`Robust Rolling PCA`: Managing Time Series and Multiple Dimensions*

Ali Hirsa[†], Federico Klinkert[‡], Satyan Malhotra[§], Ryan Holmes [¶]

March 25, 2023

Abstract

Principal Component Analysis (PCA) is an important methodology to reduce and extract meaningful signals from large data-sets. Financial markets introduce time and non stationarity aspects, where applying standard PCA methods may not give stable results. We propose `robust rolling PCA (R2-PCA)` that accommodates the additional aspects and mitigates commonly found obstacles including eigenvector sign flipping, and managing multiple dimensions of the data-set. This makes `R2-PCA` an ideal candidate for learning-based models. We demonstrate the methodology on mutual fund data-sets and illustrate its advantages in terms of applicability, and explainability.

1 Introduction

Principal Component Analysis (PCA) [1] is an important methodology used to reduce and extract meaningful signals from large data-sets. This makes it a crucial element for learning-based models. Where, it is commonplace for learning-based models to only accept data in two dimensions (N, D); where N is the number of samples and D is the number of features. The general consensus of practitioners, with respect to the size of data-sets, dictates that N has to be significantly larger than D to prevent over-fitting and other statistical issues that can arise from small sample sizes. The larger the number of features in a data-set, the larger number of samples and model parameters are required to produce meaningful results. PCA as an unsupervised learning algorithm aims to reduce the dimension of D. Unlike supervised learning algorithms, where the correct output is known, unsupervised algorithms requires interpretability and visualization to evaluate the correctness of its output. By reducing the number of D features, subsequent models require less parameters to train and thus reduce the probability of over-fitting training data-sets.

The majority of PCA research focuses on reducing data that does not depend on time. The naive approach to time series PCA is assuming the eigenvectors that represent the transformation can be computed once and, as new data is obtained, project the new data set using the historical eigenvectors. This approach implies the new data will contain a similar distribution, which may not be correct. For example financial data-sets are constantly changing leading to new principal component directions in each segment of time. A model designed for dimensionality reduction must account for these changes by updating its principal components while still retaining a relationships between components over time. Additionally, the PCA algorithm must be applied in a recursive manner through time to avoid look ahead bias and account for non-stationarity in the data. For example, if PCA is to be run on an entire time series data-set, the variation and thus principal components will be influenced by future dates that data from older periods does not have access to. Projecting the older data on principal components that incorporate future data-sets creates a look ahead bias since at that time there was no way of determining the direction of the data in future periods.

The proposed robust rolling PCA (`R2-PCA`) methodology extends standard PCA to computing reduced data using a rolling window approach while effectively tackling challenges commonly found in PCA and time series analysis including eigenvector sign switches and finite precision rounding errors. Since a learning-based model is only as powerful as the data it trains on, the more stable results of the `R2-PCA` (versus the Standard PCA) make it a better candidate for usage across AI-based applications.

*Special thanks to Miao Wang and Sikun Xu for their comments and suggestions.

[†]Industrial Engineering & Operations Research Department, Columbia University, **ah2347@columbia.edu**, Chief Scientific Officer, Ask2.ai, **ali.hirsa@ask2.ai**

[‡]Research Associate, Ask2.ai, **federico.klinkert@ask2.ai**

[§]CEO of Ask2.ai, **satyan.malhotra@ask2.ai**

[¶]Research Associate, Ask2.ai, **ryan.holmes@ask2.ai**

2 Data-sets

The main focus of this analysis into data reduction has been in relation to performance measures found in financial time series data-sets. Every performance measure is derived from some manipulation of pricing data found in a time series. For example, return captures the first moment of the distribution, volatility captures the second moment, and Sharpe ratio captures the relationship between return and volatility. There are over 100 performance measures, and each of them captures a different behavior [2].

The data-set used in this analysis is comprised of over 7,000 U.S. based Mutual funds; 40 performance measures are computed for each mutual fund that incorporate both absolute and relative performance. The primary data-sets used for demonstrative purposes in the paper come from the asset classes of Large Cap and U.S. Aggregate; however, similar results were found in other markets and asset types. For estimating the relative performance measures, S&P 500 Index and Bloomberg U.S. Aggregate Bond Index were chosen as indices for Large Cap and U.S. Aggregate markets respectively.

3 Rolling Window Input Structure

A time series data-set containing multiple assets and features must be represented as a 3-dimensional tensor denoted as X of shape (F, T, D). Each respective variable representing the number of funds F, the number of observations through time T, and the number of features D. The difference between `R2-PCA` and standard PCA is the added complexity of time dimension and the necessity to update the results of the model as new information becomes available in future time increments. Consider if standard PCA is used to fit all data available after each update; the data from the new time period would have a diminishing effect on the eigenvectors computed since there is an increasing amount of data from previous time periods. Additionally, there could be issues regarding data spillage since the data projected from older periods are influenced by data from future periods. A fourth parameter W must be introduced representing a rolling window in the time series to alleviate this issue and the final shape of the data-set X will be (W, F, T, D).

4 Time Series data-sets

Computers are only capable of computing values to a finite precision and in the case of eigendecomposition this can lead to issues with output consistency [6]. Small differences in rounding errors can change the direction of the eigenvector for covariance matrices with small elements. This can mean rerunning code utilizing PCA on the same computer can result in different outputs due to differing memory loads on the computer's processor. Due to the nature of modern computing technology, the finite precision rounding errors are unavoidable and additional functions in the `R2-PCA` algorithm is required to address these concerns. This poses a unique challenge to PCA over a time series, where the main goal is to achieve consistent principal components in which to project the data over time.

4.1 Computing the Eigenvectors

A important phenomenon found when running PCA on a time series data-set is that the projected data will sometimes jump as a result of the eigenvectors changing signs. Eigenvalue problems typically yield an infinite set of eigenvectors that will satisfy its constraints. Standard practice for PCA adds a constraint where the eigenvectors must have a length of 1. This reduces the infinite set of eigenvectors to just 2; this is where the issue of sign switching arises. Given any eigenvector which satisfies all constraints for PCA, the same eigenvector multiplied by -1 will also satisfy the problem. The sign flipping effect can happen to any principal component and each component is effected independently of another. This effectively means the eigenvector in the direction opposite of the data could potentially be the output of a computer-based PCA solver due to finite rounding precision error. Standard PCA is not as concerned with this issue since all data points are projected using the same eigenvectors. When running PCA recursively, a switch in component signs from one period to the next time will result in the entire reduced data for that period being multiplied by -1. A flip in eigenvector direction will completely alter the output of the reduced data, as shown in Figure 1 for reduced performance measures of US Large Cap mutual funds. This will void any results produced by the learning-based models using the reduced data as input.

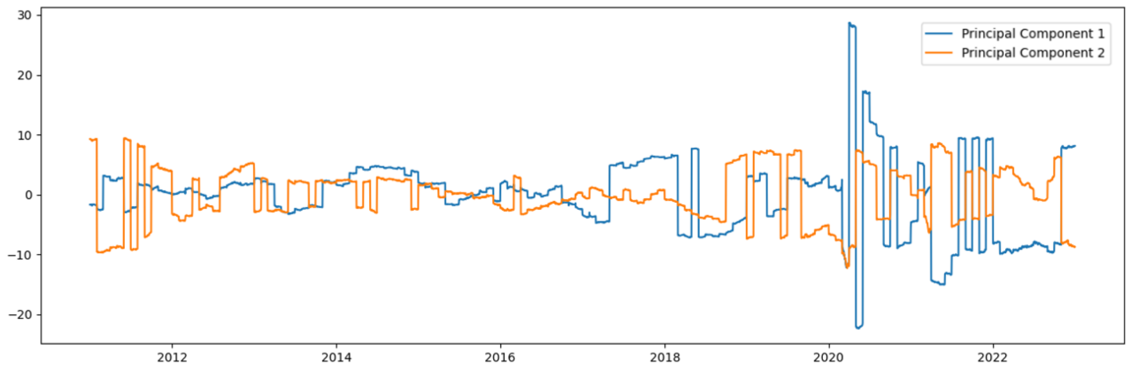


Figure 1: Standard Rolling Window PCA: Performance Measures of Large Cap Mutual Funds

Data is typically normalized over its features to ensure each feature has equal importance in the analysis since the covariance matrix is the key to determining eigenvector direction. The eigenvectors computed by PCA are usually desired to be in the direction of the data; however, this becomes numerically inconsistent when the data sets are normalized with a mean of zero and standard deviation of one. This factor combined with the finite precision errors of computing will lead to inconsistent results even when utilizing a fixed seed and other techniques to induce run consistency. A beneficial property of the eigenvector direction issue is that eigenvector sign flipping over time is much more important to later inputs of machine learning models than direction with respect to the data. As long as the direction of the components remain consistent, predictive models will simply factor this in when estimating model parameters.

4.2 Data-set Correlation

Standard PCA requires the computation of the covariance matrix, but the main challenge of modeling performance measures is their correlation can significantly change over time. The change in correlation can be attributed to the signal to noise ratio of the data-set as well as regime changes in the market. Financial data tends to be non-stationary as a whole while maintaining stationarity within certain macro economic regimes [3]. Most performance measures found in financial time series data-sets tend to be highly correlated and using all features could pose to be disadvantageous for model fitting purposes. The trade-off for pre-processing the data-set to reduce correlation could introduce a significant amount of noise in the reduced data-set. For example, picture a data-set with three features and the desire to produce two principal components from that data-set. Suppose feature 1 contains a high level of variance, feature 2 is a copy of feature 1, and feature 3 is simply random noise. The two principal components will strongly favor features 1 and 2. However, since these features are perfectly correlated, each principal component will essentially model the same characteristic found within the data-set. By analyzing and merging features 1 and 2, the first principal component will capture a meaningful direction while principal the second component will be forced in the direction of random noise due to orthogonality constraints. This factor poses a unique challenge to PCA methods computed in a rolling window; principal components computed over non-stationary data-sets can lead to principal components changing direction and orderings. For example, suppose there is one feature in a data-set which accounts for the most variation in future time increments while remaining stagnant in the current time period. Any method that directly uses information from future periods incur a look ahead bias rendering the results unusable. Thus, the effects non-stationarity data will always remain a constant consideration in recursive PCA methods.

5 `R2-PCA` Model Framework

`Robust Rolling PCA` (`R2-PCA`) is designed to address consistency and direction of principal components over time. `R2-PCA` computes the eigenvectors in the first time step and then uses these eigenvectors in future periods to ensure correct signs. If the eigenvectors computed in the first time step are in the opposite direction of the data (since the eigenvector multiplied by -1 is also a solution), the results of all future eigenvectors will also be incorrectly flipped. While predictive machine learning can account for this, in terms of model explainability, it is important to ensure the first set of principal components are computed in the correct direction of the data for visualization purposes.

5.1 Recursive Similarity of Principal Components

The sign change issue arises in all PCA problems although is most noticeable when applying PCA recursively. `R2-PCA` requires principal components to have the same sign throughout a time series (signs are allowed to change naturally due to market factors). Otherwise, the eigenvector and subsequent projected data will suddenly appear be multiplied by -1; resulting in a large and misleading jump in the output of the `R2-PCA` model. To fix eigenvector flipping, the idea emerges from establishing one principal direction for each eigenvector and then recursively comparing the eigenvectors in the next period to the current one using cosine similarity. The cosine similarity score can be computed as the dot product of eigenvectors given the eigenvectors calculated are unit vectors. The similarity scores will recursively link principal components in sequential time periods, allowing the `R2-PCA` algorithm to unflip the eigenvectors should the cosine similarity score be below 0.

5.2 Ordering of Principal Components

The cosine similarity score allows the `R2-PCA` algorithm to compare principal components in different time periods. The eigenvectors created by PCA are ordered by their corresponding eigenvalues which represent a magnitude of variation. Computing PCA in a recursive manner creates an issue where the ordering of principal components will change over time; this is typically found in data-sets where the features maximizing variance suddenly shift to new features due to market forces. This factor necessitates a reordering of the principal components in the current time period with respect to the eigenvectors in the previous period. At each step in the rolling window and each eigenvector in the current period, the absolute cosine similarity score will be computed for all eigenvectors found in the previous period. The eigenvectors in the current time period will be reordered according the their maximum cosine similarity score. The absolute value of these scores are critical to determining the eigenvectors flipping signs. Should the maximum absolute similarity score come from a negative value, then the eigenvector in the current period will unflipped by multiplying it by -1 and then reordered accordingly. To further improve the accuracy of ordering, more than the desired number of principal components can be computed and tracked between periods. Once these principal components are reordered, they can be reduced to the desired number of principal components.

5.3 Variable Number of Features

The use of the straightforward cosine similarity score allows for comparisons to be made on datasets with a variable number of features D. The number of elements in an eigenvector is equal to the number features in a dataset, rendering the dot product unusable for vectors with differing number of elements. The solution to this problem is to simply reduce the dimensionality of eigenvectors in different time periods to the features shared between each dataset. The cosine similarity score will set the norm of each of these eigenvectors to 1, ensuring the range of similarity score lies between -1 and 1. While the `R2-PCA` algorithm can be used on datasets with variable number of features, it is important make certain that a majority of features remains constant over time; otherwise the accuracy of the cosine similarity test diminished in determining the flipping of eigenvectors. Variable features can frequently arise in financial time series datasets. For example, imagine a dataset in which the LIBOR feature is gradually phased out by the SOFR feature. The ability for `R2-PCA` to transition and compare eigenvectors of variable feature length can be an important tool for adding new performance measures and features that may not exist in a past time periods.

5.4 Principal Components and Initialization

While `R2-PCA` is typically computed in the earliest time period available in the given data-set and rolled forward, it is not necessarily the only way to structure the recursion. One of the main objectives of `R2-PCA` is to produce stable results for a wide variety of financial time series metrics. This poses a potential issue when certain data-sets exhibit drastic shifts in performance measures due to market factors. Principal components for each rolling window time increment do not have the luxury of knowing what direction the next principal component will be in. Computing the cosine similarity of components between time periods give an idea of the movement of components over time and the overall stability of the data-set itself.

Ideally, data-sets that exhibit changes in regime do so gradually enough that the principal components can be slowly adjusted for each increment in the rolling window. In data-sets where that is not that case, careful analysis must be taken in choosing the time period in which the `R2-PCA` model will initialized, as the components of the first initialization will set the precedent for following components in which they will be compared to using cosine similarity. Alternative solutions to this issue could include changing the features in the data-set and rolling the window in smaller time increments. Additionally, it is possible to separate the data-sets into distinct periods (e.g., regimes) and fit the `R2-PCA` individually on each period. Potentially, each fitting would save its own principal components in the previous period, so when an already established regime is re-entered the `R2-PCA` model will continue where the same regime left off.

6 `R2-PCA` Algorithm

The design of the `R2-PCA` Algorithm is to run PCA in a recursive manner while altering the eigenvectors at each time step to diminish the negative effects of sign changes and eigenvector directions. Standard PCA requires inputs to be of dimension 2. The `R2-PCA` algorithm, depicted below, includes the technique of converting a dataset of dimension (W, F, T, D) to a two dimensional size (T, D) so standard PCA can be employed at each increment in the rolling window. This reduction in dimension size is achieved by computing the individual covariance matrix for each element in F and then averaging the results prior to the eigendecomposition step. This allows to perform PCA on the average covariance. If a model will be trained using multiple assets, it assures that the overall features going into the model are uncorrelated even it cannot be guaranteed on an asset level.

Algorithm 1 `R2-PCA` Algorithm

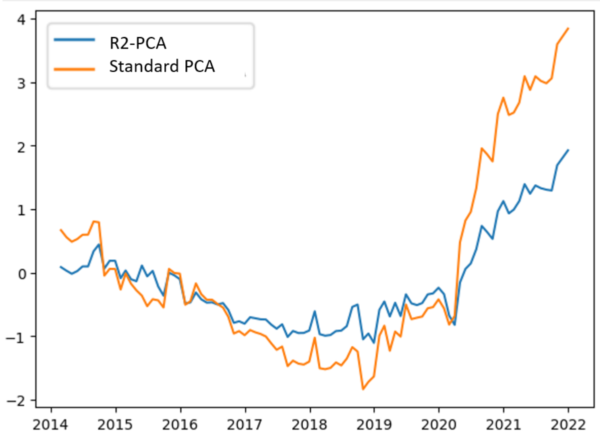
Choose an number of Principal Components p
Choose a dataset X with dimensions (F, T, D) ▷ (Funds, Time, Features)
Choose a rolling window length W
Choose w as set of all time increments up to time t with $|w| = W$
Set time $t = 1$ and rolling window $w_t = \{t, t-1, ..., t-W+1\}$ ▷ If data exists for $t < 1$, else $w_t = \{t\}$
Compute the covariance matrix C_f for each element/asset in $f \in w_t$
Compute average covariance matrix $\bar{C} = \frac{1}{|F_t|} \sum_i C_i$
Eigendecompose $\bar{C} = P \Lambda P^T$ and extract eigenvectors $V_{w_t} = \{v_1, ...v_p\}_{w_t}$
for $t = 2, ..., T$ **do**
    Set rolling window $w_t = \{t, t-1, ..., t-W+1\}$
    Compute the covariance matrix C_f for each element/asset $f \in w_t$
    Compute average covariance matrix $\bar{C} = \frac{1}{|F_t|} \sum_i C_i$
    Eigendecompose $\bar{C} = P \Lambda P^T$ and extract eigenvectors $V_{w_t} = \{v_1, ..., v_p\}_{w_t}$
    for $i = 1, ..., p$ **do**
        Set $j = \text{argmax}(|v^i_{w_t} \cdot V_{w_{t-1}}|)$ ▷ Find eigenvector with highest absolute similarity score for ordering
        if $v^i_{w_t} \cdot v^j_{w_{t-1}} < 0$ **then** ▷ Eigenvectors from current and previous rolling windows with $||v|| = 1$
            Set $v^i_{w_t} = -v^i_{w_t}$ ▷ Sign Flip
        end if
    end for
    Reorder $V_{w_t} = \{v_1, ..., v_p\}_{w_t}$ from each argmax j result ▷ Data can be projected using V_{w_t} after this step
end for

7 `R2-PCA` Results

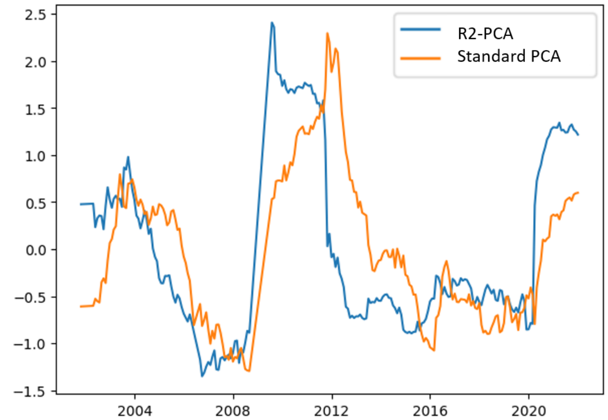
7.1 Fixing the look ahead bias of standard PCA

A key test in demonstrating the accuracy of the `R2-PCA` model is to compare the results of the standard PCA on the entire dataset with the resutls of `R2-PCA`. It should be noted that running the standard PCA is this manner will have a look ahead bias and will only contain one set of eigenvectors for the entire dataset. This look ahead PCA model should only be used for testing as the projected data in older time periods are significantly influence by future data. It should be noted that the computation of the look ahead algorithm should utilize the same methods in `R2-PCA`; essentially it will treat the entire dataset as one time period in the rolling window. When visually comparing results, the effects of normalization on a rolling window (`R2-PCA`) and applying PCA to the whole window will consistently alter the scaling of data between models. This factor is unavoidable, since it is impossible for `R2-PCA` to access the future data for normalization in the current rolling window increment without incurring look ahead bias. Instead, benchmarking requires a focus on the general shape of the outputs for both models to ensure the eigenvectors are correctly being flipped over time.

Figure 2 depicts the PCA benchmarking corresponding to the Large Cap and US Aggregate performance measures datasets.



(a) Large Cap VFTNX Fund

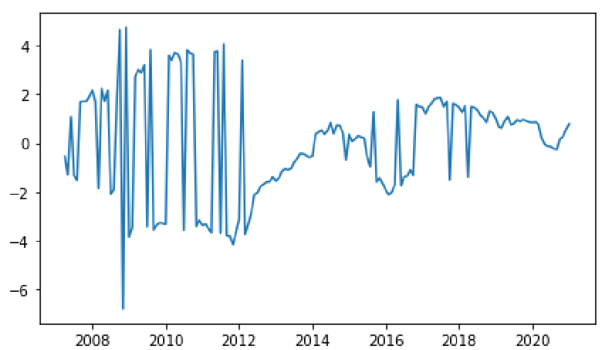


(b) US Aggregate ISHIX Fund

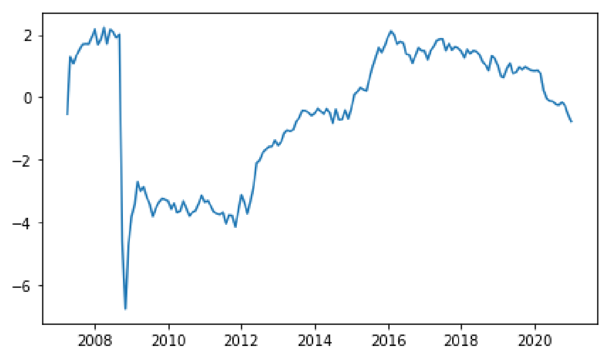
Figure 2: Look Ahead PCA Benchmark and `R2-PCA` Results

7.2 Fixing Eigenvector Sign Flips

The effects flipping eigenvectors can be viewed by plotting standard PCA on each rolling window increment with the results of the `R2-PCA` algorithm. Flips typically can be seen where the data appears to be multiplied by -1 and flipped over the axis. From a visual standpoint, it would appear as thought the reduced data would be in a period of low volatility if it were simply flipped over its axis. It is more difficult to distinguish between eigenvector flips and natural jumps in the market during periods of high volatility. As shown in Figures 3 and 4, drastic market jumps should be anticipated around the 2008 financial crisis and the COVID-19 pandemic. The characteristic of the flipping eigenvectors overtime can be better visualized at ask2-ai/insights-203. The visual highlights the abrupt sign flips of the standard PCA vesus the smoother transitions of the `R2-PCA`.



(a) Standard PCA refitting

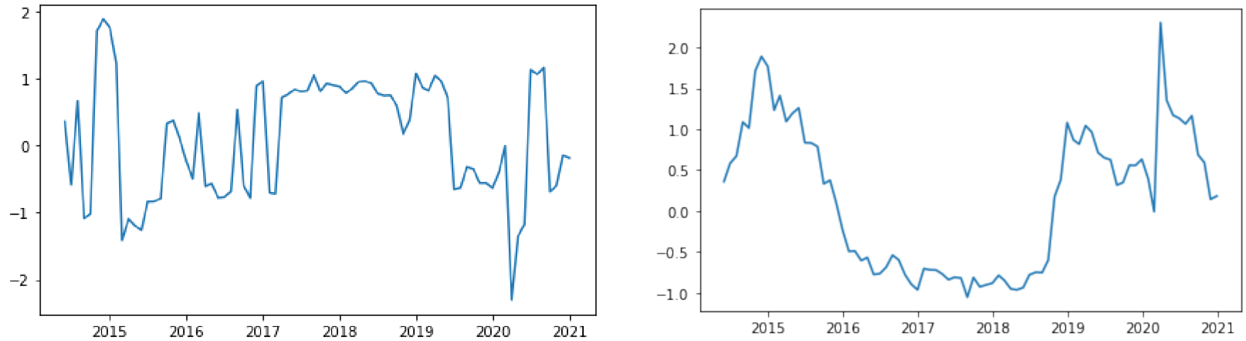


(b) `R2-PCA`

Figure 3: PCA recursive correctness: US Aggregate LSIIX Mutual Fund

7.3 Incorporating Multiple Dimensions

The facets included in the `R2-PCA` model could be composed of any group of data-sets that contains variables. For example, it could be a group of mutual funds which each mutual fund containing a data-set of performance measures. The number of facets can be variable each time that we roll the window, but the number of variables that each of the facet contain must be the same across all facets at any point in time. However, the number of variables can vary over time for all facets as mentioned in Section 5.3

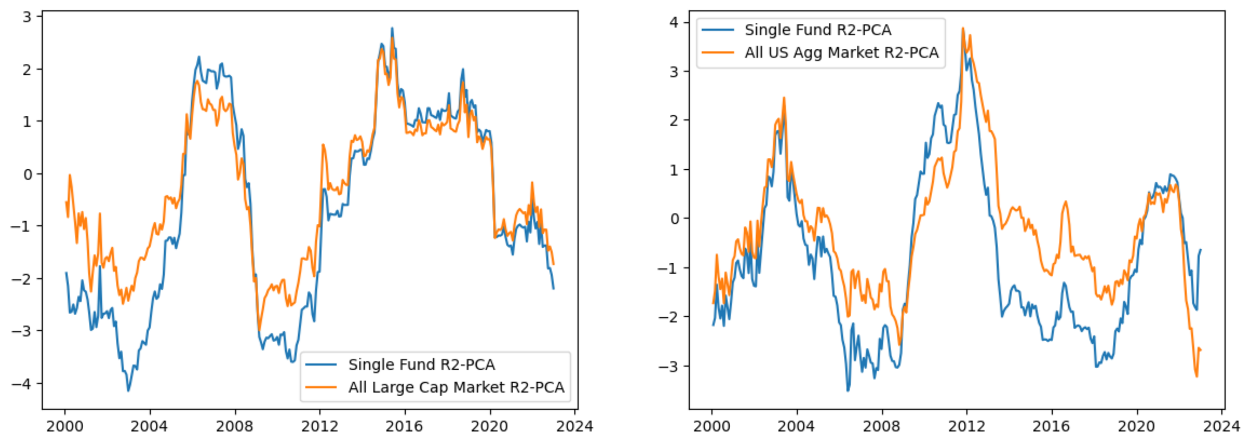


(a) Standard PCA refitting

(b) `R2-PCA`

Figure 4: PCA recursive correctness: Large Cap VVPLX Mutual Fund

The amount of facets fed into the `R2-PCA` model highly depends on the model formulation. Since fitting multiple facets takes into account the average covariance, the resulting principal components are going to be similar to the case of fitting one face at the time if that facet is highly correlated to the average of the facets. We can see this behavior in Figure 5.

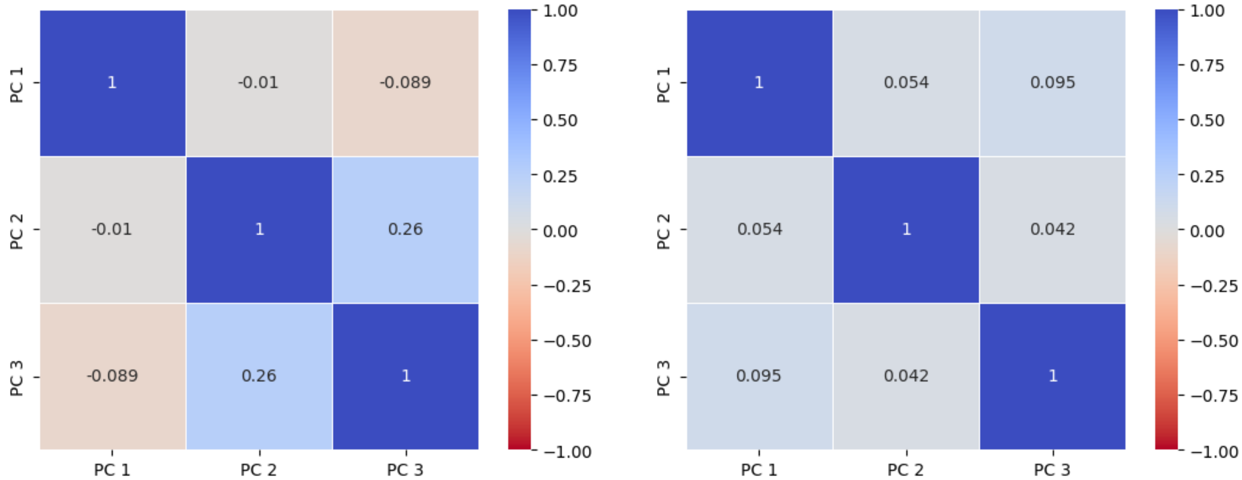


(a) Large Cap: LAFFX

(b) US Agg: SCOAX

Figure 5: First Principal component for single and multi-facets

Fitting multiple facets at the same time ensures uniformity of feature importance for subsequent model trains. Although it cannot ensure that individual facets will have uncorrelated principal components, it ensures that the combination of all facets will be uncorrelated since they will be fed into the same model, as seen in Figure 6



(a) Large Cap

(b) US Aggregate

Figure 6: `R2-PCA` Correlation of Reduced Data

7.4 Data-sets with sub grouping

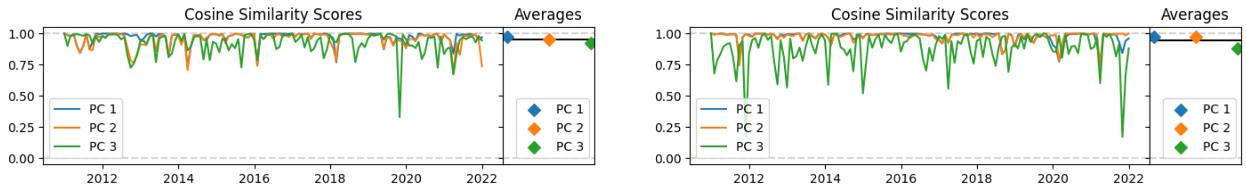
Interpretation of PCA or `R2-PCA` results is an important property to consider when assigning unique and meaningful labels to principal components. There exists a trade-off in `R2-PCA` analysis between orthogonality and explainability. Consider, for example, a data-set in which the `R2-PCA` model is run in which the data-set is concatenated from 3 labeled data-sets and the objective of the analysis is to obtain 3 principal components that represent the market. There are two choices; either run `R2-PCA` on the entire concatenated data-set or run `R2-PCA` on each labeled data-set independently. If `R2-PCA` is run on the entire data-set, the 3 principal components will be orthogonal and posses useful properties such as independence from one another. Then, consider running `R2-PCA` on the data-set split into 3 predefined groupings and extracting one principal component from each. Now there are 3 components and each one definitively belongs to a predefined group in the market, however orthogonality is no longer guaranteed to the concatenated eigenvectors from each group. This will instigate the eigenvectors toward the direction of the standard basis, making a single element of each component larger than the rest making name assignment more solid. However, doing this creates a challenge in terms of naming conventions, where each principal component could represent the same characteristic found withing the data-set thus resulting in identical names. There is a significant risk that each component could be correlated with each other, especially if certain features dominate the measure of variation shared between data-sets. Balancing explainability and orthogonality requires testing each method and weighting the costs and benefits of each case.

8 Back Testing `R2-PCA` Results

Each element of an eigenvector represents a dimension of a feature space. By choosing the largest absolute value for elements of each eigenvector, tangible labels can be assigned to each eigenvector based on the name of the feature space of that element. Care must be taken when choosing the maximum, as an eigenvector in the incorrect direction of the data could results in the minimum having the most influence on the principal direction and thus should be named after its feature. This method of naming principal components makes the most sense when each eigenvector is close to the standard basis (where typically only 1 element has a large weight). Orthogonality constraints will help ensure a diverse range of names for each component.

8.1 Application and Validation

Validation of the `R2-PCA` algorithm requires examining the cosine similarity scores, percent variation, and correlation of the reduced dataset. The cosine similarity scores, depicted in Figure 7, is a measure of the changes of eigenvectors over time. While the range of cosine similarity is between -1 and 1, the `R2-PCA` algorithm will flip the sign of eigenvectors whose score is below 0. Backtesting similarity scores should range from 0 to 1, with 0 representing a drastic shift in direction and 1 indicating a stable principal direction over time. For example, in Figure 7, a majority of components remain relatively stable over time; only PC 3 in US Aggregate (b) shows signs of instability.

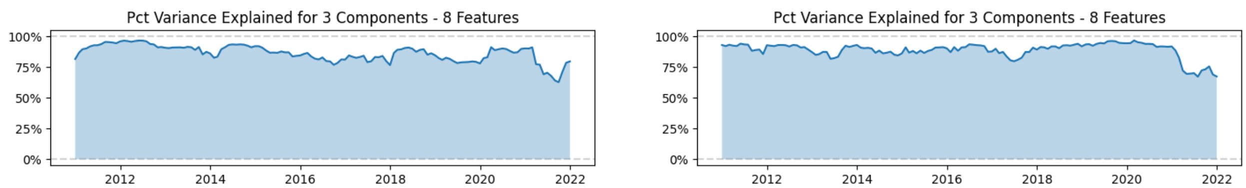


(a) Large Cap

(b) US Aggregate

Figure 7: `R2-PCA` Cosine Similarity Scores

Percent of variation illustrates the sum of variation captured by the principal components. While the percentage of variation is primarily dependent on the number of components and original features in the dataset, the change in percent of variation will give insight to the stability of the model. Algorithms and datasets with good performance will see very little changes in variation over time. Figure 8 shows the relative stability of the `R2-PCA` algorithm in both US Aggregate and Large Cap datasets.



(a) Large Cap

(b) US Aggregate

Figure 8: `R2-PCA` Percent of Variation Captured

8.2 Covariance Instability

Unsupervised learning models sometimes require back-tests using artificial data-sets with desired characteristics to best assess performance. The focus of this back-test is to examine how recursive PCA models handle principal components changing direction over time.

The setup for this back-test requires that two subsets of uniformly distributed data be generated and projected into uncorrelated sub spaces. In this example, data-sets in two dimensions with a correlation of zero will have clearly visible principal components where one subset of data lies on the x axis and the other subset on the y axis for the first time increment. To retain the uncorrelated nature of the data-set over time, the data should be centered but not normalized at each time increment.

Artificial data-sets allow for the parameterization of the change in principal component direction denoted as θ. To simulate principal component change over time, at each subsequent time period will take the previous periods data and rotate it by θ. Since the data is uncorrelated, rotating the data-set by θ will also change the principal components by θ. An optional feature of this back-test is to randomly generate and project new data at each time increment. This will have the added effect of simulating eigenvector sign flipping since the data is centered but not normalized.

Another iterative approach to PCA has been proposed as the IPCA [7] by Paul Bilokon and David Finkelstein. The IPCA model computes the eigenvectors with PCA in the first time period, then uses the covariance and Ogita-Aishima algorithm [8] to perturb the eigenvectors for all future time increments. The Ogita-Aishima algorithm relies on the assumption that the covariance matrices only change a small amount between time steps. Should the covariance matrix sharply change (such as an event that shocks the financial market), the Ogita-Aishima algorithm will leave the eigenvectors unchanged and will detach from the principal directions found within the data-set at that time period. Those eigenvectors will continue to remain in that position until a covariance within the threshold of convergence for the Ogita-Aishima be found in future time periods.

The `R2-PCA` algorithm utilizes the robustness of the cosine similarity measure to accommodate for financial time series data-sets with higher levels of covariance instability across time. By using PCA at each time increment, the principal components are guaranteed to point in the directions that maximize variation for that subset of data while incorporating the additional dimension of funds F, variable length features D. Animations depicting the results of standard PCA, IPCA and `R2-PCA` and the time series instability backtest can be found at: ask2-ai/insights-203

9 Conclusion

The formation of `Robust Rolling PCA` method stems in part from financial industry's need of stable and reliable reduced time series data-sets. From multiple runs and stress testing, the `R2-PCA` methodology serves as a robust method of computing meaningful principal components in a recursive structure while mitigating common obstacles in PCA including eigenvector sign flipping, ordering, and managing higher dimensionality. The unsupervised nature of PCA will provide practitioners with the ability to tweak and perturb the `R2-PCA` algorithm to optimize the performance of models using reduced time series data as input. Visualization tools such as look ahead benchmarking and principal component naming conventions can further enhance the interpretability and explainability of the `R2-PCA` results. Further study of recursive principal component methods could focus on anticipating regime shifts, polishing naming conventions, and improving performance on data-sets with high correlation among features.

References

[1] Jolliffe, Ian T., and Jorge Cadima. "Principal component analysis: a review and recent developments." Philosophical transactions of the royal society A: Mathematical, Physical and Engineering Sciences 374.2065 (2016): 20150202.

[2] Hirsa, Ali, Rui Ding, and Satyan Malhotra. "Explainability Index (EI): Unifying Framework of Performance Measures and Risk of Target (RoT): Variability from Target EI." Available at SSRN 4335455 (2023).

[3] Schmitt, Thilo A., et al. "Non-stationarity in financial time series: Generic features and tail behavior." Europhysics Letters 103.5 (2013): 58003.

[4] Bro, Rasmus, Evrim Acar, and Tamara G. Kolda. "Resolving the sign ambiguity in the singular value decomposition." Journal of Chemometrics: A Journal of the Chemometrics Society 22.2 (2008): 135-140.

[5] Bro, Rasmus, Riccardo Leardi, and Lea Giørtz Johnsen. "Solving the sign indeterminacy for multiway models." Journal of Chemometrics 27.3-4 (2013): 70-75.

[6] Greenbaum, Anne, and Zdenek Strakos. "Predicting the behavior of finite precision Lanczos and conjugate gradient computations." SIAM Journal on Matrix Analysis and Applications 13.1 (1992): 121-137.

[7] Bilokon, Paul, and David Finkelstein. "Iterated and exponentially weighted moving principal component analysis." arXiv preprint arXiv:2108.13072 (2021).

[8] Ogita, Takeshi, and Kensuke Aishima. "Iterative refinement for symmetric eigenvalue decomposition." Japan Journal of Industrial and Applied Mathematics 35 (2018): 1007-1035.

[9] Animation link for the methods: https://www.ask2.ai/insights-203/