대용량 문서분류에서의 비선형 주성분 분석을 이용한 특징 추출

신 형 주, 장 병 탁, 김 영 택
서울대학교 컴퓨터공학과
{hjshin, btzhang}@scai.snu.ac.kr, ytkim@comp.snu.ac.kr

Feature Selection with Non-linear PCA in Text Categorization

Hyung Joo Shin, Byoung-Tak Zhang, Yung Taek Kim
School of Computer Engineering and Science, Seoul National University

요 약

문서분류의 문제점 중의 하나는 사용하는 데이터의 차원이 매우 크다는 것이다. 그러므로 문서에서 필요한 단어만을 자동적으로 추출하여 문서데이터의 차원을 축소하는 작업이 문서분류에서는 필수적이다. DF(Document Frequency)는 문서의 차원축소의 대표적인 통계적 방법 중 하나인데, 본 논문에서는 문서의 차원축소에 DF 와 주성분 분석(PCA)을 비교하여 주성분 분석이 문서의 차원축소에 적합함을 실험적으로 보인다. 그리고 비선형 주성분 분석(nonlinear PCA)방법 중 locally linear PCA 와 kernel PCA 를 적용하여 비선형 주성분 분석을 이용하여 문서의 차원을 줄이는 것이 선형 주성분 분석을 이용하는 것 보다 문서분류에 더 적합함을 실험적으로 보인다.

1. 서 론

문서분류학습은 미리 어떤 범주(category)에 속하는 지 알려진 문서로 분류 시스템을 학습하는 감독 학습이다. 회귀 모델(regression model), 나이브 베이지안 확률 모델(Naive Bayesian probabilistic model, NB), 결정나무모델(Decision tree model), 귀납적 학습 모델(Inductive rule learning model), 신경망 모델(Neural Networks, NNets), Support Vector Machines(SVM), k-Nearest Neighbor(kNN)모델, Linear Least Square Fit(LLSF)모델 등 여러 통계적인 학습 방법이 여기에 응용되었다. [1]에서 이러한 방법들을 정리하고 실험적으로 비교하고 있다.

문서분류가 가지고 있는 큰 문제중의 하나는 사용하는 데이터의 차원이 매우 크다는 것이다. 여기서 차원이란 문서의 단어, 즉 feature 를 의미하는 것으로, 분류하고자 하는 문서의 모든 단어를 사용한다면 몇 백만의 차원을 가지는 데이터를 학습에 사용해야 할 것이다. 그러므로 문서분류에서는 문서에서 필요한 단어만을 자동적으로 추출하는 작업(Automatic feature selection, 이하 차원축소)이 필수적이다.

통계적인 차원축소를 위해 Lewis & Ringuette 는 베이지안 모델, 결정나무모델과 함께 정보 획득량(Information Gain, IG)을, Wiener 등은 신경망 모델과 함께 상관정보(Mutual Information, MI)와 χ-square(CHI)를, Yang 이나 Schutze 는 LLSF 와 함께 선형주성분분석(linear PCA)을, Yang & Wilbur 는 kNN 과 함께 클러스터링을, Lang 은 Minimum Description Length(MDL)을 사용하였다. 그리고 DF(documet frequency)는 가장 간단한 방법임에도 불구하고 대용량의 문서분류에서 IG, CHI 와 함께 좋은 성능을 나타낸다. [2]에서 이러한 방법들을 정리하고 실험적으로 비교하고 있다.

문서분류의 벤치마크 데이터 중 하나인 Reuters 21578 데이터에 대해서는 SVM, LLSF, kNN 이 NB 나 NNets 보다 더 좋은 성능을 보이는데 [1], LLSF[3]에서 차원축소에 사용된 주성분분석은 다변량(multivariate) 통계분석에 사용되는 가장 대표적인 방법이다.

주성분분석에는 선형적인 분석과 비선형적인 분석이 있는데 지금까지 문서분류에서 차원축소를 위해 사용된 주성분분석은 선형적인 분석이었다. 그러나 문서와 같은 복잡한 데이터는 다변량 변수들간에 강한 비선형 상관관계를 가질 것으로 예상되므로 비선형 주성분분석을 이용해 문서의 차원을 축소하는 것이 더 효율적일 것이다.

비선형 주성분분석 방법에는 Hebbian networks, associated multi-layer perceptrons, principal curves, locally linear PCA, kernel PCA 등이 있다. 본 논문에서는 문서의 차원축소에 DF 와 주성분 분석을 비교하여 주성분 분석이 문서의 차원축소에 적합함을 실험적으로 보일 것이다. 그리고 비선형 주성분 분석방법 중 locally linear PCA [5]와 kernel PCA [4]를 적용하여 비선형 주성분 분석이 선형 주성분 분석을 이용하는 것 보다 문서분류에서 문서의 차원을 줄이는 데에 더 적합함을 실험적으로 보인다. 분류 시스템은 n-ary 분류에 응용하기 간단하고 성능도 비교적 좋은 kNN [10]을 사용할 것이다.

2. 주성분 분석 (PCA)

주성분 분석은 고차원 데이터로부터 데이터의 구조를 밝히거나, 데이터의 차원을 낮추는 데 많이 이용되는 다변량 통계 분석 방법이다. 이는 상관행렬(correlation matrix)의 고유벡터(eigenvectors)를 찾아내는 문제로 행렬 연산으로 찾아내는 방법과 신경망 등을 사용하여 반복적으로(iteratively) 찾아내는 방법 등이 있다 [6]. 즉, 주어진 데이터를 분산이 최대가 되는 축으로 변환하는 것으로, 이 새로운 차원에서의 데이터의 벡터들을 주성분(principal components)이라고 한다. 이 때 분산이 작은 성분을 제거함으로써 데이터의 차원을 줄이는 동시에 데이터에 포함되어 있던 잡음(noise)을 제거할 수 있다. 데이터 행렬 $\mathbf{X}$ 의 차원을 k 로 낮추는 식이 다음과 같다.

$$\mathbf{X} \cdot \mathbf{V}^k \qquad (1)$$

여기서 $\mathbf{V}$ 는 $\mathbf{X}$ 의 상관행렬의 고유벡터를 해당하는 고유값(eigenvalue)의 내림차순으로 정렬한 행렬이고, k 는 이 중 k 개의 열을 사용하겠다는 의미이다.

그런데 선형 주성분 분석(linear PCA)는 데이터의 변수들 간에 비선형적인 상관관계가 있을 경우 적당하지 못하다. 비선형

주성분 분석(nonlinear PCA)은 선형 주성분 분석의 이러한 단점을 극복하는데, principal curves 의 개념으로 [7]에서 소개되었다. 그 후 [8]에서 이산적인(discrete) principal curves 를 SOM 으로 찾는 방법이, [9]에서는 4-layer MLP(Multi Layer Perceptron)으로 principal curves 를 찾는 것이 제안되었다. 또한 최근에 [4]에서 kernel PCA 가, [5]에서 locally linear PCA 가 소개되었다.

Kernel PCA [4]는 주어진 차원의 데이터를 고차원의 공간(feature space)으로 사상(mapping)하여 그 고차원의 공간에서 주성분 분석을 함으로써 결과적으로 비선형 주성분 분석을 하는 것이다. 데이터 $\mathbf{x}_i$ (i=1,…,M, M 은 데이터 개수)를 feature space 로 사상한 $\mathbf{\Phi}(\mathbf{x}_i)$를 주성분 분석으로 k 차원으로 낮추는 사상이 식 (2)의 좌변인데, 이는 계산이 매우 어렵기 때문에 이를 계산하는 대신 우변과 같이 kernel 행렬을 계산하여 이 행렬에 대해 주성분 분석을 하는 것이 kernel PCA 이다. 이를 식으로 나타내면 다음과 같다.

$$\mathbf{V}^k \cdot \mathbf{\Phi}(\mathbf{x}_i) = \sum_{j=1}^{M} \alpha_i^k \mathbf{K}(\mathbf{x}_i, \mathbf{x}_j) \qquad (2)$$

(2)에서 $\mathbf{V}$ 는 feature space 에서의 데이터의 상관행렬의 고유벡터 이고, $\boldsymbol{\alpha}$는 kernel 행렬의 상관행렬의 고유벡터 이다. 이 때 kernel 함수로 polynomial kernel, radial-basis 함수, sigmoid kernel 을 사용할 수 있다. 데이터의 개수가 M 개일 때 kernel 행렬은 M×M 이다.

Locally linear PCA [5]는 주어진 데이터의 차원을 clustering 하여 각각의 cluster 에 대해 선형 주성분 분석을 함으로서 결과적으로 비선형 주성분 분석을 하는 것이다. [5]에서는 clustering 에 Vector Quantization(VQ)을 사용하였으나 본 논문에서는 Self-Organizing Maps(SOM)을 사용하였다.

3. 실험

3.1. 데이터

실험에 사용한 데이터는 Reuters-21578 이다. Reuters-21578 은 5 개의 범주집합으로 구성되어 있고, 각 범주집합은 또 여러 개의 범주로 구성된다. 본 논문에서는 이 중 연구가 활발한 경제분야(TOPICS)의 135 개의 범주 중 데이터의 개수가 많은 10 개의 범주의 문서를 사용하였다. 원래의 데이터는 각각의 범주에 대해 그 범주에 속하는 데이터(positive data)와 속하지 않는 데이터(negative data)가 함께 있는데, 우리는 n-ary 분류를 할 것이므로 주어진 negative 데이터를 사용하지 않고, 한 범주에 속한 데이터의 negative 데이터는 다른 9 개의 범주에 속한 모든 데이터로 한다. 이를 위한 학습 데이터와 테스트 데이터의 개수가 표 1 과 같다.

표 1. 사용한 데이터의 범주별 개수

범주	학습 데이터 수	테스트 데이터 수
Corn	133	36
Ship	180	80
Wheat	185	54
Interest	275	97
Trade	304	103
Crude	334	156
Grain	370	125
Money-fx	428	130
Acq	1483	640
Earn	2706	1043
총	6397	2464

데이터를 벡터로 만들기 위해 Porter algorithm 을 사용하여 stemming 을 하고, 524 개의 stop word 를 제거한 후, 사전을 이용하여 8754 개의 feature(단어)를 뽑아내고, tf/idf(term frequency/inverse term frequency)를 이용하여 weight 를 매겼다 [1].

8754 개의 단어 중 학습 데이터 6397 개 중 모든 데이터에서 weight 가 0 이 되는 단어들을 제거하여 7218 개의 단어를 사용하였다. 결과적으로 학습 데이터는 6397×7218 의 벡터이고, 테스트 데이터는 2464×7218 의 벡터이다.

3.2. Feature Selection

다음의 방법들을 사용하여 데이터의 차원을 5000, 4000, 3000, 2000, 1000, 500 으로 낮추어 가며 분류를 하였다.

3.2.1. Document Frequency (DF)

모든 문서들에 대해 weight 가 0 이 아닌 항이 많은 feature 순으로 벡터를 구성하여 데이터의 차원을 낮췄다. 테스트 시에는 학습 데이터에서 사용한 feature 만 사용하여 테스트 데이터의 차원을 축소했다.

3.2.2. 선형 주성분 분석

선형 주성분 분석을 이용하여 데이터의 차원을 낮췄다. 이 때 변환행렬을 저장해 두어야 테스트 데이터의 차원을 낮추는데 사용할 수 있다. 데이터의 차원을 낮추는 변환은 식 (1)과 같다.

3.2.3. Locally linear PCA

Self-Organizing Map Program Package Ver. 3.1[2]을 가지고 feature 를 100 개의 cluster 로 clustering 하여 각 cluster 에 대해 선형 주성분 분석으로 데이터의 차원을 낮췄다. 7218 개의 feature 를 100 개로 clustering 했을 때 각 cluster 에 속하는 feature 의 개수가 표 2 에 나와 있다. 문서의 차원을 5000 개 이하로 축소 할 것이므로, 각 cluster 에 들어가는 feature 의 개수와 cluster 의 개수의 곱이 5000 이상이어야 한다. 그러므로 한 cluster 에 적어도 100 개 이상의 feature 가 포함되도록 SOM 에서 가까운 cluster 들을 합하여 50 개의 cluster 를 만들었다. 여기서 각 cluster 에 대한 변환 행렬을 저장해 두었다가 테스트 할 문서가 들어오면 테스트 데이터의 각 feature 가 속하는 cluster 에 대한 변환행렬에 의해 차원을 축소한다.

표 2. Feature 를 clustering 한 SOM 의 map

47	17	11	9	98	3	3	77	6	158
1	9	67	3	11	97	2	18	8	15
97	15	12	4	50	29	27	26	18	118
11	26	13	51	7	24	84	33	5	20
70	5	64	3	7	105	18	16	49	58
44	54	9	86	7	8	109	37	12	86
236	76	130	25	12	99	22	14	146	88
124	1065	76	93	35	15	104	58	98	257
465	109	109	76	12	63	40	14	158	88
201	288	165	64	83	44	93	66	22	109

3.2.4. Kernel PCA

polynomial kernel 과 radial-basis kernel 을 사용하였다. 이 때 polynomial kernel 함수에서 d=4 로 하였고, radial-basis kernel 함수에서 σ=1 로 하였다. 이렇게 만들어 진 kernel 행렬에 대해 선형 주성분 분석으로 데이터의 차원을 낮춘다. 마찬가지로 변환행렬을 저장해 두어야 한다. 테스트 데이터가 들어오면 학습 데이터를 이용하여 테스트 할 문서의 kernel 벡터를 만들고, 변환행렬에 의해 차원을 축소한다.

---

[1] 이를 위해 McCallum 등이 만든 *Bow*library 를 이용하였다. 이는 http://www.cs.cmu.edu/~mccallum/bow 에서 구할 수 있다.

[2] 이는 Teuvo Kohonen 등에 의해 1995 년에 만들어진 것으로 anonymous ftp site 인 cochlea.hut.fi 의 /pub/som_pak 에서 구할 수 있다.

3.3. 분류 시스템 및 성능 척도

분류 시스템은 n-ary 분류에 응용하기 간단하고 성능도 비교적 좋은 kNN [6]을 사용하였다. Similarity measure 로는 cosine measure 를 사용하였고 neighborhood size k 는 100 으로 하였다.

테스트 할 2464 개의 문서들도 이미 분류가 되어 있으므로, 분류 시스템이 분류한 범주와 이미 분류되어 있는 범주 중 몇 개가 일치하는가를 퍼센트로 하여 성능을 측정했다.

4. 결과 및 토의

4.1. 실험 결과

문서의 차원을 축소하지 않은 상태에서 kNN 으로 분류했을 때의 성능은 81.73%이다. 차원축소나 분류에 randomness 가 없으므로 실험은 1 번만 수행하였다.

문서의 차원을 축소할 때 DF 를 사용한 결과와 선형 주성분 분석을 사용한 결과가 표 3 과 같다.

표 3. DF 와 Linear PCA 비교 (단위: %)

	DF	Linear PCA
5000	80.03	81.24
4000	79.98	81.70
3000	79.12	82.67
2000	78.84	82.89
1000	76.19	81.31
500	66.24	76.11

표 4. Kernel PCA 와 Locally linear PCA 의 비교 (단위: %)

	Kernel PCA (polynomial)	Kernel PCA (radial basis)	Locally linear PCA
5000	80.98	81.25	82.28
4000	81.73	82.71	82.65
3000	82.22	84.13	83.18
2000	82.50	82.08	**84.33**
1000	80.18	80.22	81.17
500	76.97	78.45	76.84

표 5. 학습데이터의 개수와 성능간의 관계

범주	학습 데이터 수	성능 (단위: %)
Corn	133	76.04
Wheat	180	77.05
Ship	185	78.98
Interest	275	81.29
Trade	304	80.57
Grain	334	82.55
Money-fx	370	81.59
Crude	428	82.47
Acq	1483	85.15
Earn	2706	86.39
총	6397	84.33

표 4 는 비선형 주성분 분석 중 kernel PCA 와 locally linear PCA 를 사용하여 문서의 차원을 축소하여 분류한 결과이고, 표 5 는 locally linear PCA 로 데이터의 차원을 2000 으로 낮추어 분류했을 때 각 범주에 속하는 학습 데이터의 개수와 분류 성능의 관계이다.

4. 결론

표 3, 4 에서 알 수 있듯이 문서데이터는 차원을 축소하여도 어느 정도까지는 성능이 크게 떨어지지 않는데, 이는 문서데이터의 특성상 데이터의 weight 값 중 0 이 대부분으로, 데이터에 정보량이 적기 때문이다. 차원을 많이 축소할수록 분류를 빠르게 할 수 있지만 차원을 500 으로 축소하였을 때는 성능이 크게 떨어지는 사실에서 알 수 있듯이 문서분류 시스템을 구현할 때는 몇 개의 feature 를 사용할 지 결정하는 것이 시스템의 속도와 성능에 크게 영향을 미친다.

표 3 을 보면 DF 보다 데이터의 분포를 고려하는 주성분 분석이 차원축소로 인한 정보손실을 적게 한다는 것을 알 수 있다. 특히 feature 수가 2000 일 때는 모든 차원을 사용할 때 보다 더 나은 성능을 보이는데 이는 주성분 분석이 문서데이터의 잡음을 제거하는 효과를 가져오기 때문이다.

표 3 과 표 4 를 비교해 보면 비선형 주성분 분석이 선형 주성분 분석보다 정보손실을 줄여 문서의 차원축소에 더 적합하다는 것을 알 수 있다. 하지만 kernel PCA 를 사용하는 경우에는 선형 주성분 분석보다 오히려 성능이 떨어지는 경우도 있는데, 이는 데이터에 특성에 따라 어떤 kernel 을 선택하는 지가 성능에 영향을 미치기 때문이다.

결과적으로 locally linear PCA 는 DF, 선형 주성분 분석, kernel PCA 보다 좋은 성능을 보인다. Locally linear PCA 로 문서의 차원을 축소하는 경우에는 선형 주성분 분석과, 새로운 문서를 분류하는 데 걸리는 시간이 차이가 나지 않으면서도 성능은 더 좋기 때문에 문서분류를 위한 차원축소에 적당함을 알 수 있다.

그러나 표 4 를 보면 문서의 개수가 적은 경우 성능이 떨어지는 것을 볼 수 있다. 문서의 개수가 적은 경우 사용할 수 있는 정보도 적기 때문인데, 이를 극복하기 위한 연구가 필요하다. 그리고 비선형 주성분 분석에도 kernel PCA 나 locally linear PCA 외에 여러 가지 방법이 있는데, 어떤 것이 문서분류에 적합한지에 관한 연구가 더 필요하다.

감사의 글

본 연구는 정보통신부에서 시행한 대학 기초 연구 기술 지원 사업(c1-98-006800)에 의해 일부 지원되었음.

참고 문헌

[1] Yiming Yang and Xin Liu, "A re-examination of text categorization methods," in *Proceedings of the 22th Ann Int ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '99)*, 1999.

[2] Yiming Yang and Jan O. Pedersen, "A comparative study on feature selection in text categorization," in *Proceedings of the 14th Int Conference on Machine Learning (ICML'97)*, pp. 412-420, 1997.

[3] Yiming, Yang, "Noise reduction in a statistical approach to text categorization," in *Proceedings of the 18th Annual Int ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'95)*, pp. 256-263, 1995.

[4] Bernhard Schölkopf and Alexander Smola, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Computation*, vol. 10, no. 5, pp. 1299-1319, 1998.

[5] N. Kambhatla and Todd K. Leen, "Dimension reduction by local principal component analysis," *Neural Computation*, vol.9, no.7, pp. 1493-1516, 1997.

[6] Erkki Oja, "Principal components, minor components, and linear neural networks," *Neural Networks*, vol. 5, pp. 927-935, 1992.

[7] T. Hastie and W. Stuetzle, "Principal curves," *Journal of the American Statistical Association*, vol. 84, no. 406, pp. 502-516, 1989.

[8] Helge Ritter, Thomas Martinetz, and Klaus Schulter, *Neural Computation and Self-Organizing Maps, An Introduction*., Addison-Wesley, Reading, Massachusetts, 1992.

[9] Mark A. Kramer, "Nonlinear principal component analysis using autoassociative neural networks," *American Institute of Chemical Engineers (AIChE),* vol. 37, no. 2, pp. 233-243, 1991.

[10] B. Masand, G. Linoff, and D. Walts, "Classifying news stories using memory based reasoning," in *Proceedings of the 15th Annual Int ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'92)*, pp. 59-64, 1992.