MOBILE PRICE RANGE CLASSIFICATION USING PRINCIPAL COMPONENT ANALYSIS AND SUPPORT VECTOR MACHINES

CMSC 191 - MACHINE LEARNING

Harold R. Mansilla
Department of Physical Sciences and Mathematics
College of Arts and Sciences
University of the Philippines Manila
hrmansilla@up.edu.ph

Virgilio M. Mendoza III
Department of Physical Sciences and Mathematics
College of Arts and Sciences
University of the Philippines Manila
vmmendoza1@up.edu.ph

April 20, 2019

1 Introduction

1.1 Mobile Price Range Classification

This problem is discussed in [1]

> Bob has started his own mobile company. He wants to give tough fight to big companies like Apple,Samsung etc.
>
> He does not know how to estimate price of mobiles his company creates. In this competitive mobile phone market you cannot simply assume things. To solve this problem he collects sales data of mobile phones of various companies.
>
> Bob wants to find out some relation between features of a mobile phone(eg:- RAM,Internal Memory etc) and its selling price. But he is not so good at Machine Learning. So he needs your help to solve this problem.
>
> In this problem you do not have to predict actual price but a price range indicating how high the price is.

1.2 Principal Comoponent Analysis

Principal Component Analysis, PCA, is a simple, nonparametric method for extracting relevant information from confusing data sets; that is, it is a dimension-reduction tool that can be used to reduce a large set of variables to a smaller set of components that contains most of the information. It is also capable of quantifying the importance of each component by using the measurement of variance along each component.[2]

1.3 Support Vector Machines

The support vector algorithm finds a hyperplane in an n-dimensional space, where n is the number of features, that directly classifies the data points [**?**]. Data points that fall on either side of the hyperplane can be said to belong to different classes. Data points that are close to the hyperplane that can influence its position and orientation are called support vectors. Using these points, a hyperplane who separates the classes with the maximum margin between classes is formed.

2 The Dataset

The dataset was retrieved from [1]. It has 2 `.csv` files for training and testing. For this paper, onyl the training file was used since it has the class labels needed for prediction and for the purposes of evaluation. It has 2000 entries and 21 original features (Table **??**)

Feature	Type (Categorical, Numerical)
id	–
battery_power	Numerical
blue	Categorical
clock_speed	Numerical
dual_sim	Categorical
fc	Numerical
four_g	Categorical
int_memory	Numerical
m_dep	Numerical
n_cores	Numerical
pc	Numerical
px_height	Numerical
px_width	Numerical
ram	Numerical
sc_h	Numerical
sc_w	Numerical
talk_time	Numerical
three_g	Categorical
touch_screen	Categorical
wifi	Categorical

The target values is `price_range` column wherein the possible values range from 0 to 3.

3 Preprocessing

The column `id` was already absent from the training file. Feature scaling was applied on the numerical columns. To achieve this, the `sklearn` class `StandardScaler` was used.

4 Dimension Reduction

To implement dimension reduction on the data set, the PCA of `sklearn` was used [3]. An important parameter for PCA is `n_components` which will determine the number of principal components to be used by PCA. As mentioned earlier, PCA is a tool that allows us to quantify the importance of each component by measuring its variance. By using the `explained_variance_ratio` in PCA, it can be seen which components encompass majority of the variance of the data set. For purposes of experimentation with the number of components, it has been set to a range between 1 to 20 (the number of features).

5 SVM Classifier Construction

`scikit-learn` provides an implementation of the SVM algorithm with `sklearn.svc`. The linear kernel was used for uniformity as well as setting `gamma=0.001`. For the metrics to be explored on, the accuracy, precision, and f1-score were used.

The dataset was split in 70:30 fashion.

6 Results

Figure 1 shows the effect of increasing `n_components` with respect to the explained variance after conducting PCA. It is shown to be increasing as the number of components increase.

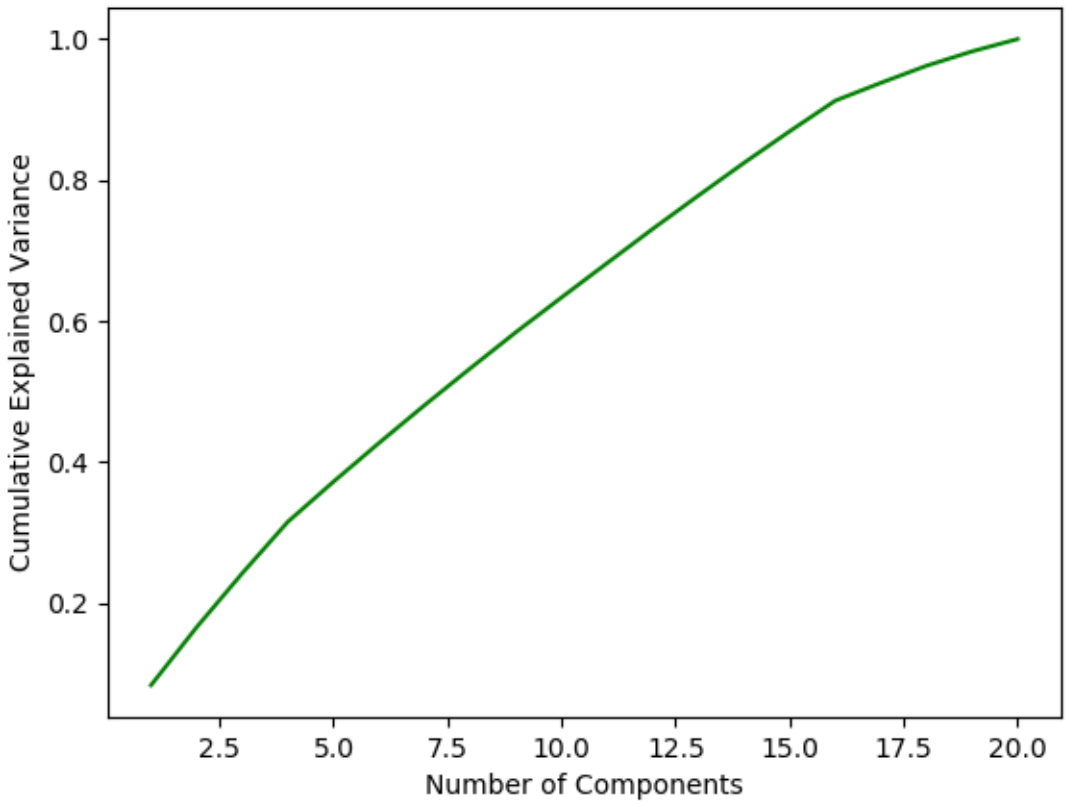

Figure 1: Plot of Explained variance against the number of principal components

Figure 2 shows the accuracies of the SVM classifier against the number of components in the preceding PCA. The highest accuracy recorded was at 96.3% at `n_components = 17` and `n_components = 19`. The respective precision and f1-scores were the same with that of the respective accuracies.

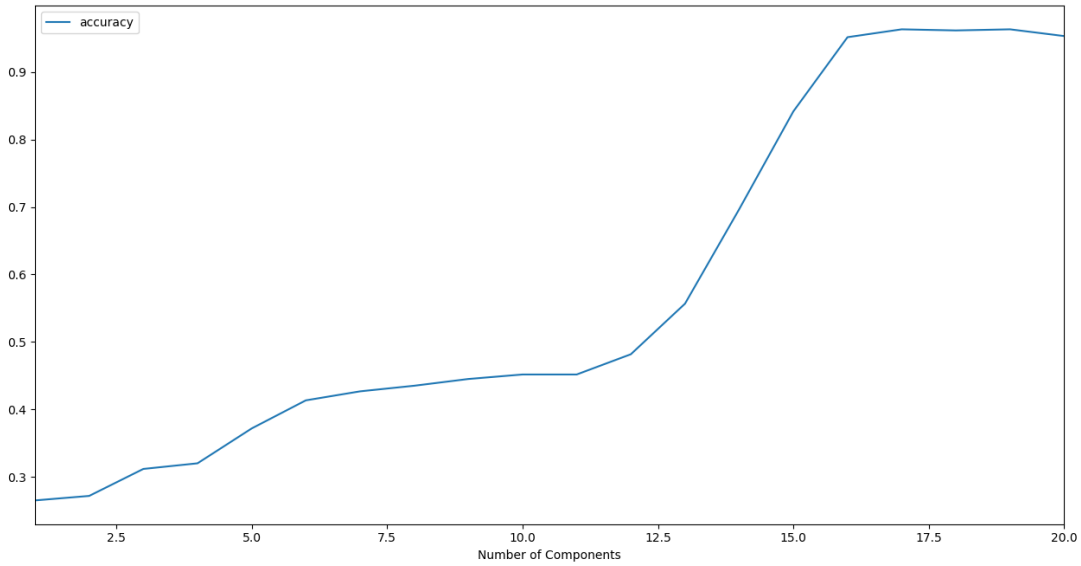

Figure 2: Plot of Accuracy against the number of principal components

7 Conclusion

The effect of the number of components in PCA to classifier performance in the SVM model was experimented on. For this dataset, 17 or 19 components proved optimal to the performance of the SVM classifier with its own cumulative explained variance of 96% and 98%, respectively.

References

[1] A. Sharma, "Mobile price classification."

[2] J. Shlens, "A tutorial on principal component analysis," *arXiv preprint arXiv:1404.1100*, 2014.

[3] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.