Information Loss During Size Reduction Depending On Structure Scale

Laurens Bogaardt[1], Romulo Goncalves[1], Raul Zurita-Milla[2], and Emma Izquierdo-Verdiguier[2,3]

[1]NLeSC Amsterdam, The Netherlands {*{l.bogaardt, r.goncalves}@esciencecenter.nl*}

[2]Faculty of Geo-Information Science and Earth Observation (ITC), University of Twente, the Netherlands

{{*{r.zurita-milla,e.izquierdoverdiguier}@utwente.nl*}}

[3]Image Processing Laboratory (IPL), Universitat de Valencia, Spain

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 faithfully. Using generated *Gaussian Random Fields* with various levels of spatial autocorrelation, we examine several exact and approximate analysis techniques. Our aim is to outline when certain techniques can be useful and to find a relation between performance and the scale of the structure described by the input datasets.

1. INTRODUCTION

This article looks at several techniques to analyse large datasets and at the inherent scale of the structure within the data. When a researcher has an idea about the scale of their data and has a target margin of error, this article can suggest which technique may be relevant for their analysis. The techniques discussed here are by no means novel [7, 10, 4, 16]. However, there are domains which are less familiar analysing large datasets, so a review may be beneficial.

1.1 Matrix Size and Rank Decomposition

Many datasets can be represented by a matrix of values. Take, for instance, a group of n idividuals who report scores on m different questions. Or take the temperatures at m locations, measured over n time periods. These values can be placed into 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 bases 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 a *Singular Value Decomposition* (*SVD*)of the matrix.

If one can find a rotation in which one of the coefficients becomes zero, the matrix seems to be able to be described by fewer parameters than are available. In a sense, it is underdetermined. It's internal dimension is smaller than what could have been guessed from its m by n size. This is the concept of matrix 'rank'. An m by n matrix has rank r if the rows and the columns both span a subspace of dimension r. If $r = \min(m, n)$, such a matrix is said to have full rank, 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 an m by n matrix of rank r, with $r \leq n \leq m$, we can decompose it into an m by r matrix and an r by n one. We can choose the first factor of this product to be an orthonormal matrix which induces a rotation, i.e. a change-of-basis. The second factor captures the 'action' of the matrix, written in the new bases. It is this second matrix, which is 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 the of coefficients, called singular values, will be zero.

As noted by Martinsson, the condition that a dataset has precisely rank r is not realistic in practice because the values originate from devices with finite precision [13]. Even though some singular values of a dataset are not zero, they may be close enough to zero to be considered *noise*. If we take the inherent imprecise nature of real-world datasets 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 [5]. 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.

So, we can identify three types of matrix 'sizes'. The first is the size of the full matrix, $m \times n$. Storing such a matrix requires $m \times n$ units of storage and computing the product with a vector requires $m \times n$ flops [13]. The second type is the rank decomposed version of the matrix. Storing such a matrix requires $m \times r + r \times n$ units of storage and an equal number of flops for the vector multiplication [13]. 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 such fields, values vary slowly and neighbouring points are not entirely independent of one another, neither in space nor in time citeEshel2011. In this case, there is a high level of autocorrelation and the structure of the field has a large scale. Due to such redundancy, it is likely that the dataset is rank deficient. In this article, we will examine various fields and exploit rank factorization to analyse the data in an efficient manner. The reported results come from calculations performed in a .. *Jupyter Notebook* [**?**].

In order to compare analysis techniques and to find the relation between performance and the structure scale, we need to be able to generate fields which resemble fields often encountered in real-world applications. In particular, the field we will concern ourselves with need to have some amount randomness and some level of spatial autocorrelation. Real-valued Gaussian Random Fields are particularly useful, as their structure scale can be captured in a single parameter. The power spectrum of such a field follows the power law described in equation 1 where k is the wave number and α the parameter which controls the level of autocorrelation. For 2D spatial fields, rotational invariance is assumed, such that k can be substituted by $|\vec{k}|$.

$$f = |\vec{k}|^{-\alpha} \tag{1}$$

In spatial data analysis, other measures of autocorrelation are often used [6, 17]. These include Moran's I and the Γ index [14, 9, 15]. Another measure comes from the singular values. These are related to the amount of variance in the original dataset explained by their associated mode. For fields with autocorrelation, the singular values decay quickly. One can try to fit a power law to them and estimate the exponent, which we'll call β. All the measures give an indication of the level of autocorrelation in the field and the scale of the structure represented in the data.

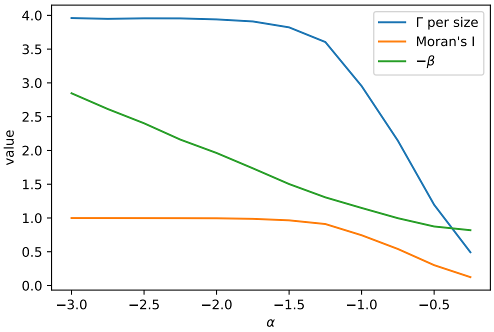

Figure 1: Various measures of autocorrelation as a function of α

When α is less negative, the Gaussian Random Field has smaller scale structure. It is closer to randomness and has less autocorrelation. As expected, this can be seen in the two measures of autocorrelation, which are now smaller.

1.3 Spatial-Temporal Fields

In many data analyses, we are interested in finding frequently occurring patterns. The *maximum covariance analysis* and *canonical correlation analysis* examine the cross-covariance matrix of two datasets and find patterns which occur frequently and simultaneously [6, 17]. 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 fields, the term SVD is often used synonymously with MCA. In an MCA, modes are found where the left- and the right-field covary maximally, wereas in an CCA, they correlate maximally [2].

In many real-world applications, the analysis of a field does not only involve a single time snapshot. In fact, it often includes data over multiple weeks, months or years, where the field over the entire time period does not change drastically. Just as there is spatial autocorrelation, there is temporal autocorrelation. In principle, there can be a different level of correlation over time than over space. However, for simplicity, we are using the same α determine the level of autocorrelation in all dimensions.

2. TECHNIQUES

This section will discuss four techniques to analyse large datasets efficiently using their singular value decomposition and by exploiting autocorrelation and rank deficiency.

2.1 Exact Norm Difference via SVD

In real-world applications, 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 calculating the norm. However, for large matrices, this may be inefficient, especially when they are rank deficient. Performing an SVD of both matrices can reduce the internal calculations.

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 2.

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

Figure 2 shows that this procedure can determine the norm in an efficient manner, provided the number of singular values is small. The result is mathematically identical to the full calculation, which means that any error will be of the order of machine-precision.

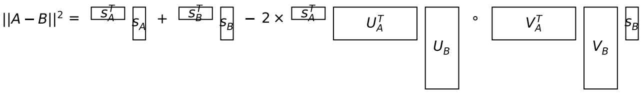

Figure 2: Exact norm difference via SVD

For very large datasets, the SVDs may be obtained using a random algorithm reviewed extensively in an article by Halko et al. [8]. The number of singular values will then be truncated to the l largest values, similar to finding an ϵ-rank approximation. In section 2.4, 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.

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 **??** rely on performing an SVD of the cross-covariance or cross-correlation matrix of the two fields. In particular, the two input datasets often have the various gridpointa as rows and will have the sample of recorded values over time as columns. Multiplying these gives the cross-correlation matrix. However, for highly rectangular matrices, when there are many spatial gridpoint but few temporal samples, the resulting cross-correlation matrix is inefficiently large. The qrProductSVD function can take such input data and perform an SVD in an efficient manner. The result is mathematically identical to the full SVD, which means that the difference will be at machine-precision.

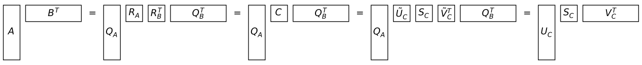

Figure 3: Caption

2.3 Approximate SVD via Spatial Coarsening

Although the qrProductSVD function works well for two rectangular matrices, sometimes the input data is large and square. Performing an SVD on such a large dataset will be time consuming and, perhaps, inefficient given the desired level of precision. When a field have large scale structure, the values of neighbouring cells do not change drastically. This is what autocorrelation means. As such, perhaps neighbouring cells can be averaged together to produce a smaller dataset which still faithfully describes the original field. The matrixToGrid function can cut a matrix into multiple smaller sections.

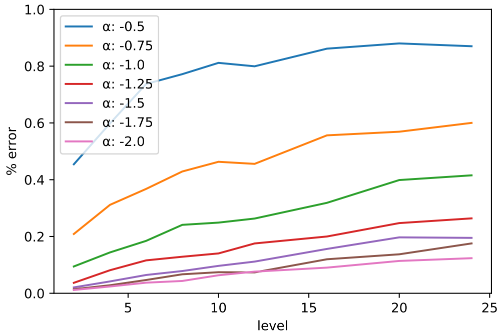

Figure 4: Caption

The testProductSpatialTemporalFieldsViaCoarsening function determines the percentage of the information lost in a coarsening process for matrices of various sizes and α's, and at various levels of coarsening. The two input matrices used here are similar, as they are generated by the same Gaussian Random Process. Therefore, they will correlate highly and the bases in which they are best described will be similar. In principle, any two datasets can be analysed and the amount of information lost during the coarsening process will likely depend on the similarity between the two datasets. This is one aspect which we do not cover here and leave for further research.

Due to the multiplication step in this analysis, the typical error due to coarsening is larger than before. As before, α plays an important part, with more negative α's leading to a less dramatic loss in information due to coarsening.

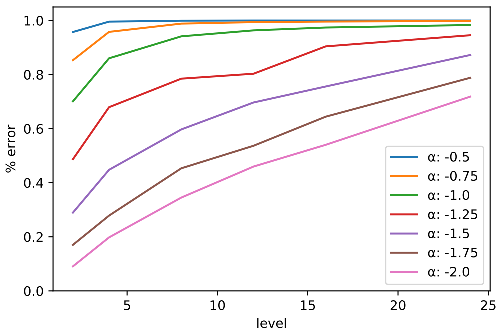

Figure 5: Caption

The additional benefit of coarsening is that data collection can be optimised.

2.4 Approximate SVD via Dimensionality 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. Dimensionality reduction refers to discarding modes which contribute little to the variance in a dataset. An SVD is precisely the procedure used to find modes which explain as much variance as possible. Discarding the smallest singular values/vectors is, therefore, the most efficient form of dimensionality reduction. Performing an SVD on a large dataset, however, is computationally costly. The Randomised Dimensionality Reduction process is far more efficient.

The reduceSizeRandomisedSquare function 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. To make the result more precise, the procedure can be repeated multiple times. The parameter i indicates how many loops are performed.

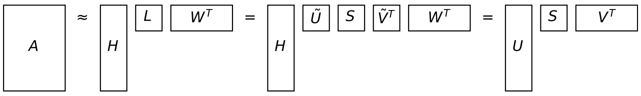

Figure 6: Caption

As seen, it is possible for some fields to be represented by matrices of much smaller sizes without losing any substantial amount of information. This is obvious when one realises the singular modes which are removed during the reduction are the smallest ones, described by the tail-end of the power law. In the review article by Halko, Martinsson and Tropp on randomised dimensionality 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.

The Randomised Dimensionality Reduction process can also be applied to the CCA or MCA analysis of two spatial-temporal fields. Similar to the QR Product SVD, it has the advantage that the SVD is applied to a small l x l matrix. The result will be an approximation, but, as we will see, can be close to the real solution.

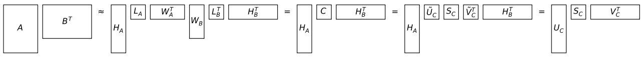

Figure 7: Caption

To see the effect of dimensionality reduction on such a matrix product, let's generate two Gaussian Random Fields and plot their corss-correlation matrix together with a reduced version. To determine precisely how much information is lost during the reduction, we should look at the variance of the datasets. The norm of the difference between the reduced matrix and the original is the amount of information lost in the reduction process.

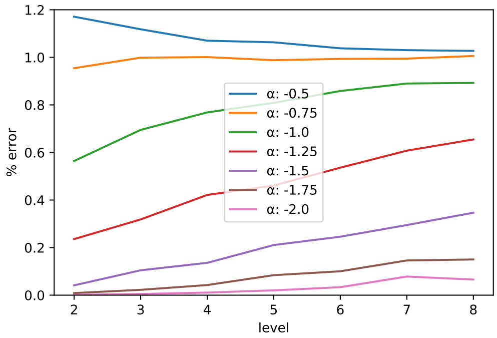

Figure 8: Caption

The testRandomisedSizeReducedMatrixProduct function determines the percentage of the information lost in a reduction process for matrices of various sizes and α's, and at various levels of reduction. The two input matrices used here are similar, as they are generated by the same Gaussian Random Process. Therefore, they will correlate highly and the bases in which they are best described will be similar. In principle, any two datasets can be analysed and the amount of information lost during the coarsening process will likely depend on the similarity between the two datasets. This is one aspect which we do not cover here and leave for further research.

Due to the multiplication step in this analysis, the typical error due to reduction is larger than before. There is an affect of temporal size on the information loss. The randomisedSquareProductSVD function is especially useful for square input matrices. To be able to do the comparisons with the full SVD quickly, we use rectangular matrices here. The larger the temporal size, the more square the input matrices. As before, the scale of the structure of the field influences the information loss. The effect in this case can be quite dramatic. Especially for more negative α, this procedure performs much better than the coarsening procedure.

Note that, unlike the coarsening procedure, the reduction is not applied on each time-slice of the spatial field, but rather on the spatially-flattened time-series. Therefore, the level of spatial autocorrelation may not be as important as the level of temporal autocorrelation. The proper analysis of this is left for further research.

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 [2]. 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. In a CCA, where the fields are standardized, this resulting mode may appear important even though it stems from noise. In an MCA, the variance of the noise will be low, so the chance that it will appear as an important mode is less [2]. One method of finding the right level of filtering is by bootstrapping/cross-validating the results [12].

3. APPLICATIONS

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.

3.1 Approximate SVD via Spatial Coarsening

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.

3.2 Approximate SVD via Dimensionality Reduction

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.

4. FURTHER QUESTIONS

It may 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. In many applications, however, the dataset does not resemble such a Gaussian Random Field.

Additionally, it would be an improvement to relax the assumption that the auto-correlation 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 x and in the y direction.

Can similar tricks be used to the generalised MCA/CCA analysis, where the input to the SVD in a concatenation of multiple cross-correlation matrices [3, 11]?

Can the dimensionality reduction be applied to the spatial part of the spatial-temporal fields, before it is flattened?

5. REFERENCES

[1] T. P. Barnett and R. Preisendorfer. Origins and levels of monthly and seasonal forecast skill for united states surface air temperatures determined by canonical correlation analysis. *Monthly Weather Review*, 115(9):1825–1850, 1987.

[2] C. S. Bretherton, C. Smith, and J. M. Wallace. An intercomparison of methods for finding coupled patterns in climate data. *Journal of Climate*, 5(6):541–560, 1992.

[3] J. D. Carroll and J.-J. Chang. Analysis of individual differences in multidimensional scaling via an n-way generalization of âĂIJeckart-youngâĂİ decomposition. *Psychometrika*, pages 283âĂŞ–319, 1970.

[4] T. F. Chan. An improved algorithm for computing the singular value decomposition. *ACM Trans. Math. Softw.*, pages 72–83, 1982.

[5] C. Eckart and G. Young. The approximation of one matrix by another of lower rank. *Psychometrika*, pages 211âĂŞ–218, 1936.

[6] G. Eshel. *Spatiotemporal Data Analysis*. Princeton University Press, 2011.

[7] G. H. Golub and C. Reinsch. Singular value decomposition and least squares solutions. *Numerische Mathematik*, 14:403–420, 1970.

[8] N. Halko, P. G. Martinsson, and J. A. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review*, 53(2):217–288, 2011.

[9] L. J. Hubert, R. G. Golledge, and C. M. Costanzo. Generalized procedures for evaluating spatial autocorrelation. *Geographical Analysis*, 13(3):224–233, 1981.

[10] ÃĚke BjÃűrck and G. H. Golub. Numerical methods for computing angles between linear subspaces. *Mathematics of Computation*, 27(123):579–594, 1973.

[11] J. R. Kettenring. Canonical analysis of several sets of variables. *Biometrika*, pages 433âĂŞ–451, 1971.

[12] R. E. Livezey and T. M. Smith. Covariability of aspects of north american climate with global sea surface temperatures on interannual to interdecadal timescales. *Journal of Climate*, 12(1):289–302, 1999.

[13] P.-G. Martinsson. Randomized methods for matrix computations and analysis of high dimensional data. *ArXiv*, 2016.

[14] P. A. P. Moran. Notes on continuous stochastic phenomena. *Biometrika*, 37(1/2):17–23, 1950.

[15] S. Rey. Pysal. `http://pysal.readthedocs.io`, 2009–2013.

[16] M. Tygert. Suggested during personal communication, 10 2017.

[17] H. von Storch and F. W. Zwiers. *Statistical Analysis In Climate Research*. Cambridge University Press, 1999.