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 20, 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 examines several such techniques in combination with 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, the field contains large scale structure and there is a high level of autocorrelation. Datasets do not need 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. As the main result, 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. Finally, these techniques and predictions, based on simulated data, 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 procedures which exploit autocorrelation and rank decomposition to analyse data in an efficient manner. Even though these techniques are not novel, a review is beneficial for domains less familiar with analysing large datasets [1–3]. In this article, we outline when certain techniques can be useful and we make predictions about the error incurred in the approximations based on the level of autocorrelation of the input data.

To arrive at these predictions, *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, based on simulated data, are verified using real-world geospatial datasets. The reported results come from calculations performed in an accompanying *Jupyter Notebook* which can be found in the online supplementary material [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; for instance, a group of n individuals who report scores on m questions or 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 row- and the column basis, which can also be changed via a rotation. A clever basis to rotate into is one where the product of the first row- and column basis vectors explains as much of the variance in the dataset as possible. Subsequent pairs of basis vectors, known as *modes*, explain as much of the remaining variance as possible while being orthogonal to all previous modes. 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 expected 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. It is rank deficient if $r < \min(m, n)$.

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 which results in a set of orthonormal column basis vectors U, a list of coefficients s and a set of row 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]. This means that the *information* contained in the data is limited by the noise level. 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 $l < r$, without losing much information. Following the Eckart-Young theorem, the best 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]. Setting a threshold ϵ, the dataset is approximate rank deficient if some singular values fall below ϵ. Then, it has an ϵ-rank of l and the spectral norm of the difference with its approximation is at most ϵ [5].

Thus, 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 achieved 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 takes up 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.

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

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.

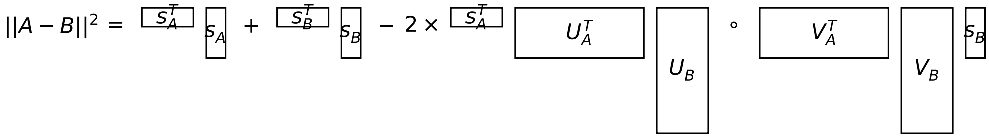


Figure 1. Visualising the calculation of the norm of a difference via SVD's

A slight variation of this concept applies to approximate SVD's. As mentioned in section 1.1, real-world data are gathered by machines with finite precision. Realising that all fields contain some level of noise, it may not be necessary to determine the mathematically exact SVD. An approximation can provide an equivalent amount of information. Then, it is no longer *efficient* to work with the full dataset, but it makes sense to reduce the data to the point where the error due to reduction is around the noise level. This requires knowledge of the precision of the data gathering equipment and setting a desired level of accuracy for the final output.

Reducing the size of a dataset can speed up and SVD in two ways. The obvious effect is that the number of calculations is decreased. The second benefit of size reduction is related to reading data. For very large datasets, which do not fit in *RAM*-memory and are saved out-of-core, reading the data becomes a major factor in determining the speed of the analysis [7]. Size reduction can either enable data to be placed into internal memory or will reduce the amount of bytes the computer has to read.

1.3. Decision Tree

In this section, we help researcher identify situations where different SVD approaches are beneficial. There are several reasons for performing an SVD analysis. When applied to 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. Researchers typically compare two such datasets using *Maximum Covariance Analysis* (MCA) and *Canonical Correlation Analysis* (CCA) to find patterns which occur frequently and simultaneously [8,9]. Such a pattern, or mode, is a combination of a row- and a column basis vector. One technique to determine 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 row- and the column vector covary maximally, whereas in a CCA, they correlate maximally [10].

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 exists which provide an approximate answer: coarsening and dimensionality reduction. These will be discussed in section 3.1 and section 3.2.

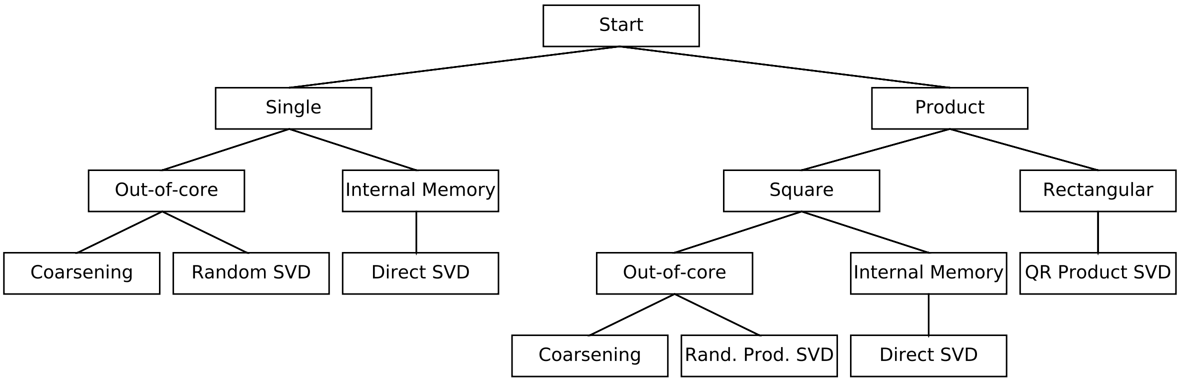


Figure 2. Decision tree describing the possible SVD techniques

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 3.7. Square matrices small enough to fit in memory can be analysed directly. Variations of coarsening and dimensionality reduction can assist analyses of larger datasets, discussed in section 3.4 and section 3.5.

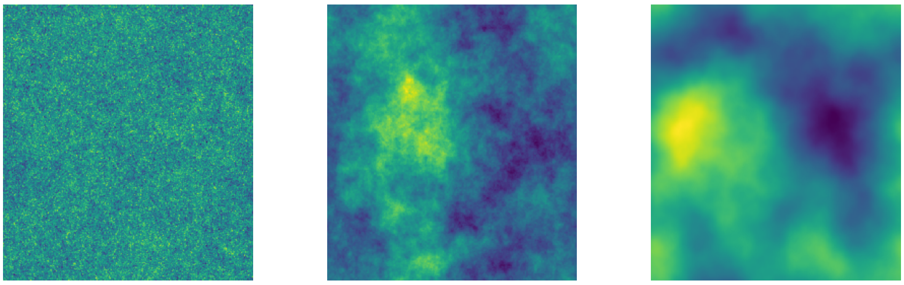
2. Materials and Methods

In domains such as climate science, datasets are typically spatio-temporal fields e.g. of global temperatures. In such fields, values vary slowly and neighbouring points are not entirely independent of one another, neither in space nor in time [8]. 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 establish 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 fields.

2.1. Spatio-Temporal Fields

As simulated spatio-temporal fields, real-valued *Gaussian Random Fields* (GRF's) 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.



(a) $\alpha = 1$ **(b)** $\alpha = 3$ **(c)** $\alpha = 5$

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 level of autocorrelation in all directions, determined by parameter α.

2.2. Autocorrelation

In the geosciences, there are additional measures of spatial autocorrelation [8,9]. One frequently used is Moran's I [11–13]. 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.

One can also devise an autocorrelation measure from the singular values of a dataset. 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. A power law can be fitted to this list, with an exponent which we call β.

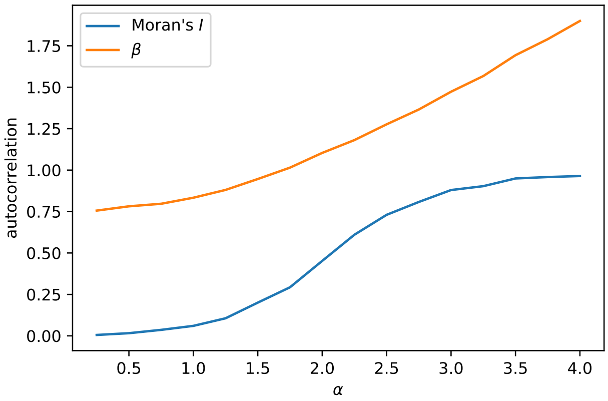


Figure 4. Measures of autocorrelation as a function of α

A high α implies a high Moran's I and a high β, which, in turn, implies some singular values are close to zero. Therefore, spatial fields with high levels of autocorrelation are described by matrices which are approximate rank deficient. This allows for data reduction without losing much information.

3. Results

This section lists several SVD related implementations to analyse large datasets efficiently by exploiting autocorrelation and rank deficiency. Additional methods exist, though the ones here cover three main benefits: the product SVD via QR decomposition provides an exact result, dimensionality reduction provides the best possible approximation for any level of reduction, while coarsening is an easy-to-implement method. Which technique to use when is described in the decision tree of figure 2.

3.1. Approximate SVD of a Single Matrix via Coarsening

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. Here, we coarsen two dimensional GRF's.

Figure 5 shows the error in a coarsening process for matrices of various α's and for different coarsening window 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]. Other measures of similarity could have been used, such as the correlation between the two datasets, but this is left for future research. Note that coarsening a two dimensional field at level 5 reduces the dataset by 25 times. For fields with high autocorrelation, e.g. $\alpha = 3$, the coarsened version differs by less than 10% from the original.

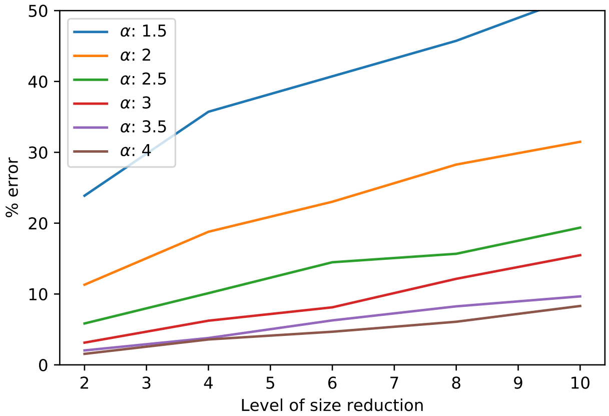


Figure 5. Error in the SVD of a coarsened spatial field for various α's

3.2. Approximate SVD of a Single Matrix via Dimensionality Reduction

An alternative method of reducing the size of a matrix is via dimensionality reduction. During dimensionality reduction, the number of basis-vectors of a matrix is truncated to the l most important ones, similar to finding an ϵ-rank approximation. Remember from section 1.1 that matrices with low rank require less storage space and fewer flops for vector multiplications. A random algorithm exists which performs an approximate rank decomposition of a large matrix efficiently. This algorithm is reviewed extensively elsewhere, here we merely examine its performance on datasets with autocorrelation [7,14]. One of its benefits is that this algorithm requires only a constant number of passes over the data. For large matrices, stored out-of-core, this reduces reading time. Additionally, the incurred error in the approximation can be made arbitrarily small by adjusting l and ϵ, giving the researcher full control over the balance between computation cost and accuracy.

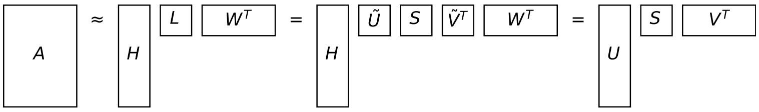


Figure 6. Visualising the calculation of an approximate SVD via dimensionality reduction

Figure 6 depicts the calculation in this process, which first reduces the input matrix to a smaller square matrix of l by l. It also provides two projection matrices which rotate the rows and columns of this smaller matrix back as close as possible to the bases of the original input. Subsequently, the SVD is applied to the small l by l matrix, which results in a fast and efficient approximation of the matrix's decomposition with an error of the order of the size of the largest truncated singular value [5,7].

The calculations in the accompanying *Jupyter Notebook* show that the errors induced by the randomised SVD procedure are very small, even for high levels of reduction [4]. The technique performs much better than coarsening. The coarsening procedure has several advantages though. For one, it is intuitive and the results are easy to interpret. It is also trivial to implement. Additionally, different coarsening levels can be applied to different directions. This is especially advantages when directions have different levels of autocorrelation or are recorded at different resolutions. Finally, the predictions of figure 5 can help researchers determine at what resolution to gather their data in the first place. In domains were 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.

3.3. Case Study of an SVD of a Single Matrix

Coarsening and dimensionality reduction are particularly useful for spatial fields with high levels of autocorrelation. As an example of this, we examine humidity and cloud cover data from the ERA5 datasets for a single time period. ERA5 is an atmospheric reanalysis of the global climate using high spatial resolution forecasts, produced by combining models with observations [15]. It contains estimates of atmospheric parameters such as air temperature, pressure and wind at different altitudes.

Clearly, the ERA5 humidity and cloud cover fields are not GRF's. Nonetheless, we can get an idea of the accuracy of an SVD after data reduction if we estimate the levels of autocorrelation. This can verify whether simulated GRF's are reasonable representations of real-world datasets. The humidity field has a Moran's $I \approx 0.98$, while the cloud cover data shows less structure with a Moran's $I \approx 0.82$. The estimations for α were unreliable, though figure 4 can help us translate the measures and suggests the fields are equivalent to GRF's with an $\alpha \sim 3.5$ and $\alpha \sim 2.5$, respectively.

The coarsening predictions of figure 5 indicate the first field is expected to incur errors around a few percent for size reductions between 2 and 8, while for the second field we should see errors between 5% and 15%. The calculations in the *Jupyter Notebook* in the supplementary material show that this prediction is fairly accurate, perhaps slightly pessimistic. For the dimensionality reduction, the errors are, as expected, very small; below 1%. If a researcher cares most about performance, dimensionality reduction is the best option.

3.4. Approximate Product SVD of Square Matrices via Coarsening

In real-world applications, researchers 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 two spatio-temporal fields. Take the input datasets with the various spatial gridpoints as rows and the sample of recorded values over time as columns, multiplying these gives the cross-covariance matrix. While section 3.1 dealt with coarsening a single spatial field, we can also coarsen two fields before analysing their cross-covariance matrix. Figure 7 shows the percentage error for various generated spatio-temporal fields. Note that only the spatial directions are coarsened in our calculation. This is because the time direction gets consumed in the matrix product of the MCA or CCA and coarsening it will not speed up the SVD. Coarsening two spatial directions means each field is reduced by the square of the coarsening level, while the cross-covariance matrix is reduced by this level to the power 4. As a result, the typical error in this product is larger than for the single field, though the speed up is also substantial. Clearly, the level of autocorrelation plays an important part, with larger α's leading to less error. Preliminary work shows the amount of error also depends on the similarity between the two spatio-temporal fields, though we leave deeper investigation of this aspect for further research.

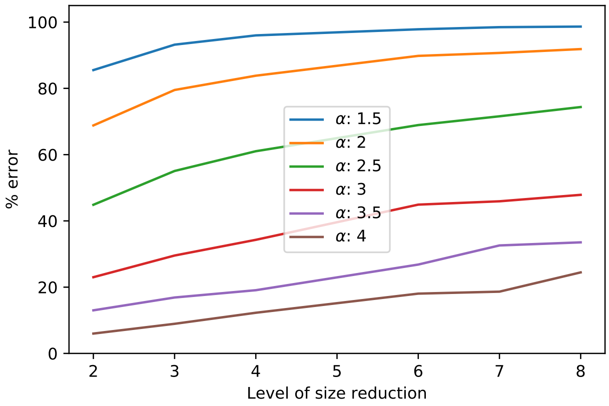


Figure 7. Error in the SVD of the product of two coarsened fields for various α's

3.5. Approximate Product SVD of Square Matrices via Dimensionality Reduction

The randomised dimensionality reduction process can be applied to two spatio-temporal fields before they are multiplied into the cross-covariance matrix. Similar to the QR decomposition of section 3.7, it has the advantage that the SVD is applied to a small l by l matrix. This calculation is visualised in figure 8. In the first step, the input datasets are decomposed using an algorithm which is reviewed extensively elsewhere [7,14]. Subsequently, the SVD is performed on the small, inner matrix.

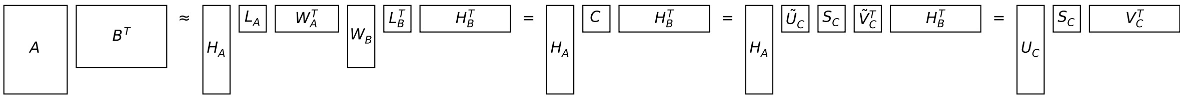


Figure 8. Visualising the calculation of an approximate SVD for the product of two fields using dimensionality reduction

After generating various GRF's, the reduced cross-covariance matrix is compared with the original. Figure 9 shows that the results are quite bad for fields with a small α, but high levels of autocorrelation allow for substantial savings in computation time without incurring much error. When performing an MCA or CCA on a spatio-temporal field, the spatial directions are flattened and some of the spatial autocorrelation is lost. This partially explains why the error is substantial for low α. Again, we performed our analysis with two generated fields which correlated to some degree. How this correlation influences the amount of error after dimensionality reduction is left for further research.

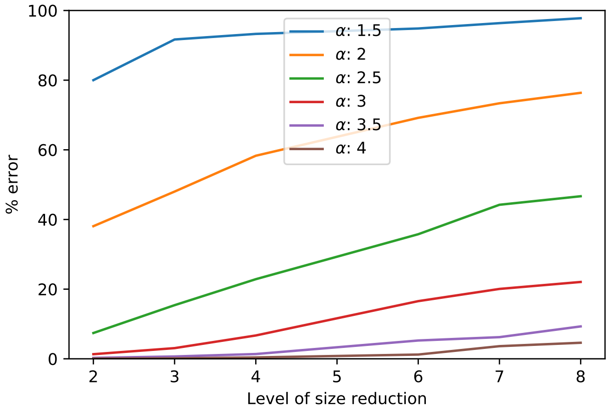


Figure 9. Error in the SVD of the product of two reduced fields for various α's

The reduction of the number of dimensions of each input dataset before an MCA or CCA is actually advised by some researchers, as a method to filter out noise [16]. Especially when the number of temporal samples is small, outliers and random fluctuations could affect the result [10]. 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.

3.6. Case Study of a Product SVD of Square Matrices

The JRA55 data is an atmosphere reanalysis product which includes quantities such as humidity, pressure and temperature [17]. Recently, these quantities were used to determine the total meridional energy transport and latent heat, measures important to understand the global climate [18]. As an example of our reduction techniques for matrix products, we are using a Mercator projection of the energy transport and latent heat, recorded monthly from 1979 to 2015. Although the spatially flattened matrices are not completely square, their high resolution in the time direction make them substantially less rectangular than the phenology data from section 3.8. Therefore, this serves as a good use case for the coarsening and dimensionality reduction techniques for a square product SVD.

The energy field has a Moran's $I \approx 0.93$, while the latent heat field has a Moran's $I \approx 0.86$. The estimations for α were unreliable, though figure 4 can help us translate the measures and suggests the fields are equivalent to GRF's with an $\alpha \sim 3.0$ and $\alpha \sim 2.5$, respectively. The analyses of section 3.4 and of section 3.5 showed that, for such α levels, the errors should be quite high. In contrast, we found that coarsening the fields before applying an SVD on their product merely resulted in an error between 8% and 16%. For the dimensionality reduction technique, as well, the predictions overstated the observed error, which were around a few percent, even for high levels of size reduction.

3.7. Exact Product SVD of Rectangular Matrices via QR Decomposition

In an MCA and CCA, an SVD is performed on the product of two spatio-temporal fields to find patterns which occur frequently and simultaneously [8,9]. For highly rectangular datasets, when there are many spatial gridpoints but few temporal samples, the resulting cross-covariance matrix is inefficiently large and obviously rank deficient. Performing a rank decomposition on each dataset before multiplying them allows the SVD to be calculated in an efficient manner [3,19]. Figure 10 depicts the calculation in this technique. Using the QR decomposition, the input datasets are first transformed into two square, full rank matrices together with rectangular orthonormal basis vectors. The SVD is then performed on the product of the small, square matrices, giving a mathematically identical result to the full SVD while never forming an unnecessarily large, intermediate matrix.

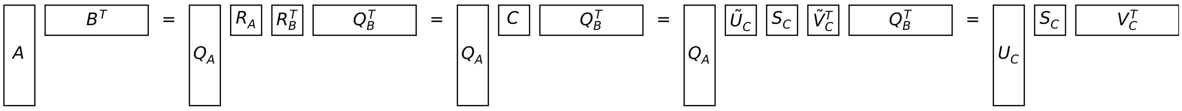


Figure 10. Visualising the calculation of the exact SVD of a product via QR decomposition

3.8. Case Study of a Product SVD of Rectangular Matrices

Let's apply the QR decomposition technique to phenological datasets. Phenology is the science that studies 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 [20]. Additionally, we use the *Extended Spring Indices* (SI-x), which are a suite of models that transform daily temperatures into consistent phenological metrics [21]. In particular, we take a version of the Bloom index which was recently generated for the US by adapting the SI-x models to a cloud computing environment [22]. Both datasets span from 1989 to 2014 and have a 1km^2 spatial resolution, meaning there are far fewer time periods than spatial gridpoints giving highly rectangular matrices. In fact, each dataset contains 30 million rows and 26 columns and is about 3.2GB large. Their cross-covariance would be 30 million rows by 30 million columns and about 3.7PB in size. Luckily, this product matrix need not be created to perform the SVD.

For the SVD via QR decomposition, the autocorrelation measures have no effect because this technique provides a mathematically exact result. Indeed, the *Jupyter Notebook* in the supplementary material, as well as work being prepared for publication, shows that this technique provides the full SVD of the cross-covariance matrix for these datasets in a matter of seconds, without ever exceeding the *RAM*-memory [4,23].

4. Discussion

4.1. Further Work

Much of the analysis here relies on knowledge of the level of autocorrelation, which may be difficult to determine for large datasets. 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 compared with the full calculation. Further work could be placed into making this algorithm more professional and more user friendly. Additional areas of research include relaxing the assumption that the autocorrelation in the time direction is similar to that in the spatial directions. In fact, it is more realistic to allow for different levels of autocorrelation in all directions and to have a version of Moran's I which can estimate these values.

Furthermore, a warning about autocorrelation and standardisation. In MCA's, the timeseries of each spatial gridpoint is centred about its mean and in CCA's, each gridpoint is standardised. These operations destroy much of the spatial autocorrelation, as it can affect neighbouring cells differently. When researcher choose the coarsening technique, this should occur before any additional data processing steps.

Unlike the coarsening procedure, the dimensionality 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. Alternative solutions, which retain the spatial structure, may include 3D tensor operations such as *Higher-Order Singular Value Decomposition* (HOSVD) [24].

4.2. Summary

In summary, randomised dimensionality reduction works best for datasets which are too large for internal memory. It requires only a constant number of passes over the data, which decreases storage reading time. It also allows the researcher to balance computation cost with accuracy, by tuning the algorithm's parameters. Performing analyses at a coarse level can be beneficial when data collection is difficult and provides an intuitive and easy-to-implement alternative. These techniques require at least some autocorrelation in the fields, which results in rank deficient datasets. In general, rank decompositions can speed up calculations by splitting datasets into smaller matrices of full rank. For rectangular matrices, this can give a mathematically exact result. Once the analysis is performed on the smaller matrix, the output can be rotated back to the original bases, saving memory usage and 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: This research was funded by the Nederlandse Organisatie voor Wetenschappelijk Onderzoek (NWO).

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 function
SOS	Start of season
SI-x	Extended spring indices
AVHRR	Advanced very-high-resolution radiometer
ERA5	European fifth generation reanalysis
JRA55	Japanese 55-year reanalysis
HOSVD	Higher-order singular value decomposition

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 techniques to speed up svd analyses. 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. 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.
8. Eshel, G. *Spatiotemporal data analysis*; Princeton University Press, 2011.
9. von Storch, H.; Zwiers, F.W. *Statistical analysis in climate research*; Cambridge University Press, 1999.
10. 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.
11. Moran, P.A.P. Notes on continuous stochastic phenomena. *Biometrika* **1950**, *37*, 17–23.
12. 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.
13. Rey, S. PySAL. http://pysal.readthedocs.io, 2009–2013.
14. Li, H.; Kluger, Y.; Tygert, M. Randomized algorithms for distributed computation of principal component analysis and singular value decomposition. *CoRR* **2016**, *abs/1612.08709*.
15. Dee, D.P.; Uppala, S.M.; Simmons, A.J.; Berrisford, P.; Poli, P.; Kobayashi, S.; Andrae, U.; Balmaseda, M.A.; Balsamo, G.; Bauer, P.; Bechtold, P.; Beljaars, A.C.M.; van de Berg, L.; Bidlot, J.; Bormann, N.; Delsol, C.; Dragani, R.; Fuentes, M.; Geer, A.J.; Haimberger, L.; Healy, S.B.; Hersbach, H.; Holm, E.V.; Isaksen, L.; Kallberg, P.; Köhler, M.; Matricardi, M.; McNally, A.P.; Monge-Sanz, B.M.; Morcrette, J.J.; Park, B.K.; Peubey, C.; de Rosnay, P.; Tavolato, C.; Thepaut, J.N.; Vitart, F. The ERA-Interim reanalysis: configuration and performance of the data assimilation system. *Quarterly Journal of the Royal Meteorological Society* **2011**, *137*, 553–597. doi:10.1002/qj.828.
16. 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.
17. Kobayashi, S.; Ota, Y.; Harada, Y.; Ebita, A.; Moriya, M.; Onoda, H.; Onogi, K.; Kamahori, H.; Kobayashi, C.; Endo, H.; Miyaoka, K.; Takahashi, K. The JRA-55 reanalysis: general specifications and basic characteristics. *Journal of the Meteorological Society of Japan* **2015**, *93*, 5–48. doi:10.2151/jmsj.2015-001.
18. Liu, Y.; Attema, J.; Moat, B.; Hazeleger, W. Synthesis and evaluation of historical meridional heat transport from midlatitudes towards the arctic. *Climate Dynamics* **2018**, *Submitted*.
19. Tygert, M. Suggested during personal communication, 2017.

20. 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.
21. 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.
22. 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.
23. Zurita-Milla, R.; Bogaardt, L.; Izquierdo-Verdiguier, E.; Gonçalves, R. Analyzing the cross-correlation between the extended spring indices and the AVHRR start of season phenometric. *EGU General Assembly: Geophysical Research Abstracts* **2018**.
24. Tucker, L.R. The extension of factor analysis to three-dimensional matrices. In *Contributions to mathematical psychology*; Gulliksen, H.; Frederiksen, N., Eds.; Holt, Rinehart and Winston, 1964; pp. 110–127.