HMM을 이용한 채팅 텍스트로부터의 화자 감정상태 분석

문현구 장병탁
서울대학교 컴퓨터공학부
{hkmoon, btzhang}@scai.snu.ac.kr

Emotional States Recognition of Text Data Using Hidden Markov Models

Hyun-Ku Moon Byoung-Tak Zhang
School of Computer Science and Engineering, Seoul National University

요 약

입력된 문장을 분석하여 미리 정해진 범주에 따라 그 문장의 감정 상태의 천이를 출력해 주는 감정인식 시스템을 제안한다. Naive Bayes 알고리즘을 사용했던 이전 방법과 달리 새로 연구된 시스템은 Hidden Markov Model(HMM)을 사용한다. HMM은 특정 분포로 발생하는 현상에서 그 현상의 원인이 되는 상태의 천이를 찾아내는데 적합한 방법으로서, 하나의 문장에 여러 가지 감정이 표현된다는 가정 하에 감정인식에 관한 이상적인 알고리즘이라 할 수 있다. 본 논문에서는 HMM을 사용한 감정인식 시스템에 관한 개요를 설명하고 이전 버전에 비해 보다 향상된 실험결과를 보여준다.

1. 서론

3D 그래픽스, TTS(Text to Speech) 시스템, 음성인식 시스템 등의 발전은 모니터 안에 가상의 캐릭터를 만들어 인간과 비슷한 형태로 사용자와 대화할 수 있도록 하는 기술을 제공한다. 예를 들어 전자우편 에이전트의 경우 해당 편지를 선택함으로써 3차원 캐릭터가 편지를 읽어 줄 수 있도록 할 수 있으며 안내 데스크에서도 캐릭터를 사용해 보다 친근감 있는 음성 안내를 할 수 있다. 만약 이러한 시스템에서 3차원 캐릭터가 자신이 말하고 있는 내용에 따라 표정을 바꾸며 이야기를 진행한다면 보다 사실감 있는 캐릭터 구현이 가능할 것이다. TexMo2 라고 이름 붙여진 이 시스템은 입력된 문장에서 감정 상태의 흐름을 출력한다. 이전 버전인 TexMo가 naive Bayes 알고리즘을 사용했기 때문에 한 문장에서 하나의 감정 상태를 출력했던 것과 달리, TexMo2는 HMM을 사용함으로써 문장 내의 각 단어마다 상태를 결정 할 수 있다. 이는 캐릭터의 에니메이션을 실시간에 조합해 보여 줄 수 있는 3D 그래픽스의 특징에 적합한 시스템임을 뜻한다. 표 1은 TexMo2에서 인식 가능한 상태를 보여주고 있다. 이러한 상태들은 가능한 적은 범주로 많은 감정을 표현할 수 있도록 결정되었으며, 따라서 이미지로 표현할 때도 가능한 여러 감정을 표현할 수 있도록 디자인되어야 한다. 이 글에서는 먼저 HMM에 관해 간략하게 리뷰를 하고, TexMo2의 시스템 구조에 대해 설명을 한 후, 감정인식을 위해 사용되는 알고리즘인 HMM이 어떻게 쓰여졌는지 볼 것이며, 마지막으로 실제 채팅 데이터를 이용한 실험에서 나온 결과를 보일 것이다.

분류 범주	입 력 예
0. 무동작	여러 가지 자연스러운 대화
1. 미소	인사, 축하혜, 사랑혜 등의 여러 가지 상황
2. 질문	질문을 하거나 모른다는 표정 등
3. 긍정	긍정, 이해, 동의 등에 사용
4. 부정	부정, 거절, 반대 등에 사용
5. 고민	고민, 체념, 한숨, 미안함, 슬픔, 지루함 등
6. 놀람	놀람, 황당함, 감탄 등의 표정
7. 화남	화나거나 흥분했을 경우
8. 울음	눈물을 흘림
9. 강조	"대단히" "정말" 같은 강조하는 말 등에 사용
10. 집중	미간을 찌푸리며 집중하는 표정

표 1. TexMo2가 현재 분류 가능한 범주

2. Hidden Markov Modeling

HMM은 각 상태(state)간 전이 확률을 가지는 유한 상태 기계(finite state machine)이며, 각 상태들은 직접적으로 관찰 가능하지는 않으나, 대신 각 상태들이 일정 확률을 가지고 만들어 내는 심벌을 보고 원래 상태를 추정하는 방법이다. HMM은 다음과 같은 3가지 문제를 해결하는 것으로 요약될 수 있다.

1. 관찰된 심벌의 시퀀스 $O=O_1O_2\cdots O_T$ 와 모델 $\lambda=(A,B,\pi)$ 가 주어졌을 때 모델에 대한 심벌의 likelihood $P(O|\lambda)$를 구하는 문제
2. 위와 같이 O와 λ가 주어졌을 때 심벌의 시퀀스 O에 부합하는 상태의 시퀀스 $Q=q_1q_2\cdots q_T$를 선택하는 문제
3. $P(O|\lambda)$를 최대로 하는 모델 파라메타 $\lambda=(A,B,\pi)$ 를 구하는 문제

이때 모델의 파라메타 A,B,π 에 대한 설명은 다음과 같다. $A=\{a_{ij}\}$, 이때 a_{ij}는 상태 i에서 상태 j로 천이할 확률 $B=\{b_j(k)\}$, 이때 $b_j(k)$는 상태 j에서 심볼 k를 볼 확률 π_i는 첫번째 상태가 i일 확률

위의 세가지 문제는 다음과 같이 각각 forward 알고리즘, Viterbi 알고리즘, Baum-Welch 알고리즘으로 해결이 가능하다.

Forward algorithm : $\alpha_t(i)$를 다음과 같이 정의한다.

$\alpha_t(i)=P(O_1O_2\cdots O_t,q_t=S_i\ |\lambda)$

즉 모델 파라메터 λ가 주어졌을 때 시간 t 에서 시퀀스 $O_1O_2\cdots O_t$를 보고 상태 S_i에 있을 확률이라 할 때 $\alpha_t(i)$는 다음과 같이 귀납적으로 구할 수 있다.

1) $\alpha_1(i)=\pi_i b_i(O_1),\ 1\le i\le N.$

2) $\alpha_{t+1}(j)=\left[\sum_{i=1}^{N}\alpha_t(i)a_{ij}\right]b_j(O_{t+1}),\ 1\le t\le T-1$
$1\le j\le N.$

3) $P(O\ |\lambda)=\sum_{i=1}^{N}\alpha_T(i).$

Viterbi Algorithm : $\delta_t(i)$를 다음과 같이 정의하고

$$\delta_t(i)=\max_{q_1,q_2,\cdots,q_{t-1}} P[q_1q_2\cdots q_t=i,O_1O_2\cdots O_t\ |\lambda]$$

$\Psi_t(j)$를 시간 t 에서 상태 j에 있을 때 $t-1$ 에 나타날 수 있는 최적 상태를 나타낸다고 할 때 최적 패스는 다음과 같이 구해진다.

1) $\delta_1(i)=\pi_i b_i(O_1),\ 1\le i\le N$

$\Psi_1(i)=0.$

2) $\delta_t(j)=\max_{1\le i\le N}[\delta_{t-1}(i)a_{ij}]b_j(O_t),\ 2\le t\le T$
$1\le j\le N$

$\Psi_t(j)=\arg\max_{1\le i\le N}[\delta_{t-1}(i)a_{ij}],\ 2\le t\le T$
$1\le j\le N.$

3) $P^*=\max_{1\le i\le N}[\delta_T(i)]$

$q_t^*=\arg\max_{1\le i\le N}[\delta_T(i)].$

4) 최적 경로 backtracking

$q_t^*=\Psi_{t+1}(q_t^*),t=T-1,T-2,\cdots,1.$

Viterbi 알고리즘 : $P(O^{(k)}|\lambda)$를 간단히 P_k로 표기하고 $O^{(k)}$를 k번째 학습데이터라고 하면 세개의 파라메터는 다음과 같은 식에 의해 갱신된다.

$$\overline{\pi_i}=\frac{\sum_{k=1}^{K}\frac{1}{P_k}\alpha_1^k(i)\beta_1^k(i)}{\sum_{k=1}^{K}\frac{1}{P_k}\sum_{i=1}^{N}\alpha_1^k(i)\beta_1^k(i)}$$

$$\overline{a_{ij}}=\frac{\sum_{k=1}^{K}\frac{1}{P_k}\sum_{t=1}^{T_k-1}\alpha_t^k(i)a_{ij}b_j(O_{t+1}^{(k)})\beta_{t+1}(j)}{\sum_{k=1}^{K}\frac{1}{P_k}\sum_{t=1}^{T_k-1}\alpha_t^k(i)\beta_t^k(i)}$$

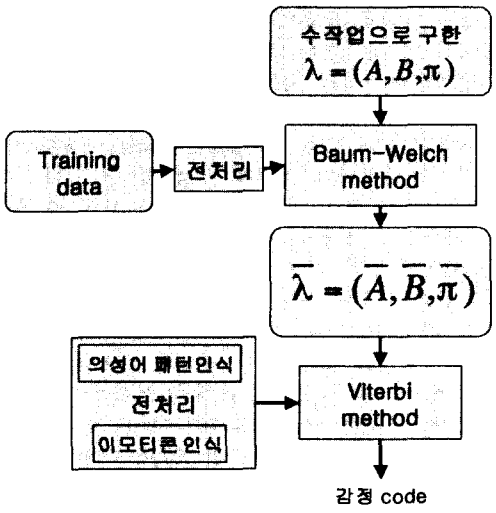
$$\overline{b_j(l)}=\frac{\sum_{k=1}^{K}\frac{1}{P_k}\sum_{s.t.O_t=v_l}^{T_k-1}\alpha_t^k(i)\beta_t^k(i)}{\sum_{k=1}^{K}\frac{1}{P_k}\sum_{t=1}^{T_k-1}\alpha_t^k(i)\beta_t^k(i)}$$

이때 $\beta_t(i)$는 backward 알고리즘으로 구할 수 있는데 이는 forward 알고리즘과 유사하나 시간 t이후의 심벌을 본다는 것이 다르다(자세한 내용은 [1] 참조)

3. 시스템 구조

전체 시스템 구조가 그림 1에 나와 있다. 전처리 모듈에서는 입력 문장에서 불필요한 각종 기호를 제거하고, 감정 표현에 결정적인 단서가 되는 그림문자를 검색해서 따로 처리하며 웃음소리와 같이 한글자가 연속되는 패턴을 찾아 한가지 패턴으로 수정해 준다. 이러한 처리를 거침으로써 사전의 크기를 크게 줄여 줄 수 있다.

시스템 구현에서 가장 어려운 부분은 모델 λ를 결정하는 일이다. 문제는 알고리즘의 특성상 파라메타의 초기 값이 제대로 결정되지 않으면 학습 결과가 local maximum에 빠져버린다는 것이다. 특히 B의 초기 행렬이 제대로 결정되지 않으면 시스템의 성능이 제대로 나오지



않았다. 이를 해결하기 위해 우리는 naive Bayes알고리즘을 사용한 TexMo의 학습에 이용됐던 통계자료를 사용했다. 즉 단어 k가 감정 상태 j에 속할 확률 $b_j(k)$를 사람이 수작업으로 한 태깅을 통해 가지고 있었고 이를 통해 A와 π 또한 구할 수 있었으므로 HMM의 구현시 큰 도움이 될 수 있었다. 또 한가지 문제는 $b_0(k)$에 대한 문제이다. 상태 0은 어떠한 감정 상태에도 속하지 않는 '무동작 상태'이다. 문제는 다른 상태와 달리 상태 0에 속하는 단어들이 다른 상태에 비해 상대적으로 그 양이 너무 많다는 것이다. 이러한 문제를 해결하기 위해 naive Bayes를 사용한 모델에서 태깅을 할 때 감정 번호를 매기지 않은 단어는 모두 무동작 상태로 간주하였으나 HMM에서는 자주 쓰이는 단어 중 무동작 상태에 해당하는 것들을 선별해서 상태 0번으로 태깅해 주었다.

위와 같이 λ가 초기화되면 EM 알고리즘을 사용하여 모아진 채팅 데이터를 학습시킨다. 학습에 사용되는 데이터는 모두 2가지 이상의 상태를 가진 문장을 사용했다. 이미 초기화 단계에서 한가지 상태를 갖는 문장에 대한 감정값을 학습했으므로 여기선 두개 이상의 상태를 가진 문장만을 학습한다. 그렇지 않은 경우 하나의 상태를 가진 문장이 월등히 많아 상태 전이의 확률이 제대로 반영되지 않았다.

학습 단계의 전처리와 달리 분석 단계에서의 전처리에서는 "하하하", "호호호"등의 의성어 패턴과 "^^, :-)" 같은 이모티콘을 검출하는 두 가지 모듈이 각각 추가된다. 이 두 가지는 감정표현을 명시적으로 나타내는 것들이기 때문에 분석 알고리즘에 들어가기 전에 미리 처리되어 표현된다. 마지막으로 테스트 데이터의 분석에는 Viterbi 알고리즘이 사용된다. 이때 사전에 없는 단어가 있을 경우 그 단어는 무시하는 것으로 했다.

4. 실험 결과

실험에서 알고리즘 성능 비교를 위해 naive Bayes를 사용한 방법이 사용되었다[7]. 표 2는 naive Bayes 방법과 HMM 방법의 성능 비교가 나와 있다. 학습을 위해 채팅 사이트에서 모은 20000여 문장이 사용되었다.

테스트에는 학습데이터와는 다른 내용의 채팅데이터 794문장이 사용되었다. 테스트 데이터는 한 단어로 이루어진 문장과, 두 단어 이상일 경우에 감정값을 하나만 갖는 경우, 그리고 같은 경우에 감정값을 두개 이상 갖는 경우로 나누어졌다. 결과에서 보듯이 한 단어를 분석한 결과는 naive Bayes 방법과 큰 차이가 없었다. 이는 입력문에 한가지 상태만 존재할 경우 상태 변화에 대한 정보가 없어 사실상 두 알고리즘의 차이가 없기 때문이다. 두 단어 이상일 경우의 비교에서 naive Bayes 방법에서는 정확도가 크게 떨어지는 반면 HMM에서는 차이가 그리 크지 않음을 알 수 있다. 특히 naive Bayes에서는 다중 상태일 경우 그중 가장 대표되는 감정을 표현하는지를 검사하는데, 이 경우 두개의 상태 중 하나를 임의로 결정했다고 볼 수 있을 정도로 정확도가 떨어졌다.

결과에서 볼 때 HMM을 사용한 감정인식이 한 문장에 둘 이상의 상태변화가 있을 경우 인식률이 높게 나옴을 알 수 있다.

		naive Bayes	HMM
한단어 분석		0.91	0.89
두단어 이상 분석	단일 상태	0.83	0.87
	다중 상태	0.54	0.92

표 2 테스트 데이터에 대한 정확도(accuracy)

5. 결론

우리는 여기서 채팅데이터의 한 문장을 분석하여 10개로 분류된 감정코드의 시퀀스로 출력해 주는 시스템의 개요와 실험 결과에 대해서 살펴보았다. 이전 버전보다 HMM을 사용한 시스템의 성능이 향상되었다는 것을 알 수 있었다. 특히 한 문장에 두 가지 이상의 감정 상태가 포함되었을 경우 높은 인식율을 나타내었다. HMM을 사용한 감정인식 시스템의 특징은 naive Bayes를 사용할 때와 달리 한 문장에서도 여러 감정값 출력이 가능하다는 것이다. 이는 보다 자연스러운 캐릭터의 표정을 만들어 낸다. 실험에서 가장 문제가 되는 부분은 채팅에서 사용되는 문장의 특성상 문법이나 띄어쓰기, 또는 맞춤법이 어긋난 데이터가 너무 많다는 것이다. 만약 실험 데이터가 정규 회화 문장으로 사용된다면 보다 좋은 성능을 얻을 수 있을 것으로 본다. 또는 채팅데이터에서 맞춤법이나 오타를 교정해 주는 모듈을 전처리에 추가한다면 더 낳은 성능을 얻을 수 있을 것이다.

감사의 글: 본 연구는 교육부 BK 21의 지원과 첨단정보기술 연구센터 (AITrc)를 통한 과학재단의 지원을 받았음.

<참고문헌>

[1] L. Rabiner and B. Juang. An Introduction to Hidden Markov Models. *IEEE ASSP Magazine*, pp. 4-16, 1996.

[2] E. Charniak. *Statistical Language Learning*. MIT Press, 1997.

[3] D. Miller, T. Leek, R. Schwartz. A Hidden Markov Model Information Retrieval System. *In SIGIR-99*, pp. 214-221, 1999.

[4] T. Mitchell. *Machine Learning*. McGraw-Hill Companies, 1997.

[5] A. McCallum and K. Nigam. A Comparison of Event Models for Naive Bayes Text Classification. In *AAAI-98 Workshop on Learning for Text Categorization*, pp. 41-48, 1998.

[6] K. Nigam, A. McCallum, S. Thrun, and T. Mitchell. Learning to Classify Text from Labeled and Unlabeled Documents. In *AAAI-98*, pp. 792-799, 1998.

[7] 문 현구, 장 병탁. "채팅 텍스트로부터의 화자 감정상태 학습", 한국정보과학회 봄 학술발표논문집, pp. 340-342, 2001.