SAMPLE SUMMARY WITH GENERATIVE ENCODING

Author1, Author2
Affiliation
Univ
City
{Author1, Author2}email@email

Author3
Affiliation
Univ
City
email@email

ABSTRACT

With increasing sample sizes, all algorithms require longer run times that scales at best logarithmically. A concept that summarises the sample space to reduce the total number of samples into a core set that can be used for regression, classification, dimensionality reduction or other tasks is introduced. This idea of summarisation is called folding - the technique for projecting data into a lower dimensional subspace, whereas unfolding projects it back into the original space. Results for a prediction task show that information is retained during folding as accuracy after unfolding is still comparable to prediction without summarisation.

***K*eywords** Sampling · Summary · Subset · Selection

1 Introduction

Large sample sizes are important in detecting meaningful effect sizes that are statistically significant. With more features, the covariance between samples contains structured information that can be analysed using machine learning methods. The limitations of methods that treat samples as being independent such as in ordinary least squares, do not account for the immense amount of structure and variance in Big Data that modern day techniques such as neural networks can. An alternative to working with Big Data is proposed by making summaries of the samples.

Sample reduction has been studied as a form of data sketching, or data transformation. Data sketching involves selecting a number of samples that are representative of other data points calculated via distance based neigbouring metrics [1], considering the spatial covering over which the points span [2], or by considering geometric segments over which the sample space covers [3]. Data transformation involves dimensionality reduction to a reduced space by accounting for minimal loss of spectral properties, carrying out analyses in the reduced space, and then projecting learned information back onto the original samples [4] [5].

Summarisation of samples takes the form of data transformation, whereby information is encoded into a reduced space, analysed, and then decoded back into the original sample space. This encoder-decoder technique follows on from the idea of Autoencoders and models the encoding as a transformation and the decoding as a decomposition. By simultaneously treating transformation and decomposition in a single model, the samples and features of the data can be generated via the learned encoding and decoding functions.

2 Method

The aim of Generative Encoding is to integrate data from several datasets, Y, X and A, into an aligned and reduced dimensional space d via a transformation. An example of Generative Encoding that is implemented here:

$$\alpha_{Y_{k,i}} Y_{i,l} \beta_{Y_{l,c}} = \alpha_{X_{k,i}} X_{i,j} \beta_{X_{j,c}} = \alpha_{A_{k,i}} A_{i,d} \beta_{A_{d,c}}$$

Here, k and c are the dimensions of the reduced dimensional encoding. The sample size i is shared between the datasets Y, X, and A. However, each dataset contains differing feature sizes l, j, d. For example, this could be of a single cell experiment where each dataset contains the profile of gene expression, protein and chromatin accessibility, respectively for Y, X and A.

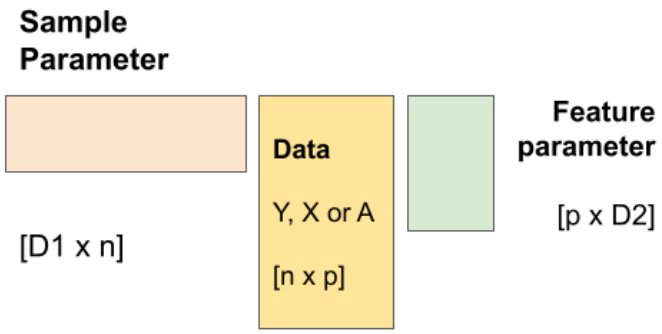

Figure 1: Encoding the dataset using two sets of parameters - an encoding of the sample space and an encoding of the feature space. Multiple sets of data can be aligned to the same space, provided there is a proportional relationship between the two encoding spaces, for example, an equality

The α parameter transforms the samples into the same reduced dimensional plane along the sample space. Furthermore, the β parameter transforms the features across the datasets into the same space. The matrix αX are the **encoded features** where the sample space has been encoded leaving only the features in the original space. The matrix $X\beta$ are the **encoded samples** where the feature space has been encoded leaving only the samples in the original space.

By simultaneously transforming the samples and features into a reduced dimensional code representation, similar to Autoencoders, Generative Encoders learn a set of parameters for both features and samples that can encode both datasets into a reduced dimensional space.

The concept here is a form of Generalised Procrustes with Canonical parameters, similar to the rigid geometric method, Procrustes analysis, which aligns a set of shapes via a rotation transform, and Canonical Correlation Analysis, which seeks to find canonical parameters which maximise the correlation transformation of two data matrices.

2.1 Further details of the Generative Encoder

Inspection of the parameters and how they are linked can be taken a further step. By expressing Y, X, and A as a function of the parameters, it is possible to extract the core common element shared between all datasets called the code. For example, let the code be Z of reduced dimension k by c.

For Y the decoded estimate is $\hat{Y}$,

$$\hat{Y} = \alpha^T_{Y_{i,k}} Z_{k,c} \beta^T_{Y_{c,l}}$$

Likewise, X can be recovered by the decoded estimate $\hat{X}$

$$\hat{X} = \alpha^T_{X_{i,k}} Z_{k,c} \beta^T_{X_{c,j}}$$

Finally, A can be recovered by the decoded estimate $\hat{A}$

$$\hat{A} = \alpha^T_{A_{i,k}} Z_{k,c} \beta^T_{A_{c,d}}$$

These expressions using the code Z can recover the original datasets, with the key important concept that the latent space shared amongst the datasets are given as Z.

Furthermore, it can be limiting to consider the datasets as matrices - they can be of any size, including three dimensional objects or tensors.

2.2 Properties of the Generative Encoder

Samples can be encoded : Sample sizes can be reduced and thus save computationl resources when running prediction or nearest-neighbour algorithms. For example, by first reducing the large sample size (several million data points) to a smaller encoded set (several thousand), an algorithm can be run on the smaller set and then decoded back into the original space.

Data can be recovered and imputed : Given all data points can be fully expressed by the model, then iterating over updates of the model while simultaneously updating any missing data points can recover the structure of the data and impute missing points. This is similar to the concept of a bootstrap, yet for prediction.

Transformations can be transferred : If of similar sample ID the sample parameter α can be transferred, if of similar feature label the feature parameter β can be transferred. For this reason, a transform learned to reduced the dimensions from a set of genes can be trained on one dataset, and transfered to the second dataset, provided the gene list is identical.

Parameters can be joined : If multiple datasets share the same sample ID, the sample parameter can be shared. Otherwise, if multiple datasets have overlapping feature labels, the feature parameter can be shared. This provides an extra level of interpretability.

2.3 Coordinate descent updates for learning parameters

The coordinate descent update to estimate the parameters, iterates through multiple steps outlined by Algorithm 1 until convergence. The full step loops over all datasets, and is run in a while loop within gcode.

Algorithm 1 Generative Encoding via Generalised Canonical Procrustes

Input: D_L Dataset for each modality L
Output: α_L sample parameters, β_L feature parameters, Z code

procedure COORDINATE DESCENT UPDATE(D_L)
  for L in D_L **do** ▷ For each dataset
    $\alpha_L = D_L^T(Z\beta_L^T)^T((Z\beta_L^T)(Z\beta_L^T)^T)^{-1}$ ▷ Update sample parameters
    $\beta_L = ((\alpha_L^T Z)^T(\alpha_L^T Z))^{-1}(\alpha_L^T Z)^T D_L$ ▷ Update feature parameters
    $Z = (\alpha_L^T\alpha_L)^{-1}\alpha_L^T D_L\beta_L^T(\beta_L\beta_L^T)^{-1}$ ▷ Update Code

2.4 Why learn in a subspace?

Learning in a reduced subspace enables a greater level of interpretability when evaluating parameters, and more computational efficiency as the parameters are learned in the smaller reduced dimension. Both of these are benefits from reducing the sample and feature size to an encoded dimension, equivalent to the encoding space.

For example, the code can represent biological information of living cells from different datasets of varying modalities (gene expression, protein etc.). Here, the sample and feature parameters represent a transform of the code of biological cell information into actual realisations of observed samples. This parameterises a full data generating process: encompassing measurement specific noise (α, sample space), relevant signal (β, feature space), and the underlying phenomena (Z, the code).

Given the dimensions of both sample and features are encoded down into a reduced dimension, it is possible to learn in the encoded space. The process involves projecting the data into this reduced dimension, after which the main code is learned in sync with the other datasets. The code allows the learning of both the dimensionally reduced sample and feature spaces after projecting to a shared space informed by the code. These sample and feature projections are parameterised by the matrices: α and β.

3 Related Work

The optimisation function outlined in Drineas, Mahoney and Muthukrishnan, is shared in this work:

$$\left\|A - CC^{\dagger}A\right\|_2^2$$

where A represents a dataset, and C represents a matrix projection and $C^{\dagger}$ indicates the inverse of the matrix projection C.

This means that the dataset A is transformed by $C^{\dagger}$ into an encoded space, which is here called a summary of A. The summary of A is then projected back into the original space via C, the matrix pseudo-inverse of $C^{\dagger}$.

In Generative Encoding, C is in place of α, and A represents the original dataset. The expression is then given as:

$$\left\|A - (\alpha^T\alpha)^{-1}\alpha^T\alpha A\right\|_2^2$$

Notice that αA represents the encoded samples, and is considered the transformation projection by α on data A into a smaller subspace. The encoded samples are considered the summary of the samples and used in any analyses that give an output from function f

$$\hat{Y} = f(\alpha A)$$

Given $\hat{Y}$, the output can be decoded back into the original subspace via a linear projection, such that:

$$(\alpha^T\alpha)^{-1}\alpha^T\hat{Y} = (\alpha^T\alpha)^{-1}\alpha^T f(\alpha A) = f((\alpha^T\alpha)^{-1}\alpha^T\alpha A) = f(A)$$

This gives $f(A)$, the function applied to the original dataset on all the samples. By running and learning the function f on the summary through αA, the learned parameters can then be reapplied to A without the need to learn on all the samples.

4 Numerical Results

The results contain two parts - a simulation to reveal the straightforward properties of sample and feature encoding, and, a showcase of sample summarisation by testing on regression algorithms.

- A simulation of graphical shapes undergoing an alignment into the same subspace
- Testing several regression algorithms on the summary of a dataset

4.1 Simulation of shape alignment

Generative encoding shares ideas with Procrustes, a geometric based method designed to match points optimally in a rigid formulaic expression. Generative Encoding can match points similarly, however, it is more flexible as the main goal is to align all datasets involved into the same subspace, rather than aligning one dataset to another.

Here, two datasets are simulated - both visually inspected as heart shape structures when represented in the euclidean plane. When these datasets are fully aligned, it is expected that the parameters transform one dataset to another such that the structures align in axis and match in location [6].

The encoded features align in location, while the encoded samples align along the vertical axis. Then, by simultaneously aligning the encoded samples (aligned along y-axis) and encoded features (aligned at center of shape), the datasets integrate and match point-by-point.

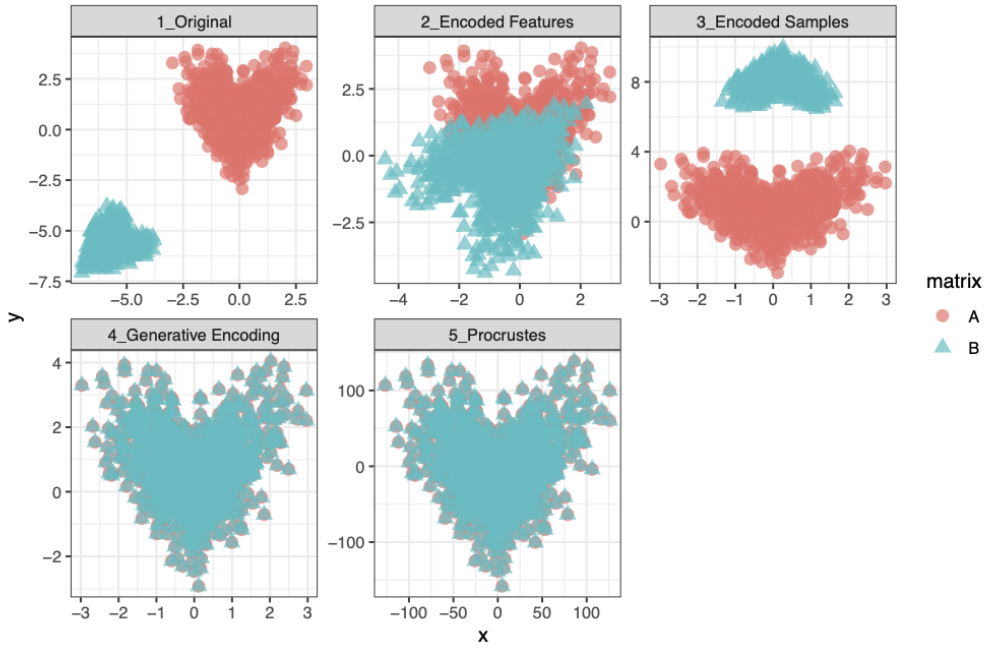

Figure 2: Simulation of two heart shapes (red and blue) with the same distribution of points - rotated and located at different regions on the plane. When taken through Generative Encoding or gcode, the heart shapes align: point to point. This similarly occurs for Procrustes, a geometric based method.

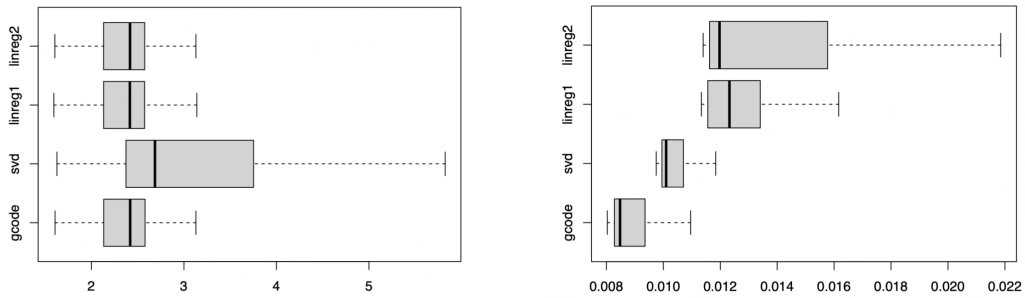

Figure 3:

5 Sample summary evaluation

6 Discussion

Acknowledgments

References

[1] Benjamin DeMeo and Bonnie Berger. Hopper: a mathematically optimal algorithm for sketching biological data. *Bioinformatics*, 36:i236–i241, 07 2020.

[2] Mostafa Rahmani and George K. Atia. Spatial random sampling: A structure-preserving data sketching tool. *IEEE Signal Processing Letters*, 24(9):1398–1402, 2017.

[3] Brian Hie, Hyunghoon Cho, Benjamin DeMeo, Bryan Bryson, and Bonnie Berger. Geometric sketching compactly summarizes the single-cell transcriptomic landscape. *Cell systems*, 8(6):483–493.e7, Jun 2019.

[4] Joel A. Tropp. *Column Subset Selection, Matrix Factorization, and Eigenvalue Optimization*, pages 978–986. Proceedings. Society for Industrial and Applied Mathematics, Jan 2009. 0.

[5] Petros Drineas, Michael Mahoney, and Senthilmurugan Muthukrishnan. Subspace sampling and relative-error matrix approximation: Column-row-based methods. volume 4110, pages 304–314, 09 2006.

[6] Meng Xu. Procrustes analysis. 2016.