Mathematics of kernel normative modeling

In this section, we provide a brief overview of methods from Graph Signal Processing that are used to define the kernel normative model.

GSP paradigm:

The GSP approach is a method that can convert an arbitrary *signal* defined over a network structure (referred to as the **graph signal**) from its original domain to a graph frequency domain.

Let's describe this mathematically using the high-resolution cortical brain example.

Any high-resolution cortical signal (such as the thickness of an individual, or the boundaries of an atlas region) can be viewed as a vector $\overrightarrow{X} \in R^{N_v}$ where N_v denotes the number of high-resolution vertices on the cortical surface.

Now, if we define a network to describe the connectivity topology between these high-resolution vertices, then any brain signal $\overrightarrow{X}$ can essentially be treated as a graph signal.

Graph Fourier Transform (GFT)

Here, we'll take a short detour to describe how GFT can be used to transform the signal from its original domain to the graph frequency domain.

Essentially, we build an orthonormal basis (a set of eigenmodes that are orthogonal and have a unit norm) for a N_v dimensional space of all possible graph signals. Hence, theoretically, this basis can explain any arbitrary graph signal. In other words, with the complete basis (of N_v eigenmodes), any graph signal can be described as a linear combination of the eigenmodes.

This basis is generated from the network's adjacency matrix $A \in R^{N_v} \times R^{N_v}$ and eigenmodes are ordered based on the graph frequency.

First, a symmetric normalized Laplacian matrix $L = L^{sim} \in R^{N_v} \times R^{N_v}$ is computed from the adjacency matrix A (we refer to this as the laplacian matrix for simplicity, but actually there are a few different ways to compute Laplacians, see this):

$$L = I - D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$$

Where D is a diagonal matrix whose diagonal elements denote the degree of every vertex (row/column sum of A).

Footnote: In GSP literature, this matrix (and other similar matrices) is referred to as the Graph Shift Operator. As it basically can be viewed as a rule to shift information on the graph. Basically, L^{sim} describes a shift operation in which information can be sourced from any vertex, and is distributed according to the connectivity weights.

Now, this laplacian matrix is decomposed using Singular Value Decomposition (SVD) to result in the eigenmodes (also referred to as eigendecompositions):

$$L = V \Lambda V^{-1}$$

In this decomposition $\Lambda \in R^{N_v} \times R^{N_v}$ is a diagonal matrix with the diagonal values being the eigenvalues, and $V \in R^{N_v} \times R^{N_v}$ is a list of N_v eigenvectors, each associated with an eigenvalue. As the eigenvectors are ordered by their corresponding eigenvalue, starting eigenmodes are topologically smooth (vertices that are connected are more likely to have similar values) and are hence referred to as lower graph frequencies. Similarly, the last eigenmodes are topologically chaotic/varying and are considered to explain higher graph frequencies. The kernel visualizations of our brain eigenmodes depict this.

Now, these eigenvectors can be used to encode a brain signal (graph signal) to a graph frequency domain. This is formally known as a Graph Fourier Transform:

$$\overrightarrow{\tilde{X}} = V^T \overrightarrow{X}$$

So $\overrightarrow{\tilde{X}}$ is called the Fourier transformed signal in the frequency domain. This transformed signal contains all the information about the original signal and can in fact be used to reconstruct the original signal:

$$\overrightarrow{X} = V \overrightarrow{\tilde{X}}$$

One interesting observation about these eigenmodes is that $V^T = V^{-1}$ which comes in very handy.

Note that *if the complete graph Fourier basis of eigenmodes were used*, then $\overrightarrow{\tilde{X}}$ would have exactly the same dimension as the original signal $\overrightarrow{X}$. Now, this brings us to the next topic which is Graph Signal Filtering. Graph signal filtering is used to **reduce** the dimensionality of a graph signal when encoding into the frequency domain.

Graph Signal Filtering

Very similar to the idea of filtering discrete time series using a Fourier transform, graph signals can be filtered by a graph Fourier transform (in fact the former is a special case of the latter, see this).

To explain succinctly, a graph signal $\overrightarrow{X}$ can be transformed to the graph frequency domain as described above, $V^T \overrightarrow{X}$. Next a frequency filter, in the form of a diagonal matrix $F \in R^{N_v} \times R^{N_v}$ can be used to filter the signal in the frequency domain, $FV^T\overrightarrow{X}$. To be exact, F is the diagonal matrix for which the diagonal elements are all in the range of $[0, 1]$. For instance, F can be a diagonal matrix with the first 10 diagonal elements equal to one, and the rest equal to zero. This would be an example of a low-pass graph filter because only the lower frequencies survive filtering.

Once the signal is filtered in the graph frequency domain, it can be converted back to the graph domain to construct a filtered graph signal:

$$\overrightarrow{X}_F = VFV^T\overrightarrow{X}$$

Here, $\overrightarrow{X}_F$ denotes the filtered signal.

Now, this brings us to how we can use graph signal filtering to perform dimensionality reduction. With the assumption that the majority of graph signals (brain signals) that we are interested in are topologically or spatially smooth, we can use a low-pass filter of the first N_{LP} eigenmodes to decode the signal. Reconstruction of this signal would thus provide a low-pass approximation of the original brain signal.

In fact, a binary low-pass filter is also computationally appealing, as:

- The SVD computation becomes easier as we only need to compute a subset of the eigendecomposition
- The subsequent normative modeling step will be less computational as instead of requiring N_v separate normative fits, we only need N_{LP} separate fits.

This not-very short introduction to GSP brings us to the main topic of discussion, which is how we can use the GSP paradigm to perform smooth high-resolution normative modeling of the developmental trajectories of the brain.

In the ensuing section we'll describe:

- The adjacency matrix used for GSP on the brain
- How normative modeling can be fitted over brain eigenmodes
- How a fitted model can be used to make general predictions

GSP on brain

To perform GSP techniques on the brain, we first need to use a network structure to describe the brain as a graph. Here, I used a very simplistic topological structure that only takes information regarding the spatial positioning of high-resolution vertices into account.

The adjacency structure on the brain

Here, to build an adjacency structure for the brain, I tried only using spatial information. This is mainly because we expect the high-resolution variations in a given brain signal to be relatively smooth with respect to the geometry. Hence, a sparse high-resolution adjacency structure was constructed in which vertices were only connected to other nearby vertices. Proximity was quantified by the geodesic distance between vertices over the surface mesh (an inflated mesh was used). Connections were weighted by a Gaussian distribution such that nodes were more strongly connected to those closer and loosely connected to those further away. We denoted this spatial adjacency structure by A earlier.

Note: we can change this structure and see if alternative structures provide a more appropriate kernel.

GFT on brain

Next, we compute the SVD of the laplacian matrix. To reduce computations, instead of all eigenvectors, we only derive the first $N_{LP} = 2000$ eigenmodes. Hence, we can use this partial decomposition $V_{LP} \in R^{N_v} \times R^{N_{LP}}$ to approximate a full signal reconstruction. This was any brain signal $\overrightarrow{X} \in R^{N_v}$ can be encoded into a dimensionality reduced spectral domain encoding $\overrightarrow{X}_{LP} \in R^{N_{LP}}$:

$$\overrightarrow{X}_{LP} = {V_{LP}}^T \overrightarrow{X}$$

It is important to note that V_{LP} is in fact akin to a low-pass GFT of the full eigendecomposition (and hence the subscript LP is used to denote a low-pass graph Fourier filter).

Normative kernel modeling

So far, by computing V_{LP} we have a way to reduce any high-resolution characteristic to a low-pass spectral representation. In this section, we describe how this dimensionality reduction kernel can be used to fit N_{LP} separate normative models that each describe the developmental trajectory of a feature (such as thickness) projected onto a specific eigenmode.

Dimensionality reduction

When dealing with the cortical thickness, the high-resolution data can be described as a matrix $T \in R^{N_v} \times R^{N_p}$ where N_p denotes the number of participants for which data has been collected. Essentially, for each participant, we have a high-resolution signal indicating the thickness of any vertex.

In the kernel-based normative modeling context, this information is first reduced by projecting the thickness information into a low-pass spectral domain:

$$\tilde{T}_{LP} = {V_{LP}}^T T$$

Now $\tilde{T}_{LP} \in R^{N_{LP}} \times R^{N_p}$ contains the projection of thickness onto the first N_{LP} eigenmodes and is used in the proceeding steps to fit a normative model.

Normative fits of eigenmodes

$\tilde{T}_{LP}(i)$ denotes the ith row of $\tilde{T}_{LP}$ and can be viewed as a vector with length N_p that contains the thickness loading of the ith eigenmode for all individuals. Now we can assume that this loading is itself a normative characteristic of individuals that changes by age, gender, and site. Hence, for every eigenmode, the following normative model can be independently fitted (for the sake of simplicity a normal distribution is used):

$$\tilde{T}_{LP}(i) \sim N(\mu_i, \sigma_i)$$

$$\mu_i = f_{\mu_i}(age, sex, site)$$

$$\sigma_i = f_{\sigma_i}(age, sex, site)$$

To put it simply, for every eigenmode, the mean and standard deviation across subjects are modeled as a function of the covariates (age, sex, and site). Hence, for normative kernel modeling, we essentially fit N_{LP} separate normative models to assess variations of different spectral frequencies of thickness.

Combining eigennorms

Now that we have fitted norms for every eigenmode, we next need to describe the rules by which these norms can be combined to answer arbitrary high-resolution queries.

Signal-based queries

A signal-based query is one that aims to answer a question regarding the normative trajectories of a specific signal. For instance, normative trajectories of the mean cortical thickness can be viewed as a specific example of a signal-based query. In general, a signal-based query starts with a brain signal of interest $\overrightarrow{X_Q} \in R^{N_v}$. In the specific case of mean thickness $\overrightarrow{X}_{MT}$, this brain signal is a vector with equal elements that sum to one:

$$\forall i \in \{1..N_v\} : \overrightarrow{X}_{MT}(i) = \frac{1}{N_v}$$

The reason that this signal explains the mean thickness operation is that for every participant i with high-resolution thickness information of $\overrightarrow{T_{(p=i)}}$ (ith column of T) the mean thickness can be computed by the dot product of $\overrightarrow{X}_{MT}.\overrightarrow{T_{(p=i)}}$.

So to generalize, for any brain signal query $\overrightarrow{X_Q}$ that describes a linear operation based on high-resolution values of thickness (e.g. mean thickness, the thickness of a region, a thickness contrast or comparison), that linear operation can be computed for all individuals in the group with a matrix multiplication with the signal vector:

$$\overrightarrow{T_{X_Q}} = \overrightarrow{X_Q}T$$

In the case of mean thickness, $\overrightarrow{T_{X_{MT}}}$ would be a vector of length N_p that stores the mean thickness of all individuals.

Approximating the signal-based query with the eigenmodes To predict the normative range of an arbitrary query, we first approximate the query signal X_Q using an eigenmode decomposition. This operation (i) reduces the dimensionality of the signal query (ii) transforms the query to the graph frequency domain. This equation describes this transformation:

$$\overrightarrow{\tilde{X}_{Q(LP)}} = {V_{LP}}^T\overrightarrow{X_Q}$$

Note that the matrix multiplication with the low-pass eigenmodes ${V_{LP}}^T$ (first N_{LP} eigenmodes) results in an approximate reconstruction of the original query which only contains the lower frequencies in the query and is hence going to be spatially smoother. Based on the GSP explanations earlier, this frequency domain query can be transformed back to the brain signal domain to reconstruct the low-pass filtered query:

$$\overrightarrow{X_{Q(LP)}} = V_{LP}\overrightarrow{\tilde{X}_{Q(LP)}}$$

Modeling the normative trajectories of signal queries We aim to use this eigenmode decomposition to relate the normative trajectories of the signal query to the precomputed normative trajectories of the eigenmodes. Essentially, $\overrightarrow{\tilde{X}_{Q(LP)}}$ is a query in the graph frequency domain that relates the low-pass approximation of the original signal query to a linear combination of eigenmodes. In other words, the signal query can be approximated by a linear sum of the eigenmodes. Given that the normative trajectories of the eigenmodes are modeled as normally distributed random variables, the linear combination of these normally distributed is also going to be a normally distributed variable. We hence use some statistical modeling to relate the normative trajectory of the low-pass signal query (a posterior distribution) to the normative trajectory of the eigennorms (the prior distribution).

Modeling the mean The mean (expected value) of a normally distributed random variable that is a linear combination of multiple normally distributed random variables is going to be the linear combination of their means. Or to put it more simply:

$E[a_1X_1 + a_2X_2 + ... + a_kX_k] = a_1E[X_1] + a_2E[X_2] + ... + a_kE[X_k]$

Hence, we could use this information to construct the mean of any arbitrary signal query. First, we construct a vector $\overrightarrow{M} \in R^{N_{LP}}$ in which the ith element contains the fitted mean of the ith eigenmode at a given (age, sex, site). Note that this was earlier denoted by μ_i, so in a sense, $\overrightarrow{M}$ is a vectorization of these mean values.

The normative mean of low-pass filtered query signal operating on a hypothetical high-resolution thickness vector $\overrightarrow{T}$ can thus be written as:

$$\begin{aligned} E[\overrightarrow{X_{Q(LP)}}.\overrightarrow{T}] &= E[\overrightarrow{T}.\overrightarrow{X_{Q(LP)}}] \\ &= E[\overrightarrow{T}V_{LP}\overrightarrow{\tilde{X}_{Q(LP)}}] \\ &= E[\overrightarrow{T}V_{LP}]\overrightarrow{\tilde{X}_{Q(LP)}} \\ &= E[\overrightarrow{\tilde{T}_{LP}}]\overrightarrow{\tilde{X}_{Q(LP)}} \\ &= \overrightarrow{M}.\overrightarrow{\tilde{X}_{Q(LP)}} \end{aligned}$$

The equation above is basically telling us that the mean of approximated (low-pass) query signal is just a linear combination of the mean of eigennorms weighted by the graph Fourier transform of the query signal.

Modeling the variance Again, we know that the low-pass filtered approximation of the query signal (as a random variable) can be written as a linear combination of the eigennorms which are assumed to be normally distributed random variables. As such, we can use statistical rules to compute the variance of a posterior normal distribution from the linear combination of variances and covariances of the priors:

$$\mathrm{Var}\left(\sum_{i=1}^{k} a_iX_i\right) = \left(\sum_{i=1}^{k} a_i^2\mathrm{Var}(X_i)\right) + \left(\sum_{i=1}^{k}\sum_{j=1(j\neq i)}^{k} \mathrm{Cov}(X_i, X_j)\right)$$

This can also be written in the following matrix form:

$$\mathrm{Var}(\overrightarrow{a}.\overrightarrow{X}) = \overrightarrow{a}K_{XX}\overrightarrow{a}$$

Where K_{XX} denotes the covariance matrix of random variables in $\overrightarrow{X}$.

Now, this makes it a bit complicated to accurately compute the variance of the query signal from the eigennorms. The main issue is that for the eigennorms, we have modeled the standard deviation (σ) of each eigenmode as a function of the covariates (age, sex, site). Variance can be easily computed from the standard deviation. However, the covariances were not modeled as a function of the covariates. One solution would be to construct a more detailed model that also fits the covariances of the eignenorms (potentially a future to do!), but for now, we could try to come up with an approximation of the covariance structure.

We know that the covariance between two random variables is related to their variances and correlation:

$$\mathrm{Cov}(X_i, X_j) = \sqrt{\mathrm{Var}(X_i)}\sqrt{\mathrm{Var}(X_j)}\mathrm{Corr}(X_i, X_j)$$

Or in a matrix notation:

$$\mathrm{cov}(X) = K_{XX} = (\mathrm{diag}(K_{XX}))^{\frac{1}{2}}\,\mathrm{corr}(X)\,(\mathrm{diag}(K_{XX}))^{\frac{1}{2}}$$

Where $(\mathrm{diag}(K_{XX}))^{\frac{1}{2}}$ is essentially a diagonal matrix from standard deviations of every random variable X_i.

So, if we assume that the correlation matrix is constant (does not vary by covariates), which may most likely be a false assumption (but we're implementing it now for the sake of simplicity), then the standard deviation of signal query operation can be expressed in terms of the standard deviations of the eigennorms.

First, the constant correlation matrix between eigenmodes is computed from the available encoded thickness information, i.e. $\overrightarrow{T_{LP}} = {V_{LP}}^T\overrightarrow{T}$. So a correlation matrix $\mathrm{corr}(\overrightarrow{T_{LP}}) \in R^{N_{LP}} \times R^{N_{LP}}$ is computed from the row-wise correlations between rows of T_{LP}.

Next, we derive the normative standard deviation of the low-pass filtered query signal operated on a hypothetical high-resolution thickness vector $\overrightarrow{T}$ from the normative standard deviations of the eigennorms:

$$
\begin{aligned}
\mathrm{var}(\overrightarrow{X_{Q(LP)}}.\overrightarrow{T}) &= \mathrm{var}(\overrightarrow{T}.\overrightarrow{X_{Q(LP)}}) \\
&= \mathrm{var}(\overrightarrow{T}V_{LP}\overrightarrow{\tilde{X}_{Q(LP)}}) \\
&= \mathrm{var}(\overrightarrow{\tilde{T}_{LP}}.\overrightarrow{\tilde{X}_{Q(LP)}}) \\
&= \mathrm{var}(\overrightarrow{\tilde{X}_{Q(LP)}}.\overrightarrow{\tilde{T}_{LP}}) \\
&= \overrightarrow{\tilde{X}_{Q(LP)}}\mathrm{cov}(\overrightarrow{\tilde{T}_{LP}})\overrightarrow{\tilde{X}_{Q(LP)}} \\
&= \overrightarrow{\tilde{X}_{Q(LP)}}\mathrm{diag}(\mathrm{std}(\overrightarrow{\tilde{T}_{LP}}))\mathrm{corr}(\overrightarrow{\tilde{T}_{LP}})\mathrm{diag}(\mathrm{std}(\overrightarrow{\tilde{T}_{LP}}))\overrightarrow{\tilde{X}_{Q(LP)}}
\end{aligned}
$$

Where diag(std($\overrightarrow{\tilde{T}_{LP}}$)) is the standard deviations of every eigennorm (σ_i) which is estimated using the fitted kernel norms at any given (age, sex, site).

To summarize, we provided two equations by which we could model the mean and deviation of any arbitrary signal query operator as a function of the mean and deviations of the eigennorms. As such, the eigennorms (along with the constant correlation matrix) give us all we need to approximate the normative trajectories of any arbitrary query signal.

Note: We need to assess whether it would be possible to actually formulate the covariance structure as a function of the covariates to further increase the accuracy and correctness of the model.

Arbitrary high-resolution queries

Apart from the aforementioned signal-based query, the eigennorms also give us the capability to estimate high-resolution (potentially smoothed) norms that could be used to explain high-resolution developmental trajectories. These norms could also be used to conduct an individualized assessment of high-resolution deviation patterns from the norm.

Before digging into the mathematics of these high-resolution queries, let's give a sensible example of what we want to achieve. Essentially, we would like to have a resolution/signal-independent method to interrogate the fitted norms. Basically, the aim is to answer general questions like: "what would the high-resolution cortical thickness of an individual with a specific age/gender or possibly from a specific site look like?"

Essentially, in this setting, we have a set of minimal constraints based on the covariates. Using the fitted normative functions (f_{μ_i}, f_{σ_i}) we can compute fixed estimates for the mean and standard deviation using the covariates constraints. Next, we need to transform this information onto a high-resolution signal domain. So, basically, similar to the signal query case, we need to model the mean and deviation. With the difference that this time we are computing mean and deviation over the high-resolution space directly.

So, to begin with, we have two vectors computed for a fixed set of constraints. A mean vector $\overrightarrow{M}$, and a standard deviation vector $\overrightarrow{S}$ both of which have a dimensionality of $R^{N_{LP}}$. ($\overrightarrow{S}$ is the vector of standard deviations and is equal to the diagonals of $\mathrm{diag}(\mathrm{std}(\overrightarrow{\tilde{T}_{LP}}))$)

Computing the high-resolution means Using the same statistical rules, we can derive the high-resolution mean (the expected value for high-resolution thickness) $E[\overrightarrow{T}]$. Furthermore, to make the high-resolution norms less noisy, we can evaluate the expected value after smoothing $E[G\overrightarrow{T}]$, where G is a row-normalized Gaussian smoothing kernel applied to the hypothetical high-resolution data. In any case, if no smoothing is intended to be modeled, G can be replaced with the identity matrix I. The following equation describes how the high-resolution mean can be modeled:

$$\begin{aligned} E[G\overrightarrow{T}] &\approx E[GV_{LP}\overrightarrow{\tilde{T}_{LP}}] \\ &= GV_{LP}\overrightarrow{M} \end{aligned}$$

Interestingly, GV_{LP} is equivalent to applying spatial smoothing on every eigenmode separately.

Computing the high-resolution variance Similarly, the high-resolution variance can be modeled from the deviations of the eigennorms:

$$\begin{aligned} \mathrm{var}(G\overrightarrow{T}) &\approx \mathrm{var}(GV_{LP}\overrightarrow{\tilde{T}_{LP}}) \\ &= \overrightarrow{\left(GV_{LP}\overrightarrow{S}\right)}\mathrm{corr}(\overrightarrow{\tilde{T}_{LP}})\overrightarrow{\left(GV_{LP}\overrightarrow{S}\right)} \end{aligned}$$

In a sense, this high-resolution method is a special case of the query signal, where there are N_v different queries, each of which, is a binary characteristic function of a single vertex (one for that vertex, zero everywhere else). (In the case of performing smoothing, that binary query is relaxed to a smoothed high-resolution mask from that binary mask.)