




Article

Dataset Reduction Techniques to Speed Up SVD Analyses

Laurens Bogaardt ¹, Romulo Goncalves ¹, Raul Zurita-Milla ^{2,*} and Emma Izquierdo-Verdiguier ³

¹ Netherlands eScience Center; l.bogaardt@esciencecenter.nl, r.goncalves@esciencecenter.nl

² Faculty ITC, University of Twente; r.zurita-milla@utwente.nl

³ Faculty IPL, Universitat de Valencia; emma.izquierdo@uv.es

* Correspondence: r.zurita-milla@utwente.nl

Academic Editor: name

Version August 1, 2018 submitted to ISPRS Int. J. Geo-Inf.

Abstract: Performing SVD analyses on large datasets can be computationally costly and time consuming. Often, techniques exist to arrive at the same output, or at a close approximation, which require far less effort. This article looks at several such techniques and at the inherent scale of the structure within the data. When the values of a dataset vary slowly, e.g. in a spatial field of temperature over a country, there is a high level of autocorrelation and the field contains large scale structure. Datasets need not have a high resolution to describe such fields. Using generated Gaussian Random Fields with various levels of autocorrelation, we examine rank decomposition, coarsening and approximate SVD procedures. This article outlines when certain techniques can be useful and makes predictions about the error incurred in the approximations based on the level of autocorrelation of the input data. Finally, these techniques and predictions are verified using real-world geospatial datasets.

Keywords: Singular value decomposition, autocorrelation, rank deficiency, data reduction, coarsening, approximate SVD, Gaussian Random Fields

1. Introduction

Performing *Singular Value Decompositions* (SVD's) on large datasets can be computationally costly and time consuming. Often, techniques exist to arrive at the same output, or at a close approximation, which require far less effort. This article examines several techniques which exploit autocorrelation and rank decomposition to analyse the data in an efficient manner. Even though these techniques are not novel, a review is beneficial for domains less familiar with analyzing large datasets [1–3]. Ultimately, this article provides researchers with a decision tree indicating which technique to use when and predicting the resulting level of accuracy based on the dataset's structure scale.

To arrive at these prediction, *Gaussian Random Fields* (GRF's) are generated with various levels of autocorrelation and are subsequently reduced in size. The amount of error incurred in this reduction is determined by comparing the SVD of the reduced dataset to that of the original. Finally, the techniques and predictions are verified using real-world geospatial datasets. The reported results come from calculations performed in an accompanying *Jupyter Notebook* [4]. In order to develop intuition, some matrix algebra is briefly reviewed first.

1.1. Matrix Size and Rank

Many datasets can be represented by a matrix. Take, for instance, a group of n individuals who report scores on m questions. Or take the temperatures at m locations, measured over n time

periods. These values can be arranged in a matrix with m rows and n columns. Like a vector, a matrix is a combination of basis vectors which indicate direction, each with a coefficient which indicates magnitude. As an extension of the vector, a matrix has two bases, the left- and the right-, or the row- and the column basis. These bases can also be changed via a rotation. A clever basis to rotate into is one where the basis vectors are orthonormal and each subsequent set of left- and right basis vectors, known as a *mode*, explains as much of the remaining variance in the dataset as possible. Such basis vectors are called *Principle Components* (PC's) or *Empirical Orthogonal Functions* (EOF's) and they are found via an SVD of the matrix.

If there exists a rotation for which some coefficients become zero, the matrix needs fewer basis vectors to describe it than are available. In a sense, it is underdetermined. Its internal dimension is smaller than what would be guessed from its m by n size. This is the concept of matrix *rank*; if the rows and columns both span a subspace of dimension r , a matrix has rank r . A matrix is said to have full rank if $r = \min(m, n)$, the maximum number of linearly independent basis vectors. If $r < \min(m, n)$, it is rank deficient.

A rank decomposition or factorization is the splitting of a matrix into a product where each factor has full rank. For example, an m by n matrix of rank r can be decomposed into an m by r matrix multiplied by an r by n one. An SVD is a special type of rank decomposition. It results in a set of orthonormal left basis vectors U , a list of coefficients s and a set of right basis vectors V . For rank deficient matrices, some of the coefficients, known as singular values, are zero.

The mathematical rank r of a dataset is usually not relevant in practice because the data originate from devices with finite precision [5]. Even though some singular values of a dataset are not zero, they may be small enough to be considered *noise*. If we take the inherent imprecise nature of real-world data into account, we can approximate a dataset by another matrix of rank l , with $l < r$. Following the Eckart-Young-Mirsky theorem, the best possible approximation is one described in the same bases as the original dataset, taking a subset of the l largest singular values and truncating the remainder [6]. Taking a threshold ϵ , the dataset is said to be approximate rank deficient if some singular values fall below ϵ . Then, it has an ϵ -rank of l and the norm of the difference with its l -rank approximation is at most ϵ [5].

So, we can identify three types of matrix *sizes*. The first is the size of the full matrix, $m \times n$. Storing the entire, original dataset requires $m \times n$ units of storage and computing the product with a vector requires $m \times n$ flops. The second type of size is the rank decomposed version, which requires $m \times r + r \times n$ units of storage and an equal number of flops for the vector multiplication [5]. If r is small, this can be a substantial improvement. The final definition of size approximates the original dataset with a matrix of rank l , resulting in even smaller storage and faster computations, while losing as little information as possible.

1.2. Efficiency

The term *efficiency* used in this article is related to the concept of rank deficiency. A calculation is called efficient if it never requires the construction of an unnecessarily large, intermediate matrix. The best way to build up intuition for this concept is via an example.

One often wants to find the norm of the difference between two fields. This can be done directly by subtracting one matrix from the other and summing the square of the elements. However, for large matrices, the direct calculation may be unnecessarily time consuming. Let's assume datasets A and B are rank deficient and stored in SVD form. As discussed in section 1.1, storage space can be reduced by saving rank deficient matrices in SVD form. Determining the norm of their difference directly requires reconstructing A and B from their SVD's. This requires additional storage, sometimes more than would fit in the RAM-memory of an ordinary computer.

Fortunately, an alternative approach exists. Let $\|\cdot\|$ indicate the Frobenius norm, $\langle\cdot\rangle$ the Frobenius inner product and the \circ operator the Hadamard product, then the norm of the difference between matrices A and B is given by equation 1.

$$\begin{aligned} \|A - B\|^2 &= \|A\|^2 + \|B\|^2 - 2\langle A, B \rangle \\ &= s_A^T s_A + s_B^T s_B - 2s_A^T (U_A^T U_B \circ V_A^T V_B) s_B \end{aligned} \quad (1)$$

Figure 1 depicts the matrix operations in this calculation and visualises the rank deficiency of A and B via the rectangular shapes of their U and V bases. It also shows that this procedure can determine the norm without ever creating a prohibitively large matrix. This is what defines the term *efficiency* as used in the present article.

$$\|A - B\|^2 = \boxed{s_A^T} \boxed{s_A} + \boxed{s_B^T} \boxed{s_B} - 2 \times \boxed{s_A^T} \boxed{U_A^T} \boxed{U_B} \circ \boxed{V_A^T} \boxed{V_B} \boxed{s_B}$$

Figure 1. Exact norm of difference via SVD

1.3. Decision Tree

There are several reasons for performing an SVD analysis. When performed on a single spatial field, it may be to find the PC's or EOF's, which describe areas that behave similarly. In many real-world applications, however, the analysis of a field does not only involve a single time period but includes data over multiple weeks, months or years. Then, researchers are interested in finding relevant patterns which appear in both datasets. The *Maximum Covariance Analysis* (MCA) and *Canonical Correlation Analysis* (CCA) examine the product matrix of two datasets and determine which patterns occur frequently and simultaneously [7,8]. Such a pattern, or mode, is a combination of a left- and a right basis vector. One technique to find these modes is to perform an SVD on the product of the standardised datasets. In some domains, the term SVD is used synonymously with MCA. In an MCA, modes are found where the left- and the right vector covary maximally, whereas in a CCA, they correlate maximally [9].

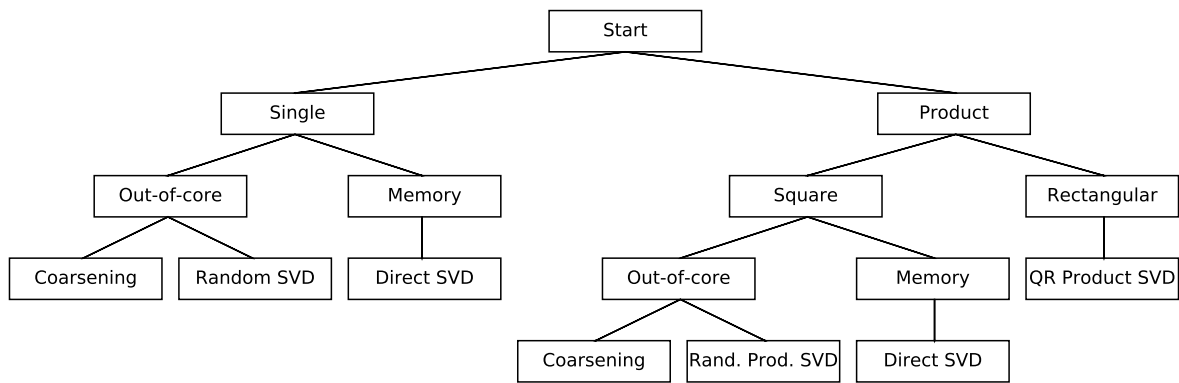


Figure 2. This is a figure.

Figure 2 shows several options a researcher has when performing an SVD. The first question to be answered is whether the SVD will be applied to a single matrix or to the product of two matrices. For single fields, the data may be small enough to fit in the memory of a computer. Then a regular SVD is the best option. If the dataset is too large, two alternatives exist which provide an approximate answer. These will be discussed in section 3.3 and section 3.4.

When the SVD is performed on the product of two matrices, the best course of action depends on whether the matrices are square or rectangular. The rank of a matrix is at most the size of the smallest side, which, for rectangular matrices, can be small. How to exploit this fact is described in section 3.1. Square matrices small enough to fit in memory can be analysed directly. What to do with larger, square datasets is discussed in section 3.6 and section 3.7.

2. Materials and Methods

In domains such as climate science and phenology, datasets are typically spatio-temporal fields, e.g. of temperature. In such fields, values vary slowly and neighbouring points are not entirely independent of one another, neither in space nor in time [7]. Then, there is a high level of autocorrelation and the field contains large scale structure. Such redundancy in the data means the matrix is rank deficient.

To compare our techniques and to find a relation between performance and structure scale, we need to be able to generate fields which resemble those often encountered in real-world applications. Additionally, we require methods to measure the autocorrelation of real-world datasets.

2.1. Spatio-Temporal Fields

As simulated spatio-temporal fields, real-valued *Gaussian Random Fields* are particularly useful because their structure scale can be captured in a single parameter. For such rotational invariant fields, the spectrum follows the power law described by $P(k) = c_0 |\vec{k}|^{-\alpha}$ where \vec{k} is the wavevector and α the parameter which controls the level of autocorrelation. Figure 3 shows fields with various α 's.

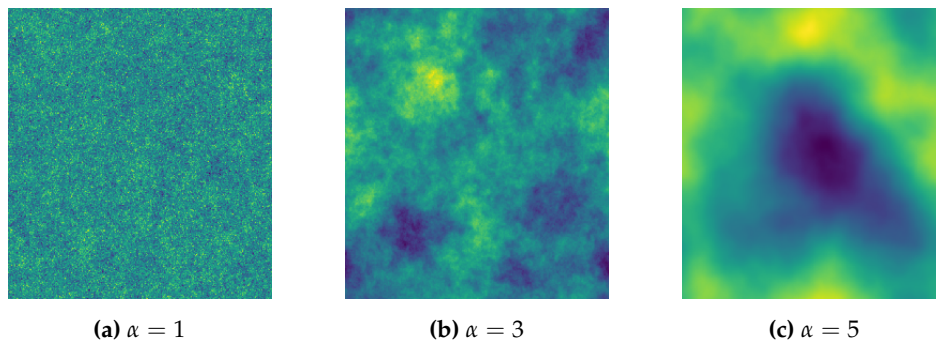


Figure 3. Gaussian Random Fields for various α 's

Just as there is spatial autocorrelation, there is temporal autocorrelation, when the values of the field over the entire time period do not change drastically. In principle, there can be different levels of autocorrelation over time and over space. However, for simplicity, in this article we will use the same α to determine the level of autocorrelation in all dimensions.

2.2. Autocorrelation

In the geosciences, there are several measures of spatial autocorrelation [7,8]. A frequently used one is Moran's I [10–12]. Figure 4 shows the relationship between Moran's I , using a uniform kernel with a bandwidth equal to 10, and the α of our generated GRF's.

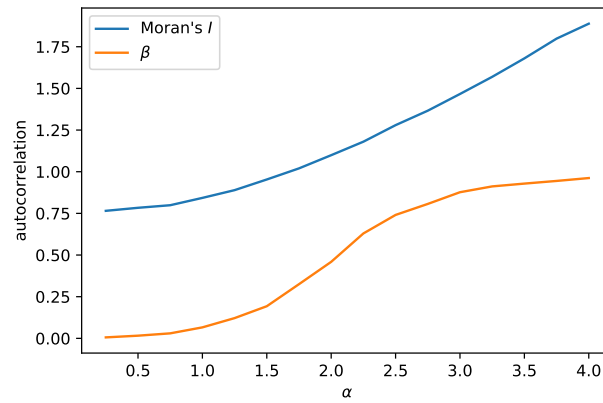


Figure 4. Measures of autocorrelation as a function of α

Additionally, one can devise a measure from the singular values. Each singular value indicates the amount of variance explained by its associated mode. For fields with autocorrelation, the sorted list of singular values decays quickly. One can try to fit a power law to this list and estimate the exponent, which we will call β . All these measures give an indication of the scale of the structure in the field and the level of autocorrelation in the data.

3. Results

This section lists several SVD related implementations to analyse large datasets efficiently by exploiting autocorrelation and rank deficiency. It includes three use cases to illustrate the procedure.

3.1. Exact Product SVD via QR Decomposition

In real-world applications, one often wants to find the relation between two fields. Analyses such as the MCA and CCA, discussed in section 1.3, rely on performing an SVD of the product matrix of the two fields. Take two input datasets with the various spatial gridpoints as rows and the sample of recorded values over time as columns. Centering and multiplying these gives the cross-covariance matrix. For highly rectangular matrices, when there are many spatial gridpoint but few temporal samples, the resulting cross-covariance matrix is inefficiently large and obviously rank deficient. Performing a rank decomposition, such as the *QR decomposition*, allows one to do the SVD in an efficient manner [3,13]. As shown in figure 5, the result is mathematically identical to the full SVD, while no unnecessarily large, intermediate matrix is ever formed.

$$\begin{bmatrix} A \\ B^T \end{bmatrix} = \begin{bmatrix} R_A \\ R_B^T \end{bmatrix} \begin{bmatrix} Q_A^T \\ Q_B^T \end{bmatrix} = \begin{bmatrix} C \\ Q_B^T \end{bmatrix} = \begin{bmatrix} \tilde{U}_C \\ S_C \\ \tilde{V}_C^T \end{bmatrix} \begin{bmatrix} Q_B^T \end{bmatrix} = \begin{bmatrix} U_C \\ S_C \\ V_C^T \end{bmatrix}$$

Figure 5. Exact SVD of a product via QR decomposition

3.2. Case Study using SI-x and AVHRR Data

Phenology is the science that studies the timings of recurring biological events such as leafing and blooming as well as their causes and variations in space and time. Spatio-temporal fields of remotely sensed images can be used to derive various phenological metrics. One of these metrics is the so-called *Start of Season* (SOS), which indicates the beginning of photosynthetic activity in plants. In this section, we use a SOS field of the US, made by processing time series of the *Advanced Very High Resolution Radiometer* (AVHRR) sensor [14]. Additionally, we use the *Extended Spring Indices* (SI-x), which are a suite of models that transform daily temperatures into consistent phenological metrics [15]. In particular, we take the long-term (1989 to 2014) and high spatial resolution (1km) version of the Bloom

index, which was recently generated for the US by adapting the SI-x models to a cloud computing environment [16].

Calculated over a subsection of the US, we estimate the Bloom field to have $\alpha \approx 3$ and Moran's $I \approx 0.97$ and the SOS field to have $\alpha \approx 2$ and Moran's $I \approx 0.37$ [4]. These measures have no further influence on the SVD via QR decomposition because this technique provides an exact result. Indeed, the accompanying *Jupyter Notebook*, as well as work being prepared for publication, shows that this technique provides the full SVD of the cross-covariance matrix in a matter of seconds, without ever exceeding the RAM-memory [?].

3.3. Approximate SVD via Coarsening

Although the QR decomposition works well for two rectangular matrices, sometimes the input data is large and square. Performing an SVD on such large datasets will be time consuming. When a spatial field has large scale structure, the values of neighbouring cells do not change drastically. Perhaps these cells can be aggregated together to produce a smaller dataset which still faithfully describes the original field. In this section, we coarsen various Gaussian Random Fields by averaging patches of neighbouring gridpoints. We then compare the result with the full calculation.

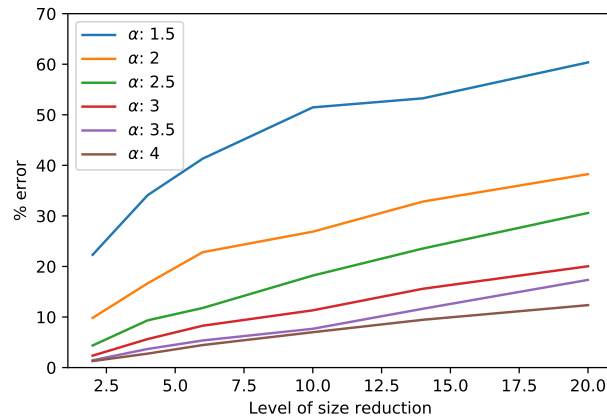


Figure 6. Error after coarsening a spatial field for various α 's

Figure 6 shows the percentage error in a coarsening process for matrices of various α 's and different windows sizes. The error is determined as the norm of the difference between the original matrix and the coarsened version, divided by the norm of the original [4]. Note that a field with high autocorrelation ($\alpha = 2$) differs only by a few percent from one 25 times smaller (level = 5).

3.4. Approximate SVD via Dimension Reduction

For very large datasets, the SVDs may be obtained using a random algorithm reviewed extensively in an article by Halko et al. [17]. The number of singular values will then be truncated to the l largest values, similar to finding an ϵ -rank approximation. In section 3.4, we will also apply this technique to our generated fields. The resulting norm will no longer be exact, but the error can be made arbitrarily small by adjusting l and ϵ .

The spatial coarsening process is intuitive and easy to implement. It is not, however, the most efficient way to reduce the size of a dataset. Dimension reduction refers to discarding modes which contribute little to the variance in a dataset. As mentioned in section 1.1, an SVD is precisely the procedure used to find modes which explain as much variance as possible. Discarding the smallest singular values, therefore, gives the best lower rank approximation [5,6]. Performing an SVD on a large dataset, however, is computationally costly. The *randomised dimension reduction* process, reviewed extensively in the article by Halko et al., is more efficient [17,18].

As depicted in figure 7, this process reduces the input matrix to a smaller square matrix of l by l . It also gives two projection matrices which can bring the rows and columns of this smaller matrix back to the bases of the original input. It is a randomised procedure to get an ϵ -rank approximation and, therefore, the error will be at the order of the size of the largest truncated singular value [5,17]. Our *Jupyter Notebook* provides more details [4].

$$A \approx H \begin{bmatrix} L \\ W^T \end{bmatrix} = H \begin{bmatrix} \tilde{U} \\ S \\ \tilde{V}^T \end{bmatrix} W^T = \begin{bmatrix} U \\ S \\ V^T \end{bmatrix}$$

Figure 7. Approximate SVD via dimension reduction

In the review article by Halko et al. on randomised dimension reduction, it is suggested to oversample the reduction. This is because the error introduced in the process is of the same order as the size of the last sampled singular value. If one is interested in the k dominant modes, reducing to a $k + l$, for some small l , rank approximation will ensure the first k modes are approximated quite well. Indeed, as seen below, the more modes one is interested in, the larger the difference compared with the original matrix.

3.5. Case Study using ERA-5 Data

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

3.6. Approximate Product SVD via Coarsening

We can also coarsen two different fields before analysing their cross-covariance matrix. Figure 8 shows the percentage error for various generated matrices. Due to the multiplication step in this analysis, the typical error as a result of coarsening is larger than before. As expected, the level of autocorrelation plays an important part, with larger α 's leading to less error. The amount of error during the coarsening process will likely also depend on the similarity between the two datasets. We leave this aspect for further research.

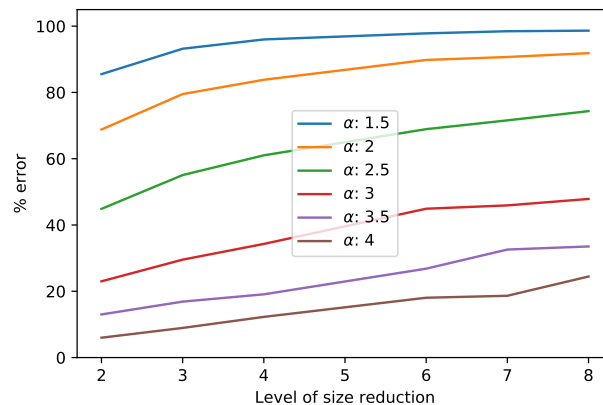


Figure 8. Error after coarsening the product of two fields

The coarsening process can speed up the calculation of the SVD, and there are additional benefits. When a target level of accuracy is set and there is an a priori estimate of the level of autocorrelation of the fields, the data collection process can be optimised. Knowing in advance at what resolution to gather data can help save time. Furthermore, in domains where satellite data is used, datasets are often not very detailed because the imaging resolution is low. Unlike local analyses of developed countries, where high resolution data is becoming more accessible, for continental or global analyses, coarse spatial resolution data may simply be the only option.

3.7. Approximate Product SVD via Dimension Reduction

The randomised dimension reduction process can also be applied to the MCA or CCA analysis of two spatio-temporal fields. Similar to the QR product SVD, it has the advantage that the SVD is applied to a small l by l matrix, as seen in figure 9.

$$\begin{bmatrix} A & B^T \end{bmatrix} \approx \begin{bmatrix} L_A & W_A^T \\ H_A \end{bmatrix} \begin{bmatrix} W_B \\ L_B^T \end{bmatrix} \begin{bmatrix} H_B^T \\ C \end{bmatrix} = \begin{bmatrix} H_A \\ C \end{bmatrix} \begin{bmatrix} U_C \\ S_C \\ \tilde{V}_C^T \end{bmatrix} \begin{bmatrix} H_B^T \\ C \end{bmatrix} = \begin{bmatrix} U_C \\ S_C \\ V_C^T \end{bmatrix}$$

Figure 9. Approximate SVD for product of two fields

To see the effect of dimension reduction on such a matrix product, let's generate various Gaussian Random Fields and compare their cross-correlation matrix with a reduced version. Figure 10 shows that the results are terrible for fields with a small α , but high levels of autocorrelation allow for substantial savings in computation time without acquiring much error.

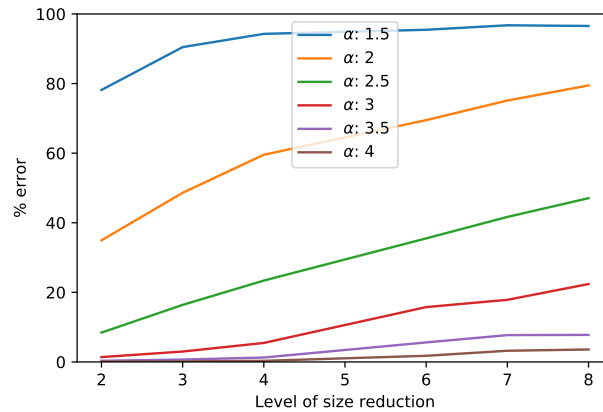


Figure 10. Error after SVD of approximated fields

Again, we performed our analysis with two generated fields which correlated highly. Whether this correlation influences the amount of error after dimension reduction is left open.

The reduction of the number of dimensions of each input dataset is actually advised by some researchers, as a method to filter out noise [19]. Especially when the number of temporal samples is small, outliers and random fluctuations could affect the result [9]. This is because any statistical analysis will choose its regression-coefficients so as to optimize the fit. It may occur that two noise-vectors in the two fields coincidentally covary and show up as dominant modes. Prefiltering can alleviate this risk.

When a dataset is too large to fit in memory, the time of transferring the matrix from storage often takes longer than the actual analysis [17]. The additional benefit of this procedure is that the randomized technique requires only a constant number of passes over the data, reducing storage communication time.

3.8. Case Study using JRA-55 Data

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

4. Discussion

4.1. Further Work

Much of the analysis here relies on a priori knowledge of the level of autocorrelation. The *Jupyter Notebook* accompanying this article includes an algorithm which estimates Moran's I based on a sample of gridpoints. This speeds up the calculation substantially for large datasets. Additional work could be placed into making this algorithm more professional and more useful to a wider audience. It may also be interesting to extend this research to fields other than GRF's. This type was chosen because its structure scale can be captured in a single parameter α . Additionally, it would be an improvement to relax the assumption that the autocorrelation in the time direction is similar to that in the spatial directions. In fact, it may even be more realistic to have different levels of autocorrelation in the various spatial directions.

Finally, note that, unlike the coarsening procedure, the dimension reduction is not applied on each spatial field for each time period, but rather on the entire spatially flattened timeseries. Therefore, the level of spatial autocorrelation may not be as important as the level of temporal autocorrelation. Further work can examine how to apply the reduction to the spatial part of the spatio-temporal fields, before it is flattened. Alternatively, 3D tensor operations could be used ??.

4.2. Conclusion

In conclusion, performing analyses at a coarse level can be beneficial when data collection is difficult. Using the randomised dimension reduction can be helpful for datasets which are too large for internal memory. And, finally, rank decomposition may speed up calculations by splitting datasets into square, full rank matrices together with left- and right orthonormal rotation matrices. Once the analysis is performed on the smaller matrix, the output can be rotated back to the original bases, often saving computation time.

Author Contributions: Conceptualization, R.Z.; Methodology, L.B.; Software, L.B.; Validation, R.G., R.Z. and E.I.; Formal Analysis, L.B.; Investigation, L.B.; Resources, R.G., R.Z. and E.I.; Data Curation, R.G., R.Z. and E.I.; Writing—Original Draft Preparation, L.B.; Writing—Review & Editing, R.G., R.Z. and E.I.; Visualization, L.B.; Supervision, R.G. and R.Z.; Project Administration, R.G.; Funding Acquisition, R.Z.

Funding: Please add: "This research received no external funding" or "This research was funded by [name of funder] grant number [xxx]." Check carefully that the details given are accurate and use the standard spelling of funding agency names at <https://search.crossref.org/funding>, any errors may affect your future funding.

Conflicts of Interest: The authors declare no conflict of interest. The founding sponsors had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, and in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

SVD	Singular value decomposition
GRF	Gaussian random field
PC	Principle component
EOF	Empirical orthogonal functions
SI-x	Extended spring indices
AVHRR	Advanced very-high-resolution radiometer
ERA-5	European reanalysis
JRA-55	Japanese 55-year reanalysis

1. Golub, G.H.; Reinsch, C. Singular Value Decomposition and Least Squares Solutions. *Numerische Mathematik* **1970**, *14*, 403–420. doi:10.1007/BF02163027.
2. Björck, Å.; Golub, G.H. Numerical Methods for Computing Angles Between Linear Subspaces. *Mathematics of Computation* **1973**, *27*, 579–594.
3. Chan, T.F. An Improved Algorithm for Computing the SVD. *ACM Trans. Math. Softw.* **1982**, pp. 72–83. doi:10.1145/355984.355990.
4. Bogaardt, L. Dataset Reduction Depending On Structure Scale. <https://github.com/phenology/>, 2018.
5. Martinsson, P.G. Randomized methods for matrix computations and analysis of high dimensional data. *ArXiv* **2016**.
6. Eckart, C.; Young, G. The approximation of one matrix by another of lower rank. *Psychometrika* **1936**, pp. 211–218. doi:10.1007/BF02288367.
7. Eshel, G. *Spatiotemporal Data Analysis*; Princeton University Press, 2011.
8. von Storch, H.; Zwiers, F.W. *Statistical Analysis In Climate Research*; Cambridge University Press, 1999.
9. Bretherton, C.S.; Smith, C.; Wallace, J.M. An Intercomparison of Methods for Finding Coupled Patterns in Climate Data. *Journal of Climate* **1992**, *5*, 541–560. doi:10.1175/1520-0442(1992)005<0541:AIOMFF>2.0.CO;2.
10. Moran, P.A.P. Notes on Continuous Stochastic Phenomena. *Biometrika* **1950**, *37*, 17–23.
11. Hubert, L.J.; Golledge, R.G.; Costanzo, C.M. Generalized Procedures for Evaluating Spatial Autocorrelation. *Geographical Analysis* **1981**, *13*, 224–233. doi:10.1111/j.1538-4632.1981.tb00731.x.
12. Rey, S. PySAL. <http://pysal.readthedocs.io>, 2009–2013.
13. Tygert, M. Suggested during personal communication, 2017.
14. Reed, B.C.; Brown, J.F.; VanderZee, D.; Loveland, T.R.; Merchant, J.W.; Ohlen, D.O. Measuring phenological variability from satellite imagery. *Journal of Vegetation Science* **1994**, *5*, 703–714. doi:10.2307/3235884.
15. Schwartz, M.D.; Ault, T.R.; Betancourt, J.L. Spring onset variations and trends in the continental United States: past and regional assessment using temperature-based indices. *International Journal of Climatology* **2013**, pp. 2917–2922. doi:10.1002/joc.3625.
16. Izquierdo-Verdiguier, E.; Zurita-Milla, R.; Ault, T.R.; Schwartz, M.D. Using cloud computing to study trends and patterns in the Extended Spring Indices. *Third International Conference on Phenology* **2015**, p. 51.
17. Halko, N.; Martinsson, P.G.; Tropp, J.A. Finding Structure with Randomness: Probabilistic Algorithms for Constructing Approximate Matrix Decompositions. *SIAM Review* **2011**, *53*, 217–288. doi:10.1137/090771806.
18. Li, H.; Kluger, Y.; Tygert, M. Randomized algorithms for distributed computation of principal component analysis and singular value decomposition. *CoRR* **2016**, abs/1612.08709.
19. Barnett, T.P.; Preisendorfer, R. Origins and Levels of Monthly and Seasonal Forecast Skill for US surface Air Temperatures Determined by Canonical Correlation Analysis. *Monthly Weather Review* **1987**, *115*, 1825–1850. doi:10.1175/1520-0493(1987)115<1825:OALOMA>2.0.CO;2.