Filomat 32:5 (2018), 1499–1506
https://doi.org/10.2298/FIL1805499Z

Published by Faculty of Sciences and Mathematics,
University of Niš, Serbia
Available at: http://www.pmf.ni.ac.rs/filomat

A method of dimensionality reduction by selection of components in principal component analysis for text classification

Yangwu Zhang[a,b], Guohe Li[a,c], Heng Zong[b]

[a] *College of Geophysics and Information Engineering, China University of Petroleum-Beijing, China*
[b] *Department of Science and Technology Teaching, China University of Political Science and Law, China*
[c] *Beijing Key Lab of Data Mining for Petroleum Data, China University of Petroleum-Beijing, China*

Abstract. Dimensionality reduction, including feature extraction and selection, is one of the key points for text classification. In this paper, we propose a mixed method of dimensionality reduction constructed by principal components analysis and the selection of components. Principal components analysis is a method of feature extraction. Not all of the components in principal component analysis contribute to classification, because PCA objective is not a form of discriminant analysis (see, e.g. Jolliffe, 2002). In this context, we present a function of components selection, which returns the useful components for classification by the indicators of the performances on the different subsets of the components. Compared to traditional methods of feature selection, SVM classifiers trained on selected components show improved classification performance and a reduction in computational overhead.

1. Introduction

In recent years, internet resources have seen rapid growth, and information available on the web is increasingly rich. Information media have diversified and grown in recent years, including in the formats of text, sound, animation, image, and video. Although multimedia information frequently appears on a variety of websites, text will always be a main component of webpages (see, e.g. CNNIC, 2017).

Machine learning, described by Minsky (1969) and Christopher (2007), is the field of artificial intelligence that solves the problems of effective learning by computer and information processing, which imitate human learning processes and experiences.Machine learning endows computers with abilities in automatic learning, rather than being indirectly programmed for a task. Alongside increases in training knowledge and experience, computer programs have seen automatic improvements in information processing capacity. Users are able to obtain insight and interpretation results that are closer to facts and objectives from learning models and data based experiences (see, e.g. Mitchell, 2014).

Under the machine learning model (see, e.g. Rumelhart, 1986; Quinlan, 1993; Vapnik, 1998; Hinton, 2015; Sudderth, 2015), marked text is input into the model for training to obtain a stable classifier. Then one category for unmarked text is determined on the trained machine learning model. Text auto-classification technology is a process of text auto-classification based on the features of class-unknown text under a

2010 *Mathematics Subject Classification*. Primary 68U15; Secondary 68T05.
Keywords. Principal components analysis; Dimensionality reduction; Text classification.
Received: 28 August 2017; Accepted: 30 January 2018
Communicated by Hari M. Srivastava
Supported by Science Foundation of China University of Petroleum-Beijing At Karamay under Grant No. RCYJ2016B-03-001
Email addresses: yangwuzh@cupl.edu.cn (Yangwu Zhang), lgh1022@sina.com (Guohe Li), hengz@cupl.edu.cn (Heng Zong)

given classification system. It has wide application prospects in the fields of natural language processing, information retrieval, mail classification, topic tracking, and digital libraries (see, e.g. Tan, 2006; Shang, 2007).

2. Text classification problems

Traditional manual text classification has great difficulties in scenarios where there is a large amount of text. Manual text classification requires a great deal of manpower in the form of field experts and knowledge engineers. This reliance on humans means that it is hard to guarantee the correctness and identity of rules. Automatic text classification would address these issues. The Vector Space Model (VSM) expresses text as a single vector in the space of high dimension feature words (see, e.g. Chen, 2016; Haddoud, 2016; Junejo, 2016). Each vector dimension represents the weight of the corresponding word that has been marked and sorted in the dictionary in the text, i.e. the weight of the feature word. Originally the weight of the feature word was expressed by term frequency (TF). Then term frequency and inverse document frequency were used to indicate the weight of the feature word (TF-IDF). The vector space model was built by transforming the problem of text classification into a calculation problem using the similarity of vectors in vector space to express semantic similarity (see, e.g. Salton, 1975, 1983; Smith, 2015), which is simple, intuitive, and easy to understand.

In text preprocessing, we find that text vector space is a sparse matrix of high dimensional features, which causes long processing times for classification and curse of dimensionality. Therefore, it seems important to reduce the feature dimension. There are currently two methods used to reduce the feature dimension: feature selection and feature extraction (see, e.g. Kilic, 2015). Feature selection selects features that have the best ability to distinguish categories based on certain rules to form a new subset in the original feature space. Feature extraction maps the original high-dimensional feature space to low-dimensional space. This change retains the information for distinguishing categories between the original feature space and the low-dimensional space.

3. Component selection in principal components analysis

3.1. Method of principal component analysis

In statistics, in order to solve problems objectively, various influential factors must be comprehensively analyzed. These factors form high-dimensionality vectors. These high-dimensionality vectors reflect certain information for studying data in different degrees. When there is a certain level of correlation between two space vectors, it can be assumed that the two vectors contain overlapping information. Principal component analysis (PCA) (see, e.g. Jolliffe, 2002) is a mathematical method of converting a high-dimensionality vector with correlation into a group of linear uncorrelated low-dimensionality vectors by orthogonal transformation (see, e.g. Jonathon, 2014; Srivastava, 2011).

Principal component analysis uses an orthogonal transformation to convert related original high dimension space into new low dimension space. The orthogonal transformation transforms the covariance matrix of the original random vector into a diagonal form matrix through algebra. x_i represents the $i-th$ sample in the dataset. n is the dimensionality of the feature space. Text samples preprocessed on the basis of the vector space model constitute a matrix X, whose entries of x_{ij} are the feature values of corresponding word j in the dictionary, i.e. the $j-th$ dimension of the sample of i. Therefore, x_i is denoted as:

$$x_i = (x_{i1}, x_{i2}, \cdots, x_{in}). \tag{1}$$

X needs to perform a matrix transformation for its centered version. The standard transformation should be performed on the data set. Therefore, the mean and the variance of the column vector of j are:

$$\bar{x_j} = \frac{\sum_{i=1}^{m} x_{ij}}{m}, \tag{2}$$

$$s_j{}^2 = \frac{\sum_{i=1}^{m} (x_{ij} - \bar{x_j})^2}{m-1}. \tag{3}$$

In which, m is the quantity of samples or row vectors in the matrix, $\bar{x_j}$ is the mean value of the $j-th$ dimension column vector, and ${s_j}^2$ is the variance of this vector. Finally, the standardized data set is:

$$Z_{ij} = \frac{x_{ij}-\bar{x_j}}{s_j}, i = 1, 2, \cdots, m; j = 1, 2, \cdots, n. \tag{4}$$

It should be noted that principal component analysis searches for the projected direction of the maximum deviation. When v is assumed to be the unit column vector of the maximum deviation, the objective function is:

$$argmax_v \frac{1}{2}\sum_{i=1}^{m} \frac{1}{m-1}(Z_i v)^2, \tag{5}$$

Which yields:

$$v^T v = 1. \tag{6}$$

The Lagrange multiplier is constructed from:

$$\mathcal{L}(v) = \frac{1}{2}\sum_{i=1}^{m} \frac{1}{m-1}(Z_i v)^2 + \lambda(1 - v^T v). \tag{7}$$

We can solve the partial derivative and obtain the equation of the covariance matrix:

$$|Cov - \lambda I| = \vec{0}, \tag{8}$$

where

$$Cov = \frac{1}{m-1}(Z^T \times Z). \tag{9}$$

From Eq. (8) and Eq. (9) we find that the maximum deviation direction corresponds to the eigenvector of the maximum eigenvalue of covariance matrix. In fact, the covariance matrix Cov is the matrix of $n \times n$, whose rank is assumed as K:

$$Rank(Cov) = K. \tag{10}$$

Thus, its quantity of eigenvectors is K, and eigenvalues fit:

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_K > 0, \tag{11}$$

$$V = [v_1\, v_2\ \cdots\ v_K]. \tag{12}$$

It is clear that the eigenvector of v_1 corresponds to the eigenvalue of λ_1 and the variance of v_1 is the maximum, followed by that of v_2, and so on. Therefore, v_1 is called the first principal component while v_2 is called the second principal component. In view of the optimum classification intervals, small sample intervals appear between the same class while large intervals emerge between the different classes. Therefore, the maximum deviation of the first principal component reflects that information loss is at the minimum (see, e.g. Louis, 1995; Nils, 2005; Liu, 2009). The first few Principal Components (PCs) of a set of components are derived vectors with optimal properties in terms of approximating the original Vector Space Model. Criteria for selecting components are often ill-defined, and may produce inappropriate subsets. Indications of the performance of different subsets of the components are adopted as described below. These criteria are used in a stepwise selection-type algorithms to choose good subsets (see, e.g. Jolliffe, 2002).

3.2. Method of principle components selection

Normally, there are tens of thousands of feature words in text classification. As mentioned above, the number of all principal components is K, which is far less than N of N-dimensions and also less than or

equal to m of the sum of all samples. Then, the calculation of the accumulative contribution rate of the top components can be performed using Eq. (11) and the accumulative contribution rate (ACR) is defined as:

$$ACR(p) = \frac{\sum_{i=1}^{p} \lambda_i}{\sum_{i=1}^{k} \lambda_i}, p = 1, 2, \cdots, K. \tag{13}$$

Since the monotonicity of the ACR function is increasing, the independent variable p should be assigned appropriate values to make sure the accumulative contribution rate reaches a higher level. Two factors determine which components are chosen. It should be guaranteed that the component is beneficial for classification, and the component should not introduce noise. When the accumulative contribution rate reaches a certain level, the classification interval information in the selected principal components maintains the most important information of the original feature space. Further, there might be an introduction of noise which misleads samples.

The method mentioned above consists of two steps. In the first step, from the original Vector Space Model, we divide the dataset into three parts: train set, validation set, and test set, and then we project N-dimensional original space onto K-dimensional extracted space, which follows as Eq. (14). Generally, it is assumed that the sample size of the train set is $K1$, the sample size of the validation set is $K2$, and the sample size of the test set is $K3$. Here, the sum of $K1$,$K2$,and $K3$ is equal to m. The dataset is defined as:

$$\begin{aligned}
Y_{trainset} &= X_{trainset}V = [X_{trainset}\, v_1 \;\; X_{trainset}\, v_2 \;\; \cdots X_{trainset}\, v_i \;\; \cdots X_{trainset}\, v_{K1NZ}], \\
Y_{validationset} &= X_{validationset}V = [X_{validationset}\, v_1 \;\; X_{validationset}\, v_2 \;\; \cdots X_{validationset}\, v_i \;\; \cdots X_{validationset}\, v_{K2NZ}], \\
Y_{testset} &= X_{testset}V = [X_{testset}\, v_1 \;\; X_{testset}\, v_2 \;\; \cdots X_{testset}\, v_i \;\; \cdots X_{testset}\, v_{K3NZ}].
\end{aligned} \tag{14}$$

$X_{trainset}$ means that the dataset is the original word vector space model for training. $Y_{trainset}$ denotes that the dataset is a mapping space by PCs(principal components selection), and used for training. $X_{validationset}$ means that the dataset is the original word vector space model for validating. $Y_{validationset}$ denotes that the dataset is a mapping space by PCs for validating. $X_{testset}$ means that the dataset is the original word vector space model for testing. $Y_{testset}$ denotes that the dataset is a mapping space by PCs for testing. v_{K1NZ} means that its eigenvalue is minimum but greater than zero in the train set, and v_{K2NZ} validation set, v_{K3NZ} test set. Then, the classifier is trained on the matrix of $Y_{trainset}$. Next, samples from the validation dataset are classified by the classifier. We obtain classification performance of the classifier on all of the components.

The second step depends on the evaluation of classification performance to determine whether or not to choose a specific component. The selection of the principal component adopts indices of classification performance to evaluate the contribution of the component for classification in top K dimensions overall. The components remain which can promote the classification performance of the classifier and the principal components that will disturb the classification are deleted. The result of selection of principal components is one subset of original dataset, i.e. the samples matrix of K-dimensional space. In fact, the corresponding column vectors are reduced for the negative principal components.

Usually the interpretation of principal components is more or less fuzzy, unlike the original vector interpretation, which is clear and exact. However, it is hard to seek an optimal solution in a large original feature space. In order to obtain the best results, we need to create a list of useful classification components. Enumeration and heuristic strategies are useful for this. The enumerable range lies in combinations of selecting components from all of the components. Taking into consideration that the corresponding eigenvalues of all of the components are decreasing, we sort all the components by their eigenvalues. The value of enumeration is one of the Y_{p_train} subset when p is assigned from one to K. Y_{p_train} and $Y_{p\_validation}$ are defined as:

$$\begin{aligned}
Y_{p_train} &= X_{trainset}V_p = [X_{trainset}\, v_1 \;\; X_{trainset}\, v_2 \;\; \cdots X_{trainset}\, v_i \;\; \cdots X_{trainset}\, v_p], \\
Y_{p_validation} &= X_{validationset}V_p = [X_{validationset}\, v_1 \;\; X_{validationset}\, v_2 \;\; \cdots X_{validationset}\, v_i \;\; \cdots X_{validationset}\, v_p].
\end{aligned} \tag{15}$$

Here, Y_{p_train} means that $K1$ samples of the train set are projected on the top p components. $Y_{p\_validation}$ means that $K2$ samples of the validation set are projected on the top p components. Y_{p_train} is the p-dimensional

subspace in K-dimensional total space of all the components when $p = 1, 2, \cdots, K$. Y_{p_train} is the union:

$$\begin{aligned}
Y_{1_train} &= [X_{trainset}\ v_1], \\
Y_{2_train} &= [X_{trainset}\ v_1 \ \ X_{trainset}\ v_2], \\
&\vdots \\
Y_{K_train} &= [X_{trainset}\ v_1 \ \ X_{trainset}\ v_2 \ \ \cdots X_{trainset}\ v_i \ \ \cdots X_{trainset}\ v_K], \\
Y_{1_validation} &= [X_{validationset}\ v_1], \\
Y_{2_validation} &= [X_{validationset}\ v_1 \ \ X_{validationset}\ v_2], \\
&\vdots \\
Y_{K_validation} &= [X_{validationset}\ v_1 \ \ X_{validationset}\ v_2 \ \ \cdots X_{validationset}\ v_i \ \ \cdots X_{validationset}\ v_K].
\end{aligned} \tag{16}$$

Here, Y_{1_train} means that $K1$ samples of the train set are projected on the top 1 component. Y_{2_train} means that $K1$ samples of the train set are projected on the top 2 components. Then, the classifier is trained on Y_{1_train}. We now classify $Y_{1\_validation}$. Therefore, we can obtain corresponding classification performance CP_{t1}, which means the classification performance on the model of the top component. CP_{tp} means classification of performance on the model of the top p components. The remaining can be deduced in the same way, accordingly, the list of $CP_{t1}, CP_{t2}, \cdots, CP_{tp}, \cdots, CP_{tK}$ is output from the model on different combinations of top components sorted by eigenvalues. $ACR(p)$ is calculated by Eq. (13) when $p = 1, 2, \cdots, K$. If we choose it as a horizontal coordinate, the performance CP_{tp} on the top p components is a longitudinal coordinate. If the trend of this curve is at its lowest at a point whose horizontal coordinate is equal to $ACR(p)$, then the $p - th$ component has disrupted the classification. The $p - th$ component should be deleted from the list of sorted components. Therefore, the function of selecting the components is constructed as:

$$f_{pcs}(l) = \{1, 2, \cdots, L\} - \sum_{i=2}^{L} \{i | CP_{i-1} - CP_i \geq threshold\}. \tag{17}$$

Here, $f_{pcs}(l)$ is the function whose return value is the list of the index of components by Eq. (17). In other words, $f_{pcs}(l)$ means that the $i - th$ component is removed from the top L components when the $i - th$ component satisfies $CP_{i-1} - CP_i \geq threshold$. CP_i refers to the classification performance on the model of top i components. U is the set of two-tuples whose elements are two-tuples, which is defined as :

$$U = \{(p, E(p)) | p \in 1, 2, \cdots, K\}. \tag{18}$$

$E(p)$ represents the list of components by selection on the top p components, and identifies the subset of components which best approximate the full top p set of components (see, e.g. Jolliffe, 2002).

4. Experiments

4.1. Text data set

The 20 Newsgroups collection is the standard data set for experiments in text applications of machine learning techniques, including text classification and text clustering. In order to make the experimental results universal and repeatable, the 20 Newsgroups data set is adopted in the experiments. The 20 Newsgroups data set contains 18,828 pieces of documents from 20 different news groups. The directory of the dataset includes one metadata set and three subdirectories, i.e. test, train, and raw, in which, the proportion of test data stands at 60% and that of train data is 40% (see, e.g. Jrennie, 2008). We selected 2,189 pieces of documents as the data set for experiments. The documents consist of a total of 14,257 words after text segmentation. According to the English dictionary in the natural language processing toolkit in python, there are 9,974 words remaining after the non-English words are removed by python.

4.2. Experimental results

Support Vector Machine (SVM) is chosen as the classifier for the experiments. SVM obtains the minimum structural risk to improve the ability of model learning and generalization by searching for the maximum margin hyperplane (see, e.g., Vapnik, 2006; Tsochantaridis, 2005; Joachims, 2006, 2009; Chapelle, 2012; Swaminathan, 2015;Tanveer, 2017). Performance evaluation of the classifier on the dataset includes the precision, recall, and F-measure using the confusion matrix given in Table 1.

Table 1: Confusion matrix

Confusion matrix		Predicted As	
		Positive	Negative
Actual result	True	TP	FN
	False	FP	TN

Here, precision is the portion of correctly classified positives (true positives) out of all instances classified as positive, reflecting the probability of success when one sample is predicted as positive. Recall (also called the true positive rate or sensitivity) is the percentage of true positives of all actual positives. Precision and Recall are individually defined as:

$$Precision = \frac{TP}{TP+FP}, \tag{19}$$

$$Recall = \frac{TP}{TP+FN}. \tag{20}$$

The traditional F-measure or balanced F-score ($F1$ score) is the harmonic mean of precision and recall, which gives the precision and the recall equal weight, such as:

$$F1 = \frac{2\times p\times r}{p+r}. \tag{21}$$

According to the text vector space model, samples from the 20 newsgroups dataset are mapped into a feature space using python, which is denoted as X to show the $m \times n$ sample data matrix in Eq.(1). X includes three parts: train set $X_{trainset}$, validation set $X_{validationset}$, and test set $X_{testset}$. If SVM is trained directly on $X_{trainset}$ and the validation set of $X_{validationset}$ is performed by SVM, the precision is 85%. Using PCA transformation from high-dimensional space to low-dimensional space, i.e. [coefftrainset, score, latent]=princomp($X_{trainset}$) in MATLAB, N-dimensional feature space is transformed to K-dimensional space in Eq.(12). Here, $coeff_{trainset}$ is the matrix of full components of $X_{trainset}$. A coeff matrix column is a component which is a unit column vector. All the columns in coeff matrix are ordered by corresponding eigenvalues. $coeff_{trainset}(:, 1 : p)$ means the subset of the top p column in $coeff_{trainset}$. We train the support vector machine classifier on $X_{trainset} \times coeff_{trainset}(:, 1 : p)$ when p ranges from 1 to K, where $K \geq min(K1NZ, K2NZ, K3NZ)$. We achieve K classifiers denoted by $Classifier_1, Classifier_2, ..., Classifier_p, ..., Classifier_K$ (see Eq.(14) and (15)). Then, the validation set of $X_{validationset} \times coeff_{trainset}(:, 1 : p)$ is performed by Classifierp when p ranges from 1 to K. The classification performance is shown in Table2.

Table 2: Relationship between classification performance and top components

Top	1	2	3	4	5	6	7	8	9	10	26	35	65	78	100	131	132
Precision	0.53	0.55	0.56	0.57	0.66	0.67	0.67	0.64	0.64	0.70	0.90	0.81	0.93	0.85	0.94	0.93	0.66
Recall	0.22	0.29	0.32	0.32	0.32	0.35	0.33	0.28	0.28	0.39	0.81	0.66	0.88	0.71	0.91	0.88	0.33
F1	0.31	0.38	0.41	0.41	0.43	0.46	0.44	0.39	0.39	0.50	0.85	0.73	0.90	0.77	0.92	0.90	0.45

According to Table 2, the precision curves are plotted for different top components, which are shown as Figure 1. X coordinates are all defined as the corresponding accumulative contribution rate of different top components (see Eq. (13)), while Y coordinates are defined as Precision. In Figure 1, when the number of principal components is the top 35, 78, 132 and the accumulative contribution rate reaches 0.1886, 0.3363, and 0.4762, the precision will reach a downtrend where $CP_{i-1} - CP_i \geq 0.08$. $X_{testset}$ is used for model assessment, and $X_{testset} \times coeff_{trainset}(:, 1 : p)$ are projected on the top p components in the $X_{trainset}$ matrix. When p is equal to 40, the 35th component is removed from $coeff_{trainset}(:, 1 : 40)$ in order to optimize model

selection, i. e. $f_{pcs}(40)$ by Eq. (17) or $coeff_{trainset}(:, 35) = []$ in MATLAB. In other words, when principal components selection is performed $Y_{testset}$ by PCs is equal to:

$$Y_{testset} \times f_{pcs}(40) = [X_{testset}v_1\ X_{testset}v_2\ \cdots\ X_{testset}v_{34}\ X_{testset}v_{36}\ \cdots\ X_{testset}v_{40}]. \tag{22}$$

When p is equal to 90, the 78th component is removed from the top 90, i.e. $f_{pcs}(90)$. When p is equal to 140, the 132th component is removed from top 140, i.e. $f_{pcs}(140)$.

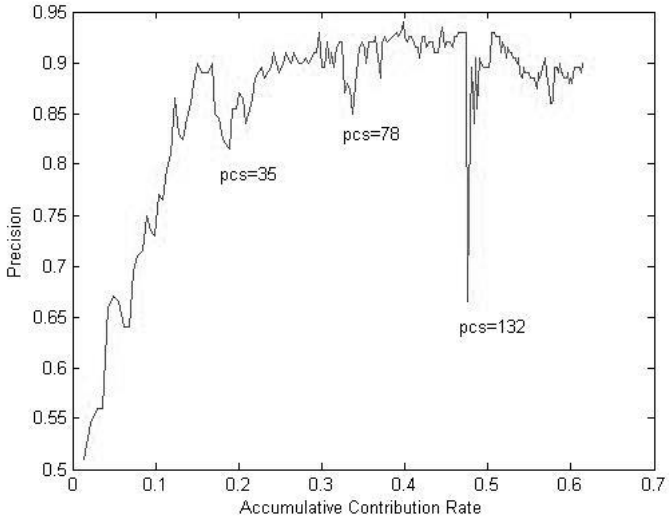

Figure 1: Curve of ACR-Precision

We retrain SVM on $X_{trainset} \times coeff_{pcs}(40)$, which refers to the top 40 column vectors from which the 35th column vector is removed in the matrix of $coeff$ by principal components selection. Then the test set of $X_{testset} \times coeff_{pcs}(40)$ is performed, $X_{trainset} \times coeff_{pcs}(90)$ versus $X_{testset} \times coeff_{pcs}(90)$, $X_{trainset} \times coeff_{pcs}(140)$ versus $X_{testset} \times coeff_{pcs}(140)$, and so on. The boosted results are shown on Table 3.

Table 3: Precision on top components by PCs

Top components	40	90	140
Precision	0.84	0.87	0.89
Precision by PCs	0.87	0.89	0.91

5. Conclusion

One of the main problems in the field of text classification has been dimensionality reduction. In this paper, we have presented a method for dimensionality reduction by component selection in principal component analysis. The function of component selection is constructed from the evaluation of classification performance on different sub datasets from combinations of certain components. We train the SVM classifier on selected components after negative components to the classification were removed. The experimental results show that the method of components selection can help improve the classification precision and reduce dimensionality from N-K dimensions to K-R dimensions.

6. Acknowledgement

We are greatly indebted to colleagues at Data and Knowledge Engineering Center, School of Information Technology and Electrical Engineering, the University of Queensland, Australia. We thank Prof. Xiaofang

Zhou, Prof. Xue Li, Dr. Shuo Shang and Dr. Kai Zheng for their special suggestions and many interesting discussions. This work is partly supported by Science Foundation of China University of Petroleum-Beijing At Karamay under Grant No. RCYJ2016B-03-001.

References

[1] Jolliffe I.T.. Principal Component Analysis, Second Edition. Spring-Verlag, 2002.
[2] CNNIC. China Statistical Report on Internet Development. http://www.cnnic.net.cn/hlwfzyj/hlwxzbg, 2017.
[3] Minsky M. L.; Papert S. A.. Perceptrons. MIT Press, 1969.0
[4] Christopher Bishop. Pattern Recognition and Machine Learning. Spring-Verlag, 2007.
[5] Mitchell T.. Never-Ending Language Learning. Proceedings of IEEE International Conference on Big Data, pp: 1-1, 2014.
[6] Rumelhart D.E.; Hinton Geoffery E.; Williams Ronald J.. Learning Representations by Back-Propagating Errors. Nature, 323(6088): 533-536, 1986.
[7] Quinlan J. R.. C4.5:Programs for Machine Learning. Morgan Kaufmann Publishers, 1993.
[8] Vapnik V.. Statistical Learning Theory. Wiley, 1998.
[9] Hinton G.;LeCun Y.; Bengio Y.. Deep learning. Nature, 521(7553): 436-444, 2015.
[10] Adams RP.; Fox EB.; Sudderth EB.; Guest Editors' Introduction to the Special Issue on Bayesian Nonparametrics. IEEE Transaction on Pattern Analysis and Machine Intelligence, 37(2): 209-211, 2015.
[11] Tan Songbo. Research on High-performance Text Categorization. Doctoral thesis of Chinese Academy of Sciences, 2006.
[12] Wenqian Shang. Research on Text Classification & Its Related Technology. Doctoral thesis of Beijing Jiaotong University, 2007.
[13] Chen KW.; Zhang ZP.; Long J.. Turning from TF-IDF to TF-IGM for term weighting in text classification. Expert Systems With Application, 18:245-260, 2016.
[14] Haddoud M.; Mokhtari A.; Lecroq T.. Combining supervised term-weighting metrics for SVM text classification with extended term representation. Knowledge and Information Systems, 49(3): 909-931, 2016.
[15] Junejo KN.; Karim A.; Hassan MT.. Terms-based discriminative information space for robust text classification. Information Sciences, 372: 518-538, 2016.
[16] Salton G.. A Vector Space Model for Automatic Indexing. Communication of ACM, 18: 623-620, 1975.
[17] Salton G.; McGill M.. Introduction to Modern Information Retrieval. McGraw-Hill, 1983.
[18] Smith DA.; McManis C.. Classification of text to subject using LDA. Proceedings of IEEE 9th International Conference on Semantic Computing (ICSC), pp: 131-135, 2015.
[19] Kilic E.; Ates N.; Karakaya A.. Two New Feature Extraction Methods for Text Classification: TESDF and SADF. Proceedings of 23nd Signal Processing and Communications Applications Conference, pp: 475-478, 2015.
[20] Jonathon Shlens. A Tutorial on Principal Component Analysis. EprintArxiv, 51(3): 219-226, 2014.
[21] Srivastava H. M.. Some Generalizations and Basic (or q-) Extensions of the Bernoulli, Euler and Genocchi Polynomials. Applied Mathematics & Information Sciences, 5(3):390-444, 2011.
[22] Louis Ferre. Selection of components in principal component analysis: A comparison of methods. Computational Statistics & Data Analysis, 19: 669-682, 1995.
[23] Nils Lehmann. Principal components selection given extensively many variables. Statistics & Probability Letters, 74: 51-58, 2005.
[24] Haifeng Liu et al.. Mixed Method of Reducing Feature in Text Classification. Computer Engineering, 35(3): 194-196, 2009.
[25] Jolliffe I.T.. Choosing a Subset of Principal Components or Variables. Principal Component Analysis, pp: 92-104, Springer-Verlag, 2002
[26] Jrennie. 20 Newsgroups. http://people.csail.mit.edu/jrennie/20Newsgroups, 2008.
[27] Vapnik V.. Learning hidden information: SVM+. Proceedings of 2006 IEEE International Conference on Granular Computing, pp:22-22, 2006.
[28] Tsochantaridis I.; Joachims T.; Hofmann T.; Altun Y.. Large margin methods for structured and interdependent output variables. Journal of Machine Learning Research, 6: 1453-1484, 2005.
[29] Joachims T.. Structured output prediction with Support Vector Machines. Lecture Notes in Computer Science. 4190: 1-7, 2006.
[30] Joachims T.; Finley Thomas; Yu CNJ.. Cutting-plane training of structural SVMs. Machine Learning, 77(1): 27-59, 2009.
[31] Chapelle O.; Joachims T.; Radlinski F.; Yue YS.. Large-Scale Validation and Analysis of Interleaved Search Evaluation. ACM Transactions on Information Systems, 30(1): 6-9, 2012.
[32] Swaminathan A.; Joachims T.. Counterfactual Risk Minimization. Proceedings of the 24th International Conference on World Wide Web, pp:939-941, 2015.
[33] Tanveer M.;Shubham K. Smooth Twin Support Vector Machines via Unconstrained Convex Minimization. FILOMAT,31(8):2195-2210,2017.