Size Reduction Depending on Structure Scale

Laurens Bogaardt
Netherlands eScience Center
l.bogaardt@esciencecenter.nl

Romulo Goncalves
Netherlands eScience Center
r.goncalves@esciencecenter.nl

Raul Zurita-Milla
University of Twente
r.zurita-milla@utwente.nl

Emma Izquierdo-Verdiguier
University of Twente
e.izquierdoverdiguier@utwente.nl

ABSTRACT

The analysis of large datasets can be time consuming and costly. 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 structure of the field has a large scale. 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 finds a relation between performance and the scale of the structure described by the input datasets.

ACM Reference Format:
Laurens Bogaardt, Romulo Goncalves, Raul Zurita-Milla, and Emma Izquierdo-Verdiguier. 2018. Size Reduction Depending on Structure Scale. In *Proceedings of SSDBM Conference (SSDBM).* ACM, New York, NY, USA, Article x, 4 pages. https://doi.org/xxx

1 INTRODUCTION

This article looks at several analysis techniques, such as the *singular value decomposition* (SVD), which can be hard to apply when working with large datasets. We suggest a number of alternative implementations that may be of use when researchers have an idea about the structure scale of their data and about the desired level of accuracy. We will generate various datasets and exploit autocorrelation and rank decomposition to analyse the data in an efficient manner. The reported results come from calculations performed in an accompanying *Jupyter Notebook* [3]. Note that the code was not optimised for speed, but merely serves to illustrate the procedures discussed here. These techniques are by no means novel [2, 5, 8]. However, there are domains which are less familiar analysing large datasets, so a review may be beneficial.

SSDBM, 2018, Italy

ACM ISBN xxx…$x
https://doi.org/xxx

1.1 Matrix Size and Rank Decomposition

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. Similarly, these bases can be changed via a rotation. Then, the coefficients will also change, leaving the resulting matrix untouched. A clever basis to rotate into is one where the basis vectors are orthonormal and each subsequent set of left- and right basis vectors explains as much of the remaining variance in the dataset as possible. Such basis vectors are called *Principle Components* or *Empirical Orthogonal Functions* and they may be found via an SVD of the matrix.

If a rotation can be found in which some coefficients become zero, the matrix can be described by fewer basis vectors 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 it into an m by r matrix multiplied by an r by n one. Furthermore, we can choose the first factor to be an orthonormal matrix which induces a rotation, i.e. a change-of-basis. Then, the second factor captures the 'action' of the matrix, written in the new bases. It is this second matrix, often smaller than the original, which is most relevant for further analyses. 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, called singular values, will be zero.

As noted by Martinsson, the mathematical rank r of a dataset is not relevant in practice because the data originate from devices with finite precision [12]. 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

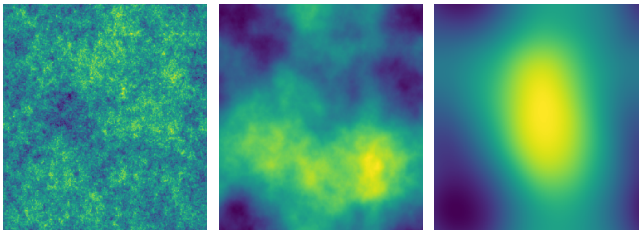
is said to be approximate rank deficient if some singular values fall below ϵ. Then, it has an ϵ-rank of l and the norm of the difference with its l-rank approximation is at most ϵ [12].

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 [12]. 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 Spatial Fields

In domains such a climate science and phenology, datasets are typically spatial fields, e.g. of temperature. In these 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 has large scale structure. Such redundancy means the dataset 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. In particular, we will concern ourselves with fields which combine some level of autocorrelation with some randomness. Real-valued *Gaussian Random Fields* are particularly useful because their structure scale can be captured in a single parameter, as shown in figure 1. For two dimensional spatial fields, rotational invariance is assumed. Then, the spectrum of such fields 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.



(a) $\alpha = 1$ **(b)** $\alpha = 2$ **(c)** $\alpha = 4$

Figure 1: Gaussian Random Fields for various α's

In spatial data analysis, other measures of autocorrelation are used [7, 16]. These include Moran's I and the Γ index [10, 13, 14]. Finally, 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. Figure 2 plots them as a function of α for various generated Gaussian Random Fields.

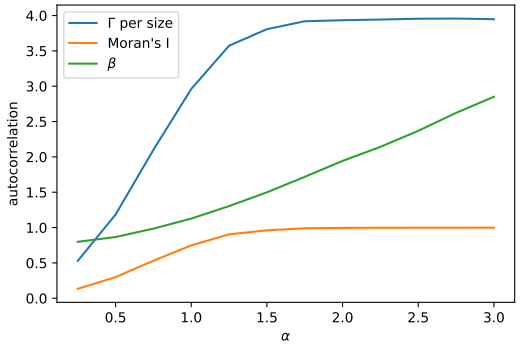


Figure 2: Measures of autocorrelation as a function of α

1.3 Spatio-Temporal Fields

In many real-world applications, the analysis of a field does not only involve a single time period but includes data over multiple weeks, months or years. Then, we are interested in finding patterns which occur frequently. The *maximum covariance analysis* (MCA) and *canonical correlation analysis* (CCA) examine the cross-covariance matrix of two datasets and find patterns which occur frequently and simultaneously [7, 16]. 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 [4].

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 TECHNIQUES

This section lists four SVD related implementations to analyse large datasets by exploiting autocorrelation and rank deficiency.

2.1 Exact Norm of Difference via SVD

One often wants to find the norm of the difference between two fields. This can be done by subtracting one matrix from the other and summing the square of the elements. However, for large matrices, this may be inefficient, especially when they are rank deficient and when their SVDs are already known.

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 3 shows that this procedure can determine the norm in an efficient manner, provided the number of singular values is small.

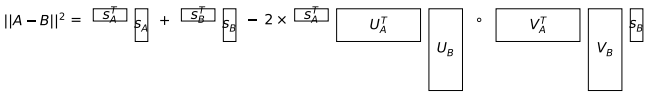


Figure 3: Exact norm of difference via SVD

2.2 Exact 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 cross-covariance 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 [5, 15]. As shown in figure 4, the result is mathematically identical to the full SVD, which means that the difference will be at machine-precision.

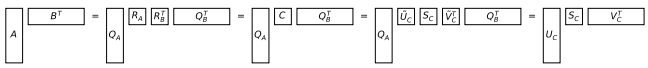


Figure 4: Exact SVD of a product via QR decomposition

2.3 Approximate SVD via Spatial 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.

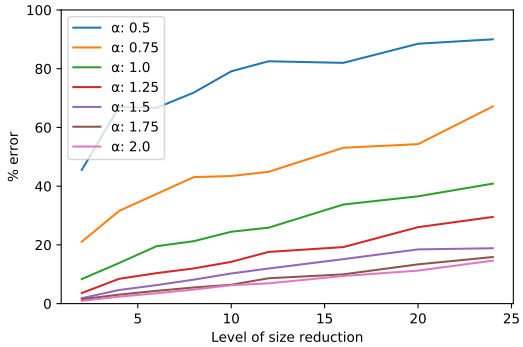


Figure 5: Error after coarsening a spatial field for various α's

Figure 5 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 [3]. Note that a field with high autocorrelation ($\alpha = 2$) differs only by a few percent from one 25 times smaller (level = 5).

We can also coarsen two different fields before analysing their cross-covariance matrix. Figure 6 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.

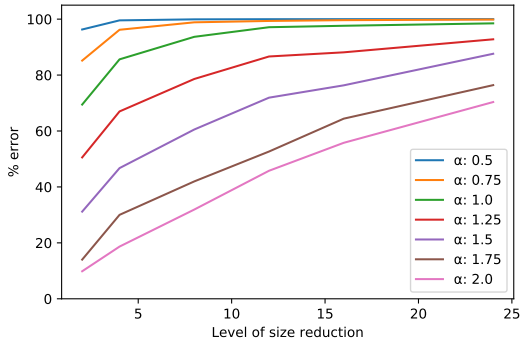


Figure 6: Error after coarsening the product of two fields

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 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.

2.4 Approx. SVD via Dimension Reduction

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 [6, 12]. 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 [9, 11].

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 [9, 12]. Our *Jupyter Notebook* provides more details [3].

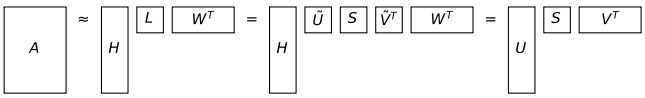


Figure 7: Approximate SVD via dimension reduction

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 8.

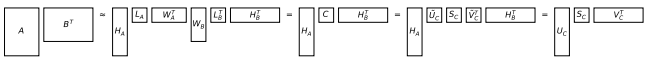


Figure 8: Approximate SVD for product of two fields

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 9 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.

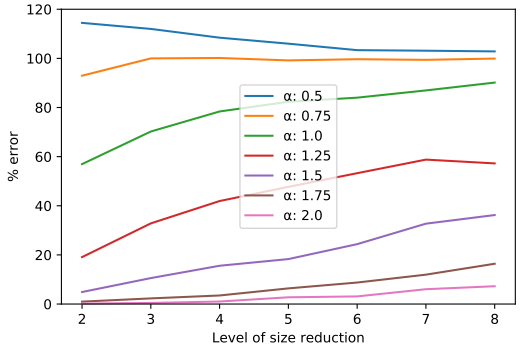


Figure 9: Error after SVD of approximated fields

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 [1]. Especially when the number of temporal samples is small, outliers and random fluctuations could affect the result [4]. 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 [9]. 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 CONCLUSION AND FURTHER WORK

Much of the analysis here relies on a priori knowledge of the level of autocorrelation. If datasets are large, it would be nice to have an algorithm to estimate Moran's I, or another measure of autocorrelation, using only a small sample of the data. It may also be interesting to extend this research to fields other than the Gaussian Random Field. 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 North-South and in the East-West direction.

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 if better results are obtained when the dimension reduction is applied to the spatial part of the spatio-temporal fields, before it is flattened.

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.

REFERENCES

[1] T. P. Barnett and R. Preisendorfer. 1987. Origins and Levels of Monthly and Seasonal Forecast Skill for US surface Air Temperatures Determined by Canonical Correlation Analysis. *Monthly Weather Review* 115, 9 (1987), 1825–1850. https://doi.org/10.1175/1520-0493(1987)115<1825:OALOMA>2.0.CO;2

[2] Åke Björck and Gene H. Golub. 1973. Numerical Methods for Computing Angles Between Linear Subspaces. *Math. Comp.* 27, 123 (1973), 579–594.

[3] Laurens Bogaardt. 2018. Size Reduction Depending On Structure Scale. https://github.com/phenology/. (2018).

[4] Christopher S. Bretherton, Catherine Smith, and John M. Wallace. 1992. An Intercomparison of Methods for Finding Coupled Patterns in Climate Data. *Journal of Climate* 5, 6 (1992), 541–560. https://doi.org/10.1175/1520-0442(1992)005<0541:AIOMFF>2.0.CO;2

[5] Tony F. Chan. 1982. An Improved Algorithm for Computing the SVD. *ACM Trans. Math. Softw.* (1982), 72–83. https://doi.org/10.1145/355984.355990

[6] Carl Eckart and Gale Young. 1936. The approximation of one matrix by another of lower rank. *Psychometrika* (1936), 211–218. https://doi.org/10.1007/BF02288367

[7] Gidon Eshel. 2011. *Spatiotemporal Data Analysis.* Princeton University Press.

[8] G. H. Golub and C. Reinsch. 1970. Singular Value Decomposition and Least Squares Solutions. *Numer. Math.* 14 (1970), 403–420. https://doi.org/10.1007/BF02163027

[9] N. Halko, P. G. Martinsson, and J. A. Tropp. 2011. Finding Structure with Randomness: Probabilistic Algorithms for Constructing Approximate Matrix Decompositions. *SIAM Rev.* 53, 2 (2011), 217–288. https://doi.org/10.1137/090771806

[10] L. J. Hubert, R. G. Golledge, and C. M. Costanzo. 1981. Generalized Procedures for Evaluating Spatial Autocorrelation. *Geographical Analysis* 13, 3 (1981), 224–233. https://doi.org/10.1111/j.1538-4632.1981.tb00731.x

[11] Huamin Li, Yuval Kluger, and Mark Tygert. 2016. Randomized algorithms for distributed computation of principal component analysis and singular value decomposition. *CoRR* abs/1612.08709 (2016). http://arxiv.org/abs/1612.08709

[12] Per-Gunnar Martinsson. 2016. Randomized methods for matrix computations and analysis of high dimensional data. *ArXiv* (2016). https://arxiv.org/abs/1607.01649

[13] P. A. P. Moran. 1950. Notes on Continuous Stochastic Phenomena. *Biometrika* 37, 1/2 (1950), 17–23.

[14] Sergio Rey. 2009–2013. PySAL. http://pysal.readthedocs.io. (2009–2013).

[15] Mark Tygert. 2017. Suggested during personal communication. (18 10 2017).

[16] Hans von Storch and Francis W. Zwiers. 1999. *Statistical Analysis In Climate Research.* Cambridge University Press.