Comparison of Classifiers for Natural Language Processing.

2017312576 수학과 이동준

요 약

현재까지 기계학습 분야에서 딥러닝이 뜨면서 여러 상황에서 사용이 되고 있다. 하지만, 사용하는 분야 및 환경에 따라서 분류기의 성능이 달라 아직까지도 K-NN, SVM 등의 인공신경망이 아닌 고전의 분류기들이 아직까지는 유효하다는 말도 존재한다. 그래서 본 기술보고서에서는 자연어 처리의 기사 분류에서 영문의 BBC 기사라는 상황을 한정시켜서 K-NN, NB, SVM, MLP 알고리즘을 이용한 분류기들을 매개변수를 변경해가며 학습시키고, 성능을 테스트하여 특정 환경에서 어떠한 분류기가 성능이 가장 좋은지 확인을 한다. 본 기술 보고서의 결과에서는 데이터 셋이 이렇게 한정되었을때 NB가 가장 좋은 성능을 내었으며, 매개변수에 따른 차이를 고려하였을때 SVM도 최적의 선택지 중 하나가 될 수 있다는 것을 보았다.

1. 서론

지도학습의 방법을 사용한 분류기는 입력으로 할 데이터와 분류기의 출력에서 목적으로 하는 라벨을 가지고 분류기를 학습시켜 이 분류기에 새로운 데이터를 입력하여 이에 해당되는 라벨을 얻어낸다. 과거부터 최근까지의 여러 연구로 인해 K-Nearest Neighbors(K-NN), Naïve Bayes(NB) Support Vector Machine(SVM), Multi Layer Perceptron(MLP) 등과 같이 여러 분류기가 등장하였으며, 이들은 자연 언어 처리(NLP), 컴퓨터 비전(CV)등의 분야에서 각각 기사 분류, 음성인식, TTS, 패턴인식, 디노이징 등에서 사용이 되고 있다. 그러나 같은 분야에서 사용하더라도 분류기의 성능이 달라질 수가 있는데, 같은 자연 언어 처리에서 기사의 주제를 분류하는 방식이더라도 작성된 언어, 기사가 작성된 플랫폼의 성향, 문화, 언어에 따라서 성능이 달라질 수가 있다.

본 기술 보고서에서는 BBC에서 영문으로 작성된 기사들을 데이터로 하여 각 기사들의 주제를 얻어내는 것을 목적으로 학습 데이터셋의 범위를 정한다. 이 상황에서 각각 NB, K-NN, SVM, MLP를 기반으로 한 분류기를 구성하여 학습 데이터를 통해 학습시키고, 테스트 데이터를 가지고 정확도를 측정하여 각각의 분류기의 성능을 비교한다.

....

2. 본론

본 기술 보고서에서는 기사에서 tf-idf 값을 추출하여 이를 분류기의 입력값으로 두고, K-NN, NB, SVM, MLP 분류기를 사용하였으며, 정확도를 내는 방식으로는 accuracy가 기본으로 사용되고 부가적으로 precision, recall, f1-score 방식을 Macro/Micro averaging 사용하여 각각 분류기들의 성능을 비교할 것이다.

2.1. tf-idf

tf-idf는 term frequency - inverse document frequency의 약자로, 특정 단어가 문서에서 얼마나 중요한지 반영하기 위한 수치 통계이며, 문서에서 나온 빈도 수를 나타내는 tf와 흔히 사용되는 단어인지 특수한 것인지 나타내기 위해 총 문서 수에서 단어가 나온 문서의 수를 나눈 값에 로그를 취해 얻는 idf를 곱하여 얻는다.

2.2. 분류기

2.2.1 K-NN : 간단한 게으른 기계학습 알고리즘 중 하나이다. 학습을 통해 생성된 샘플 그룹에서 입력된 데이터를 기준으로 k개의 가장 가까운 데이터를 가지고 가장 유사한 결과를 도출해내는 알고리즘 [1].

2.2.2 Naïve Bayes(NB): 특징들이 독립적이라고 가정하여 베이즈 정리를 기반으로 한 간단한 확률기반 분류 모델이다. 본 기술 보고서에서는 다항 분포를 사용하는 Multinormal NB를 사용할 것이다.

2.2.3 SVM: SVM 분류기는 학습할 데이터를 특정 공간으로 매핑 시킨 후 두 분류 사이의 마진을 최대화하는 초평면을 찾아내도록 학습시킨다. 그리고 이렇게 찾아낸 초평면을 기준으로 입력된 데이터의 라벨을 특정시키는 분류이다[2]. 본 기술 보고서에서는 선형 SVM을 사용할 것이다.

2.2.4 MLP: 인공신경망을 통한 알고리즘 중 하나로, 여러 계층의 퍼셉트론으로 구성된 네트워크를 사용한다.

입력, 히든, 출력의 최소 3개의 레이어로 구성이 되며, 역전파 알고리즘을 사용하여 모델을 학습시킨다[3].

....

3. 실험

본 기술 보고서에서는 BBC에서 작성된 기사 중에서 Business, Politics, Tech 주제들에서 각각 100개를 가져와 사용을 할 것이다. 학습 데이터로는 각 주제에서 80개의 데이터를 사용할 것이며, 남은 20개를 테스트 데이터로 사용하여 분류기의 성능을 측정할 것이다.

실행 환경은 Colab환경에서 Python 3.6.9 버전을 사용하였으며, tfidf 계산, 분류기 생성 및 학습, 정확도 측정은 파이썬 sklearn 라이브러리를 사용하였다.

3.1 전처리

기사를 파이썬 스트링으로 받은 후, 각각의 단어들을 nltk 라이브러리에서 word_wokenize 라이브러리를 사용하여 토큰을 만들고 이 중에서 명사, 동사의 태그를 가진 단어들만을 가지고 sklearn의 TfidfVectorizer 를 사용하여 tfidf를 계산하여 데이터의 입력에 맞춘다.

3.2 분류기

3.2.1. K-NN: sklearn의 KNeighborsClassifier를 사용하며, 매개변수 거리공간은 유클리디안 거리를 사용하고 n_neighbors 를 1에서 5까지 사용할 것이다.

3.2.2. NB: sklearn의 MultinomialBN를 사용하며, 매개변수 alpha 를 0.0001, 0.001, 0.002, 0.005, 0.01 를 사용할 것이다,

3.2.3. SVM: sklearn의 LinearSVC를 사용하며, 매개변수 C는 0.1, 0.5, 1.0, max_iter 은 500, 1000 를 사용할 것이다.

3.2.4. MLP: sklearn의 MLPClassifier를 사용하며, 매개변수는 각각 activation: tanh, solver: adam, batch_size : auto, hidden_layer_sizes : (100,), learning_rate_init를 0.0005, 0.001, 0.005, max_iter 은 200, 500 를 사용할 것이다.

3.3 성능평가

기본적으로는 accuracy를 볼것이나 부가적으로 precision, recall, f1-score 방식을 Macro/Micro averaging를 측정할 것이다. 또한, 성능이외의 요소인 학습속도 또한 측정할 것이다.

3.4 실험결과

표 1. K-NN

n_neighbors	accuracy	macfro precision	micro precision	macro recall	micro recall	macro f1	micro f1
1	0.9667	0.9683	0.9667	0.9667	0.9667	0.9662	0.9667
2	0.9667	0.9683	0.9667	0.9667	0.9667	0.9662	0.9667
3	0.9	0.9021	0.9	0.9	0.9	0.9	0.9
4	0.9	0.9007	0.9	0.9	0.9	0.9	0.9
5	0.9	0.904	0.9	0.9	0.9	0.900	0.9
		2				6	

표 2. Naive Bayes

alpha	accuracy	macfro precision	micro precision	macro recall	micro recall	macro f1	micro f1
0.0001	0.8667	0.8677	0.8667	0.8667	0.8667	0.8661	0.8667
0.001	0.9	0.9	0.9	0.9	0.9	0.9	0.9
0.002	0.9167	0.9174	0.9167	0.9167	0.9167	0.9166	0.9167
0.005	0.9333	0.934	0.9333	0.9333	0.9333	0.9333	0.9333
0.01	0.9333	0.934	0.9333	0.9333	0.9333	0.9333	0.9333

표 3. Support Vector Machine

C, max_iter	accuracy	macfro precision	micro precision	macro recall	micro recall	macro f1	micro f1
0.1, 500	0.9333	0.9406	0.9333	0.9333	0.9333	0.9316	0.9333
0.1, 1000	0.9333	0.9406	0.9333	0.9333	0.9333	0.9316	0.9333
0.5, 500	0.9333	0.9406	0.9333	0.9333	0.9333	0.9316	0.9333
0.5, 1000	0.9333	0.9406	0.9333	0.9333	0.9333	0.9316	0.9333
1, 500	0.95	0.9538	0.95	0.95	0.95	0.949	0.95
1, 1000	0.95	0.9538	0.95	0.95	0.95	0.949	0.95

표 4. Multi Layer Perceptron

lr, max_iter	accuracy	macfro precision	micro precision	macro recall	micro recall	macro f1	micro f1
0.0005, 200	0.9	0.9069	0.9	0.9	0.9	0.9006	0.9
0.0005, 500	0.9167	0.9228	0.9167	0.9167	0.9167	0.9168	0.9167
0.001, 200	0.9167	0.9228	0.9167	0.9167	0.9167	0.9168	0.9167
0.001, 500	0.9167	0.9228	0.9167	0.9167	0.9167	0.9168	0.9167
0.005, 200	0.9	0.9029	0.9	0.9	0.9	0.8998	0.9
0.005,	0.9	0.9029	0.9	0.9	0.9	0.8998	0.9

500							

표 5. 학습시간(초)

K-NN 1	K-NN 2	K-NN 3	K-NN 4	K-NN5	
0.1577	0.153	0.1428	0.171	0.1499	
NB 1	NB 2	NB 3	NB 4	NB 5	
0.0874	0.1312	0.1362	0.132	0.1338	
SVM 1	SVM 2	SVM 3	SVM 4	SVM 5	SVM 6
0.1391	0.1397	0.1393	0.1411	0.1431	0.1456
MLP 1	MLP 2	MLP 3	MLP 4	MLP 5	MLP 6
15.79	15.78	9.938	10.07	2.616	2.514

5. 결론

이 각각의 분류기에서 매개변수를 조절해가며 학습 데이터셋으로 학습을 시키고 테스트 데이터를 통하여 정확도를 측정하였다. K-NN에서는 n_neighbors가 1, 2일때, NB에서는 alpha가 0.05, 0.1일때, SVM에서는 C가 1일때, MLP 에서는 (learning_rate_init, max_iter)가 (0.0005, 500), (0.001, 200), (0.001, 500)일때 가장 크게 나왔다.

분류기 끼리의 비교로는 가장 정확도가 높을때를 기준으로는 K-NN이 0.9667로 가장 높게 나왔으며, SVM이 0.95, NB가 0.9333, MLP가 0.9167의 순서로 분류기의 성능이 비교가 되었다.

학습 시간으로는 다른 분류기들의 시간의 차이는 큰 의미가 없으나 MLP의 경우 다른 분류기들에 비해 100배정도의 차이로 다른 분류기들이 학습하는 것에 비해 오래 걸리는 것이 보인다. 또한, learning_rate_init 이 클 수록 목표에 더 빠르게 도달하여 학습 시간이 감소하는것이 보이며, 값이 0.001 이상일 때는 max_iter 이전에 수렴하는 것 또한 알 수 있다. 또한, MLP는 이 분류에서 시간이 오래걸리면서 좋은 성능을 내지 못하는 것을 확인 할 수 있었는데 이를 통해 MLP의 분류기가 다른 분류기들에 비해 떨어지는 것을 알 수 있다.

다른 유사한 연구들의 결과[4, 5, 6]들의 경우는 SVM이 다른 분류기들에 비하여 성능이 좋게 나온 것에 비하여 본 기술 보고서의 결과에서는 가장 좋은 성능을 내지는 못하였다. 하지만, 보았을때 다른 분류기들에 비해 성능이 좋은 편인것은 맞으며, 다른 K-NN, NB 분류기는 매개변수의 설정에 따라 성능 차이가 나는편이지만 SVM은 매개변수에 따라 성능 차이가 크지 않은것으로 보아 다른 논문들 처럼 SVM의 분류기가 좋은 성능이라는 것은 알 수 있다.

6. 참고문헌

[1] Bruno Trstenjaka, Sasa Mikacb, Dzenana Donkoc "KNN with TF-IDF Based Framework for Text Categorization", DAAAM International Vienna, 24th, p1358, 2013

[2]

- 기술 순서는 저자, 제목, 학술지명, 권, 호, 쪽수, 발행년도 순으로 작성.

[2] 이수용, 손소영, 김철응, 이일병 "Fuzzy Support Vector Machine for Pattern Classification of Time Series Data of KOSPI200 Index" 퍼지 및 지능 시스템학회 논문지 Vol. 14, No. 1 p52~56, 2004

[3] Lecun, Y., L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. Proceedings of the IEEE Vol. 86, No. 11 p2278–2324, 1998

[4] Ananthi Sheshasaayee and G. Thailambal. "Comparison of Classification Algorithms in Text Mining" International Journal of Pure and Applied Mathematics Vol. 116 No. 22 p425-433, 2017

[5] Fabrice COLAS and Pavel BRAZDIL "Comparison of SVM and Some Older Classification Algorithms in Text Classification Tasks" IFIP International Federation for Information Processing, Vol. 217, p169~178, 2006

[6] Adel Hamdan Mohammad, Tariq Alwada'n, Omar Al-Momani, "Arabic Text Categorization Using Support vector machine, Naïve Bayes and Neural Network" GSTF Journal on Computing Vol. 5 No. 1 p108~115, 2016