Normative Modeling of Neuroimaging Data using Scalable Multi-Task Gaussian Processes

Seyed Mostafa Kia[1,2] and Andre Marquand[1,2]

[1] Department of Cognitive Neuroscience, Radboud University Medical Centre,
Nijmegen, The Netherlands
[2] Donders Centre for Cognitive Neuroimaging, Donders Institute for Brain, Cognition
and Behaviour, Radboud University, Nijmegen, The Netherlands
{s.kia,a.marquand}@donders.ru.nl

Abstract. Normative modeling has recently been proposed as an alternative for the case-control approach in modeling heterogeneity within clinical cohorts. Normative modeling is based on single-output Gaussian process regression that provides coherent estimates of uncertainty required by the method but does not consider spatial covariance structure. Here, we introduce a scalable multi-task Gaussian process regression (S-MTGPR) approach to address this problem. To this end, we exploit a combination of a low-rank approximation of the spatial covariance matrix with algebraic properties of Kronecker product in order to reduce the computational complexity of Gaussian process regression in high-dimensional output spaces. On a public fMRI dataset, we show that S-MTGPR: 1) leads to substantial computational improvements that allow us to estimate normative models for high-dimensional fMRI data whilst accounting for spatial structure in data; 2) by modeling both spatial and across-sample variances, it provides higher sensitivity in novelty detection scenarios.

Keywords: Gaussian Processes, Multi-Task Learning, Normative Modeling, Neuroimaging, fMRI, Clinical Neuroscience, Novelty Detection

1 Introduction

Understanding the underlying biological mechanisms of psychiatric disorders constitutes a significant step toward developing more effective and individualized treatments (*i.e.*, *precision medicine* [11]). Recent advances in neuroimaging and machine learning provide an exceptional opportunity to employ brain-derived biological measures for this purpose. While symptoms and biological underpinnings of mental diseases are known to be highly heterogeneous, data-driven approaches play an important role in stratifying clinical groups into more homogeneous subgroups. Currently, off-the-shelf clustering algorithms are the most predominant approaches for stratifying clinical cohorts. However, the high-dimensionality and complexity of data beside the use of heuristics to find optimal clustering solutions negatively affect the reproducibility and reliability of resulting clusters [10]. Normative modeling [9] offers an alternative approach to model biological variations within clinical cohorts without needing to assume cleanly separable clusters or

cohorts. This approach is applicable to most types of neuroimaging data such as structural/functional magnetic resonance imaging (s/fMRI).

Normative modeling employs Gaussian process regression (GPR) [16] to predict neuroimaging data on the basis of clinical and/or behavioral covariates. GPR, and in general Bayesian inference, can be seen as an indispensable part of the normative modeling as it provides coherent estimates of predictive confidence. These measures of predictive uncertainty are important for quantifying centiles of variation in a population [9]. GPR also provides the possibility to accommodate both linear and nonlinear relationships between clinical covariates and neuroimaging data.

The variant of GPR originally employed for normative modeling aims to model only a single output variable. Thus in normative modeling, one should independently train separate GPR models for each unit of measurement (*e.g.*, for each voxel in a mass-univariate fashion). Such a simplification ignores the possibility of modeling and capitalizing on the existing spatial structure in the output space. However, GPR can be extended to perform a joint prediction across multiple outputs in order to account for correlations between variables in neuroimaging data (for example different voxels in fMRI data). Boyle and Frean [6] proposed to employ convolutional processes to express each output as the convolution between a smoothing kernel and a latent function. This idea is later adopted by Bonilla *et al.* [5] to extend the classical single-task GPR (STGPR) to multi-task GPR (MTGPR) by coupling a set of latent functions with a shared GP prior in order to directly induce correlation between output variables (tasks). They proposed to disentangle the full cross-covariance matrix into the Kronecker product of the sample (in input space) and task (in output space) covariance matrices. This technique provides the possibility to model both across-sample and across-task variations. Despite its effectiveness in modeling structures in data, MTGPR comes with extra computational overheads in time and space, especially when dealing with high-dimensional neuroimaging data. We briefly review recent efforts toward alleviating these computational burdens.

1.1 Toward Efficient and Scalable MTGPR

For N samples and T tasks, the time and space complexity of MTGPR are $\mathcal{O}(N^3 T^3)$ and $\mathcal{O}(N^2 T^2)$, respectively. These high computational demands (compared to STGPR with $\mathcal{O}(N^3 T)$ and $\mathcal{O}(N^2 T)$) are mainly due to the need for computing the inverse cross-covariance matrix in learning and inference phases. In neuroimaging problems that we consider, these can both be relatively high where N in generally in the order of $10^2 - 10^4$ and T is in the order of $10^4 - 10^5$ or even higher. Therefore, improving the computational efficiency of MTGPR is crucial for certain problems, and there have been several approaches proposed for this in the machine learning literature [13,3]. Here we briefly review two main directions to address the computational tractability issue of MTGPR.

In the first set of approaches, approximation techniques are used to improve estimation efficiency. Bonilla *et al.* [5] made one of the earliest efforts in this direction, in which they proposed to use Nyström approximation on M inducing

inputs [13] out of N samples in combination with the probabilistic principal component analysis, in order to approximate reduced M-rank and P-rank sample and task covariance matrices, respectively. Their approximation reduced the time complexity of hyperparameter learning to $\mathcal{O}(NTM^2P^2)$. Elsewhere, Alvarez and Lawrence [2] proposed to approximate a sparse version of MTGPR, assuming conditional independence between each output variable with all others given the input process. This assumption besides using M out of N input samples as inducing inputs reduces the computational complexity of MTGPR to $\mathcal{O}(N^3T + NTM^2)$ and $\mathcal{O}(N^2T + NTM)$ in time and storage, where for $N = M$ is the same as a set of T independent STGPRs. Alvarez *et al.* in [4] extended their previous work by developing the concept of inducing function rather than inducing input. Their new approach so-called variational inducing kernels achieves time complexity of $\mathcal{O}(NTM^2)$.

The second set of approaches utilize properties of Kronecker product [8] to reduce the time and space complexity in computing the exact (and not approximated) inverse covariance matrix. Stegle *et al.* [15] proposed to use these properties in combination with eigenvalue decomposition of input and task covariance matrices for efficient parameter estimation, and likelihood evaluation/optimization in MTGPR. In this method, the joint covariance matrix is defined as a Kronecker product between the input and task covariance matrices. This approach reduces the time and space complexity of MTGPR to $\mathcal{O}(N^3 + T^3)$ and $\mathcal{O}(N^2 + T^2)$, respectively. To account also for structured noise, Rakitsch *et al.* [14] extended this method by using two separate Kronecker products for the signal and noise. Importantly, this provides a significant reduction in computational complexity using all samples (*i.e.*, not just inducing inputs), and is exact in the sense that it does not require any approximation or relaxing assumptions.

Our contribution: In spite of all aforementioned efforts, applications of MT-GPR in encoding neuroimaging data from a set of clinically relevant covariates remained very limited, mainly due to the high dimensionality of the output space (*i.e.*, very large T). Our main contribution in this text addresses this problem and extends MTGPR to the normative modeling of neuroimaging data. To this end, we use a combination of low-rank approximation of the task covariance matrix with algebraic properties of Kronecker product in order to reduce the computational complexity of MTGPR. Furthermore, on a public fMRI dataset, we show that: 1) our method makes MTGPR possible on very high-dimensional output spaces; 2) it enables us to model both across-space and across-subjects variations, hence provides more sensitivity for the resulting normative model in novelty detection.

2 Methods

2.1 Notation

Boldface capital letters, $\mathbf{A}$, and capital letters, A, are used to denote matrices and scalar numbers. We denote the vertical vector which is resulted from collapsing columns of a matrix $\mathbf{A} \in \mathbb{R}^{N \times T}$ with $vec(\mathbf{A}) \in \mathbb{R}^{NT}$. In the remaining text, we use $\otimes$ and $\odot$ to respectively denote Kronecker and the element-wise matrix

products. We denote an identity matrix by $\mathbf{I}$; and the determinant, diagonal elements, and the trace of matrix $\mathbf{A}$ with $|\mathbf{A}|$, $diag(\mathbf{A})$, and $Tr[\mathbf{A}]$, respectively.

2.2 Scalable Multi-Task Gaussian Process Regression

Let $\mathbf{X} \in \mathbb{R}^{N \times F}$ be the input matrix with N samples and F covariates. Let $\mathbf{Y} \in \mathbb{R}^{N \times T}$ represent a matrix of response variables with N samples and T tasks (here, neuroimaging data with T voxels). The multi-task Kronecker Gaussian process model (MT-Kronprod) [15] is defined as:

$$p(\mathbf{Y} \mid \mathbf{D}, \mathbf{R}, \sigma^2) = \mathcal{N}(\mathbf{Y} \mid \mathbf{0}, \mathbf{D} \otimes \mathbf{R} + \sigma^2 \mathbf{I}) \quad , \tag{1}$$

where $\mathbf{D} \in \mathbb{R}^{T \times T}$ and $\mathbf{R} \in \mathbb{R}^{N \times N}$ are respectively the task and sample covariance matrices (here, modeling correlations across voxels and samples separately). Despite its effectiveness in modeling both samples and tasks variations, the application of MT-Kronprod is limited when dealing with very large output spaces, such as neuroimaging data, mainly due to the high computational complexity of matrix diagonalisation operations in the optimization and inference phases. We propose to address this problem by using a low-rank approximation of $\mathbf{D}$.

Let $\Phi : \mathbf{Y} \rightarrow \mathbf{Z}$ be an orthogonal linear transformation, *e.g.*, principal component analysis (PCA), that transforms $\mathbf{Y}$ to a reduced latent space $\mathbf{Z} \in \mathbb{R}^{N \times P}$, where $P < T$, and we have $\mathbf{Z} = \Phi(\mathbf{Y}) = \mathbf{YB}$. Here, columns of $\mathbf{B} \in \mathbb{R}^{T \times P}$ represent a set of P orthogonal basis functions. Assuming a zero-mean matrix normal distribution for $\mathbf{Z}$, by factorizing its rows and columns we have:

$$p(\mathbf{Z} \mid \mathbf{C}, \mathbf{R}) = \mathcal{MN}(\mathbf{0}, \mathbf{C} \otimes \mathbf{R}) = \frac{\exp(-\frac{1}{2}Tr[\mathbf{C}^{-1}\mathbf{B}^\top \mathbf{Y}^\top \mathbf{R}^{-1}\mathbf{YB}])}{\sqrt{(2\pi)^{NP}\,|\mathbf{C}|^P\,|\mathbf{R}|^N}} \quad , \tag{2}$$

where $\mathbf{C} \in \mathbb{R}^{P \times P}$ and $\mathbf{R} \in \mathbb{R}^{N \times N}$ are column and row covariance matrices of $\mathbf{Z}$. Using the trace invariance property under cyclic permutations, the noise-free multivariate normal distribution of $\mathbf{Y}$ can be approximated from Eq. 2:

$$p(\mathbf{Y} \mid \mathbf{D}, \mathbf{R}) \approx p(\mathbf{Y} \mid \mathbf{C}, \mathbf{B}, \mathbf{R}) = \frac{\exp(-\frac{1}{2}Tr[\mathbf{BC}^{-1}\mathbf{B}^\top \mathbf{Y}^\top \mathbf{R}^{-1}\mathbf{Y}])}{\sqrt{(2\pi)^{NT}\,\left|\mathbf{BCB}^\top\right|^T\,|\mathbf{R}|^N}} \quad , \tag{3}$$

where $\mathbf{D}$ is approximated by $\mathbf{BCB}^\top$. Our scalable multi-task Gaussian process regression (S-MTGPR) model is then derived by marginalizing over noisy samples:

$$p(\mathbf{Y} \mid \mathbf{D}, \mathbf{R}, \sigma^2) \approx p(\mathbf{Y} \mid \mathbf{C}, \mathbf{B}, \mathbf{R}, \sigma^2) = \mathcal{N}(\mathbf{Y} \mid \mathbf{0}, \mathbf{BCB}^\top \otimes \mathbf{R} + \sigma^2 \mathbf{I}) \quad . \tag{4}$$

Predictive Distribution: Following the standard GPR framework [16] and setting $\tilde{\mathbf{D}} = \mathbf{BCB}^\top$, the mean and variance of the predictive distribution of unseen samples, *i.e.*, $p(vec(\mathbf{Y})^* \mid vec(\mathbf{M}^*), \mathbf{V}^*)$, can be computed as follows:

$$vec(\mathbf{M}^*) = (\tilde{\mathbf{D}} \otimes \mathbf{R}^*)(\tilde{\mathbf{D}} \otimes \mathbf{R} + \sigma^2 \mathbf{I})^{-1}vec(\mathbf{Y}), \tag{5a}$$

$$\mathbf{V}^* = (\tilde{\mathbf{D}} \otimes \mathbf{R}^{**}) - (\tilde{\mathbf{D}} \otimes \mathbf{R}^*)(\tilde{\mathbf{D}} \otimes \mathbf{R} + \sigma^2 \mathbf{I})^{-1}(\tilde{\mathbf{D}} \otimes \mathbf{R}^{*\top}), \tag{5b}$$

where $\mathbf{R}^{**} \in \mathbb{R}^{N^* \times N^*}$ is the covariance matrix of N^* test samples , and $\mathbf{R}^* \in \mathbb{R}^{N^* \times N}$ is the cross-covariance matrix between test and training samples.

Efficient Prediction and Optimization: For efficient prediction, and fast optimization of the log-likelihood, we extend the approach proposed in [15,14] by exploiting properties of Kronecker product, and eigenvalue decomposition for diagonalizing the covariance matrices. Then the predictive mean and variance can be efficiently computed by:

$$\mathbf{M}^* = \mathbf{R}^*\mathbf{U_R}\tilde{\mathbf{Y}}\mathbf{U_C^\top}\mathbf{CB}^\top, \tag{6a}$$

$$\mathbf{V}^* = (\tilde{\mathbf{D}} \otimes \mathbf{R}^{**}) - (\mathbf{BCU_C} \otimes \mathbf{R}^*\mathbf{U_R})\tilde{\mathbf{K}}^{-1}(\mathbf{U_C^\top}\mathbf{CB}^\top \otimes \mathbf{U_R^\top}\mathbf{R}^{*\top}), \tag{6b}$$

where $\mathbf{C} = \mathbf{U_C}\mathbf{S_C}\mathbf{U_C^\top}$ and $\mathbf{R} = \mathbf{U_R}\mathbf{S_R}\mathbf{U_R^\top}$ are eigenvalue decomposition of covariance matrices, $\tilde{\mathbf{K}} = \mathbf{S_C} \otimes \mathbf{S_R} + \sigma^2\mathbf{I}$, and $vec(\tilde{\mathbf{Y}}) = diag(\tilde{\mathbf{K}}^{-1}) \odot vec(\mathbf{U_R^\top}\mathbf{YB}\mathbf{U_C})$.[3] Based on our assumption on the orthogonality of components in $\mathbf{B}$, we set $\mathbf{B}^{-1} = \mathbf{B}^\top$ and $\mathbf{B}^\top\mathbf{B} = \mathbf{I}$. Note that in the new parsimonious formulation, heavy time and space complexities of computing the inverse kernel matrix is reduced to computing the inverse of a diagonal matrix, *i.e.*, reciprocals of diagonal elements of $\tilde{\mathbf{K}}$. For the predictive variance, explicit computation of the Kronecker product is still necessary but this can easily be overcome by computing the predictions in mini-batches. For the negative log marginal likelihood of Eq. 4, we have:

$$\mathcal{L} = -\frac{N \times T}{2}\ln(2\pi) - \frac{1}{2}\ln\left|\tilde{\mathbf{K}}\right| - \frac{1}{2}vec(\mathbf{U_R^\top}\mathbf{YB}\mathbf{U_C})^\top vec(\tilde{\mathbf{Y}}) \quad . \tag{7}$$

The proposed S-MTGPR model has three sets of parameters plus one hyper-parameter: 1) reduced task covariance matrix parameters $\Theta_\mathbf{C}$, 2) input covariance matrix parameters $\Theta_\mathbf{R}$, 3) noise variance σ^2 that is parametrized on Θ_{σ^2}, and 4) P that decides the number of components in $\mathbf{B}$. While the latter should be decided by means of model selection, the first three sets are optimized by maximizing $\mathcal{L}$.

Computational Complexity: The time complexity of the proposed method is $\mathcal{O}(N^2T + NT^2 + N^3 + P^3)$. The first two terms are related to the matrix multiplication in computing the squared term in Eq.7. The last two terms belong to the eigenvalue decomposition of $\mathbf{R}$ and $\mathbf{C}$. The P^3 term can be excluded because always $P \leq min(N,T)$. Thus, for $N > T$ and $N < T$ the time complexity is reduced to $\mathcal{O}(N^3)$ and $\mathcal{O}(NT^2)$, respectively. Thus when $N > T$ or $N < T < N^2$, our approach is analytically even faster than the baseline STGPR approach applied independently to each output variable in a mass-univariate fashion. For $N \ll T$, our method is faster than other Kronecker based MTGPRs by a factor of T/N. Such improvement not only facilitates the application of MTGPR on neuroimaging data but also it provides the possibility of accounting for the existing spatial structures across different brain regions. In comparison to the related work, the proposed method provides a substantial speed improvement, especially when dealing with a large number of tasks. This is while unlike other approximation approaches, we fully use the potential of all available samples.

3 Experiments and Results

3.1 Experimental Materials and Setup

In our experiments, we use a public fMRI dataset collected for reconstructing visual stimuli (black and white letters and symbols) from fMRI data [12]. In this

---

[3] See supplementary materials for more descriptive derivations of all equations.

Table 1. Three benchmarked methods in our experiments.

Method	Time Complexity	No. Parameters	Parameter Description
STGPR	$\mathcal{O}(N^3 T)$	21752	1 for linear and 2 for squared exponential kernels, 1 for Gaussian likelihood; multiplied by the number of tasks (5438).
MT-Kronprod	$\mathcal{O}(T^3)$	9	1 for linear, 2 for squared exponential, and 1 for diagonal isotropic kernels; multiplied by 2 (for sample and task covariance functions); plus 1 for Gaussian likelihood.
S-MTGPR	$\mathcal{O}(NT^2)$	10	Same as MT-Kronprod, plus 1 hyperparameter for the number of PCA bases.

dataset, fMRI responses were measured while 10×10 checkerboard patch images were presented to subjects according to a blocked design. Checkerboard patches constituted random (1320 trials) and geometrically meaningful patterns (720 trials). We use the preprocessed data available in Nilearn package [1] wherein the fMRI data are detrended and masked for the occipital lobe (5438 voxels).[4] Whilst our approach is quite general, we demonstrate S-MTGPR by simulating normative modeling for novelty detection. Therefore, we aim to predict the masked fMRI 3D-volume from the presented visual stimuli in an encoding setting. To this end, we randomly selected 600 random pattern trials, for training the encoding model. The model then learns to represent this reference or normative class such that anomalous or abnormal samples can be detected and characterised. The rest of non-random patterns (720 trials) and random patterns (720 trials) are used for evaluating the encoding model and testing anomaly-detection performance, achieved by fitting a generalised extreme value distribution to the most deviating voxels. In our experiments, we use PCA to transform the fMRI data in the training set from the voxel space to **Z**, and the resulting $P = 10, 25, 50, 100, 250, 500, 1000$ PCA components are used as basis matrix **B** in the optimization and inference.

We benchmark the proposed method against the STGPR (*i.e.*, mass-univariate) and MT-Kronprod models in terms of their runtime, performance of the regression, and quality of resulting normative models. In all models, we use a summation of a linear, a squared exponential, and a diagonal isotropic covariance functions for sample and task covariance matrices in order to accommodate both linear and non-linear relationships. In all cases, we use an isotropic Gaussian likelihood function. This likelihood function has different functionality in the STGPR versus MTGPR settings. In STGPR, it is defined independently for each voxel, thus it handles heteroscedastic, *i.e.*, spatially varying noise. While in MTGPR a single noise parameter is shared for all voxels, hence it merely considers homoscedastic, *i.e.*, spatially stationary, noise. The truncated Newton algorithm is used for optimizing the parameters. Table 1 summarizes the time complexity and the number of parameters of three benchmarked methods in our experiments.

We use the coefficient of determination (R^2) to evaluate the explained variance by regression models. In normative modeling, the top 5% values in normative probability maps are used to fit the generalized extreme value distribution (see [9]). To evaluate resulting normative models, we employ area under the curve (AUC) to measure the performance of the model in distinguishing between normal (here random patterns) from abnormal samples (here non-random patterns). All the steps (random sampling, modeling, and evaluation) are repeated 10 times in order to estimate the mean and standard deviation of the runtime, R^2, and AUC. All

---

[4] See http://nilearn.github.io/auto_examples/02_decoding/plot_miyawaki_reconstruction.html.

Fig. 1. Comparison between S-MTGPR, STGPR, and MT-Kronprod in terms of: a) optimization and prediction runtime, b) average regression performance (R^2) across all voxels, and c) AUC in abnormal sample detection using normative modeling.

experiments are performed on a system with Intel®Xeon®E5-1620 0 @3.60GHz CPU and 16GB of RAM.[5]

3.2 Results and Discussion

Fig. 1 compares the runtime, R^2, and AUC of STGPR and MT-Kronprod, with those of S-MTGPR for different number of bases. As illustrated in Fig. 1(a) S-MTGPR is faster than other approaches where the total runtime of MT-Kronprod (3 days) and STGPR (6 hours) can be reduced to 16 minutes for $P = 25$. This difference in runtime is even more pronounced in case of the optimization time where S-MTGPR is at least (for $P = 1000$) 33 and 89 times faster than STGPR and MT-Kronprod, respectively. The multi-task approaches are slower than STGPR in the prediction phase mainly due to the mini-batch implementation of the prediction variance computation (to avoid memory overflow). Fig. 1(b) shows this computational efficiency is achieved without penalty to the regression performance; where for certain number of bases the S-MTGPR shows equivalent and even better R^2 than STGPR and MT-Kronprod. Furthermore, Fig. 1(c) demonstrates that multi-task learning, by considering spatial structures, generally provides a more accurate normative model of fMRI data in that it more accurately detects samples that were derived from a different distribution to those used to train the model. This fact is well-reflected in higher AUC values for S-MTGPR at $P = 25, 100, 250, 500, 1000$. It is worthwhile to emphasize that these improvements are achieved by reducing the degree-of-freedom of the normative model from 21752 for STGPR to 10 for S-MTGPR (see Table 1).

4 Conclusions and Future Work

Assuming a matrix normal distribution on a reduced latent output space, we introduced an efficient and scalable multi-task Gaussian process regression approach to learning complex association between external covariates and high-dimensional neuroimaging data. Our experiments on an fMRI dataset demonstrate the superiority of the proposed approach against other single-task and multi-task

---

[5] The experimental codes are available at `https://github.com/smkia/MTNorm`.

alternatives in terms of the computational time complexity. This superiority was achieved without compromising the regression performance, and even with higher sensitivity to abnormal samples in the normative modeling paradigm. Our methodological contribution advances the current practices in the normative modeling from the single-voxel modeling to multi-voxel structural learning. For future work, we will consider enriching the proposed approach by embedding more biologically meaningful basis functions [7], structural modeling of non-stationary noise, and applying our method to clinical cohorts.

References

1. Abraham, A., Pedregosa, F., Eickenberg, M., Gervais, P., Mueller, A., Kossaifi, J., Gramfort, A., Thirion, B., Varoquaux, G.: Machine learning for neuroimaging with scikit-learn. Frontiers in Neuroinformatics 8, 14 (2014)
2. Alvarez, M., Lawrence, N.D.: Sparse convolved Gaussian processes for multi-output regression. In: Advances in neural information processing systems. pp. 57–64 (2009)
3. Álvarez, M.A., Lawrence, N.D.: Computationally efficient convolved multiple output Gaussian processes. Journal of Machine Learning Research 12, 1459–1500 (2011)
4. Alvarez, M.A., Luengo, D., Titsias, M.K., Lawrence, N.D.: Efficient multioutput Gaussian processes through variational inducing kernels. In: International Conference on Artificial Intelligence and Statistics. pp. 25–32 (2010)
5. Bonilla, E.V., Chai, K.M., Williams, C.: Multi-task Gaussian process prediction. In: Advances in neural information processing systems. pp. 153–160 (2008)
6. Boyle, P., Frean, M.: Multiple output Gaussian process regression. Tech. rep. (2005)
7. Huertas, I., Oldehinkel, M., van Oort, E.S., Garcia-Solis, D., Mir, P., Beckmann, C.F., Marquand, A.F.: A bayesian spatial model for neuroimaging data based on biologically informed basis functions. NeuroImage 161, 134 – 148 (2017)
8. Loan, C.F.: The ubiquitous kronecker product. Journal of Computational and Applied Mathematics 123(1), 85 – 100 (2000)
9. Marquand, A.F., Rezek, I., Buitelaar, J., Beckmann, C.F.: Understanding heterogeneity in clinical cohorts using normative models: beyond case-control studies. Biological psychiatry 80(7), 552–561 (2016)
10. Marquand, A.F., Wolfers, T., Mennes, M., Buitelaar, J., Beckmann, C.F.: Beyond lumping and splitting: A review of computational approaches for stratifying psychiatric disorders. Biological Psychiatry 1(5), 433 – 447 (2016)
11. Mirnezami, R., Nicholson, J., Darzi, A.: Preparing for precision medicine. New England Journal of Medicine 366(6), 489–491 (2012), pMID: 22256780
12. Miyawaki, Y., Uchida, H., Yamashita, O., aki Sato, M., Tanabe, H.C., Sadato, N., Kamitani, Y.: Visual image reconstruction from human brain activity using a combination of multiscale local image decoders. Neuron 60(5), 915 – 929 (2008)
13. Quinonero-Candela, J., Williams, C.K.: Approximation methods for gaussian process regression. Large-scale kernel machines pp. 203–224 (2007)
14. Rakitsch, B., Lippert, C., Borgwardt, K., Stegle, O.: It is all in the noise: Efficient multi-task Gaussian process inference with structured residuals. In: Advances in neural information processing systems. pp. 1466–1474 (2013)
15. Stegle, O., Lippert, C., Mooij, J.M., Lawrence, N.D., Borgwardt, K.M.: Efficient inference in matrix-variate gaussian models with iid observation noise. In: Advances in neural information processing systems. pp. 630–638 (2011)
16. Williams, C.K., Rasmussen, C.E.: Gaussian processes for regression. In: Advances in neural information processing systems. pp. 514–520 (1996)

Supplementary Materials

Throughout the supplementary materials we use the same notation introduced in the main text.

Useful Equations

For $\mathbf{A} \in \mathbb{R}^{M \times N}$, $\mathbf{B} \in \mathbb{R}^{P \times Q}$, and $\mathbf{C}$, $\mathbf{D}$ (with appropriate size) we have:

1. $\mathbf{A} = \mathbf{U_A S_A U_A^\top}$ is the eigenvalue decomposition of $\mathbf{A}$,
2. $(\mathbf{ACB})^{-1} = \mathbf{B^{-1} C^{-1} A^{-1}}$,
3. $(\mathbf{A} \otimes \mathbf{B})(\mathbf{C} \otimes \mathbf{D}) = \mathbf{AC} \otimes \mathbf{BD}$,
4. $(\mathbf{A} \otimes \mathbf{B})^{-1} = \mathbf{A^{-1}} \otimes \mathbf{B^{-1}}$,
5. the eigenvalue decomposition of $\mathbf{A} \otimes \mathbf{B} + \mathbf{I}$ is:
   $(\mathbf{U_A} \otimes \mathbf{U_B})(\mathbf{S_A} \otimes \mathbf{S_B} + \mathbf{I})(\mathbf{U_A^\top} \otimes \mathbf{U_B^\top})$,
6. $(\mathbf{A} \otimes \mathbf{B})vec(\mathbf{C}) = vec(\mathbf{BCA^\top})$,
7. $\ln|\mathbf{AC}| = \ln(|\mathbf{A}||\mathbf{C}|) = \ln|\mathbf{A}| + \ln|\mathbf{C}|$,
8. for $\mathbf{C} \in \mathbb{R}^{N \times N}$, $\frac{\mathrm{d}}{\mathrm{d}x} \ln|\mathbf{C}| = Tr[\mathbf{C}^{-1}\frac{\mathrm{d}\mathbf{C}}{\mathrm{d}x}]$,
9. $Tr[\mathbf{ACBD}] = Tr[\mathbf{CBDA}] = Tr[\mathbf{BDAC}] = Tr[\mathbf{DACB}]$.

Efficient Mean Prediction

Eq. 6(a) is derived from Eq. 5(a) as follows:

$$
\begin{aligned}
vec(\mathbf{M}^*) &= (\mathbf{BCB^\top} \otimes \mathbf{R}^*)(\mathbf{BCB^\top} \otimes \mathbf{R} + \sigma^2\mathbf{I})^{-1}vec(\mathbf{Y}) \\
&= (\mathbf{BCB^\top} \otimes \mathbf{R}^*)(\mathbf{BU_C S_C U_C^\top B^\top} \otimes \mathbf{U_R S_R U_R}^\top + \sigma^2\mathbf{I})^{-1}vec(\mathbf{Y}) \\
&= (\mathbf{BCB^\top} \otimes \mathbf{R}^*)[(\mathbf{BU_C} \otimes \mathbf{U_R})(\mathbf{S_C} \otimes \mathbf{S_R} + \sigma^2\mathbf{I})(\mathbf{U_C^\top B^\top} \otimes \mathbf{U_R^\top})]^{-1}vec(\mathbf{Y}) \\
&= (\mathbf{BCB^\top} \otimes \mathbf{R}^*)(\mathbf{BU_C} \otimes \mathbf{U_R})(\mathbf{S_C} \otimes \mathbf{S_R} + \sigma^2\mathbf{I})^{-1}(\mathbf{U_C^\top B^\top} \otimes \mathbf{U_R^\top})vec(\mathbf{Y}) \\
&= (\mathbf{BCB^\top} \otimes \mathbf{R}^*)(\mathbf{BU_C} \otimes \mathbf{U_R})(\mathbf{S_C} \otimes \mathbf{S_R} + \sigma^2\mathbf{I})^{-1}vec(\mathbf{U_R^\top Y B U_C}) \\
&= (\mathbf{BC}\underbrace{\mathbf{B^\top B}}_{\mathbf{I}}\mathbf{U_C} \otimes \mathbf{R}^*\mathbf{U_R})\underbrace{diag[(\mathbf{S_C} \otimes \mathbf{S_R} + \sigma^2\mathbf{I})^{-1}] \odot vec(\mathbf{U_R^\top Y B U_C})}_{vec(\tilde{\mathbf{Y}})} \\
&= \mathbf{R}^*\mathbf{U_R}\tilde{\mathbf{Y}}\mathbf{U_C^\top CB^\top} \quad .
\end{aligned}
$$

Efficient Variance Prediction

Eq. 6(b) is derived from Eq. 5(b) as follows:

$$
\begin{aligned}
\mathbf{V}^* &= (\mathbf{BCB^\top} \otimes \mathbf{R}^{**}) - (\mathbf{BCB^\top} \otimes \mathbf{R}^*)\underbrace{(\mathbf{BCB^\top} \otimes \mathbf{R} + \sigma^2\mathbf{I})^{-1}}_{\mathbf{K}^{-1}}(\mathbf{BCB^\top} \otimes \mathbf{R}^{*\top}) \\
&= (\mathbf{BCB^\top} \otimes \mathbf{R}^{**}) - (\mathbf{BCB^\top} \otimes \mathbf{R}^*)(\mathbf{BU_C} \otimes \mathbf{U_R})(\mathbf{S_C} \otimes \mathbf{S_R} + \sigma^2\mathbf{I})^{-1} \\
&\quad (\mathbf{U_C^\top B^\top} \otimes \mathbf{U_R^\top})(\mathbf{BCB^\top} \otimes \mathbf{R}^{*\top}) \\
&= (\mathbf{BCB^\top} \otimes \mathbf{R}^{**}) - (\mathbf{BCU_C} \otimes \mathbf{R}^*\mathbf{U_R})\underbrace{(\mathbf{S_C} \otimes \mathbf{S_R} + \sigma^2\mathbf{I})^{-1}}_{\tilde{\mathbf{K}}^{-1}}(\mathbf{U_C^\top CB^\top} \otimes \mathbf{U_R^\top R}^{*\top}) \quad .
\end{aligned}
$$

Efficient Log Marginal Likelihood Evaluation

Eq. 7 is derived as follows:

$$
\begin{aligned}
\mathcal{L} &= -\frac{N \times T}{2} \ln(2\pi) - \frac{1}{2} \ln |\mathbf{K}| - \frac{1}{2} vec(\mathbf{Y})^\top \mathbf{K}^{-1} vec(\mathbf{Y}) \\
&= -\frac{N \times T}{2} \ln(2\pi) - \frac{1}{2} \ln \left| \mathbf{BCB}^\top \otimes \mathbf{R} + \sigma^2 \mathbf{I} \right| - \frac{1}{2} vec(\mathbf{Y})^\top (\mathbf{BCB}^\top \otimes \mathbf{R} + \sigma^2 \mathbf{I})^{-1} vec(\mathbf{Y}) \\
&= -\frac{N \times T}{2} \ln(2\pi) - \frac{1}{2} \ln \left| (\mathbf{BU_C} \otimes \mathbf{U_R})(\mathbf{S_C} \otimes \mathbf{S_R} + \sigma^2 \mathbf{I})(\mathbf{U_C^\top B^\top} \otimes \mathbf{U_R^\top}) \right| \\
&\quad - \frac{1}{2} vec(\mathbf{Y})^\top (\mathbf{BU_C} \otimes \mathbf{U_R})(\mathbf{S_C} \otimes \mathbf{S_R} + \sigma^2 \mathbf{I})^{-1}(\mathbf{U_C^\top B^\top} \otimes \mathbf{U_R^\top}) vec(\mathbf{Y}) \\
&= -\frac{N \times T}{2} \ln(2\pi) - \frac{1}{2} \underbrace{\ln \left| (\mathbf{U_C^\top B^\top} \otimes \mathbf{U_R^\top})(\mathbf{BU_C} \otimes \mathbf{U_R}) \right|}_{\ln |\mathbf{I}| = 0} - \frac{1}{2} \ln \left| (\mathbf{S_C} \otimes \mathbf{S_R} + \sigma^2 \mathbf{I}) \right| \\
&\quad - \frac{1}{2} vec(\mathbf{U_R^\top Y B U_C})^\top \underbrace{diag[(\mathbf{S_C} \otimes \mathbf{S_R} + \sigma^2 \mathbf{I})^{-1}] \odot vec(\mathbf{U_R^\top Y B U_C})}_{vec(\tilde{\mathbf{Y}})} \\
&= -\frac{N \times T}{2} \ln(2\pi) - \frac{1}{2} \underbrace{\ln \left| (\mathbf{S_C} \otimes \mathbf{S_R} + \sigma^2 \mathbf{I}) \right|}_{|\tilde{\mathbf{K}}|} - \frac{1}{2} vec(\mathbf{U_R^\top Y B U_C})^\top vec(\tilde{\mathbf{Y}}) \quad .
\end{aligned}
$$

Derivatives of $\mathcal{L}$ with Respect to Parameters

In the optimization process, the derivatives of $\mathcal{L}$ with respect to $\theta_\mathbf{C} \in \Theta_\mathbf{C}$, $\theta_\mathbf{R} \in \Theta_\mathbf{R}$, and $\theta_{\sigma^2} \in \Theta_{\sigma^2}$ can be efficiently computed as follows:

Gradients of $\mathcal{L}$ with Respect to $\theta_\mathbf{C}$:

$$
\frac{\partial \mathcal{L}}{\partial \theta_\mathbf{C}} = -\frac{1}{2} diag(\tilde{\mathbf{K}}^{-1})^\top [diag(\mathbf{U_C^\top} \frac{\partial \mathbf{C}}{\partial \theta_\mathbf{C}} \mathbf{U_C}) \otimes diag(\mathbf{S_R})] + \frac{1}{2} vec(\tilde{\mathbf{Y}})^\top vec(\mathbf{S_R} \tilde{\mathbf{Y}} \mathbf{U_C^\top} \frac{\partial \mathbf{C}}{\partial \theta_\mathbf{C}} \mathbf{U_C}),
$$

where the determinant term of the above equation is derived by computing the derivative of $\ln |\mathbf{K}|$:

$$
\begin{aligned}
\frac{\partial \ln |\mathbf{K}|}{\partial \theta_\mathbf{C}} &= \frac{\partial}{\partial \theta_\mathbf{C}} [\ln \left| \mathbf{BCB}^\top \otimes \mathbf{R} + \sigma^2 \mathbf{I} \right|] = Tr[(\mathbf{BCB}^\top \otimes \mathbf{R} + \sigma^2 \mathbf{I})^{-1} \frac{\partial}{\partial \theta_\mathbf{C}} (\mathbf{BCB}^\top \otimes \mathbf{R} + \sigma^2 \mathbf{I})] \\
&= Tr[(\mathbf{BU_C} \otimes \mathbf{U_R})(\mathbf{S_C} \otimes \mathbf{S_R} + \sigma^2 \mathbf{I})^{-1}(\mathbf{U_C^\top B^\top} \otimes \mathbf{U_R^\top})(\mathbf{B} \frac{\partial \mathbf{C}}{\partial \theta_\mathbf{C}} \mathbf{B}^\top \otimes \mathbf{R})] \\
&= Tr[(\mathbf{S_C} \otimes \mathbf{S_R} + \sigma^2 \mathbf{I})^{-1}(\mathbf{U_C^\top B^\top} \otimes \mathbf{U_R^\top})(\mathbf{B} \frac{\partial \mathbf{C}}{\partial \theta_\mathbf{C}} \mathbf{B}^\top \otimes \mathbf{R})(\mathbf{BU_C} \otimes \mathbf{U_R})] \\
&= Tr[\tilde{\mathbf{K}}^{-1}(\mathbf{U_C^\top B^\top B} \frac{\partial \mathbf{C}}{\partial \theta_\mathbf{C}} \mathbf{B}^\top \mathbf{BU_C} \otimes \mathbf{U_R^\top R U_R})] = Tr[\tilde{\mathbf{K}}^{-1}(\mathbf{U_C^\top} \frac{\partial \mathbf{C}}{\partial \theta_\mathbf{C}} \mathbf{U_C} \otimes \mathbf{S_R})] \\
&= diag(\tilde{\mathbf{K}}^{-1})^\top [diag(\mathbf{U_C^\top} \frac{\partial \mathbf{C}}{\partial \theta_\mathbf{C}} \mathbf{U_C}) \otimes diag(\mathbf{S_R})] \quad ,
\end{aligned}
$$

and for the squared term we have:

$$\frac{\partial}{\partial \theta_{\mathbf{C}}}[vec(\mathbf{Y})^\top \mathbf{K}^{-1} vec(\mathbf{Y})] = \frac{\partial}{\partial \theta_{\mathbf{C}}}[vec(\mathbf{Y})^\top (\mathbf{BCB}^\top \otimes \mathbf{R} + \sigma^2 \mathbf{I})^{-1} vec(\mathbf{Y})]$$

$$= -vec(\mathbf{Y})^\top (\mathbf{BCB}^\top \otimes \mathbf{R} + \sigma^2 \mathbf{I})^{-1}[\frac{\partial}{\partial \theta_{\mathbf{C}}}(\mathbf{BCB}^\top \otimes \mathbf{R} + \sigma^2 \mathbf{I})](\mathbf{BCB}^\top \otimes \mathbf{R} + \sigma^2 \mathbf{I})^{-1} vec(\mathbf{Y})$$

$$= -vec(\mathbf{Y})^\top (\mathbf{BU_C} \otimes \mathbf{U_R})(\mathbf{S_C} \otimes \mathbf{S_R} + \sigma^2 \mathbf{I})^{-1}(\mathbf{U_C^\top B^\top} \otimes \mathbf{U_R^\top})(\mathbf{B}\frac{\partial \mathbf{C}}{\partial \theta_{\mathbf{C}}}\mathbf{B}^\top \otimes \mathbf{R})$$

$$(\mathbf{BU_C} \otimes \mathbf{U_R})(\mathbf{S_C} \otimes \mathbf{S_R} + \sigma^2 \mathbf{I})^{-1}(\mathbf{U_C^\top B^\top} \otimes \mathbf{U_R^\top})vec(\mathbf{Y})$$

$$= -[vec(\mathbf{U_R^\top Y B U_C})^\top \odot diag(\tilde{\mathbf{K}}^{-1})](\mathbf{U_C^\top B^\top B}\frac{\partial \mathbf{C}}{\partial \theta_{\mathbf{C}}}\mathbf{B^\top B U_C} \otimes \mathbf{U_R^\top R U_R})$$

$$\underbrace{[diag(\tilde{\mathbf{K}}^{-1}) \odot vec(\mathbf{U_R^\top Y B U_C})]}_{vec(\tilde{\mathbf{Y}})} = -vec(\tilde{\mathbf{Y}})^\top vec(\mathbf{S_R}\tilde{\mathbf{Y}}\mathbf{U_C^\top}\frac{\partial \mathbf{C}}{\partial \theta_{\mathbf{C}}}\mathbf{U_C}) \quad .$$

Gradients of $\mathcal{L}$ with Respect to $\theta_{\mathbf{R}}$:

$$\frac{\partial \mathcal{L}}{\partial \theta_{\mathbf{R}}} = -\frac{1}{2}diag(\tilde{\mathbf{K}}^{-1})^\top[diag(\mathbf{S_C}) \otimes diag(\mathbf{U_R^\top}\frac{\partial \mathbf{R}}{\partial \theta_{\mathbf{R}}}\mathbf{U_R})] + \frac{1}{2}vec(\tilde{\mathbf{Y}})^\top vec(\mathbf{U_R^\top}\frac{\partial \mathbf{R}}{\partial \theta_{\mathbf{R}}}\mathbf{U_R}\tilde{\mathbf{Y}}\mathbf{S_C}), \quad (8)$$

where the determinant term of the above equation is derived by computing the derivative of $\ln |\mathbf{K}|$:

$$\frac{\partial \ln |\mathbf{K}|}{\partial \theta_{\mathbf{R}}} = \frac{\partial}{\partial \theta_{\mathbf{R}}}[\ln \left|\mathbf{BCB}^\top \otimes \mathbf{R} + \sigma^2 \mathbf{I}\right|] = Tr[(\mathbf{BCB}^\top \otimes \mathbf{R} + \sigma^2 \mathbf{I})^{-1}\frac{\partial}{\partial \theta_{\mathbf{R}}}(\mathbf{BCB}^\top \otimes \mathbf{R} + \sigma^2 \mathbf{I})]$$

$$= Tr[(\mathbf{BU_C} \otimes \mathbf{U_R})(\mathbf{S_C} \otimes \mathbf{S_R} + \sigma^2 \mathbf{I})^{-1}(\mathbf{U_C^\top B^\top} \otimes \mathbf{U_R^\top})(\mathbf{BCB}^\top \otimes \frac{\partial \mathbf{R}}{\partial \theta_{\mathbf{R}}})]$$

$$= Tr[(\mathbf{S_C} \otimes \mathbf{S_R} + \sigma^2 \mathbf{I})^{-1}(\mathbf{U_C^\top B^\top} \otimes \mathbf{U_R^\top})(\mathbf{BCB}^\top \otimes \frac{\partial \mathbf{R}}{\partial \theta_{\mathbf{R}}})(\mathbf{BU_C} \otimes \mathbf{U_R})]$$

$$= Tr[\tilde{\mathbf{K}}^{-1}(\mathbf{U_C^\top B^\top BCB^\top B U_C} \otimes \mathbf{U_R^\top}\frac{\partial \mathbf{R}}{\partial \theta_{\mathbf{R}}}\mathbf{U_R})] = Tr[\tilde{\mathbf{K}}^{-1}(\mathbf{S_C} \otimes \mathbf{U_R^\top}\frac{\partial \mathbf{R}}{\partial \theta_{\mathbf{R}}}\mathbf{U_R})]$$

$$= diag(\tilde{\mathbf{K}}^{-1})^\top[diag(\mathbf{S_C}) \otimes diag(\mathbf{U_R^\top}\frac{\partial \mathbf{R}}{\partial \theta_{\mathbf{R}}}\mathbf{U_R})] \quad ,$$

and for the squared term we have:

$$\frac{\partial}{\partial \theta_{\mathbf{R}}}[vec(\mathbf{Y})^\top \mathbf{K}^{-1} vec(\mathbf{Y})] = \frac{\partial}{\partial \theta_{\mathbf{R}}}[vec(\mathbf{Y})^\top (\mathbf{BCB}^\top \otimes \mathbf{R} + \sigma^2 \mathbf{I})^{-1} vec(\mathbf{Y})]$$

$$= -vec(\mathbf{Y})^\top (\mathbf{BCB}^\top \otimes \mathbf{R} + \sigma^2 \mathbf{I})^{-1}[\frac{\partial}{\partial \theta_{\mathbf{R}}}(\mathbf{BCB}^\top \otimes \mathbf{R} + \sigma^2 \mathbf{I})](\mathbf{BCB}^\top \otimes \mathbf{R} + \sigma^2 \mathbf{I})^{-1} vec(\mathbf{Y})$$

$$= -vec(\mathbf{Y})^\top (\mathbf{BU_C} \otimes \mathbf{U_R})(\mathbf{S_C} \otimes \mathbf{S_R} + \sigma^2 \mathbf{I})^{-1}(\mathbf{U_C^\top B^\top} \otimes \mathbf{U_R^\top})(\mathbf{BCB}^\top \otimes \frac{\partial \mathbf{R}}{\partial \theta_{\mathbf{R}}})$$

$$(\mathbf{BU_C} \otimes \mathbf{U_R})(\mathbf{S_C} \otimes \mathbf{S_R} + \sigma^2 \mathbf{I})^{-1}(\mathbf{U_C^\top B^\top} \otimes \mathbf{U_R^\top})vec(\mathbf{Y})$$

$$= -[vec(\mathbf{U_R^\top Y B U_C})^\top \odot diag(\tilde{\mathbf{K}}^{-1})](\mathbf{U_C^\top B^\top BCB^\top B U_C} \otimes \mathbf{U_R^\top}\frac{\partial \mathbf{R}}{\partial \theta_{\mathbf{R}}}\mathbf{U_R})$$

$$\underbrace{[diag(\tilde{\mathbf{K}}^{-1}) \odot vec(\mathbf{U_R^\top Y B U_C})]}_{vec(\tilde{\mathbf{Y}})} = -vec(\tilde{\mathbf{Y}})^\top vec(\mathbf{U_R^\top}\frac{\partial \mathbf{R}}{\partial \theta_{\mathbf{R}}}\mathbf{U_R}\tilde{\mathbf{Y}}\mathbf{S_C}) \quad .$$

Gradients of $\mathcal{L}$ with Respect to θ_{σ^2}:

$$\frac{\partial \mathcal{L}}{\partial \theta_{\sigma^2}} = -\frac{1}{2}\frac{\partial \sigma^2}{\partial \theta_{\sigma^2}}[Tr[\tilde{\mathbf{K}}^{-1}] + vec(\tilde{\mathbf{Y}})^\top vec(\tilde{\mathbf{Y}})] \quad, \tag{9}$$

where the determinant term of the above equation is derived by computing the derivative of $\ln|\mathbf{K}|$:

$$\frac{\partial \ln|\mathbf{K}|}{\partial \theta_{\sigma^2}} = \frac{\partial}{\partial \theta_{\sigma^2}}[\ln\left|\mathbf{B}\mathbf{C}\mathbf{B}^\top \otimes \mathbf{R} + \sigma^2 \mathbf{I}\right|] = Tr[(\mathbf{B}\mathbf{C}\mathbf{B}^\top \otimes \mathbf{R} + \sigma^2 \mathbf{I})^{-1}\frac{\partial \sigma^2}{\partial \theta_{\sigma^2}}]$$

$$= \frac{\partial \sigma^2}{\partial \theta_{\sigma^2}}Tr[(\mathbf{B}\mathbf{U_C} \otimes \mathbf{U_R})(\mathbf{S_C} \otimes \mathbf{S_R} + \sigma^2 \mathbf{I})^{-1}(\mathbf{U_C^\top}\mathbf{B}^\top \otimes \mathbf{U_R^\top})]$$

$$= \frac{\partial \sigma^2}{\partial \theta_{\sigma^2}}Tr[(\mathbf{S_C} \otimes \mathbf{S_R} + \sigma^2 \mathbf{I})^{-1}(\mathbf{U_C^\top}\mathbf{B}^\top \otimes \mathbf{U_R^\top})(\mathbf{B}\mathbf{U_C} \otimes \mathbf{U_R})]$$

$$= \frac{\partial \sigma^2}{\partial \theta_{\sigma^2}}Tr[\tilde{\mathbf{K}}^{-1}(\mathbf{U_C^\top}\mathbf{B}^\top\mathbf{B}\mathbf{U_C} \otimes \mathbf{U_R^\top}\mathbf{U_R})] = \frac{\partial \sigma^2}{\partial \theta_{\sigma^2}}Tr[\tilde{\mathbf{K}}^{-1}] \quad,$$

and for the squared term we have:

$$\frac{\partial}{\partial \theta_{\sigma^2}}[vec(\mathbf{Y})^\top \mathbf{K}^{-1}vec(\mathbf{Y})] = \frac{\partial}{\partial \theta_{\sigma^2}}[vec(\mathbf{Y})^\top(\mathbf{B}\mathbf{C}\mathbf{B}^\top \otimes \mathbf{R} + \sigma^2 \mathbf{I})^{-1}vec(\mathbf{Y})]$$

$$= -vec(\mathbf{Y})^\top(\mathbf{B}\mathbf{C}\mathbf{B}^\top \otimes \mathbf{R} + \sigma^2 \mathbf{I})^{-1}[\frac{\partial}{\partial \theta_{\sigma^2}}(\mathbf{B}\mathbf{C}\mathbf{B}^\top \otimes \mathbf{R} + \sigma^2 \mathbf{I})](\mathbf{B}\mathbf{C}\mathbf{B}^\top \otimes \mathbf{R} + \sigma^2 \mathbf{I})^{-1}vec(\mathbf{Y})$$

$$= -vec(\mathbf{Y})^\top(\mathbf{B}\mathbf{U_C} \otimes \mathbf{U_R})(\mathbf{S_C} \otimes \mathbf{S_R} + \sigma^2 \mathbf{I})^{-1}(\mathbf{U_C^\top}\mathbf{B}^\top \otimes \mathbf{U_R^\top})\frac{\partial \sigma^2}{\partial \theta_{\sigma^2}}$$

$$(\mathbf{B}\mathbf{U_C} \otimes \mathbf{U_R})(\mathbf{S_C} \otimes \mathbf{S_R} + \sigma^2 \mathbf{I})^{-1}(\mathbf{U_C^\top}\mathbf{B}^\top \otimes \mathbf{U_R^\top})vec(\mathbf{Y})$$

$$= -\frac{\partial \sigma^2}{\partial \theta_{\sigma^2}}[vec(\mathbf{U_R^\top}\mathbf{Y}\mathbf{B}\mathbf{U_C})^\top \odot diag(\tilde{\mathbf{K}}^{-1})](\mathbf{U_C^\top}\mathbf{B}^\top\mathbf{B}\mathbf{U_C} \otimes \mathbf{U_R^\top}\mathbf{U_R})$$

$$\underbrace{[diag(\tilde{\mathbf{K}}^{-1}) \odot vec(\mathbf{U_R^\top}\mathbf{Y}\mathbf{B}\mathbf{U_C})]}_{vec(\tilde{\mathbf{Y}})} = -\frac{\partial \sigma^2}{\partial \theta_{\sigma^2}}vec(\tilde{\mathbf{Y}})^\top vec(\tilde{\mathbf{Y}}) \quad.$$

Normative Modeling

Let $\hat{y}_{ij}$ and σ_{ij}^2 be the prediction mean and variance of the ith test sample at the jth voxel. Further, let σ_{nj}^2 be the variance of the noise that is estimated by GPR at the jth voxel. Then the normative probability map (NPM) for the ith sample at jth voxel is defined as follows:

$$NPM_{ij} = \frac{y_{ij} - \hat{y}_{ij}}{\sqrt{\sigma_{ij}^2 + \sigma_{nj}^2}} \quad,$$

where y_{ij} is the true output. Having computed NPMs for all samples and brain locations, the abnormality index of each sample can be computed by fitting a generalized extreme value distribution (GEVD). We fit GEVD on the distribution of robust means of top 5% voxels (in absolute value) across all NPMs. The resulting distribution is used to compute the probability of each sample being abnormal.