LINEAR MIXED FORMS

David Banh
Data Science
AskExplain
St Lucia, Brisbane, Australia
david.b@askexplain.com

ABSTRACT

An alternative approach to linear mixed models on matrix data. By decomposing all variables in the model into two latent components: a common fixed effect and common random effect it is possible to learn a flexible model. Furthermore, through learning an encoding of all variables through learnable parameters similar to linear regression coefficients the model is fully interpretable. The setup structure of the model also enables a covariance of the interpretable parameters to reconstruct the observed data represented by each variable. Coupling the model with a convolutional layer in a deep learning system also gives comparable results to a sequential layer of linear layers with activation functions.

1 Introduction

Linear mixed models are able to represent informative signals from the data as both the expectation of the original data generating process and the covariance from interacting signals that share an associative structure with the original generation process. By coupling the original data generating process with interacting processes, informative signals from each process can be decoupled with greater acuity and used in the model to improve fit and reduce errors.

This paper introduces linear mixed forms as the idea of integrating multiple modalities in a linear mixed model via transformations. The idea focuses on a transform to a common space via a latent code that constructs an idealised data generation process which generates samples with multiple modal measurements. It shares ideas with the Generative Encoder a method that transforms multiple modalities into a similar region defined by a common latent space via matrix regression [1] [2]. Convolution of the common latent space with the feature components of each dataset generates the individual modal observations.

The concepts of transformations via encoding and decoding are used often in the machine learning literature on neural networks - especially in transformers [3]. The main idea is to find an efficient and effective way to learn relevant signals from associative structures across features. The idea of associative structures exists within the standard linear mixed model. In addition, this paper posits that transformations via encoding information within the fixed, random and estimated structures can extend the model to:

- transform the variable of interest to and from other modalities
- enable integration of multimodal data (e.g. images) into the mixed model
- improve computational efficiency and runtime when learning the model

2 Model

Given the response variable Y to be estimated using the main data X as the fixed effects, and the covariates Z as the random effects with residuals $e \sim N(0, D)$ and diagonal D, with independent random effects $u \sim\ N(0, G)$ where the full covariance of Y is given as $Cov(Y) = D + ZGZ^T$:

To transform the fixed and random effects into an estimation of a similarly transformed response by α to form $Y\alpha$ via β and u in $X\beta$ and Zu in the expression:

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

Drawing inspiration from the ideas in Generative Encoders, each dataset is decomposed into a common latent code, and the transpose of the transformation projection parameter.

Thus if the latent code is given as K or H, and α transforms $Y \longrightarrow Y\alpha$, then find parameters K or H such that

$$Y = (K + H)\alpha^T$$

. The common latent K and H are then shared across all steps for learning of the other parameters:

$$Y = (K + H)\alpha^T$$

$$X = K\beta^T$$

$$Z = Hu^T$$

The full expression when the decomposition into a latent code and an extra transformation parameter is given as:

$$(K + H)(\alpha^T\alpha) = K(\beta^T\beta) + H(u^T u) + e$$

Algorithm 1: Linear Mixed Forms

Input: Response Y, Data X, Covariates Z
Output: α, β, u, K

Initialise α, β, u, K

for $t \in timesteps$ **do**
  $\alpha = (((K + H)^T(K + H))^{-1}(K + H)^T Y)^T$
  $\beta = ((K^T K)^{-1}K^T X)^T$
  $u = ((H^T H)^{-1}H^T Z)^T$
  if *remainder(t / 3) = 0* **then**
    (K+H) = Y α $(\alpha^T\alpha)$
  if *remainder(t / 3) = 1* **then**
    K = X β $(\beta^T\beta)$
  if *remainder(t / 3) = 2* **then**
    H = Z u $(u^T u)$

3 Properties

There are several important properties that linear mixed forms share with generative encoding:

-

3.1 Orthogonality

When parameters α, β, or u are orthogonal it is possible to rewrite an expression that can be included in a neural network without need for generalised inverse operations. When parameters are orthogonal, the fixed effects part can be expressed as:

$$\beta^T\beta = I$$
$$K = K\beta^T\beta = X\beta$$
$$X\beta = X\beta(\beta^T\beta) = X\beta(\beta^T\beta)(\beta^T\beta) = X\beta(\beta^T\beta)(\beta^T\beta)(\beta^T\beta)$$

Here a series is created where the orthogonality property of β creates a series of equalities that implies the structure of a loss function ready for a neural network. For approximation and appropriate runtime with adequate convergence, in classification image tasks of labels Y and images X with convolution $f(X)$, the loss function would then be:

$$Cross\ \ Entropy\ \ (Y, f(X)\beta) + Mean\ \ Squared\ \ Error\ \ (f(X)\beta, f(X)\beta(\beta^T\beta))$$

This also sprevents prohibitive inverse operations over dimension sizes of approximately greater than 1000 components and allows neural networks to extend to sizes limited only by computational time of matrix multiplication (or standard neural network operations).

4 Acknowledgements

David Banh would like to acknowledge:

Alan Huang for relevant discussions of Generative Encoders, and work relating to linear mixed models. Also for being open and generous enough to discuss ideas, even when it was just a small one.

Ryan Deslandes (University of Queensland, Australia) for assistance with software development of the genecode dashboard (no longer in use): `https://board.askexplain.com/genecode`. This idea was presented in the NanoString Hackathon earning third place and bonus prize for best images (first and second place went to the teams from Dr Omer Bayraktar's Lab from the Wellcome Sanger Institute, one of which presented cell2location).

Cameron Gordon and Olivia Ou (University of Adelaide, Australia and University of Queensland, Australia) for consistent support and advice throughout the project.

Alex Alsaffar (University of Queensland, Australia) for relevant discussions on the model of Generalised Canonical Procrustes and Generative Encoding as well as work on a preliminary Python version of gcproc (a prior version of gcode): `https://github.com/thisismygitrepo/gcprocpy`.

Dr Quan Nguyen (University of Queensland, Australia) for a brief discussion on the model of corevec (a prior version of gcproc, which is an earlier version of gcode) regarding imputation and alignment in early 2021 https://github.com/AskExplain/corevec

References

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

[2] Cinzia Viroli. On matrix-variate regression analysis. *Journal of Multivariate Analysis*, 111:296–309, 2012.

[3] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.