Evaluation of the Performance of Different Covariance Matrix Estimators in the Context of CNN Image Classification

1st Luiz Fernando Medeiros
Department of Computer Science
Aalto University
Espoo, Finland
luiz.medeiros@aalto.fi

***Abstract*—The goal of this work was to evaluate the performance of three different covariance matrix estimators in the context of Principal Component Analysis (PCA), Convolutional Neural Networks (CNNs), and classification. Namely, the different estimators were used to compute PCA features from images in the Fashion-MNIST dataset. These features were then fed to a simple CNN for a classification exercise. The different estimators, Sample Covariance, Ledoit-Wolf Linear Shrinkage, and Rotationally Invariant estimators are evaluated on the basis of the final classification performance. Results showed that instead of using the predominant Sample Covariance estimator, it is better to use the novel Rotationally Invariant estimators.**

***Index Terms*—Covariance Matrix Estimation, Machine Learning, PCA**

I. Introduction

Principle Component Analysis (PCA) is a technique that is widely used for reducing the dimension of datasets. When it is not used on its own for data separation purposes, its paired with machine learning models for regression and/or classification. In order to get a feeling of how widespread this practice has been for the past decade, all which one must do is search "PCA Images CNN" in any search engine. Results range from smaller experiments and publications, to large, significant projects. One project which stands out is the "Labeled Faces in the Wild" project, by University of Massachusetts, where faces of people are used in a classification exercise.

Now, the success and applicability of PCA is highly dependent on a few factors: whether variance is a significant factor in separating data, the order of correlation between variables, and sample sizes (to name a few). Now, an important step in PCA, which reflects the importance of these factors is the computation (or estimation) of the covariance matrix. It follows that the quality of the covariance matrix estimation will directly impact all conclusions that follow (more specifically the eigenvalue decomposition is based on the computed covariance matrix).

Considering the widespread of the PCA method, and the current Deep Learning trend, this paper aims to evaluate how different covariance estimation methods tend to impact a classification exercise. More specifically, given a flattened image $\mathbf{x} \in \Re^N$ as input, how will the different covariance estimation methods used in the PCA procedure to reduce the dimension of this image affect the eventual classification results? In order to answer this question three different estimators are explored: the sample covariance estimator (SCE), Ledoit-Wolf Linear Shrinkage Estimator (LWE) [1], and the Rotationally Invariant Estimators (RIE) [2].

In order to get some insight into what can be expected, the latter two estimators, LWE and RIE, are considered state of the art methods, and generally accepted as better estimators than SCE. However, RIE is a complex method that only works for input data where the number of observations (or samples) is larger than the number of variables. This can be a problematic constraint when attempting to reduce the dimension of large images, whose sample numbers are small. On the other hand, LWE is a well accepted method, whose constraints are not restricted by sample size. This makes LWE the method which is expected to produce the best results under a broader set of conditions.

In order to maintain focus on the estimation methods, a simple Convolutional Neural Network (CNN) model will be used to classify the images. Moreover, the images used are from the well known Fashion-MNIST dataset [3]. With this setup, the aim is then to evaluate how the classification results will vary as a function of estimator and sampling size.

The structure of the text will be as follows: After this introduction, the background to the different estimators will be presented along with some of the successful related work that has been proposed in the realm of PCA, images, and CNNs. Following that, a description about the experiments and results will be presented. Finally, a conclusion with thoughts about the final results and future work will be posed.

II. Background and Related Work

In this section, we will present the basic theory behind the different covariance estimators being used, along with some of the work which has pioneered the use of PCA in feature extraction.

A. Sample Covariance Estimator

The definition of SCE [4] can be defined as follows:

$$\mathbf{S} = \frac{1}{T} \cdot \mathbf{X}^\top \mathbf{X} - \bar{\mathbf{r}}^\top \bar{\mathbf{r}} \tag{1}$$

where $\mathbf{X} \in \mathbb{R}^{T \times N}$ is the data matrix with T observations, and N variables (dimensions). In the case of this experiment, our N variables is equivalent to the effective size of the image. For example, if the image has 28×28 pixels $= 784$. Therefore, for this problem, $N = 784$, since we flatten the input images. Moreover, $\bar{\mathbf{r}} = \frac{1}{T} \sum_{t=1}^{T} \mathbf{r}_t$, where $\mathbf{r}_t \in \mathbf{X}$ is the row vector at row t of $\mathbf{X}$.

Now the general notion behind SCE is that as the number of samples that we have approaches ∞, our Sample Covariance Matrix (SCM) estimate $\mathbf{S}$ of the covariance matrix approaches the real covariance matrix $\mathbf{\Sigma}$ [4]. However, as our number of samples $T \rightarrow N$, we have a situation where our estimate becomes worse. Consequently, the eigenvalues and eigenvectors become distorted. This is where the state of the art methods such as RIEs attempt to optimize the estimate.

B. Ledoit-Wolf Linear Shrinkage Estimator

LWE builds on SCE, adn the idea to minimize the difference between a true covariance matrix $\mathbf{\Sigma}$ and an estimate $\mathbf{\Sigma}^\star$. More specifically, it aims at solving the following optimization problem:

$$\begin{aligned} &\underset{\zeta_1,\zeta_2}{\text{minimize}} \ \mathbb{E}\Big\{ \|\mathbf{\Sigma}^\star - \mathbf{\Sigma}\|_F^2 \Big\} \\ &\text{s.t. } \mathbf{\Sigma}^\star = \zeta_1 \mathbf{I} + \zeta_2 \mathbf{S} \end{aligned} \tag{2}$$

where $\|\cdot\|_F^2$ is the Frobenius norm [1] , $\mathbf{I}$ is the identity matrix, $\mathbf{S}$ is the sample covariance matrix (as defined in Equation 1), and ζ_1 and ζ_2 are constants. The aim is to find an optimal linear combination of the SCM and the identity matrix. Further details can be found in Lemma 2.1 of [1].

C. Rotationally Invariant Estimator

Now, in the case of RIE, a number of pre-conditions must be sufficed, in order to be able to use this method. The first, is that data matrix $\mathbf{X}$ must be normalized, which means that a specific data point in the new $\mathbf{X}_\eta$ will be defined as follows: $\mathbf{X}_\eta[j,i] = \frac{\mathbf{r}_{j,i} - \bar{\mathbf{r}}_i}{\sigma^2_{\mathbf{r}_i}}$. The second, is the fact that RIEs are directly targeted to correlation matrices. However, we can derive covariance matrices from correlation matrices. As such, Bun et al defined the sample based estimate for the correlation matrix as follows:

$$\begin{aligned} \mathbf{E} &:= \frac{1}{T} \mathbf{X}_\eta \mathbf{X}_\eta^\top \\ &:= \sum_{k=1}^{N} \lambda_k \mathbf{u}_k \mathbf{u}_k^* \in \mathbb{R}^{N \times N} \end{aligned} \tag{3}$$

where $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_N \geq 0$ are the eigenvalues of $\mathbf{E}$. Moreover, it follows that when the number of samples $T \rightarrow \infty$, $\mathbf{E}$ approaches the true sample correlation matrix $\mathbf{C}$ [5].

In the spirit of estimating the correlation matrix, we shift our efforts to the form where the eigenvalues and eigenvectors of this estimate are in evidence. Namely, we can infer that if we would somehow do a better job in the eigenvalues estimate, we should be able to produce an estimate $\mathbf{\Xi}$ of the correlation matrix that is close to the true correlation matrix $\mathbf{C}$. Namely:

$$\mathbf{\Xi}^{\mathbf{RIE}} := \sum_{\mathbf{k=1}}^{\mathbf{N}} \xi_{\mathbf{k}}^{\mathbf{RIE}} \mathbf{u_k} \mathbf{u_k^*} \tag{4}$$

The details of how this is achieved is thoroughly described in [2]. In short, one re-scales the eigenvalues closer to an established norm according to Marcenko & Pastur's [6] description of how small eigenvalues become smaller and large eigenvalues become larger when comparing the spectrum of a sample correlation matrix estimate $\mathbf{E}$ and the true correlation matrix $\mathbf{C}$.

D. PCA and Convolutional Neural Networks

Now we will tie in the background presented thus far, and its relevance, to PCA and Images. We will start by mentioning the seminal worked produced by Turk and Pentland on face recognition [7]. And then follow to consider a couple of papers which attempted different classification methods while using PCA to decompose the images.

In the work presented by Turk and Pentland [7], the authors aimed at producing a feature space which better encoded the variation in the images they were presented. In contrast to the images used in this work, the images used in [7] were face images, eventually used for face recognition. The general procedure was to flatten the images, and then compute the SCM of the images as presented in Equation 1. The next step was then to produce the eigenvalues and eigenvectors which were to be used in the classification exercise. These set of steps are generic and were also used for the papers [8] and [9], which will be discussed shortly.

Now, the authors in reference [8] attempted several classification methods, while employing the same initial steps mentioned above. Their main goal was to evaluate the how different distance metrics would affect the classification exercise on the FERET dataset (a dataset composes of labeled face images). This latter work is the one which more closely relates to which we are attempting to produce in this paper. An evaluation of different methods, and how they perform in the eventual classification exercise. The most significant take from the latter work is the specific statistical tests used, such as McNemar's Test.

Finally, an additional work presented by Ruiz et al [9] demonstrates another approach where PCA, is paired with other eigenspace decomposition approaches with similarity matching methods such as Self Organizing Maps (SOMs), Fuzzy Feature Contrast. This latter work aims at producing a comprehensive evaluation on how these combinations perform in the exercise of similarity matching.

As of the time of this writing, we are not aware of an empirical evaluation that explores the exact points in which this work reports. However, it is clear from the previously discussed references, that there is quality work which we are able to leverage in our work.

[1] The Frobenius norm is defined as: $\|\mathbf{A}\|_F = \sqrt{\text{tr}\Big(\mathbf{A}^\top \mathbf{A}\Big)/N}$, for $\mathbf{A} \in \mathbb{R}^{T \times N}$

III. Study Description

In this section we will discuss the data that was used in the research, along with the test setup. In our discussion regarding the test setup, we will explain in more depth which CNN model was used and other crucial variables used in evaluating the performance of the different estimators.

A. Data

In this experiment we used the Zalando Fashion-MNIST [3] dataset. The dataset contains images from articles that are used in Zalando's website. Each image was set to grayscale (meaning there is only one channel), and reduced to a resolution of 28x28 pixels. This means that when flattened, each input $\mathbf{x}$ is equivalent to an array with 784 features ($\mathbf{x} \in \mathbb{R}^{1\times N}$).

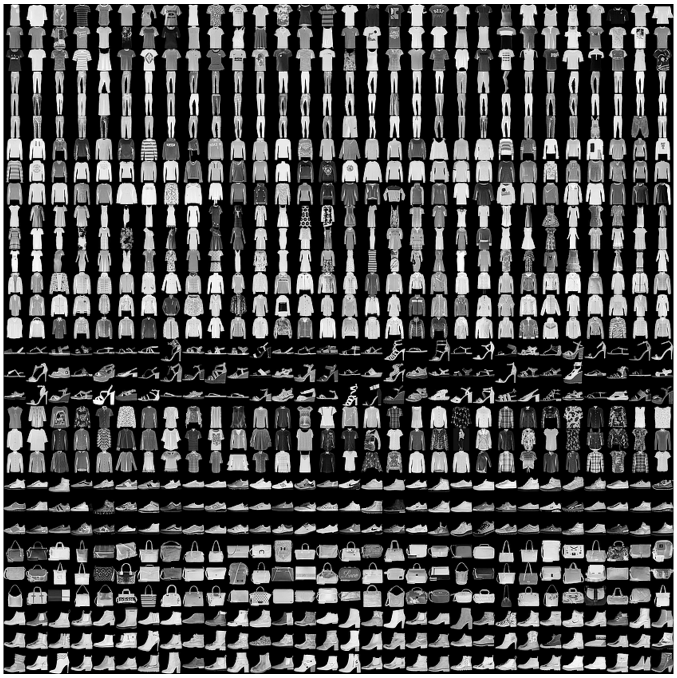

Fig. 1. Example of the data contained in the fashion-mnist dataset.

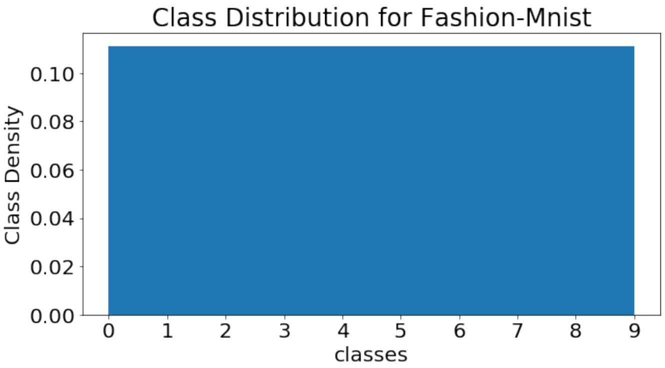

Fig. 2. Image demonstrating the evenly distributed classes in the MNIST dataset.

The dataset contains 60000 example images for training, and 10000 images for testing. In addition, the images are divided into 10 classes: t-shirt/top, trouser, pullover, dress, coat, sandal, shirt, sneaker, bag, and ankle boot. The classes are distributed evenly, as Figure 2 shows.

B. Test Setup

The test setup is divided in multiple parts: data processing, CNN model parameters, and classification performance. Each part is built in a pipeline format, where each component receive parameters, and produces output that is then used by the following component, until the final result is outputted. The final output is the classification report for the estimator used to produce the input data.

The first part, data processing, is composed of structuring the data in three different sets: training, validation, and test. The initial number of examples is 70000, as described in the previous section. From this number of we extract 25% of examples, or 17500 images and place in the test data matrix $\mathbf{X}_{test} \in \mathbb{R}^{17500\times 784}$. Next, from the remaining examples we create a validation and training set, in which 90% is training and 10% is validation. Namely: $\mathbf{X}_{training} \in \mathbb{R}^{47250\times 784}$ and $\mathbf{X}_{validation} \in \mathbb{R}^{5250\times 784}$. This ensures that we have distinct test examples to verify that our model is in fact learning, in addition distinct validation examples which ensure that we are considering each training epoch against distinct examples. Following this input data setup is the application of PCA with one of the estimators: SCE, LWE, or RIE. This means that we will compute transform the input data into the PCA domain, such that each $\mathbf{X}_{.}$ will be equivalent to the transformation of the respective PCA produced by an estimator of interest. Figure 3 provides a high level diagram of the different actions involved, where data formatting and PCA composed the data processing component of the testing.

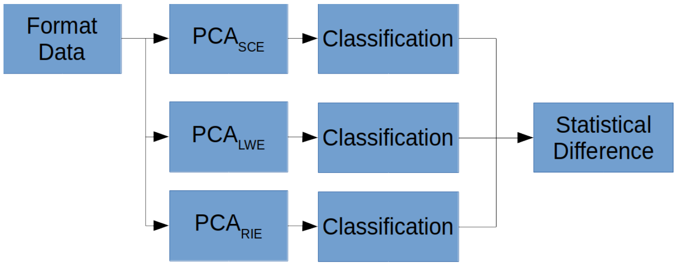

Fig. 3. High level view of the different actions that are performed during testing.

Now, the classification component is equivalent to CNN model parameters in the test setup. In a similar fashion to the data formatting, this component will have the same parameters for every statistical estimator used to compute PCA. This means that we use the same model, with its respective parameters. The model used is a Lenet-5 model, as defined in [10], with the difference that this is applied to 1-dimensional input datasets, and therefore contain 1-D convolutions. The number of training epochs are the same for every estimator. This component will output the classification results for the specific transformed dataset.

The classification output for each different estimator will be considered, where the higher metrics (precision, recall, F1-score, and accuracy) will indicate the best performer. In addition, the test prediction output from the different estimators will also be compared in a Pearson's Chi-squared test [11] to evaluate the statistical difference between the results. In case metric reports are very close, the latter will indicate whether or not there exists a difference between the different estimators in terms of predictive output. Finally, the goal is to perform the aforementioned steps for different numbers of observations and record results.

IV. Results

The results presented in the following images reflect statistics that were computed over 1500 tests. This means that each data point that was used in computing the mean f1-score [12] was extracted from a classification result. The f1-score used here is defined as follows:

$$F_1 = 2 \cdot \frac{P \cdot R}{P + R} \tag{5}$$

$$P = \frac{\text{true positives}}{\text{true positives + false positives}} \tag{6}$$

$$R = \frac{\text{true positives}}{\text{true positives + false negatives}} \tag{7}$$

whereby true positives are basically when the model was able to classify properly, whereas false positive is when the model classified as the class of interest, however it was not. Finally, false negatives are when the model classifies as the opposing class, and the result was incorrect.

These results were passed through the pipeline, where the test procedure described in subsection III-B was followed. The specific reason why f1-score was chosen as the performance metric was due to the fact that it produces a harmonic mean between precision and recall.

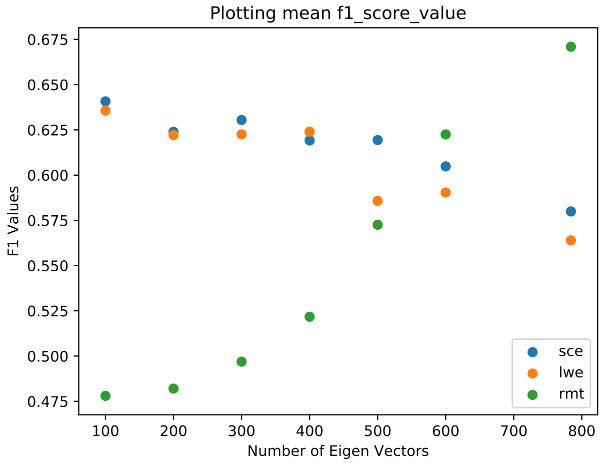

Fig. 4. Mean f1-score values achieved throughout all tests

Now, Figure 4 provides insight into how the different number of eigen vectors affect the performance of the different estimators. It should be noted that Figure 4 also reflects the results from using different number of training samples. The main trend that is possible to highlight is how RIE starts by producing the worst performance with small number of eigen vectors, and as we increase the dimension of the decorrelated input, we are able to see its performance increase.

In addition to insight in how the overall models performed, on average, it was also possible to investigate how these models performed with varying numbers of training examples. Figure 5 demonstrates that RIE with small number of training samples produces the lowest average performances, also bringing about the highest variance in results. However, as we increase the number of samples, these performances increase, on average, and also vary much less. In fact, page 5 indicates that with all the training samples, RIE (or rmt as indicated in the plot), produces the best performance, with the smallest variance.

Finally, Figure 6 shows a more granular view of the performances. It demonstrates the performance as we vary the number of training samples for every number of eigen vectors tested. As consequence, it's possible to note that if we maintain all the eigen vectors in the decorrelated data matrix, RIE again produces the best performances (even with small number of samples).

V. Conclusion

In the beginning of this work, the initial assumption was that LWE was going to be the best performing estimator. However, as the results in section IV demonstrates, RIE produced the best performances in scenarios where plenty of training data is available, as well as scenarios where it is possible to keep the original dimension of the input dataset. As a consequence, the general conclusion is that instead of using the widely used SCE, it is a better idea to use RIE, for the purposes and of producing higher performance metrics and minimizing the variance of the input data.

References

[1] O. Ledoit and M. Wolf, "A well-conditioned estimator for large-dimensional covariance matrices," *Journal of Multivariate Analysis*, vol. 88, no. 2, pp. 365–411, 2004.

[2] J. Bun and J. P. Bouchaud, "Cleaning correlation matrices," *Risk.net*, no. April, 2016.

[3] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms," 2017.

[4] B. Hajek, *Random Processes for Engineers*. Cambridge University Press, 2015.

[5] J. Bun, J. P. Bouchaud, and M. Potters, "Cleaning large correlation matrices: Tools from Random Matrix Theory," *Physics Reports*, vol. 666, pp. 1–109, 2017.

[6] V. A. Marčenko and L. A. Pastur, "Distribution of Eigenvalues for Some Sets of Random Matrices," *Mathematics of the USSR-Sbornik*, vol. 1, no. 4, pp. 457–483, 1967.

[7] M. A. Turk and A. P. Pentland, "Face recognition using eigenfaces," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 810, (Maui), pp. 586–591, IEEE, 1991.

[8] W. S. Yambor, B. A. Draper, and J. R. Beveridge, "Analyzing pca-based face recognition algorithms: Eigenvector selection and distance measures," in *Empirical evaluation methods in computer vision*, pp. 39–60, World Scientific, 2002.

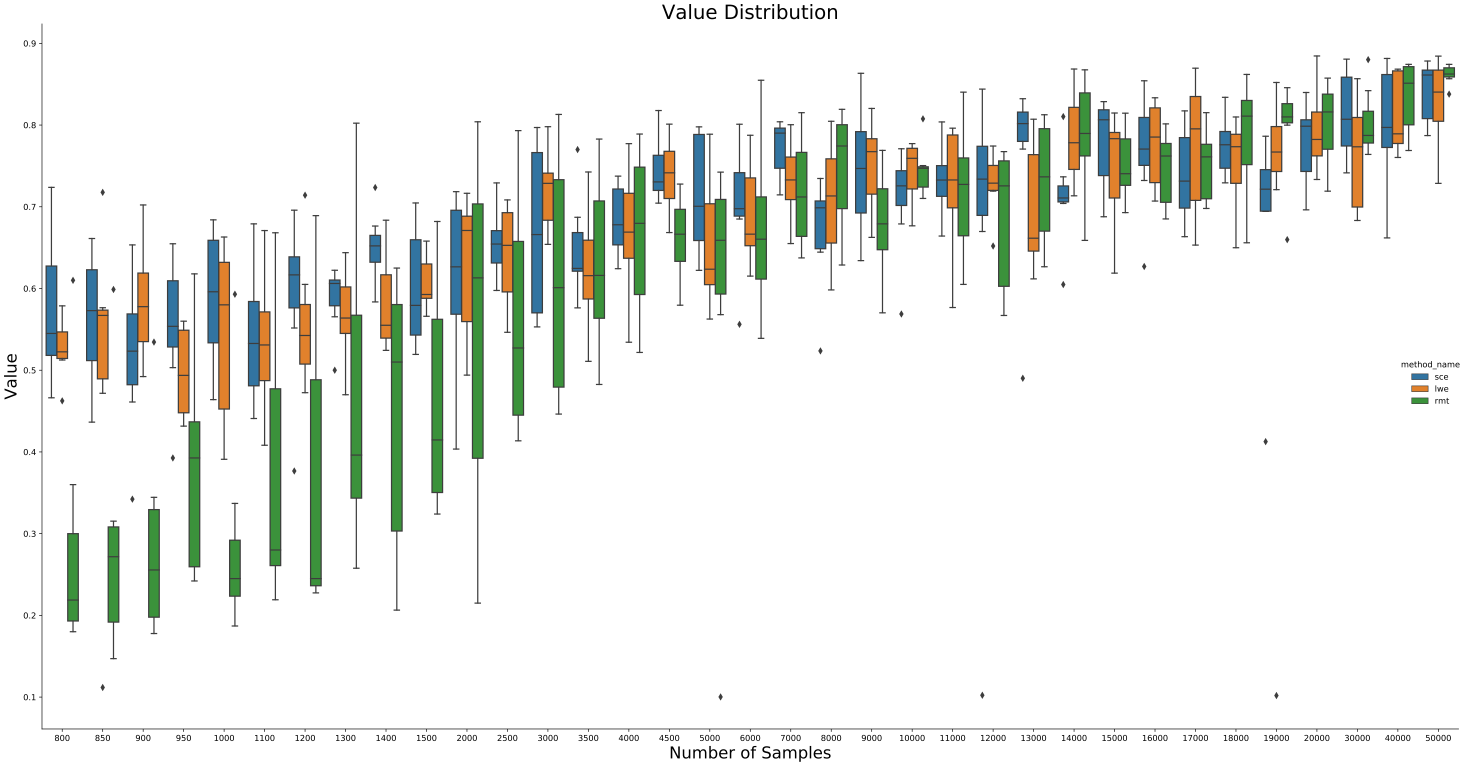

Fig. 5. Distribution of f1-score values for different number of training samples. The smallest number of training samples is 800, and largest is 50000.

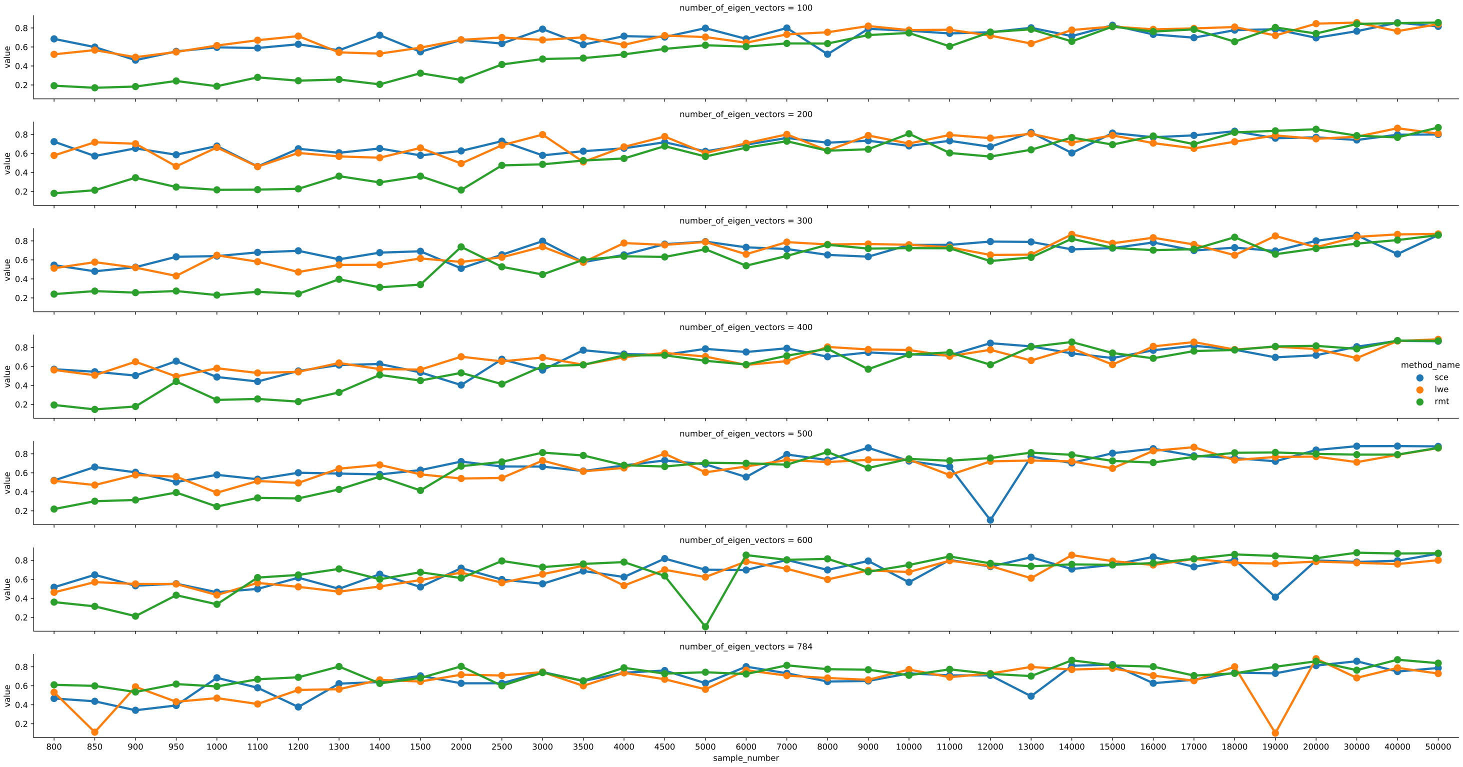

Fig. 6. F1-score performance divided on the basis of number of eigen vectors.

[9] J. Ruiz-del Solar and P. Navarrete, “Eigenspace-based face recognition: a comparative study of different approaches,” *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 35, no. 3, pp. 315–325, 2005.

[10] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, *et al.*, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[11] K. Pearson, “X. on the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling,” *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 50, no. 302, pp. 157–175, 1900.

[12] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, *et al.*, “Scikit-learn: Machine learning in python,” *Journal of machine learning research*, vol. 12, no. Oct, pp. 2825–2830, 2011.