangholee@kpu.ac.kr
논문접수 : 2004년 7월 16일
심사완료 : 2005년 3월 11일

포가 스팸/비스팸 집합에서 서로 다르다는 점에 착안한 방법이며, 가장 널리 사용되고 있는 방법은 이 중 베이지안 필터링 방법이다[7].

스팸 필터링 방법에는 이와 같이 메일의 내용을 조사하는 방법 이외에도 스팸 메일의 발송 패턴을 이용하여 필터링할 수 있다. 스팸 발송자들이 메일을 보내는 방식들을 살펴보면, 초기에는 단순하게 특정한 컴퓨터(특정한 IP 주소)에서 대량의 메일을 보내는 방식이 주류를 이루었다. 이러한 방식으로 보내진 메일들에 대해 대부분의 스팸 메일 차단 시스템들은 특정 IP에서의 과다 메일발송 감지 방법을 통해 메일들을 차단할 수 있었다. 이러한 차단 방식을 깨달은 스팸 발송자들은 좀 더 정상메일처럼 보이기 위해서 특정한 하나의 IP 주소가 아닌 다수의 발송 IP(다수의 PC)에서 스팸을 보내기 시작했다. 이 방법은 특정 프로그램을 인터넷에 유포시켜 그 프로그램을 내려 받은 사용자의 컴퓨터는 사용자가 모르는 상황에서 스팸 메일의 발송지가 되는 것이다. 하지만 이러한 방법에 대해서도 최근의 스팸 필터링 시스템들은 IP 추적을 통해 대부분 스팸 메일을 구별해내기에 이르렀다.

결국, 최근의 스팸 발송자들은 좀 더 발전된 형태의 스팸 발송 방법을 고안해 내었는데, 그것은 모든 정상적인 절차에 따라 스팸을 발송하는 것이다. 즉 대형 메일 서비스 업체(예: yahoo, hotmail, hanmail) 웹메일 계정으로 스팸메일을 보내는 방법이다. 유명사이트에 발송용 계정을 대량으로 생성한 후에 그 계정을 이용해서 메일을 발송하는 것으로서, 수신자 측에서 보면 그 사이트에서 보내오는 일반 메일과 구분이 되지 않기 때문에, 기존의 스팸 차단 방식으로는 스팸 메일을 구분해낼 수 없다. 이 방법의 특징을 살펴보면, 스팸 메일 발송자들은 하룻밤 사이에 무료 이메일 계정을 10만개 이상 만들고, 그 계정을 이용해서 한 두통씩 보내기 때문에, 스팸 차단 시스템 쪽에서는 사이트 자체를 막기도 힘들고 특정 ID가 반복되는 것으로도 막기 힘든 어려움이 있다. 다만 자동적으로 이메일 계정들을 만들기 때문에, 계정 문자열이 의미 없는 문자열에 가깝고 또한 일반적인 계정보다 다소 문자열이 길다는 특징이 있다. 인간은 이렇게 자동 생성된 계정 문자열을 보면, 거의 확실하게 자동 생성되었는지를 판단할 수 있지만, 차단 시스템에 이러한 능력을 부여하는 것은 매우 도전적인 과제라고 할 수 있다.

본 논문의 목적은 발송자의 이메일 계정을 입력으로 받아 이를 프로그램에 의해 자동 생성된 메일 계정인지를 판단하고자 한다. 이를 위해 트리 구조 분류기인 CART(Classification and Regression Trees)[8]를 이용하여 해결하고자 한다. CART는 예측 변수에 의해 크게 결정 트리와 회귀 트리로 나뉘어지는데, 이 중 결정 트리[9]를 사용하여 문제를 해결한다. 본 논문의 구성은 다음과 같다. 2장에서는 가장 널리 사용되고 있는 나이브 베이지안 분류기를 이용한 필터링 방법과 그 방법의 문제점에 대해 논의한다. 3장에서는 본 연구에서 사용하고 있는 CART 패턴 분류기의 소개와 이를 이용한 계정 분류 방법을 설명하고, 4장에서 본 연구의 의의와 함께 결론을 맺는다.

2. 관련 연구

현재 가장 널리 알려진 스팸 메일 필터링 방법은 나이브 베이지안 분류기를 이용한 방법이다[1-3]. 이 방법은 입력 메일에서 나타나는 n개의 단어 열 $w_1 w_2 \ldots w_n$ 이 주어졌을 때 입력 메일이 스팸 메일일 확률과 정상 메일일 확률을 조사하여 그 값이 큰 메일로 분류하는 방법이다. 우리가 알고 싶은 목적 클래스 $m^* \in \{spam, non-spam\}$ 는 다음과 같이 전개될 수 있다.

$$
\begin{aligned}
m^* &= \arg \max_{m \in \{spam, non-spam\}} P(m \mid w_{1..n}) \\
&= \arg \max_{m \in \{spam, non-spam\}} P(w_{1..n} \mid m) P(m) / P(w_{1..n}) \\
&= \arg \max_{m \in \{spam, non-spam\}} P(w_{1..n} \mid m) P(m) \\
&\cong \arg \max_{m \in \{spam, non-spam\}} P(m) \sum_{i=1}^{n} P(w_i \mid m)
\end{aligned}
$$

위 식들 중 마지막 식을 관찰해보면, 임의의 클래스로부터 단어 열이 발생할 확률 $P(w_{1..n} \mid m)$ 을 $\sum_{i=1}^{n} P(w_i \mid m)$ 로 근사화한 것을 볼 수 있다. 이는 메일을 분류하는데 단어의 위치 정보가 많은 영향을 주지 않을 것이라는 가정에 기인한 것이다. 실제로 이 방법은 구현이 용이하면서도 효과적인 방법으로 알려져 있다[7].

한편, 본 연구의 대상인 스팸 필터링 분야는 컴퓨터 바이러스 분야와 유사하게, 새로운 스팸 필터링 기술이 소개될 때마다 스팸 발송자들은 그 방법을 교묘히 빠져나가는 방법을 고안해낸다. 기존의 필터링 시스템들이 위의 베이지안 방법을 이용한다는 것을 알아낸 스팸 발송자들은 최근에 "word salad"라는 방법을 이용하여 스팸 메일을 정상 메일처럼 보이게 하였다[10]. 이 방법은 스팸 메일 내에 정상적인 단어들을 추가하는 방법이다. 즉, 정상 메일에서 자주 발생하는 단어들을 모은 뒤, 그 단어들을 원래 메일에서 사용하는 단어들의 개수보다 훨씬 많이 넣게 되면, 필터링 시스템은 그 메일을 정상

메일로 간주하게 되는 것이다. 이러한 메일은 html 형식으로 작성되어져 보내지고, 삽입된 단어들의 색깔을 바탕색과 동일하게 하여 그 단어들을 감춘다. 결론적으로 사용자는 그 메일을 읽는데 전혀 어려움이 없게 되고, 필터링 시스템은 스팸 메일을 정상 메일로 잘못 분류하게 된다.

이외에도 스팸 발송자들은 다음과 같은 여러 가지 방법들을 동원하여 스팸 메일을 정상 메일처럼 보이게 하고 있다[11]. "Daily News" 방법은 스팸 메일을 보낼 때 그 날의 뉴스를 메일 앞 단에 붙이고, 그 부분에 잘못된 html 태그를 붙이는 방법이다. 베이지안 필터링 시스템은 이러한 메일을 정상메일로 간주하게 되고, 웹 브라우저는 잘못된 태그 안의 뉴스 내용을 화면에 출력하지 않게 된다. "Slice and Dice" 방법은 메일 내 모든 글자들 사이에 태그를 넣어서 화면에는 정상적인 스팸 메일 내용이 출력되게 하고, 필터링 시스템은 완전한 형태의 단어를 하나도 인식하지 못하게 하는 방법이다. 이와 비슷한 방법으로 모든 글자들의 사이에 공란을 넣거나, '*'를 넣는 "lost in space" 방법이 있다. 이 방법은 모든 글자 내에 '*'를 넣어도 원래 단어를 유추할 수 있다는 인간의 능력을 이용한 방법으로, 물론 필터링 시스템은 완전한 형태의 단어를 하나도 입력받지 못하게 된다.

지금까지 설명한 방법들은 모두 스팸 메일의 내용을 조작하여 베이지안 필터를 통과시키는 방법들이다. 대부분의 필터링 시스템들은 스팸을 찾아내기 위해서 각각의 스팸 메일 형태에 대응하는 필터링 방법을 복합적으로 이용하고 있다. 본 연구는 메일 형태를 분석하는 기존 시스템의 성능을 높이기 위해서 스팸 메일의 발송 패턴을 분석하는 방법으로, 기존의 필터링 시스템을 통과하는 스팸 메일들을 분류하는 것에 목적이 있다.

3. CART를 이용한 메일 계정 분류

패턴 분류를 위해서는 기본적으로 특징 벡터를 정의하고 그 특징 벡터를 분류하는 패턴 분류기가 필요하다. 본 논문에서는 이메일 계정 문자열을 관측 대상으로 하고, 트리 구조 분류기의 일종인 CART를 이용하여 계정 문자열을 분류하고자 한다.

3.1 CART

비모수적 통계 분류기인 트리 구조 분류기는 트리의 예측 오류를 최소화하는 방향으로 특징 벡터 $\boldsymbol{x}$의 공간을 연속적으로 나누는 기법으로[8], 예측될 변수 y가 카테고리 변수인 경우와 실변수인 경우, 각각에 대해 결정 트리와 회귀 트리를 생성하게 된다. 이 방법의 장점들로는 특징 벡터 $\boldsymbol{x}=(x_1 x_2 \ldots x_n)$를 이루는 각각의 원소 x들의 변수 타입에 무관하게 동일한 틀 안에서 학습이 이루어진다는 점과 학습된 트리의 해석이 매우 용이하다는 점이다.

본 연구에서 사용하는 트리 기반 모델링 기법은 CART (classification and regression trees)를 기반으로 한다. 트리 생성의 기본 절차인 노드의 분할과정과 트리 제거 방법은 각각 Chou의 분할 알고리즘[12]과 CART의 비용-복잡도 제거(cost-complexity pruning) 방법을 적용하였다[8]. 노드의 불순도를 나타내는 함수로는 Gini 인덱스를 사용하였으며, 최적 트리는 CART에서 제안하고 있는 10차 교차 시험(10-fold cross-validation) 방법에 의해 구해졌다. 최적 트리를 구하는 다른 방법으로는 C4.5에서 제안하는 노드 오류율의 신뢰도를 이용하는 방법[13]이 있고, CART의 선택 방법은 오류의 교차 시험 추정치 $R^{cv}(T)$가 최소인 트리 T를 선택하는 방법이다. 이 값은 총 열 개 트리들의 오류율 평균값이며, 동시에 오류율의 표준편차를 구하여 새로운 실험 자료에서의 성능을 미리 예측할 수 있다. 실험에 사용된 CART 학습 프로그램은 C 언어로 구현하였으며, 본 실험에서의 트리 학습 시간은 2.8 GHz의 Pentium PC에서 약 10시간 정도였다.

CART 트리에 관한 표현법은 다음과 같다. 구해진 트리와 트리의 노드는 각각 T와 t로 표현하고, 트리의 단말 노드 집합은 $\tilde{T}$, 단말 노드의 개수는 $|\tilde{T}|$로 표현한다. 학습 혹은 실험에 사용된 N개의 데이타에 대해 오류율 $R(T)$는 $R(T)=1/N\sum_{t\in\tilde{T}} M(t)$ ($M(t)$는 노드 t에서 잘못 분류된 데이타 개수)이다. 이 오류율들이 교차 시험에 의해 구해졌을 경우는 $R^{cv}(T)$ 혹은 $\hat{R}(T)$로 표현하고, 실험 데이타에 적용한 후 구한 값이면 $R^{ts}(T)$로 표현한다.

3.2 자료 수집

본 연구의 목표는 이메일 계정을 만든 이가 인간인지 혹은 프로그램인지를 판단하여 스팸 메일을 필터링하는 것이고 이 연구의 개념도가 그림 1에 보인다. 그림에서

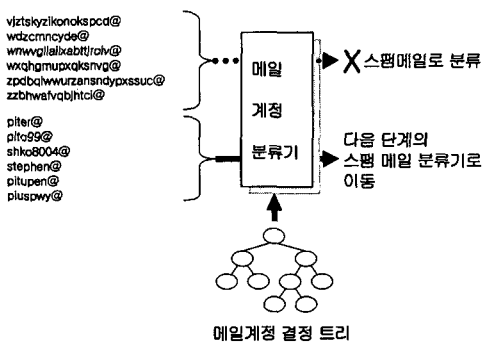

그림 1 스팸 차단 개념도

메일계정 분류기는 이메일 계정을 입력 받은 후, 그 계정이 프로그램에 의해 생성되었다고 판단되면 메일의 제목이나 혹은 메일의 내용을 살펴보지 않고 바로 그 메일을 스팸 메일로 판단하게 된다. 한편, 계정 문자열이 인간에 의해 생성된 것이라고 판단될 때에는 다음 단계의 스팸 메일 차단기를 구동하게 된다.

본 연구를 수행하기 위해 우선 이메일 계정을 수집해야 했다. 이는 국내 웹 메일 사이트의 로그 파일로부터 자료를 수집하였다. 자료를 제공한 사이트는 이미 기존의 스팸 메일 차단 프로그램에 의해 메일을 스팸과 비스팸 메일로 분류하고 있었다. 수집된 로그 파일에는 발송자와 수신인의 메일 계정, 제목 등이 적혀 있었다. 수집된 로그 파일에서 스팸으로 분류된 메일의 발송자 계정과 일반 메일로 분류된 메일의 발송자 계정을 수집하였다. 수집된 자료에는 동일 계정 문자열이 존재하므로 이를 중복 허용하지 않도록 한 결과 총 2,155,104개의 메일 계정을 수집할 수 있었다. 이 중 스팸 메일을 발송하는데 사용된 불량 메일계정은 133,996개이었으며, 정상 메일계정은 2,021,108개이었다. 본 실험에서는 스팸 메일을 발송한 메일 계정을 모두 자동 생성된 메일 계정으로 판단한다. 스팸 메일 계정이 모두 자동 생성된 계정이라고 간주하기에는 어려움이 있지만, 사람이 계정들을 수작업으로 분류하기에는 너무나 많은 노력이 필요하므로 우선 별도의 작업없이 수집된 자료에 대해 결정 트리를 학습 및 실험하였다.

3.3 특징 변수 선정

일반적으로 패턴 분류기는 n개의 특징 변수를 입력으로 하고 있으며, 본 연구에서도 총 15개의 특징 변수를 이용하고 있다. 각 특징 변수들은 수집된 메일 계정들을 관찰하여 선정되었으며, 이산 (discrete) 변수 혹은 연속 (continuous) 변수의 형태이다. 입력 스트링으로부터 특징 변수 값을 구할 때는 시간 복잡도를 최대한 줄이기 위해서 유한 상태 오토마타를 주로 이용하였으며, 사용된 특징 변수와 그 특징 변수를 선택한 이유는 다음과 같다.

1. 계정 문자열의 길이 (연속 변수) : 일반적으로 계정 문자열이 길면, 불량 메일 계정일 가능성이 높다. 웹 메일 사이트의 경우, 대부분의 짧은 메일 계정들은 일반 사용자가 이미 사용하고 있을 가능성이 높다. 그러므로 스팸 발송자들은 새로운 메일 계정을 만들기 위해서 문자열이 긴 계정을 이용하게 된다.
2. 문자열 패턴 (이산 변수) : 이 특징 변수는 문자열 패턴에 대한 이산 값으로 다음의 총 여덟 개의 패턴 값 중 하나를 가지게 된다. (1) 계정이 모두 숫자로만 이루어져 있는가, (2) 계정이 모두 문자 (알파벳)로만 이루어져 있는가, (3) 계정의 형식이 숫자열 다음에 문자열이 오는가, (4) 문자열 다음에 숫자열이 오는가, (5) 숫자열 + 문자열 + 숫자열 형식인가, (6) 문자열 + 숫자열 + 문자열 형식인가, (7) 문자열 내에 특수문자가 존재하는가, (8) 그 이외의 경우인가. 대부분의 정상 메일들은 모두 문자열로 이루어지거나, 혹은 문자열 다음에 숫자가 오는 경우가 많다.
3. 숫자열이 계정 문자열 내에서 떨어져 나타난 회수 (연속 변수) : 예를 들어 "0fdcf8a8"이란 문자열이 있다면 이 특징 변수 값은 3이다. 사람이 계정을 만들 때 이렇게 숫자를 분산하여 삽입하는 경우는 드물다. 그러므로 이 값이 클수록 불량 메일계정일 가능성이 높다.
4. 계정 내 알파벳 문자의 총 개수 (연속 변수) : 이 값이 크면 불량 메일 계정일 가능성이 높다.
5. 계정 내 숫자의 총 개수 (연속 변수) : 이 값이 비정상적으로 크다면 불량 메일 계정일 가능성이 높다.
6. 문자열 내 특수 문자의 총 개수 (연속 변수) : 특수 문자는 알파벳과 숫자가 아닌 문자를 의미하고 예를 들면 '*', '^', '_'와 같은 문자를 말한다. 이러한 문자가 자주 사용되면 불량 메일 계정일 가능성이 높다.
7. 문자열 내 연속으로 나타나는 자음 문자 길이 중 가장 큰 값 (연속 변수) : 예를 들어 "arphlxhuoo-jpzjuuaaely"라는 문자열이 있다면 연속 자음군은 "rphlxh", "jpzj", "ly" 세 개이다. 이 중 가장 긴 자음 군은 "rphlxh"로 총 길이가 6이다. 이 값이 클수록 불량 메일계정일 가능성이 높다. 일반적으로 메일 계정은 자신의 이름을 이용하여 만들게 되는데, 아무리 첫 글자만을 이용한 계정이더라도 자음열이 6개 정도나 계속 나온다면 정상적인 메일로 간주하기에 무리가 따른다.
8. 문자열 내 최대 연속 모음 수 (연속 변수)
9. 문자열 내 최대 연속 숫자 수 (연속 변수)
10. 문자열 내 최대 특수 문자 수 (연속 변수)
11. 문자열 내 총 자음 수 (연속 변수)
12. 문자열 내 총 모음 수 (연속 변수)
13. 특수 문자와 숫자를 제외한 나머지 문자열 대비 자음 개수의 비율 (연속 변수) : 이 값은 자음수/(자음수+모음수) 값이며 값이 크게 되면 불량 메일 계정일 가능성이 높다.
14. 문자 타입을 알파벳, 숫자, 특수 문자로 나누었을 때, 이들의 변화 횟수 (연속 변수) : 예를 들어 "abc^_^347ab26^*"라는 문자열이 있다면, 문자 군은 "abc", "^_^", "347", "ab", "26", "^*"이다. 그러므로 문자군의 이동 횟수는 5이다. 정상적인 메일 계정일수록 이 값은 작은 경향이 있다.
15. 문자열의 로그 우도비(log likelihood ratio) (연속 변수)

위 특징 변수 중 마지막 특징 변수인 로그 우도비(*lr*)는 *n*개의 문자열 $c_1c_2 \ldots c_n$에 대해 다음과 같이 정의된다.

$$\begin{aligned} lr &= \log(p(c_{1..n} \mid \text{Good}) / p(c_{1..n} \mid \text{Bad})) \\ &= \log p(c_{1..n} \mid \text{Good}) - \log p(c_{1..n} \mid \text{Bad}) \\ &= \sum_{i=1}^{n} \log p_{\text{Good}}(c_i \mid c_{i-1}) - \sum_{i=1}^{n} \log p_{\text{Bad}}(c_i \mid c_{i-1}) \\ &= \sum_{i=1}^{n} (\log p_{\text{Good}}(c_i \mid c_{i-1}) - \log p_{\text{Bad}}(c_i \mid c_{i-1})) \end{aligned}$$

위 식에서 $p(c_{1..n} \mid \text{Good})$은 정상 계정 집합에서 해당 문자열 $c_{1..n}$이 발생할 확률을 뜻하고, $p(c_{1..n} \mid \text{Bad})$는 불량 계정 집합에서 해당 문자열이 발생할 확률을 뜻한다. 로그 우도비의 정의로부터 만약 해당 문자열이 정상 계정 집합에서 발생할 가능성이 더 크면, *lr*은 양의 값을 가지게 되고, 그렇지 않으면 음의 값을 가지게 된다. 로그 우도비의 최종 식은 임의의 문자열 $c_{1..n}$에 대해 확률 $p(c_{1..n})$이 바이그램(bigram) 확률의 곱 $\prod_{i=1}^{n} p(c_i \mid c_{i-1})$으로 근사화될 수 있다는 점을 이용하여 구해졌다.

일반적으로 바이그램 확률을 이용하게 되면, 학습 자료에서 발견되지 못한 바이그램이 실험 자료에서 나타날 가능성이 있다. 이런 경우 1) 유니그램(unigram)과의 확률 선형 결합 방법을 사용하거나, 혹은 2) back-off 방식을 이용하게 된다[14]. 본 실험에서는 다음과 같은 back-off 방법을 이용하였다.

$$p(c_i \mid c_{i-1})^{*} = \begin{cases} (C(c_i, c_{i-1}) - b) / C(c_{i-1}), & \text{if } C(c_i, c_{i-1}) > 0 \\ \beta p(c_i), & \text{if } C(c_i, c_{i-1}) = 0 \end{cases}$$

이 방법에서 $C(c_i, c_{i-1})$은 인접한 두 문자가 발생된 횟수를 의미한다. 수식에서 *b*의 값으로 1.0을 선택하였고, β는 확률의 총 합이 1.0이어야 되는 조건에 의해 구해지는 값이다

3.4 실험

트리 분류기 방법의 성능을 알아보기 위해, 앞에서 설명한 15개의 특징 변수를 이용하여 실험하였다. 총 수집된 이메일 계정 2,155,104개 중 1/4에 해당하는 538,776개로 CART 트리를 학습하고 나머지 자료로 트리의 성능을 실험하였다. 로그 우도비를 구하기 위해 사용되는 바이그램 확률들은 학습 자료로부터 구하였으며, 최종적으로 76개의 단말 노드를 가지는 트리가 얻어졌다. 이 트리의 10차 교차 시험 오류율은 0.036468±0.000255이었다. 학습 자료와 실험 자료에서의 각 성능은 표 1과 같다.

표 1 CART 트리의 불량 메일계정 분류 성능

학습 (N=538,776)	실험 (N=1,616,328)
$\lvert\tilde{T}\rvert = 76$. $\hat{R}(T) = 0.036468 \pm 0.000255$ 정상 계정 분류 정확률 : 502734/505232 = 0.9950 불량 계정 분류 정확률 : 16667/ 33544 = 0.4968	$R^{ts}(T) = 0.0364$ 정상 계정 분류 정확률 : 1507856/1515876 = 0.9947 불량 계정 분류 정확률 : 49449/100452 = 0.4922

위 표에서 나타내는 두 종류의 분류 정확률 중 정상 메일 계정의 분류 정확률은 특별한 의미를 가진다. 만약 정상 계정을 불량 계정으로 오인식하게 되면 사용자는 자신이 원하는 메일을 스팸 메일 분류함에서 찾게 되므로 위 두 정확률의 중요도가 서로 다르다고 말할 수 있다. 다시말해 불량 계정으로 판단했으나 실제로는 정상 계정일 확률인 오경보율(false-alarm rate)을 낮추어야 한다. 그러므로 본 연구에서는 기존의 스팸 필터링 시스템과의 유연한 연동을 위해, 분류 결정에 대한 신뢰도를 제공하기로 하였다. 결정에 대한 신뢰도는 학습 트리의 노드에 포함되는 자료들의 정상/불량 계정간의 비율을 제공하는 것으로 대신한다.

본 연구에서 최종적으로 구해진 트리의 일부분이 그림 2에 보인다. 그림에서 노드 중앙에는 분류 결과를 보이고 있고, 노드 옆에는 사용된 질문과 분류 결과에 대한 신뢰도를 보인다. 그림에서 루트 노드 질문은 로그 우도비에 관한 것이다. 이 값이 -3.81보다 클 경우는 일단 계정이 정상적이라고 가정할 수 있고, 만약 이 값이 작다면 루트 노드의 왼쪽으로 움직여 불량 계정이라고 생각할 수 있다. 로그 우도비의 정의에 따르면 원래 0보다 크면 정상 계정으로 간주되어야 했다. 하지만 그림에서 어느 정도의 음수 값이 되어도 정상 계정으로 판단한다는 것은 트리가 보수적인 결정을 한다는 것을 의미하고, 이것은 사용자가 느끼는 오경보율을 낮춘다는 점에서 올바른 결정이라고 생각한다. 루트 노드의 오른쪽 노드에서는 문자열 길이에 관한 질문이 사용되었고, 그 길이가 15보다 작으면 정상 계정, 15보다 크면 불량 계정으로 판단한다. 이러한 판단은 인간이 판단하기에도 메일계정이 길 경우 불량 메일계정으로 판단하게 되는데, 이 판단 과정을 그대로 반영하고 있다.

본 실험에서 구해진 CART 트리를 이용하여 실제 스팸 메일을 보낸 불량 계정을 분류해 본 결과 표 2를 얻었다. 표에 나타난 메일 계정 중 세번째 계정의 경우 비록 스팸 메일을 발송한 계정이지만 이를 정상 메일계정

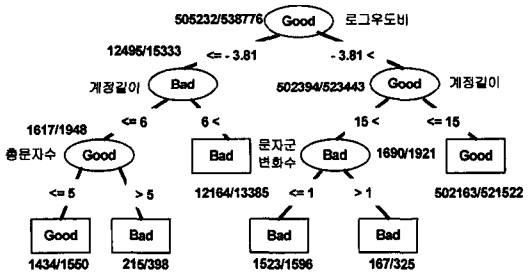

그림 2 불량 계정을 판단하는 결정 트리

으로 분류하였다. 하지만 불량 계정이라고 판단하기에는 계정의 길이가 충분히 짧기 때문에 이와 같이 보수적인 판단을 하는 것이 본 논문의 목적에 맞다고 생각한다.

표 2 불량 계정을 판단한 결과

fafrdptcdnikecaa@naver.com	→ BAD	신뢰도:	0.975057
fajgapqoxtsdlgis@freechal.com	→ BAD	신뢰도:	0.975057
fbbvju@korea.com	→ GOOD	신뢰도:	0.918963
fbedeuberyyzpnvvlyo@naver.com	→ BAD	신뢰도:	0.975057
fbgobyrccujdwox@naver.com	→ BAD	신뢰도:	0.975057
019buclor@sexyhotxxchicx.com	→ GOOD	신뢰도:	0.896239
020890@mx106.mf.home.ne.jp	→ GOOD	신뢰도:	0.978471

4. 결 론

본 논문에서는 최근 급증하고 있는 새로운 형태의 스팸 발송 방법에 대응할 수 있는 메일 계정 분류기를 구현하였다. 외관상으로는 극히 정상적인 절차에 의해 발송되는 스팸 메일에 대해, 본 연구에서 제안하는 분류기는 계정이 자동 생성되었는지를 판단하므로 메일의 내용을 분석하기에 앞서 메일을 차단할 수 있다. 이 방법은 기존의 메일 필터링 시스템에서 차단하기 힘든 메일에 대해서도 효과적인 대응 방법을 제공한다.

대부분의 필터링 시스템에서는 정상 메일을 스팸 메일로 잘못 분류하는 비율인 오경보율을 최대한 낮춘 상황에서 스팸 메일 분류 정확도를 최대한으로 높이고자 한다. 본 연구의 경우도 이를 위해 학습 자료의 구성이 불량 계정보다 정상 계정을 많게 하여 정상 계정의 정확률에 좀 더 높은 비중을 두었다. 입력 메일 계정에 대해 총 15개의 특징 값을 정의하였고, 트리 구조 분류기인 CART를 이용하여 패턴을 분류하였다. 학습에 참여하지 않은 총 160만개의 메일 계정에 대해 3.6%의 분류 오류율을 나타내었으며, 정상 계정 151만개에 대해 1만개를 불량 계정으로 판단하여 0.53%의 오경보율을 나타내었다. 실제로 기존의 시스템과의 연동 시에는 이 오경보율을 더 낮출 수 있도록 하기 위해, 분류 결과와 함께 그 결과의 신뢰도를 제공하였다.

앞에서 구현한 계정 분류기를 기존의 스팸 차단 시스템과 연동하여 운용해본 결과 기존 방식으로 검출되지 않던 스팸 메일을 차단할 수 있었는데, 현재는 높은 신뢰도의 불량 계정 판단일 경우에만 메일을 차단하고 있다. 이는 오경보율을 최대한 낮추기 위해 취한 방법이며, 향후에는 계정 분류기와 기존 차단 시스템의 분류 특성이 고려된 연동 규칙을 연구할 계획이다.

참 고 문 헌

[1] 조 한철, 조 근식, 나이브 베이지안 분류자와 메세지 규칙을 이용한 스팸메일 필터링 시스템, *한국정보과학회 봄 학술발표논문집*, 제 29권, 제 1호, pp. 223-225, 2002.

[2] Paul Graham, A plan for spam, http://www.paulgrapham.com/spam.html, 2003.

[3] Ion Androutsopoulos, Georgios Paliouras, Vangelis Karkaletsis, Georgios Sakkis, Constantine D. Spyropoulos, and Panagiotis Stamatopoulos, Learning to filter spam e-mails: A comparison of a naive bayesian and a memory-based approach, *Proceedings of Machine Learning and Textual Information Access*, pp. 1-13, 2000.

[4] 신 경식, 안 수산, 데이타 마이닝 기법을 활용한 스팸 메일 분류 및 예측모형 구축에 관한 연구, *경영논총*, 제 20권, 제 2호, pp. 89-104, 2002.

[5] 민 도식, 송 무희, 손 기준, 이 상조, SVM 분류 알고리즘을 이용한 스팸메일 필터링, *한국정보과학회 봄 학술발표논문집*, 제 30권, 제 1호, pp. 552-554, 2003.

[6] Aleksander Kolcz and Joshua Alspector, SVM-based filtering of e-mail spam with content-specific misclassification costs, *Proceedings of the TextDM'01 Workshop on Text Mining*, 2001.

[7] William S. Yerazunis, The spam-filtering accuracy plateau at 99.9% accuracy and how to get past it, *MIT Spam Conference*, 2004.

[8] L. Breiman, J.H. Friedman, R.A. Olshen, and C.J. Stone, *Classification and Regression Trees*, Wadsworth Statistics/Probability Series, Belmont, CA, 1984.

[9] S.R. Safavian and D. Landgrebe, A survey of decision tree classifier methodology, *IEEE Trans. on Systems, Man, And Cybernetics*, Vol. 21, No. 3, pp. 660-674, 1991.

[10] J. Graham-Cumming, How to beat an Adaptive Spam Filter, *MIT Spam Conference*, 2004.

[11] J. Graham-Cumming, The Spammer's Compendium, *MIT Spam Conference*, 2004.

[12] P.A. Chou, Optimal partitioning for classification and regression trees, *IEEE Trans. on PAMI*, Vol. 13, No. 4, pp. 340-354, 1991.

[13] J.R. Quinlan, *C4.5: Programs for Machine Learning*, San Mateo: Morgan Kaufmann, 1993.

[14] H. Ney, S. Martin, and F. Wessel, Statistical language modeling using leaving-one-out, Steve

Young and Gerrit Bloothooft, editors, *Corpus-Based Methods in Language and Speech Processing*, pp. 174-207. Kluwer Academic Publishers, 1997.

이 상 호

1993년 2월 동국대학교 전자계산학과(공학사). 1995년 2월 한국과학기술원 전산학과(공학석사). 2000년 2월 한국과학기술원 전산학과(공학박사). 2000년 2월~2004년 2월 LG전자기술원 모바일멀티미디어 연구소 선임연구원. 2004년 3월~현재 한국산업기술대학교 게임공학과 전임강사. 관심분야는 음성 처리, 자연언어 처리, 패턴 인식, 게임 인공지능