Journal of the Korean
Data & Information Science Society
2000, Vol. 11, No. 1, pp. 19 ~ 30

베이지안 학습을 이용한 문서의 자동분류

김진상 [1] · 신양규[2]

요약

정보통신기술의 비약적인 발전은 온라인으로 생성되는 전자문서의 양을 폭발적으로 증가시키고 있다. 따라서 수동으로 문서를 분류하던 종래의 방법 대신 문서의 자동분류 기술 개발이 특별히 요구되고 있다. 본 논문에서는 베이지안 학습 기법을 이용하여 문서를 자동으로 분류하는 방법을 연구하고, 20개의 유즈넷 뉴스그룹 문서들을 분류하도록 시험하였다. 사용한 알고리즘은 Naive Bayes Classifier이며, 구현한 시스템을 이용해 유즈넷 문서를 대상으로 자동분류를 실험한 결과 분류의 정확률이 약 77%로 나타났다.

주제어: 문서분류, 베이지안 학습, 기계학습.

1. 서론

최근 컴퓨터를 통해 접할 수 있는 전자우편, 뉴스그룹 아티클, 웹 문서 등 온라인 전자 문서의 양은 폭발적으로 증가하고 있으며 문서의 양이 늘어날수록 원하는 정보를 정확하고 신속하게 발견하기 위한 검색, 여과, 관리의 필요성 또한 더불어 커진다. 예를 들어, 1979년 미국에서 시작되어 특정 분야의 정보 교환을 위해 사용하고 있는 뉴스그룹은 현재 전 세계적으로 약 15,000개에 이르고 있으며[1], 이들 뉴스그룹은 대부분 수동으로 관리되어 사용자는 미리 분류된 뉴스그룹에 자신의 문서를 게시(posting)하거나 혹은 다른 사람의 문서를 읽고 있다. 만약 사용자가 게시할 문서를 내용에 따라 자동으로 분류하여 준다면, 여러 단계를 거쳐 해당 뉴스그룹으로 찾아가는 번거로움을 피할 수 있고 또 효율적인 뉴스그룹의 관리가 가능할 것이다.

문서의 자동분류란 일반적으로 기계학습을 이용하여 미리 학습하여 둔 범주 중 하나로 문서를 분류하는 처리를 말하며[4], 이 때 사용하는 기계학습은 개개의 사례를 분석하여 일반적인 규칙이나 함수를 찾아내는 귀납학습법이 보편적이다. 문서 자동분류로 널리 알려진 방법은 결정 트리 학습법인데. 이 방법의 대표적인 예는 1986년에 개발된 ID3[5]와 1993년에 개발된 C4.5 및 Cubist[6]등이 있다. 결정 트리 학습법은 트리를 학습에 적용한 것으로 구현이 쉽고, 'IF - THEN'

[1]대구시 달서구 신당동 계명대학교 컴퓨터 전자공학부교수
[2]경북 경산시 점촌동 경산대학교 정보과학부교수

형태의 간편한 규칙으로 표현되기 때문에 많이 사용되지만, 깊이나 가지의 수를 제한하기 위한 부가적인 처리가 필요하다는 단점이 있다[7]. 본 논문에서는 결정 트리의 대안으로 확률적 기법을 학습에 적용한 베이지안 학습법을 이용하여 문서의 자동분류를 해결하고자 한다. 여기서 사용할 베이지안 학습법은 NBC(Naive Bayes Classifier)라고 부르는 방법으로서, 범주별로 주어진 문서의 각 단어마다 확률을 계산해 분류를 수행한다. 이 방법의 효율성을 테스트하기 위해 20개의 뉴스그룹을 선택하여 각각의 뉴스그룹에 1,000개의 문서를 저장하여 학습을 시키고 학습의 결과로 만들어진 분류기를 이용하여 문서의 자동분류를 수행해 분류의 정확성을 살펴본다.

본 논문의 구성은 다음과 같다. 2절에서는 기계학습을 이용한 문서분류와 베이지안 학습법 및 NBC알고리즘에 대해서 알아보고, 3절에서는 베이지안 알고리즘을 이용한 문서분류의 시험과정과 결과에 대해서 기술할 것이다. 끝으로 4절에서는 본 논문의 의의 및 향후 연구 방향에 대해 언급할 것이다.

2. 기계학습을 이용한 문서분류

기계학습이란 컴퓨터 프로그램이 일련의 작업 T를 수행하면서 축적된 경험 E를 토대로 성능 P를 이전보다 증가 시켰다면, 그 프로그램은 T와 P에 관한 경험 E로부터 학습을 했다고 한다[6]. 기계학습은 컴퓨터의 자동학습 모형을 연구하여 컴퓨터가 스스로 학습할 수 있는 시스템을 구축하는 연구 분야이며, 연구 방향은 일반적인 자동학습 알고리즘에 대한 이론적 분석과 개발, 인간의 학습과정 계산 원리 개발, 그리고 특정 응용프로그램에 학습 기능을 추가하는 일 등이다.

2.1 문서분류

문서의 자동분류란 문서 파일을 미리 정의된 여러 개의 범주 중 하나에 속하도록 대응시키는 것을 말한다. 텍스트 분류를 위해서는 훈련예제를 이용하여 학습을 시키는데 학습의 결과로 문서분류기가 생성되며 이를 통해 새로운 문서가 어떤 범주에 속하는지를 결정하는 것이다. 그림 1은 일반적인 문서 분류기의 구조를 나타낸다.

문서를 자동으로 분류하기 위해서는 문서가 특정 형태로 변환이 되어야 한다. 왜냐하면 하나의 문서는 통상 수백 혹은 수천의 단어로 구성되어 있기 때문에 이를 모두 처리하는 것은 매우 비효율적이므로 문서를 대표할 수 있는 일부의 단어만으로도 자동분류가 가능해야 한다. 일반적으로 문서는 속성과 값의 형태로 표현하는데, 어떤 단어가 훈련 데이터에서 세 번 이상 나타났을 때 그 단어를 특성 단어라 하며 특성 단어들을 벡터 형태로 표현한다[3, 5]. 특성 벡터의 요소는 실수나 이름 값을 사용하지만 표현력의 한계가 있기 때문에, 이를 확장하여 문자열을 허용하는 특성 벡터 표현을 최근에는 많이 이용한다.

2.2 베이지안 학습법

확률 이론을 기계학습에 적용한 것으로, 특정 데이터 집합 D를 조사했을 때 가설 h가 사실일 확률은 $P(h|D)$가 된다. 그리고 가설이 사실일 경우 데이터 D의 확률이 $P(D|h)$일 때 베이즈 정

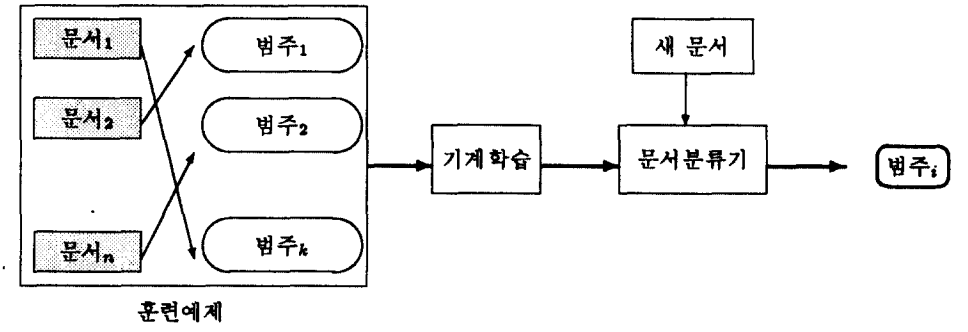

그림 1: 문서 분류기의 구조

리는 다음과 같다.

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)} \tag{1}$$

여기서 $P(h)$는 데이터에 관한 정보가 주어져 있지 않을 때 가설이 사실일 사전확률이다. 기계학습에서 관심을 가지는 값은 $P(h|D)$인데, 베이지안 학습법은 가설집합 H에 포함된 가설 중 최대 확률을 가지는 가설 h를 구하는 것이다.

최대확률을 구하기 위해서는 최대사후확률(MAP, maximum a posterior probability)을 계산하면 되는데, 이 확률은 데이터를 조사했을 때 가장 가능성이 높은 가정으로서 다음 식을 이용하여 구한다.

$$h_{MAP} = \text{argmax}_{h \in H} P(h|D) = \text{argmax}_{h \in H} \frac{P(D|h)P(h)}{P(D)} \tag{2}$$

식 (2)에서 $\text{argmax}_{h \in H} P(...)$은 확률 $P(...)$가 최대가 되는 H 내의 가설 h를 나타내는데, 이 때 분모 $P(D)$는 h와 무관하기 때문에는 상수 값으로 간주하여 다음과 같이 표현한다.

$$h_{MAP} = \text{argmax}_{h \in H} P(D|h)P(h) \tag{3}$$

최대사후확률보다 더 구체적인 것으로 최대우도(ML, maximum likelihood)를 사용하며, 이를 구하기 위해 먼저 가설 공간에 존재하는 모든 가설들이 같은 확률을 가진다고 간주한다. 즉, 가설 집합 H의 원소의 수를 $|H|$로 나타낼 때 아래와 같이 가정한다.

$$\forall h (\in H) P(h) = p = 1/|H| \tag{4}$$

이 가정을 이용하면 $P(D|h)P(h)$에서 $P(h)$의 값은 상수가 되므로 $P(D|h)$를 극대화시키는 최대우도 확률을 h_{ML}이라 할 때 다음의 식으로 표시된다.

$$h_{ML} = \text{argmax}_{h \in H} P(D|h) \tag{5}$$

식 (??)의 $P(h|D)$는 훈련예제 D가 주어질 때 가설 h가 성립할 확률을 말하는데, 식 (??)에서 나타내는 바와 같이 베이지안 학습법은 가설집합 H에 포함된 가설 중 가장 큰 확률을 가지는 h를 찾아 최종 가설로 설정하는 것이다.

2.3 Naive Bayes Classifier 알고리즘

기계학습의 여러 가지 방법 중에서 귀납 학습의 범주에 속하는 베이지안 학습법, 신경망, 그리고 결정트리 학습법은 서로 대등한 성능을 보여주는데, 특히 자료의 양이 많을 때는 베이지안 학습법의 정확도가 다른 방법 보다 훨씬 더 높다고 알려져 있다[6]. 베이지안 학습법은 각 사례 x가 속성값들의 벡터로 표시되고, 기계학습의 결과로서 구하고자 하는 타겟함수 $f(x)$가 범주들의 유한 집합 V의 원소인 경우의 학습에 잘 적용된다. 즉, 기계학습을 위해 사용하는 훈련 예제인 속성벡터 x 각각에 대해 가설 $h(x)$가 주어질 때 $\forall x\ h(x) = f(x)$인 함수 f를 타겟함수라고 하며 미리 정의된 범주들의 집합 V에서 함수 f를 대신할 수 있는 확률이 가장 큰 원소 $v \in V$를 구하는 것이 최종 목표가 된다.

훈련예제 집합이 주어지고 새로운 사례가 $\langle a_1, a_2, ..., a_n \rangle$과 같이 속성값들의 벡터로 주어지면, 학습기는 이 사례에 대한 타겟함수의 값 혹은 분류를 예측할 수 있다. 새로운 사례에 대한 분류를 예측하는 방법은 주어진 속성벡터 $\langle a_1, a_2, ..., a_n \rangle$에 대응되는 가장 가능성이 높은 타겟함수의 값 v_{MAP}을 다음 식과 같이 구하는 것이다.

$$v_{MAP} = \text{argmax}_{v_j \in V} P(v_j | a_1, a_2, ..., a_n, D) \tag{6}$$

여기서 V는 미리 정의된 범주들의 집합을 나타낸다. 식 (6)에다 베이즈 정리를 적용하면

$$\begin{aligned} v_{MAP} &= \text{argmax}_{v_j \in V} \frac{P(a_1, a_2, ..., a_n | v_j, D) P(v_j | D)}{P(a_1, a_2, ..., a_n | D)} \\ &= \text{argmax}_{v_j \in V} P(a_1, a_2, ..., a_n | v_j, D) P(v_j | D) \end{aligned} \tag{7}$$

과 같이 된다. 이 때 훈련예제에 대한 각 범주의 확률 $P(v_j|D)$는 아래 식과 같이 계산되는데, $|v_j|$가 범주 v_j에 속한 속성의 수이고 N_{Tot}가 전체 속성 데이터의 수일 때 훈련집합 전체를 해당 범주에 속한 데이터의 개수로 나누면 된다.

$$P(v_j|D) = |v_j| / N_{Tot}$$

$P(a_1, a_2, ..., a_n | v_j, D)$는 추정하기가 쉽기 않기 때문에 '베이즈 독립성' 가정을 적용하여 해결을 한다. 적용할 베이즈 독립성 가정은 "주어진 범주에 대해 속성값들은 모두 독립이다"라는 것이며, 식 (8)과 같이 표현할 수 있다.

$$P(a_1, a_2, ...a_n | v_j, D) = \prod_i P(a_i | v_j, D) \tag{8}$$

베이즈 독립성 가정인 식 (8)을 식 (7)에 적용하면 다음의 식이 구해진다.

$$v_{NB} = \text{argmax}_{v_j \in V} \frac{|v_j|}{N_{Tot}} \prod_i P(a_i | v_j, D) \tag{9}$$

LEARN-NAIVE-BAYES-TEXT(*Egs, V*)

1. *Egs*에 나타난 모든 단어와 토큰을 모은다.

   $Voc \leftarrow$ *Egs*에 나타난 모든 상이한 단어 및 토큰들

2. 필요한 확률 $P(v_j)$와 $P(w_k|v_j)$를 계산

   - V 내의 각 타겟값 v_j에 대해 다음을 시행
     - $docs_j \leftarrow$ 타겟값이 v_j인 *Egs*의 부분집합
     - $P(v_j) \leftarrow docs_j/Egs$
     - $Text_j \leftarrow docs_j$의 모든 요소를 나열하여 만든 하나의 문서
     - $n \leftarrow Text_j$ 에 있는 단어의 총 수(중복된 단어는 중복 회수 만큼 센다.)
     - *Voc*에 있는 w_k에 대해 다음을 시행
       * $n_k \leftarrow Text_j$에 나타난 w_k의 횟수
       * $P(w_k|v_j) \leftarrow (n_k + 1)/(n + Voc)$

그림 2: Naive Bayes Classifier 알고리즘

식 (9)에서 $P(a_i|v_j, D)$를 구하기 위해서는 각 범주에 속한 훈련집합에서 각각의 속성값이 발생하는 빈도 수를 헤아리면 된다. 즉,

$$P(a_i|v_j, D) = |a_i|/\text{범주 } v_j\text{에 속한 사례들의 수}$$

위에서 $|a_i|$는 속성 a_i의 수를 나타낸다.

2.4 Naive Bayes Classifier의 구현

앞에서 본 Naive Bayes Classifier를 이용한 학습 알고리즘은 그림 2와 같으며, 이를 프로그램으로 구현한 후 훈련예제를 입력 데이터로 적용하여 학습시킨 결과 확률들은 디스크에 저장해 둔다. 학습 결과를 이용하여 새로운 문서를 분류할 때는 학습시 미리 수집한 데이터 중 일부를 이용해서 분류작업을 수행하는데, Naive Bayes Classifier의 분류 알고리즘은 그림 3과 같다.

3. 문서분류 시험 및 평가

3.1 훈련예제

범주가 20개이고, 각 범주별로 1,000의 문서를 훈련데이터로 사용할 때 필요한 문서의 수는 총 20,000개가 된다. 표 1은 실험에 사용된 20개의 뉴스그룹을 나타내며 각각의 뉴스그룹 이름은 컴

CLASSIFY-NAIVE-BAYES-TEXT(Doc)

- $position \leftarrow Voc$에서 발견된 토큰들의 Doc내의 모든 단어들의 위치
- 다음 식에 의해 계산된 v_{NB}

$$v_{NB} = \text{argmax}_{v_j \in V} P(v_j) \prod_{i \in position} P(a_i|v_j)$$

그림 3: Naive Bayes Classifier의 분류 알고리즘

alt.atheism	rec.sport.hockey
comp.graphics	sci.crypt
comp.os.ms-windows.misc	sci.electronics
comp.sys.ibm.pc.hardware	sci.med
comp.sys.mac.hardware	sci.space
comp.windows.x	soc.religion.christian
misc.forsale	talk.politics.guns
rec.autos	talk.politics.mideast
rec.motorcycles	talk.politics.misc
rec.sport.baseball	talk.religion.misc

표 1: 20개의 뉴스그룹

```
Path: cantaloupe.srv.cs.cmu.edu!das-news.harvard.edu!ogicse!uwm.edu!
From: jbarrett@aludra.usc.edu (Jonathan Barrett)
Newsgroups: rec.sport.hockey
Subject: Re: This year's biggest and worst (opinion)...
Keywords: NHL, awards
Message-ID: <1pqgq3INN2vn@aludra.usc.edu>
Date: 5 Apr 93 09:53:39 GMT
Article-I.D.: aludra.1pqgq3INN2vn
References: <C4zCII.Ftn@watserv1.uwaterloo.ca>
Sender: nntp@aludra.usc.edu
Organization: University of Southern California, Los Angeles, CA
Lines: 15
NNTP-Posting-Host: aludra.usc.edu

I can only comment on the Kings, but the most obvious candidate for pleasant
surprise is Alex Zhitnik. He came highly touted as a defensive defenseman, but
he's clearly much more than that. Great skater and hard shot (though wish he
were more accurate). In fact, he pretty much allowed the Kings to trade away
that huge defensive liability Paul Coffey. Kelly Hrudey is only the biggest
```

그림 4: 뉴스그룹 문서 예

퓨터에서 처리될 디렉터리의 이름이 되고, 각 디렉터리 내에 1,000개 씩의 문서가 존재한다. 그림 4는 훈련예제로 사용된 뉴스그룹 아티클 문서 중 하나이다.

하나의 문서는 헤더(header)부분과 바디(body)부분으로 나뉘는데, 헤더부분은 문서의 경로, 제목, 키워드, 글쓴이 등을 나타내며, 바디부분은 이 문서의 내용을 나타낸다. 자동분류 시스템에서는 헤더부분을 무시하고 바디부분만 이용하는데, 문서를 하나씩 읽으면서 바디부분은 그림 5와 같은 빈도수 테이블로 표현된다. 단어들을 읽으면서 빈도수 테이블을 추가하고 수정하는데, TID는 현재 문서의 범주인 CID에 속하는 문서에 출현하는 단어이고, PCID는 계층 구조로 확장 할 때를 대비하여 CID의 부모 범주를 나타낸다. 그리고 추가로 네 개의 숫자 필드가 있는데 SF1과 SF2로 표시된 수치 ①과 ②는 각각 $\sum_{d\in c}\frac{n(d,t)}{n(d)}$ 및 $\sum_{d\in c}(\frac{n(d,t)}{n(d)})^2$ 이다.

이와 같은 빈도수 테이블을 만드는 방법은 각 문서 d와 단어 t에 대해서, 빈도수 테이블에 한 행씩 추가하면 된다. 먼저 SMC는 1로, SNC는 d에서의 t수를 나타내는 $n(d,t)$, SF1과 SF2는 위에서 언급한 값을 사용한다. 같은 값이 중복되어 들어오면, 각 필드의 값들을 수정하면서 통계치들을 수집한다. (TID, PCID, CID)는 중복 없이 유일한 값을 가지는 키가 되는데, 결국 SMC는 범주 c에서의 문서들의 총 수이고, SNC는 범주 c에서 단어 t의 총 수가 되고, SF1과 SF2는 부가적인 통계치 계산을 위한 것으로서 특성 단어 선택과 분류 작업을 할 때 사용된다.

이 문서의 범주는 파일이 존재하는 디렉터리의 이름이 되는데, 문서에도 나타나듯이 'rec.sport.hockey'가 된다. 그림 6은 훈련예제로 사용된 범주들의 구조를 나타낸다.

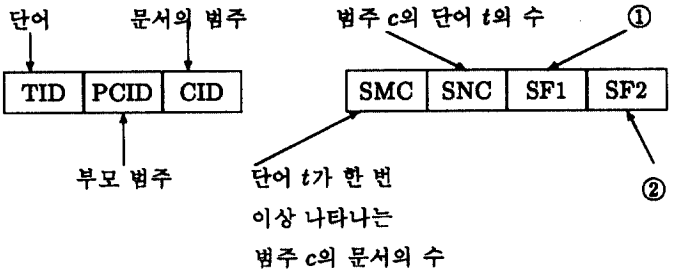

그림 5: 빈도수 테이블 구조

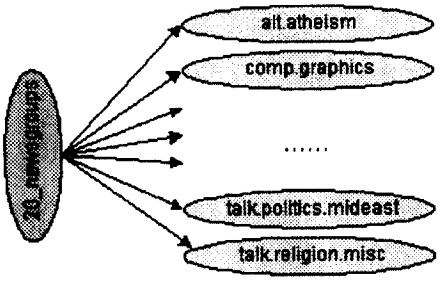

그림 6: 훈련예제 범주구조

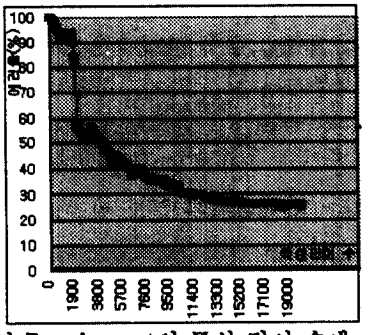

그림 7: sci.crypt의 특성 단어 수에 따른 에러율의 변화

Dataset:.\20_nesgroups\

Test ratior: 30%

- Row is actual, Column is predicted

class	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	accuracy
0	197	0	0	0	0	0	0	0	0	0	1	0	0	0	1	8	2	7	14	70	65.67%
1	0	234	4	4	3	18	0	0	0	0	0	18	2	0	4	0	1	3	6	1	78.00%
2	0	27	182	19	4	50	2	0	0	0	0	7	5	0	1	0	1	0	2	0	60.67%
3	0	21	14	207	11	8	3	0	0	0	0	17	7	0	5	0	1	2	4	0	69.00%
4	0	14	6	11	213	6	3	0	0	0	0	11	3	2	6	1	5	3	15	1	71.00%
5	0	28	7	1	2	243	0	0	0	1	0	7	0	0	2	0	1	1	3	0	81.00%
6	0	16	1	27	11	2	182	9	2	1	3	1	9	3	4	0	4	10	15	0	60.67%
7	0	2	0	1	0	0	4	242	2	0	0	4	1	0	4	0	7	3	30	0	80.67%
8	0	2	0	0	0	1	1	11	261	0	0	2	1	2	0	0	5	5	9	0	87.00%
9	1	2	0	0	0	2	0	2	0	259	14	1	0	1	0	0	1	0	17	1	86.33%
10	0	0	1	0	0	0	0	0	0	2	287	2	0	0	0	0	0	1	7	0	95.67%
11	0	2	0	0	0	2	0	0	0	0	0	285	0	1	1	0	2	1	6	0	95.00%
12	1	14	1	9	4	1	2	7	0	0	0	57	156	10	14	2	7	6	9	0	52.00%
13	5	8	1	0	0	0	0	2	0	0	2	1	1	238	3	2	5	7	20	5	79.33%
14	0	8	0	0	0	0	0	0	0	0	0	1	0	3	266	0	1	0	21	0	88.67%
15	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	297	0	0	2	0	99.00%
16	0	0	0	0	0	0	0	0	0	0	0	8	0	0	0	0	255	4	28	5	85.00%
17	1	0	0	0	0	0	0	0	0	1	0	2	0	0	0	2	0	279	10	0	93.00%
18	2	0	0	0	0	0	0	0	0	0	0	4	0	0	1	0	37	12	226	16	75.33%
19	36	0	0	0	0	0	0	0	0	0	0	1	0	0	2	38	15	8	43	157	52.33%

Correct : 4666 out of 6000

Accuracy average : 77.77%

표 2: 분류 결과

0	alt.atheism	10	rec.sport.hockey
1	comp.graphics	11	sci.crypt
2	comp.os.ms-windows.misc	12	sci.electronics
3	comp.sys.ibm.pc.hardware	13	sci.med
4	comp.sys.mac.hardware	14	sci.space
5	comp.windows.x	15	soc.religion.christian
6	misc.forsale	16	talk.politics.guns
7	rec.autos	17	talk.politics.mideast
8	rec.motorcycles	18	talk.politics.misc
9	rec.sport.baseball	19	talk.religion.misc

표 3: 분류 범주

3.2 뉴스 그룹 문서의 분류 및 결과

그림 7은 어떤 범주를 특별히 대표한다고 볼 수 있는 특성 단어를 추출하여 이것의 수를 변화시키면서 분류를 수행해 본 것으로, 둘 다 특성 단어의 수가 20,000개 정도에서 최소의 에러율을 가졌다. 본 연구에서도 분류의 정확성을 분석하기 위해 특성 단어의 수를 중요한 요인으로 설정하여 보았는데, 실험 환경에 따른 처리 속도의 제약 때문에 특성 단어의 수를 20,000개 이내로 제한하였다.

수집한 훈련예제들 중 70%는 학습에 사용하고 30%는 분류하는데 사용하였으며, 분류결과는 표 2와 같으며, 사용된 범주는 표 3과 같다. 분류 결과인 표 2에서 보여 주는 것처럼 범주 15와 같이 특성 단어(속성 값)들이 국한되어 나타나는 범주의 분류 정확도는 매우 높지만, 범주 12와 같이 특성 단어가 다른 범주에도 자주 나타나는 경우는 분류의 정확도가 상대적으로 떨어진다.

한편 이와 같은 분류의 결과는 Chakrabarti 등이[2] 시도한 특성 단어를 이용하여 Fisher 분포 테이블에서 문서의 범주를 분류하는 방법이 보인 평균 정확률 64.4% 보다 약 13% 정도 정확율이 높음을 확인 하였다. 다만 Chakrabarti 등의 방법은 뉴스 그룹 문서와 같은 비교적 정형화된 데이터 보다는 웹 문서처럼 정형화와는 거리가 먼 데이터를 분류할 목적으로 개발되었다는 차이점이 있다.

4. 결론

빠르게 변하는 인터넷 환경 하에서 급증하는 정보를 효과적으로 구하는 방법이 절실히 요구된다. 특히 컴퓨터에 의해 생성되는 전자 문서의 경우 양적인 면에서 사람이 수동으로 분류할 수 있는 범위를 벗어났으며 따라서 기계적인 처리를 통한 자동분류가 피할 수 없는 상황이다. 본 논문에서는 문서의 자동분류 기법 중 하나인 베이지안 학습 알고리즘을 이용하여 시스템을 구현하였으며, 20개의 유즈넷 뉴스그룹 아티클을 대상으로 분류를 수행하였다. 그 결과 분류의 평균 정확률은 약 77% 였으며, 이 수치는 경제적 가치가 충분히 있는 것으로 보고 있다.

통계학적 기법과 자연어 처리 및 인공지능 기술의 결합을 통해 최근 새롭게 대두되는 문제에 대한 효과적인 해결책을 찾을 수 있으며, 그 중 하나가 본 논문 에서 살펴 본 문서의 자동분류이다. 그러나 본 연구에서 다룬 내용은 뉴스그룹 문서라는 비교적 정형화에 가까운 데이터를 처리하였기 때문에 웹 문서나 기타 일반적인 문서처럼 비정형 문서의 처리를 위해서는 베이지안 알고리즘의 개선과 자연 언어에 관한 처리 기법을 더욱 정교하게 개발할 필요가 있다.

참 고 문 헌

1. –, 인드라넷, http://indra.indranet.net/internet/journal/jounal-2-f.htm.
2. S. Chakrabarti, B. Dom, R. Agrawal, P. Raghavan(1998). Scalable feature selection, classification and signature generation for organizing large text databases into hierarchical topic taxonomies, *VLDB Journal* 7: 163 – 178.
3. W. Cohen(1995). Fast effective rule induction, *Twelfth International Conference on Machine Learning.*
4. W. Cohen(1996). Context-sensitive learning methods for text categorization, SIGIR-96.
5. W. Cohen(1996). Learning Trees and Rules with Set-valued Features, *AAAI-96.*
6. T. Mitchell(1997). *Machine Learning*, McGraw-Hill.
7. S. Weiss, C. Kulikowski(1991). *Computer Systems That Learn: Classification and Prediction Methods from Statistics, Neural Nets, Machine Learning, and Expert Systems*, Morgan Kaufmann.

An Automatic Document Classification with Bayesian Learning

Jinsang Kim[1] · Yangkyu Shin[2]

Abstract

As the number of online documents increases enormously with the expansion of information technology, the importance of automatic document classification is greatly enlarged. In this paper, an automatic document classification method is investigated and applied to UseNet 20 newsgroup articles to test its efficacy. The classification system uses Naive Bayes classification algorithm and the experimental result shows that a randomly selected newsgroup arcicle can be classified into its own category over 77% accuracy.

Key Words and Phrases: Document Classification, Bayesian Learning, Machine Learning

[1]School of Computer and Electronics Engineering, Keimyung University, Taegu, Korea

[2]School of Information Science, Kyungsan University, Kyungpook, Korea