Principal Component Analysis for Semantic Classification

Benjamin Liu, Kelsey Maass, Riley Molloy; Department of Applied Mathematics, University of Washington

Introduction

Principal component analysis (PCA) and classification via supervised learning are two popular topics in data science today. In our project, we combine techniques from both areas in order to classify news articles based on their word frequency content. We find that we can accurately classify the data by projecting onto a small subset of principal components, reducing the feature space from nearly 10,000 elements to only 4. We also compare results from the traditional and robust PCA formulations, and discuss what additional semantic information can be inferred from our results.

Data

Our dataset consists of 2225 BBC articles labeled by topic, including business (510), entertainment (386), politics (417), sports (511), and technology (401). The data, collected for a paper presented at the 2006 International Conference on Machine Learning [1], consists of integer word counts for 9635 distinct words subject to the following pre-processing steps:

1. Stemming to identify like-words (e.g. fish, fisher, fishing, fishes)
2. Removal of stop words (common words such as: a, about, am, any, are, etc.)
3. Low-count (less than 3) removal

To perform supervised learning, we first convert the word counts into word frequencies by dividing by the total number of words for each article. Next we separate the data into a training set (80%) and a cross validation set (20%).

Principal Component Analysis

Principal component analysis is a powerful tool for isolating low-dimensional structure embedded in high-dimensional data. The goal of PCA is to find the best linear transformation of the data which maximizes the variance in the first principal component direction and in each subsequent orthogonal directions. To perform PCA on our dataset, we compute the singular value decomposition of our data matrix,

$$A = U\,\Sigma V^T.$$

In Figure 1 we plot the first five column vectors of V, where we note that there is a clear distinction in how each of the article classes express the first four principal components. Since projecting the data onto the first four principal components separates the classes, we can run our classification algorithms on this 4-dimensional principal component space rather than on the 9635-dimensional word space.

Robust PCA is an extension of PCA that first begins by separating the matrix into two parts

$$A = L + S,$$

where L is low rank and S is sparse. The goal of robust PCA is to remove important but uncommon elements of the data, the outliers, that artificially increase its dimensionality beyond that of its common underlying structure. In Figure 2, we compare the energy percentages of each of the first ten singular values for both PCA and robust PCA, where we note that in the robust PCA results the first four principal components increase in significance.

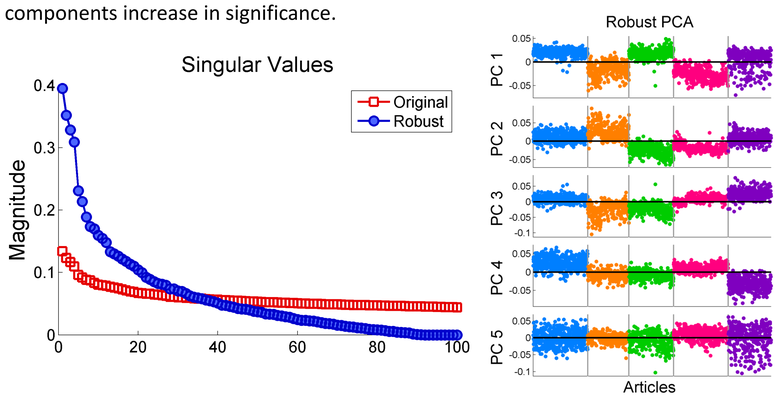

Figure 1. Principal component analysis of the article data. **Left**. Comparison of first 100 normalized singular values derived from the data with and without robust PCA. **Right.** Contribution of principal components to word frequencies for articles in each class.

Article Classification

To classify articles in our cross validation set, we first project both datasets onto a subset of the training data's principal components. Next we compare our classification accuracy as we increase the number of principal components used for the following classification methods:

- Nearest Mean: For the nearest mean method, we first project the training data onto the first four principal components and calculate the average projection for each class. To classify new articles, we assign the label of the closest mean projection.
- *K*-Nearest Neighbors: The *K*-nearest neighbors algorithm classifies new articles by finding their *K* nearest neighbors and assigning the label shared by the most neighbors. For our article classification, we let $K = 5$.
- Logistic Regression: Logistic regression classifies articles by first drawing linear decision boundaries between the classes (one-vs-all) that maximize the log-likelihood of the probabilities that the training articles belong to their respective classes. The final label is chosen from the class with the highest probability.
- Neural Network: To classify with neural networks, we first build a network with three layers: input, hidden, output. We then train the network to minimize a cost function on the training data and assign classes to new articles by choosing the class with the highest output value.

In the four classification methods we tested, we found that the classification accuracy did not significantly increase when we added more than four principal components.

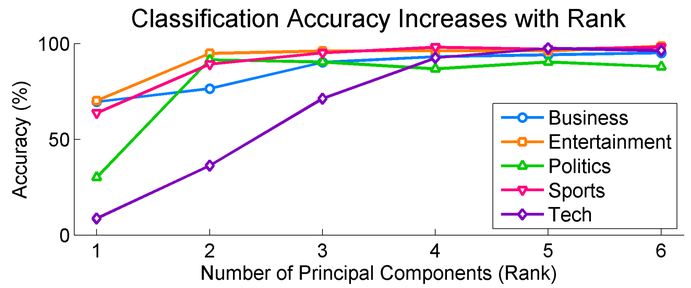

Figure 2. Comparison of classification accuracy for each article class as we vary the number of principal components (rank) used.

	Rank 1		Rank 2		Rank 3		Rank 4		Rank 5	
Nearest Mean	59%	48%	77%	78%	82%	89%	96%	95%	96%	96%
KNN (K=5)	56%	52%	77%	77%	86%	92%	96%	96%	97%	96%
Logistic	45%	45%	75%	74%	78%	77%	94%	93%	94%	93%
Neural Net	46%	45%	75%	74%	79%	79%	95%	93%	95%	93%

Table 1. Comparison of classification accuracy for four classification algorithms on our PCA space (white) and robust PCA spaces (gray).

Misclassification

While projecting onto the principal components of the training set does a good job at separating the article by class, it does not separate them perfectly. Therefore articles near the decision boundaries may be misclassified.

For example, Figure 3 shows how articles from each class are classified by the K-nearest neighbor algorithm as we vary the number of principal components used. It is interesting to note that articles from business and politics are likely to be mistakenly cross-classified. Furthermore, technology articles are prone to misclassification if the fourth principal component, which contains many technology-related words, is ignored (see Semantic Analysis).

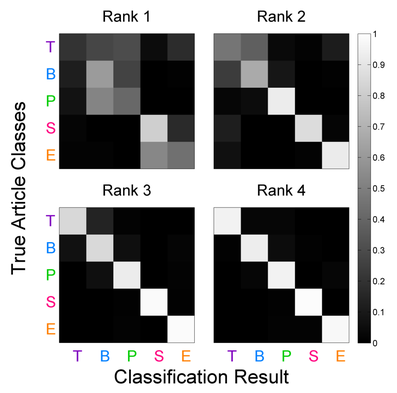

Figure 3. Classification of each article by type. Diagonal elements indicate percent of articles classified correctly; off-diagonal elements correspond to misclassified articles.

Semantic Analysis

While data methods are powerful tools for discovering trends and structure of large in large datasets, inferring semantic meaning from these models can be difficult. For example, principal component analysis determines the best basis to represent our article data in terms of word frequency values and variance, with no regard to semantic meaning. Nevertheless, our article classification analysis uncovers some trends with meaningful interpretation in terms of the thematic content of the articles.

Below, we list the 25 words with the highest weights in each of the first four principal components. Since each principal component is a linear combination of words, it is difficult to interpret their meaning. However, we note that the fourth principal component has many technology-related words, which could explain why the inclusion of the fourth principal component is necessary to correctly classify technology articles.

PC 1 game, play, win, player, firm, England, company, market, govern, year, against, team, growth, share, side, best, sale, price, club, bank, economy, go, cup, people, coach

PC 2 film, award, year, best, music, party, labour, star, elect, govern, Blair, include, sale, ministry, Tori, top, show, actor, nomination, told, number, album, 2004, people, band

PC 3 film, game, party, sale, labour, year, best, award, elect, market, firm, Blair, people, company, growth, price, share, Tori, profit, bank, music, ministry, England, star, player

PC 4 people, game, phone, user, mobile, year, tech, music, service, software, Microsoft, computer, firm, best, site, program, govern, search, labour, economy, elect, digital, net, online, Blair

Robust PCA yields a sparse matrix containing elements which represent words important to a small number of articles. Our sparse matrix contained 252 non-zero elements, compared with 227,814, the total number of non-zero elements in the training dataset. We notice that these words correspond to subtopics within a given category (e.g.: last names of athletes, music terminology, "Yahoo"). Some of the most frequent of these words are listed below.

Business	Sports	Entertainment	Politics	Tech
call	roddick	song	wage	game
centre	nadal	music	minimum	ink
	dallaglio	urban	forsythe	yahoo
		best	increase	gadget

Conclusions

By performing principal component analysis we decreased the article feature space from 9635 dimensions to just four. By projecting articles into the principal component space, we achieved high classification accuracy with several classical classification algorithms. While the traditional PCA formulation performs slightly better than the robust PCA formulation in terms of classification accuracy, the sparse matrix generated from the robust PCA analysis provides additional thematic information for particular articles.

Future Work

An important issue for classification is how sensitive results are to the particular dataset used. In our case, we had a fairly large data set composed of a small number of categories. Future analyses should examine data sets with different categories or more granular subcategories. If distinctions between article classes become less clear, attempts to classify them may fail or behave unpredictably.

Our work focused on supervised learning algorithms, but the use of unsupervised learning algorithms to uncover unknown categories as well as to discover ways to discriminate between subcategories could greatly enhance the inferential power of these methods.

References

1. D. Greene and P. Cunningham, *Practical Solutions to the Problem of Diagonal Dominance in Kernel Document Clustering*, Proc. ICML, 2006.
2. Y. Ma, E. J. Candes, X. Li, and J. Wright. *Robust Principal Component Analysis?* Technical report, Stanford University , Stanford CA, 2009.
3. J. N. Kutz. *Data-Driven Modeling & Scientific Computation: Methods for Complex Systems and Big Data*. Oxford University Press, 1st edition, 20013.
4. N. Boyd, S. Boyd, *Proximal Algorithms,* Foundations and Trends in Optimization, Vol. 1, No.3, 2013.