TUTORIAL FOR GENERATIVE ENCODING

David Banh [1], Alan Huang [2]

[1] AskExplain
[2] The University of Queensland
Corresponding email: david.b@askexplain.com

ABSTRACT

Large information sizes in samples and features can be encoded to speed up the learning of statistical models based on linear algebra and remove unwanted signals. Encoding information can reduce both sample and feature dimension to a smaller representational set. Here two examples are shown on linear mixed models and mixture models speeding up the run time for parameter estimation by a factor defined by the user's choice on dimension reduction (can be linear, quadratic or beyond based on dimension specification).

GitHub at: `https://github.com/AskExplain/encoding_information`

1 Introduction

Large sizes in the samples and features in statistics creates large matrix objects in linear algebra. Large information structures generally occur when the dimensional size is large enough such that the (generalised) inverse of the cross product of two matrices is computationally infeasible to calculate quickly (this generally occurs for sizes greater than n = 10,000).

Here, an encoding of the information is proposed to reduce the matrix dimensions to a tractable size such that after inverting and decoding, gives a representational structure similar to when the operation runs on the full matrix. Examples are used for mixture models [1] and linear mixed models [2].

Encoding information can be done with Singular Value Decomposition (SVD), or, a similar method based on SVD by same authors called Generative Encoding [3] [4].

2 Methods

2.1 CoreVec

$$Y = \alpha X \beta$$

The basic model based on linear regression. Extending the standard linear regression, α, a full-rank matrix, rotates the samples to align with Y.

2.2 Generalised Canonical Procrustes

$$\alpha_Y Y \beta_Y = \alpha_X X \beta_X$$

Extending the CoreVec model, the linear regression is turned into a projection to an equivalent space with lower dimensions than the original X or Y. Neither α nor β of X or Y are of full-rank and are thus projections to a lower dimensional subspace. The standard Procrustes Analysis method assumes α is a full-rank rotation matrix.

2.3 Generative Encoding

$$\alpha_Y Y \beta_y = \alpha_X X \beta_X = \alpha_Z Z \beta_Z$$

This is similar to the Generalised Canonical Procrustes method where multiple modalities of data can be projected or "encoded" into the same subspace - the extension is the factorisation by learned parameters, as shown here:

$$Y = \alpha_Y C_Y \beta_Y + i_Y + \epsilon_Y$$
$$X = \alpha_X C_X \beta_X + i_X + \epsilon_X$$
$$Z = \alpha_Z C_Z \beta_Z + i_Z + \epsilon_Z$$

The projections are equivalent to a factorisation of the data. This implies that the projections of the data are also a component of the data - when optimally learned it is equivalent to Singular Value Decomposition (SVD).

Notice, the data can be modelled via a generalised model, where ϵ can take any distribution with residual noise equivalent to a Gaussian, Poisson, or Negative Binomial.

An intercept is added, although not necessary (as per SVD).

2.4 Joining

Setting some parameters to be common increases the degrees of freedom for a more flexible model that can be more generalisable.

For example, setting a joined code C projects the data into the same subspace:

$$Y = \alpha_Y C \beta_Y + i_Y + \epsilon_Y$$
$$X = \alpha_X C \beta_X + i_X + \epsilon_X$$
$$Z = \alpha_Z C \beta_Z + i_Z + \epsilon_Z$$

Notice that all datasets X, Y and Z are now in a joint feature and sample subspace - projected to the space of C, or joined code. A joined feature parameter can project the feature information into a common space:

$$Y = \alpha_Y C_Y \beta + i_Y + \epsilon_Y$$
$$X = \alpha_X C_X \beta + i_X + \epsilon_X$$
$$Z = \alpha_Z C_Z \beta + i_Z + \epsilon_Z$$

Here, all of X, Y and Z must be of the same feature dimension p. parameter fixes the information into the same subspace

Likewise, a joined sample parameter can project the sample information into the same space:

$$Y = \alpha C_Y \beta + i_Y + \epsilon_Y$$
$$X = \alpha C_X \beta + i_X + \epsilon_X$$
$$Z = \alpha C_Z \beta + i_Z + \epsilon_Z$$

A covariance join assumes the data has a covariance structure, and joins both the sample and feature parameters together. Assume X, Y and Z are sample covariance structures with the same sample size :

$$Y = \alpha C_Y \alpha$$
$$X = \alpha C_X \alpha$$
$$Z = \alpha C_Z \alpha$$

2.5 Signals and Components

Components are similar to SVD components and can be defined as orthogonal information. Components are also considered as signals - where signals extract meaningful objects that sum together to compose the data generating process. Signals that are orthogonal are also considered as filters - see the concept of Gabor Filters.

A novel extension of how signals are generated would be to see how a Gaussian random initialisation transforms into a signal through iterative updates.

2.6 Combining signals

The learned parameters which are signal components can be combined together to form new signals. A combination can compose through summation or multiplication. A new signal from a combination of signals contains more information shared between spaces that enrich the contained informational content of the signals.

2.7 Feature associations

The parameters in each signal are weights that compose together the dataset. The weights can be considered how strong the association is between feature or sample to the corresponding signal. Parameter weights are scores assigned to signals - the greater the score, the more aligned (positive) or opposing (negative) the sample or feature is to the signal. The code is then the association between signals of features and samples. The encode is then a projection of the data to the signal space.

2.8 Dimension Reduction

Encoding information involves reducing the information space down to a smaller subspace. This involves a parameter that acts as an encoding function to code the samples or features into a reduced dimension.

Here, the sample encoder α reduces the sample dimension down from n to m, keeping the feature dimension untouched. Here αY is of m by p dimensions where p is the original feature size.

Alternatively, the feature encoder β reduces the feature dimension down from p to r, where the sample dimension is left to be. The encoded data is given as: $Y\beta$ of n by r dimensions, where n is the original sample size.

2.9 Encoding

Using the same expression as dimension reduction, statistical models can be encoded into a smaller dimension (sample or feature), reducing the information structure contained into a smaller subspace. Statistical models that are encoded contain a covariance matrix of reduced size, leading to learning estimates faster.

2.10 Factorisation

Parameter encoders are more than weights, data points for sample i and feature j are composed of weights where the contribution by each weight has a sample signal (alpha), and a feature signal (beta), linked by the association between sample signals and feature signals (code)

2.11 Transformation

A common code which links sample and feature signals into a common space via the joined code coordinate system enables multiple modalities to be transformed from one to the other. The common coordinate system is given as follows for two datasets:

$$X = \alpha^T C \beta_X^T$$
$$Y = \alpha^T C \beta_Y^T$$

Then the transformation of modality X to Y is given as:

$$X\beta_X = \alpha^T C \sim \alpha^T C \beta_Y = X\beta_X \beta_Y^T$$

Like lego or a modular machine, feature or sample parameters can be transformed from one modality by another.

2.12 Generative Sampling

Learning the model of $X = \alpha^T C \beta^T$ extending the sample signal α would extend the sample observations of X. For example take α as n by k where the extended sample signals is given by γ of s by k, where $s > n$. Here, to extend n to s, the covariance of α is taken as k by k and re-sampled from a Gaussian distribution with $\mu = \Sigma_n \alpha$ and $\Sigma = \alpha^T \alpha$.

2.13 Prediction and Imputation

Learning the encoded and coded structure of the data can impute or predict missing points by iteratively recovering the missing information, and repeatedly update the encoded and coded structure. This bootstrap iterative form of prediction can improve the prediction accuracy of a standard KNN or linear regression model.

2.14 Model and Parameter transfer

The sample parameters of a model of a given dataset of sample size n can be transferred to a new model given the dimensions of the new dataset is also n. Alternatively, it can be possible that the feature size of p from the original dataset and model can be transferred to the new dataset and model of feature size p.

3 Encoding of Statistical models

3.1 Linear Mixed Model

For a linear mixed model to be encoded, first consider the mixed model equation:

$$Y = X\beta + Zu + e$$

To encode sample information, a function is introduced as a parameter α to re-weight the samples into a reduced dimension:

$$\alpha Y = \alpha X\beta + \alpha Zu + e$$

Provided each of Y, X and Z are of n samples, and α is a parameter that transforms the n samples into m samples (where $n > m$), then the final model will be learned via the covariance of a smaller size.

3.1.1 Genetic Relatedness Matrix

For example, the general model in genetics to measure the heritability of a trait is given as:

$$Y \sim N(X\beta, G\sigma_g + D_1\sigma_e)$$

Reducing the sample size would introduce to this model, the following:

$$\alpha Y \sim N(\alpha X\beta, (\alpha G\alpha^T)\sigma_g + D_2\sigma_e)$$

Notice that α is of m dimensions rather than n, enabling the linear mixed model to learn with a smaller Genetic Relatedness Matrix, yet still retaining the ability to learn σ_g the heritability of the phenotype.

3.2 Mixture Model

Rather than encoding sample information, the dimensions of the feature structure can be encoded. For example for a mixture model to be encoded, first consider the expression:

$$Y \sim \pi_i N(\mu_i, \Sigma_i)$$

To encode sample structure, a function is introduced as a parameter β to re-weight the samples into a reduced dimension:

$$Y\beta \sim \pi_i N(\mu_i\beta, \beta^T\Sigma_i\beta)$$

By reducing the dimensions of the mixture model to learn a model from r features rather than p features (where $p > r$) the model can be learned faster as the feature information p is learned in the reduced dimensions r.

3.2.1 Factor analytic models

For example, the general model for mixtures of factor analyses in psychology or economic studies used to measure the scores of particular behaviours or events is given as:

$$X \sim \pi_i N(\mu_i + \Lambda_i z, D_i)$$
$$X|z \sim \pi_i N(\mu_i, \Lambda_i\Lambda_i^T + D_i)$$

where the conditional expression based on the latent features z is found in the second expression above.

Taking into account the feature encoding to reduce the dimensional size of p takes the dimensions of the features to r dimensions

$$X\beta \sim \pi_i N(\mu_i\beta + \beta\Lambda_i z_\beta, D_{i_\beta})$$

$$X\beta|z_\beta \sim \pi_i N(\mu_i\beta, (\beta\Lambda_i\Lambda_i^T\beta^T) + D_{i_\beta})$$

4 Results

4.1 Linear Mixed Model

To test the results on a linear mixed model, 1000 permutations were run on a simulated dataset with a heritability of 0.5.

The package GMMAT in R was used to run the analysis with a Genetic Relatedness Matrix. The table below shows the results comparing the encoded and original linear mixed models comparing the heritability estimates and runtime of the full mixed model with the encoded model (including the time to learn the encoding).

This is for 100 permutation runs of a simulation with 1000 samples and 100 SNPs simulated according to

$$Z \sim Binom(m = 2, p = 0.5)$$
$$\lambda_s \sim N(0, \sqrt{(h_2/p)})$$
$$Y \sim N(Z\lambda_s, \sqrt{(1 - h_2)})$$

Where the Genetic Relatedness Matrix (1000 by 1000 dimensions) is given by:

$$GRM = (ZZ^T)/p$$

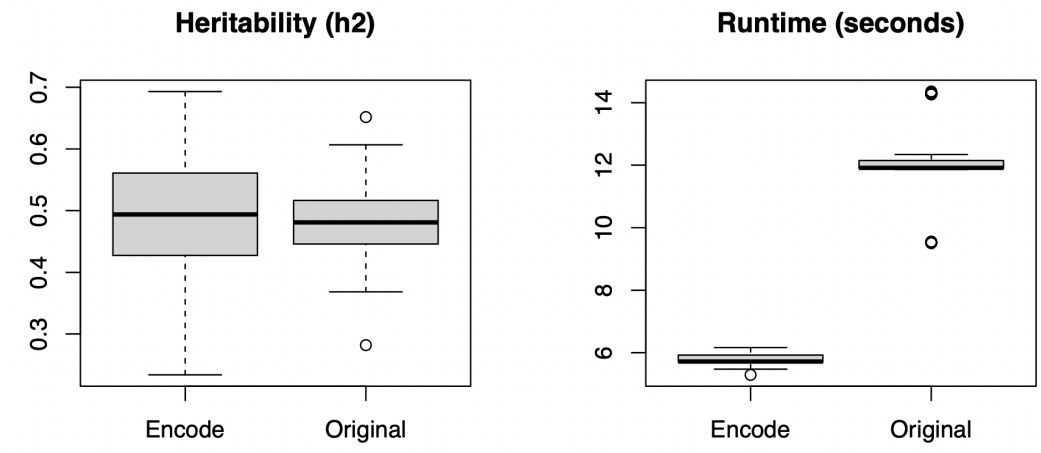

Figure 1: Comparing an encoded and standard linear mixed model for a heritability estimate distributed around 0.5 (true is at $h^2 = 0.5$), and runtime in seconds

4.2 Mixture Model

To test the results on a linear mixed model, approximately 1000 permutations were run on a test dataset from the pdfCluster [5] package using the OliveOil dataset only on the numerical dataset.

Given the OliveOil dataset has two categorical features structured hierarchically, one as a subset of the other, the categorical feature with the fewest number of categories was used. This equates to 3 known clusters to label.

The features were encoded ranging from 2 to 8 (the original number of numerical features in the OliveOil dataset).

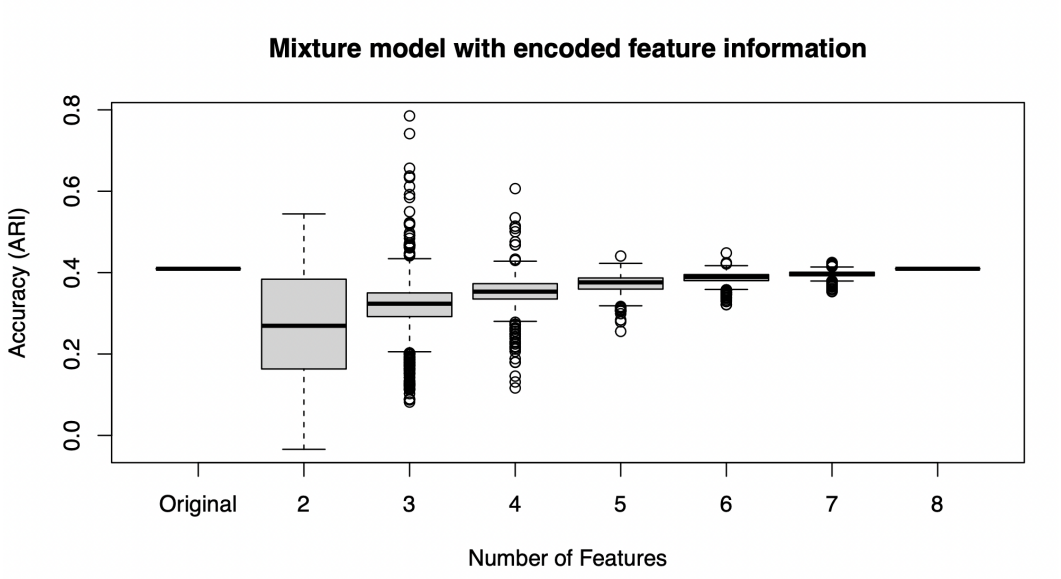

Figure 2: Mixture model with an encoding to 2 to 8 features, compared with the original features (p = 8).

5 Discussion

Given information is being encoded, it is expected for there to be information loss leading to higher variability compared to the standard linear mixed model. However, due to reduced sample size via an encoding the runtime is faster - almost half the speed of the original mixed model according to the R package GMMAT [6].

Notice that with an encoding of the features into a lower dimensional space - mixture model clustering accuracy has more flexibility due to an increase the degrees of freedom leading to fit worse or better models. On average, the larger the encoding - up to the original number of features, the higher the accuracy of the final clustering.

6 Conclusion

Information in a vector, matrix, or tensor object can be encoded by learning representational features that simultaneously: encodes the object and represents the object via factorisation. Through considering functions that encodes and re-represents the information via factorisation, an optimal model is learned that extracts relevant signals from the data object to manipulate the feature and, or sample structure.

This has implications on linear algebra and statistical methods - encoding the samples can reduce the computational run time for mixed models when a sample covariance matrix is used. Alternatively, features can be encoded to reduce the computational run time of a feature covariance in mixture models.

7 Acknowledgements

Professor Geoff McLachlan has been a tremendous help in the guidance of past work on mixture models (see Deep Gaussian Mixture Models). Professor Jian Yang has also been an inspiration for the mixed model work. Also a thanks to Yuna Zhang for jump starting the work by providing preliminary scripts on Average Information with mixed models.

References

[1] Cinzia Viroli and Geoffrey J. McLachlan. Deep gaussian mixture models. *Statistics and Computing*, 29(1):43–51, Jan 2019.

[2] Jian Yang, S. Hong Lee, Michael E. Goddard, and Peter M. Visscher. Gcta: a tool for genome-wide complex trait analysis. *American journal of human genetics*, 88(1):76–82, Jan 2011. 21167468[pmid].

[3] David Banh and Alan Huang. Scalable parametric encoding of multiple modalities. *bioRxiv*, 2022.

[4] David Banh. Sample summary with generative encoding. *CoRR*, abs/2201.08233, 2022.

[5] Adelchi Azzalini and Giovanna Menardi. Clustering via nonparametric density estimation: The r package pdfcluster. *Journal of Statistical Software*, 57(11):1–26, 2014.

[6] Han Chen, Chaolong Wang, Matthew P. Conomos, Adrienne M. Stilp, Zilin Li, Tamar Sofer, Adam A. Szpiro, Wei Chen, John M. Brehm, Juan C. Celedón, Susan Redline, George J. Papanicolaou, Timothy A. Thornton, Cathy C. Laurie, Kenneth Rice, and Xihong Lin. Control for population structure and relatedness for binary traits in genetic association studies via logistic mixed models. *American journal of human genetics*, 98(4):653–666, Apr 2016. 27018471[pmid].