Dimension Reduction – Linear and Ridge Regression

Sofia Nahir Sapienza – Master OSS OS14 Final Assignment

***Abstract*— The purpose of this assignment was to implement dimension reduction techniques, such as PCA and Kernel PCA, in order to perform a regression using Linear Regression and Ridge Regression on the different latent spaces obtained from reduction. Standard Scaler and Min Max Scaler were used to find the best combination of techniques to deliver the best results.**

I. INTRODUCTION

The dataset used for this assignment was California housing from scikit-learn. The target variable, which predicting its value was the main goal of this work, is the median house value for California districts, expressed in hundreds of thousands of dollars ($100,000).

This dataset was derived from the 1990 U.S. census, using one row per census block group. A block group is the smallest geographical unit for which the U.S. Census Bureau publishes sample data (a block group typically has a population of 600 to 3,000 people).

A household is a group of people residing within a home. Since the average number of rooms and bedrooms in this dataset are provided per household, these columns may take surprisingly large values for block groups with few households and many empty houses, such as vacation resorts. In this dataset, we have 8 variables, that describes information regarding the demography (income, population, house occupancy) in the districts, the location of the districts (latitude, longitude), and general information regarding the house in the districts (number of rooms, number of bedrooms, age of the house). Since these statistics are at the granularity of the district, they correspond to averages or medians.

II. METHODS

A. Train Test Split

The first step was making Train-test-split using train_test_split from python library Sklearn, Model Selection. 500 samples were used for train, and the other 20.140 were let for test.

B. Scaling Data

Two methods were used for this step, StandardScaler and MinMaxScaler, both from Sklearn Preprocessing, and both applied on train and test data. This step was necessary in order to normalize data, so that each feature contributes approximately proportionately to the final distance, specially knowing that unit measurements of each feature are different. Figure 1 shows the train set before scaling, where it can be seen that feature number 4 has a big range, and Figures 2 shows the same data after MinMaxScaler.

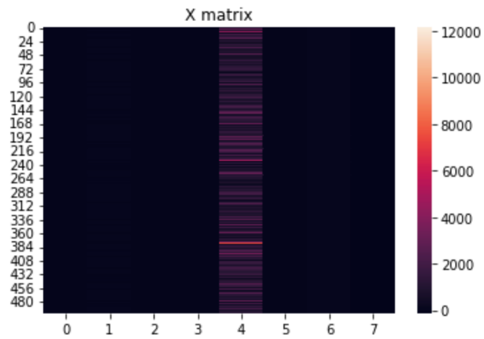

Fig.1 *Train Set before scaling*

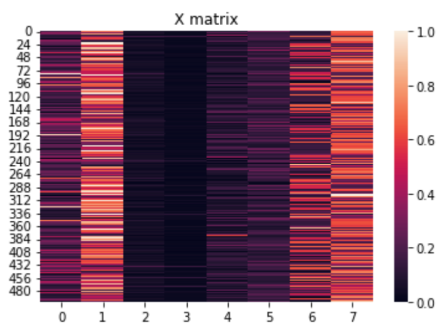

Fig.2 *Train Set after MinMaxScaler*

C. Dimension Reduction

Three techniques were performed: Principal Component Analysis (PCA) [1], a gaussian Kernel PCA, and linear Kernel PCA.

1) Principal Component Analysis (PCA)

The co-variance matrix was obtained using the function cov from python library Numpy, and eigen-decomposition was performed using function LA from Numpy. After choosing the desired dimension of the latent space, in this case, 2, 4 and 6, Beta matrix was bult (one for each desired dimension). Finally, the original dataset was transformed using beta by performing an np.dot, obtaining Z subspace.

2) Kernel PCA

The Kernel PCA [2] differs from PCA by finding a non-linear projection of the samples, and instead of using the co-variance matrix, the gram Matrix is used. This matrix contains on each position the pairwise similarity measurements done by the RKHS function between each pair of samples.

The eigen-decomposition was then applied in this matrix, and a set of eigenvalues and eigenvectors were obtained. Then, by selecting the dimensions (2,4 and 6) the data was projected using the top eigenvectors associated to the top eigenvalues. This was performed using a Gaussian Kernel and a Linear one.

D. *Regression Training*

Linear and Rigde regression where trained.

1) *Linear Regression*

Using cross validation technique with cross_val_score, and LinearRegression from Sklearn, a linear regression was trained using scaled data, both for MixMaxScaler and StandarScaler normalized train datasets. Each procedure was repeated for every dimension 2,4,6 both for linear KPCA and PCA latent spaces.

2) *Ridge Regression*

By using GridSearchCV [3] from Sklearn, 7 values of Alpha were tried in order to find the best parameter, which gave the best score in train, and those values where 0.001, 0.01, 0.1, 1, 10, 100 and 1000. Ridge regression was trained both for MixMaxScaler and StandarScaler normalized train datasets and repeated for every dimension 2,4,6 both for linear KPCA and PCA latent spaces.

E. *Regression Test*

In order to test models trained with PCA and KPCA latent space, reducing test data was necessary. PCA on test was performed by multiplying the Beta matrix built when reducing train data by the test data, and Linear KPCA with Sklearn KernelPCA function. Using the models trained in the previous step, target values were predicted and then compared with real values. The metrics used were mean absolute error (MAE), mean squared error (MSE) and root mean squared error (RMSE), all calculated by the functions available in Sklearn metrics.

Additionally, and with the objective of revealing if reducing the dimension was or not worth it to improve the prediction, linear and ridge models were performed on the original space (8 features) and then tested.

III. EXPERIMENTS

A. *Visualization of reduced subspace*

Figure 3 shows reduced subspace in 2 dimensions, where it can be noticed that data follows a linear relationship.

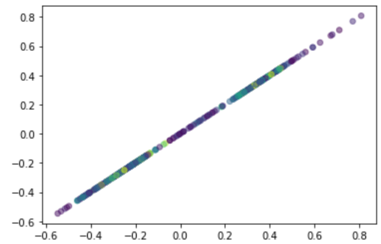

Fig.3 *PCA Visualization*

B. *Training and Test Results*

Table 1 shows R2 metric in train and MAE, MSE and RMSE for data scaled using Min Max Scaler and for each combination of model (Linear or Ridge), technique for reduction (PCA, KPA and no reduction) and dimension of subspace if reduced.

TABLE 1

Reduced	Model	Dimension	*Min Max Scaler*			
			R^2	*MAE*	*MSE*	*RMSE*
Linear Kernel PCA	Linear	2	-0,008	0,92	1,33	1,15
		4	0,566	1,08	1,87	1,37
		6	0,578	1,14	4,88	2,21
	Ridge	2	-0,006	0,92	1,33	1,15
		4	0,566	1,08	1,86	1,36
		6	0,59	1,13	4,27	2,07
PCA	Linear	2	-0,008	0,91	1,32	1,15
		4	0,58	0,57	0,63	0,79
		6	0,619	0,55	0,72	0,85
	Ridge	2	-0,005	0,91	1,33	1,15
		4	0,581	0,57	0,62	0,79
		6	0,62	0,55	0,67	0,82
Original Space	Linear	8	0,664	0,54	6,57	2,56
	Ridge	8	0,665	0,54	6,81	2,61

Table 2 shows the same results as table 1 but for standard scaled data.

TABLE 2

Reduced	Model	Dimension	*Standard Scaler*			
			R^2	*MAE*	*MSE*	*RMSE*
Linear Kernel PCA	Linear	2	-0,009	0,92	1,33	1,15
		4	0,400	0,93	1,37	1,17
		6	0,575	0,93	1,36	1,17
	Ridge	2	-0,008	0,92	1,33	1,15
		4	0,400	0,93	1,37	1,17
		6	0,575	0,93	1,36	1,17
PCA	Linear	2	-0,009	0,9	1,31	1,15
		4	0,400	0,7	0,95	0,98
		6	0,575	0,61	0,71	0,85
	Ridge	2	0,022	0,91	1,31	1,14
		4	0,402	0,7	0,95	0,97
		6	0,576	0,61	0,71	0,84
Original Space	Linear	8	0,664	0,54	0,59	0,77
	Ridge	8	0,665	0,54	0,59	0,77

IV. Discussion

MAE sums the absolute values of the difference between actual value and predicted one and gets their mean. It gives a notion of the overall error for each prediction of the model, the smaller (closer to 0) the better. MSE, squares the absolute values of the errors, making large errors even larger. Finally, RMSE gets the square root of MSE, to scale it back to the same units of the data. Taking all of this in account, best model appeared to be either both Rigde and Linear models with all 8 features using Standard Scaler for normalization, since they have the same metrics in test, and the closest to 0 values if compared with the other models.

Using Min Max Scaler, the best model was Ridge, with PCA reduced subspace dimension 6, with metrics a little bit higher but not very far from the best ones in Standard Scaler. It is interesting that, when using linear kernel PCA, adding more dimensions reduced the performance in test, whereas when using PCA, adding more dimensions improved it.

Finally, taking test metrics into account, best models were built using the original space with all the features. This could be explained by the fact that not always reducing dimensions would lead to better results.

V. Conclussion

It can be concluded that changing small parameters and techniques when reducing dimensions, training and testing models can make great difference among the results obtained.

Although some models showed better results than others, the techniques used performed well for the desired task and can be easily replicated for other regression studies.

References

[1] Bro, R., & Smilde, A. K. (2014). Principal component analysis. Analytical methods, 6(9), 2812-2831.

[2] Schölkopf, B., Smola, A., & Müller, K. R. (2005, June). Kernel principal component analysis. In Artificial Neural Networks—ICANN'97: 7th International Conference Lausanne, Switzerland, October 8–10, 1997 Proceeedings (pp. 583-588). Berlin, Heidelberg: Springer Berlin Heidelberg.

[3] Lerman, P. M. (1980). Fitting segmented regression models by grid search. Journal of the Royal Statistical Society: Series C (Applied Statistics), 29(1), 77-84.