Generative Framework

A generative modelling framework for individual brain organization across a number of different data. The Model is partitioned into a model that determines the probability of the spatial arrangement of regions in each subject s, $p(\mathbf{U}^{(s)};\theta_A)$ and the probability of observing a set of data at each given brain location. We introduce the Markov property that the observations are mutually independent, given the spatial arrangement.

$$p(\mathbf{Y}^{(s)}|\mathbf{U}^{(s)};\theta_E)=\prod_i p(\mathbf{y}_i^{(s)}|\mathbf{u}_i^{(s)};\theta_E) \tag{1}$$

Inference and learning

We will learn the model, by maximizing the ELBO (Evidence lower bound). For clarity, I am dropping the index for the subject (s) for now,

$$\begin{aligned}\log p(\mathbf{Y}|\theta)&=\log\sum_{\mathbf{U}}p(\mathbf{Y},\mathbf{U}|\theta)\\&=\log\sum_{\mathbf{U}}q(\mathbf{U})\frac{p(\mathbf{Y},\mathbf{U}|\theta)}{q(\mathbf{U})}\\&\geqslant\sum_{\mathbf{U}}q(\mathbf{U})\log\frac{p(\mathbf{Y},\mathbf{U}|\theta)}{q(\mathbf{U})} && (\text{Jensen's inequality})\\&=\langle\log p(\mathbf{Y},\mathbf{U}|\theta)-\log q(\mathbf{U})\rangle_q\triangleq\mathcal{L}(q,\theta)-\log\langle q(\mathbf{U})\rangle_q\end{aligned}$$

Given the markov property, we can break the expected complete log likelihood into two pieces, one containing the parameters for the arrangement model and one containing the parameters for the emission model.

$$\begin{aligned}\langle\log p(\mathbf{Y},\mathbf{U}|\theta)\rangle_q&=\langle\log(p(\mathbf{Y}|\mathbf{U};\theta_E)p(\mathbf{U}|\theta_A))\rangle_q\\&=\langle\log p(\mathbf{Y}|\mathbf{U};\theta_E)\rangle_q+\langle\log p(\mathbf{U}|\theta_A)\rangle_q\\&\triangleq\mathcal{L}_E+\mathcal{L}_A\end{aligned}$$

We will refer to the first term as the expected emission log-likelihood and the second term as the expected arrangement log-likelihood. We can estimate the parameters of the emission model from maximizing the expected emission log-likelihood, and we can estimate the parameters of the arrangement model by maximizing the expected arrangement log-likelihood.

Arrangement models

This is a generative Potts model of brain activity data. The main idea is that the brain consists of K regions, each with a specific activity profile $\mathbf{v}_k$ for a specific task set. The model consists of a arrangement model that tells us how the K regions are arranged in a specific subject s, and an emission model that provides a probability of the measured data, given the individual arrangement of regions.

Independent Arrangement model

This is the simplest spatial arrangement model - it simply learns the probability at each location that the node is part of cluster K. These probabilities are simply learned as the parameters $\pi_{ik}=p(u_i=k)$, or after a re-parameterization in log space: $\eta_{ik}=\log\pi_{ik}$. Vice versa (not assuming that the etas are correctly scaled):

$$\pi_{ik}=\frac{\exp(\eta_{\mathrm{ik}})}{\sum_j\exp(\eta_{\mathrm{ij}})} \tag{2}$$

This independent arrangement model can be estimated using the EM-algorithm.

In the Estep, we are integrating the evidence from the data and the prior:

$$p(u_i = k|\mathbf{y}_i) = \langle u_{ik}\rangle = \frac{\exp(\log(\mathrm{p}(\mathbf{y}_\mathrm{i}|\mathrm{u}_\mathrm{i} = \mathrm{k})) + \eta_\mathrm{ik})}{\sum_j \exp(\log(\mathrm{p}(\mathbf{y}_\mathrm{i}|\mathrm{u}_\mathrm{i} = \mathrm{j})) + \eta_\mathrm{ij})} \tag{3}$$

Or in vector notation:

$$\langle \mathbf{u}_i \rangle = \mathrm{softmax}(\log(\mathrm{p}(\mathbf{y}_\mathrm{i}|\mathbf{u}_\mathrm{i})) + \boldsymbol{\eta}_\mathrm{i})$$

For the M-step, we use the derivative of the expected arrangement log-likelihood in respect to the parameters η:

$$\begin{aligned} \mathcal{L}_A &= \sum_i \sum_k \langle u_{i,k} \rangle \log(\pi_\mathrm{ik}) \\ &= \sum_i \sum_k \langle u_{ik} \rangle (\eta_{ik} - \log \sum_\mathrm{j} \exp(\eta_\mathrm{ij})) \\ &= \sum_i \sum_k \langle u_{ik} \rangle \eta_{ik} - \sum_i \log \sum_j \exp(\eta_{i,j}) \end{aligned} \tag{4}$$

So the derivative is

$$\begin{aligned} \frac{\partial \mathcal{L}_A}{\partial \eta_{ik}} &= \langle u_{ik} \rangle - \frac{\partial}{\partial \eta_{ik}} \log \sum_j \exp(\eta_{ij}) &(5) \\ &= \langle u_{ik} \rangle - \frac{1}{\sum_j \exp(\eta_{ij})} \frac{\partial}{\partial \eta_{ik}} \sum_j \exp(\eta_{ij}) &(6) \\ &= \langle u_{ik} \rangle - \frac{\exp(\eta_{ik})}{\sum_j \exp(\eta_{ij})} &(7) \\ &= \langle u_{ik} \rangle - \pi_{ik} &(8) \end{aligned}$$

We can also get the same result directly by the application of chain rule: For a good introduction, see: [https://eli.thegreenplace.net/2016/the-softmax-function-and-its-derivative/].

$$\frac{\partial \pi_k}{\eta_k} = \pi_k(\delta_{kj} - \pi_j) \tag{9}$$

Simple Potts model

The brain is sampled in P vertices (or voxels). Individual maps are aligned using anatomical normalization, such that each vertex refers to a (roughly) corresponding region in each individual brain. The assignment of each brain location to a specific parcel in subject s is expressed as the random variable $u_i^{(s)}$.

Across individual brains, we have the overall probability of a specific brain location being part of parcel k.

$$p(u_i = k) = \pi_{ki} \tag{10}$$

The spatial interdependence of brain locations is expressed as a Potts model. In this model, the overall probability of a specific assignment of brain locations to parcels (the vector $\mathbf{u}$) is expressed as the product of the overall prior and the product of all possible pairwise potenenials (ψ_{ij}).

$$p(\mathbf{u}) = \frac{1}{Z(\theta)} \prod_i \pi_{u_i,i} \prod_{i \neq j} \psi_{ij}(u_i, u_j) \tag{11}$$

Each local potential is defined by an exponential over all other that are connected to node i, i.e. nodes with connectivity weights of $w_{ji} = w_{ij} > 0$.

$$\psi_{ij} = \exp(\theta_{\mathrm{w}} \mathbf{u}_{\mathrm{i}}^{\mathrm{T}} \mathbf{u}_{\mathrm{j}} \mathrm{w}_{\mathrm{ij}}) \tag{12}$$

Where we have introduced a one-hot encoding of u_i with a K vector of indicator variables $\mathbf{u}_i$, such that $\mathbf{u}_i^T \mathbf{u}_j = 1$ if $u_i = u_j$ and 0 otherwise.

The spatial co-dependence across the entire brain is therefore expressed with the pairwise weights w that encode how likely two nodes belong to the same parcel. The temperature parameter θ_w determines how strong this co-dependence overall influences the local probabilities (relative to the prior). We can use this notation to express local co-dependencies by using a graph, where we define

$$w_{ij} = \begin{cases} 1; \text{if i and j are neighbours} \\ 0; \text{otherwise} \end{cases} \tag{13}$$

This formulation would enforce local smoothness of the map. However, we could also express in these potential more medium range potentials (two specific parietal and premotor areas likely belong to the same parcel), as well as cross-hemispheric symmetry. Given this, the matrix $\mathbf{W}$ could be simply derived from the underlying grid or be learned to reflect known brain-connectivity.

The expected arrangement log-likelihood therefore becomes:

$$\mathcal{L}_A = \sum_i \langle \mathbf{u}_i \rangle^T \log \boldsymbol{\pi}_i + \theta_w \sum_i \sum_j w_{ij} \langle \mathbf{u}_i^T \mathbf{u}_j \rangle - \log Z$$

Inference using stochastic maximum likelihood / contrastive divergence

We can approximate the gradient of the parameters using a contrastive divergence-type algorithm. We view the arrangement log likelihood as a sum of the unnormalized part ($\tilde{\mathcal{L}}_A$) and the log partition function. For each parameter θ we then follow the gradient

$$\begin{aligned} \nabla_\theta \mathcal{L}_A &= \nabla_\theta \tilde{\mathcal{L}}_A - \nabla_\theta \log Z \\ &= \nabla_\theta \tilde{\mathcal{L}}_A - \mathrm{E}_p[\nabla_\theta \tilde{\mathcal{L}}_A] \\ &= \nabla_\theta \langle \log \tilde{p}(\mathbf{U}|\theta) \rangle_q - \nabla_\theta \langle \log \tilde{p}(\mathbf{U}|\theta) \rangle_p \end{aligned}$$

Thus, we can use the gradient of the unnormalized expected log-likelihood (given a distribution $q(\mathbf{U}) = p(\mathbf{U}|\mathbf{Y};\theta)$, minus the gradient of the unnormalized expected log-likelihood in respect to the expectation under the model parameters, without seeing the data, $q(\mathbf{U}) = p(\mathbf{U}|\mathbf{Y};\theta)$. This motivates the use of sampling / approximate inference for both of these steps. See Deep Learning (18.1).

E-step: sampling from prior or posterior distribution

The problem is that the two expectations under the prior (p) and the posterior (q) distribution of the model cannot be easily be computed. We can evaluate the prior probability of a parcellation $p(\mathbf{U})$ or the posterior distribution $p(\mathbf{U}|\mathbf{Y})$ up to a constant of proportionality, with for example

$$p(\mathbf{U}|\mathbf{Y};\theta) = \frac{1}{Z(\theta)} \prod_i \mu_{u_i,i} \prod_{i \neq j} \psi_{ij}(u_i, u_j) \prod_i p(\mathbf{y}_i|u_i) \tag{14}$$

Calculating the normalization constant $Z(\theta)$ (partition function, Zustandssumme, or sum over states) would involve summing this probability over all possible states, which for P brain locations and K parcels is K^P, which is intractable.

However, the conditional probability for each node, given all the other nodes, can be easily computed. Here the normalizaton constant is just the sum of the potential functions over the K possible states for this node

$$p(u_i|u_{j \neq i}, \mathbf{y}_i; \theta) = \frac{1}{Z(\theta)} \mu_{u_i,i}\, p(\mathbf{y}_i|u_i) \prod_{i \neq j} \psi_{ij}(u_i, u_j) \tag{15}$$

With Gibbs sampling, we start with a pattern $\mathbf{u}^{(0)}$ and then update $u_1^{(1)}$ by sampling from $p(u_1|u_2^{(0)}\dots u_P^{(0)})$. We then sample $u_2^{(1)}$ by sampling from $p(u_2|u_1^{(1)}, u_3^{(0)}\dots u_P^{(0)})$ and so on, until we have sampled each node once. Then we return to the beginning and restart the process. After some burn-in period, the samples will come from desired overall distribution. If we want to sample from the prior, rather than from the posterior, we simply drop the $p(\mathbf{y}_i|u_i)$ term from the conditional probability above.

Gradient for different parametrization of the Potts model

For the edge-energy parameters θ_w we clearly want to use the natural parametrization with the derivate:

$$\frac{\partial\tilde{\mathcal{L}}_A}{\partial\theta_w} = \sum_i\sum_j w_{ij}\langle\mathbf{u}_i^T\mathbf{u}_j\rangle \tag{16}$$

For the prior probability of each parcel k at each location i (π_{ik}) we have a number of options.

First ,we can use the probabilities themselves as parameters:

$$\frac{\partial\tilde{\mathcal{L}}_A}{\partial\pi_{ik}} = \frac{\langle u_{ik}\rangle}{\pi_{ik}} \tag{17}$$

This is unconstrained (that is probabilities do not need to sum to 1), and the normalization would happen through the partition function.

Secondly, we can use a re-parameterization in log space, which is more natural: $\eta_{ik} = \log\pi_{ik}$. In this case the derivative of the non-normalized part just becomes:

$$\frac{\partial\tilde{\mathcal{L}}_A}{\partial\eta_{ik}} = \langle u_{ik}\rangle \tag{18}$$

Finally, we can implement the constraint that the probabilities at each location sum to one by the following re-parametrization:

$$\begin{aligned}
\pi_{iK} &= 1 - \sum_{k=1}^{K-1}\pi_{ik}\\
\eta_{ik} &= \log(\frac{\pi_{ik}}{\pi_{iK}}) = \log\pi_{ik} - \log(1-\sum_{k=1}^{K-1}\pi_{ik})\\
\pi_{ik} &= \frac{\exp(\eta_{ik})}{1+\sum_{k=1}^{K-1}\exp(\eta_{ik})}\\
\pi_{iK} &= \frac{1}{1+\sum_{k=1}^{K-1}\exp(\eta_{ik})}
\end{aligned}$$

In the implementation, we can achieve this parametrization easily by defining a non-flexible parameter $\eta_{iK} \triangleq 0$. Then we can treat the last probability like all the other ones.

With this constrained parameterization, we can rewrite the unnormalized part of the expected log-likelihood as:

$$\begin{aligned}
\tilde{\mathcal{L}}_A &= \sum_i\sum_k^{K-1}\langle u_{ik}\rangle\log\pi_{ik} + [1-\sum_k^{K-1}\langle u_{ik}\rangle]\log\pi_{iK} + C\\
&= \sum_i\sum_k^{K-1}\langle u_{ik}\rangle(\log\pi_{ik} - \log\pi_{iK}) + \log\pi_{iK} + C\\
&= \sum_i\sum_k^{K-1}\langle u_{ik}\rangle\eta_{ik} - \log(1+\sum_{k=1}^{K-1}\exp(\eta_{ik})) + C
\end{aligned}$$

where C is the part of the normalized log-likelihood that does not depend on π. Taking derivative in respect to η_{ik} yields:

$$\begin{aligned}\frac{\partial \tilde{\mathcal{L}}_A}{\partial \eta_{ik}} &= \langle u_{ik} \rangle - \frac{\exp(\eta_{ik})}{1 + \sum_k^{K-1} \exp(\eta_{ik})} \\ &= \langle u_{ik} \rangle - \pi_{ik}\end{aligned}$$

So in this parameterization in the iid case, $Z = 1$ and we don't need the negative step. In general, however, we cannot simply set the above derivative to zero and solve it, as the parameter θ_w will also have an influences on $\langle u_{ik} \rangle$.

Probabilistic multinomial restricted Boltzmann machine

As an alternative to a Potts model, we are introducing here a multivariate version of a restricted Boltzmann machine. A restricted Boltzmann machine consists typically out a layer of binary visible and a layer of binary hidden units ($\mathbf{h}$) with J nodes h_j. Here, we are replacing the input with the spatial arrangement matrix $\mathbf{U}$, with each column of the matrix $\mathbf{u}_i$ representing a one-hot encoded multinomial random variable, that assigns the brain location i to parcel k.

The hidden variables is still a vector of binary latent variables

$$p(h_j|\mathbf{U}) = \sigma(vec(\mathbf{U})^T \mathbf{W}_{.,j} + \mathbf{b}_j) \tag{19}$$

Where σ is the sigmoid function.

The probability of a brain location then is given by:

$$p(\mathbf{u}_i|\mathbf{h}) = \mathrm{softmax}([\mathbf{h}^\mathrm{T}\mathbf{W}^\mathrm{T}]_\mathrm{i} + \boldsymbol{\eta}_\mathrm{i}) \tag{20}$$

Where $[.]_i$ selects the element for $\mathbf{u}_i$ from vectorized version of $\mathbf{U}$.

Positive Estep: Expectation given the data

The advantage of a Boltzmann machine is that we can efficiently do inference and sampling in a blocked fashion. In the positive E-step, the expectation can be passed - and we can do one or more iteration between the $\mathbf{h}$ and the $\mathbf{U}$ layer.

We intialize the hidden layer with

$$\langle \mathbf{h} \rangle_q^{(0)} = \mathbf{0} \tag{21}$$

An then alternate:

$$\langle \mathbf{u}_i \rangle_q^{(t)} = \mathrm{softmax}([\mathbf{W}\langle \mathbf{h} \rangle^{(\mathrm{t})}]_\mathrm{i} + \boldsymbol{\eta}_\mathrm{i} + \log \mathrm{p}(\mathbf{y}_\mathrm{i}|\mathbf{u}_\mathrm{i})) \tag{22}$$

$$\langle h_j \rangle_q^{(t+1)} = \sigma(vec(\langle \mathbf{U} \rangle_q^{(t)})^T \mathbf{W}_{.,j} + \mathbf{b}_j) \tag{23}$$

Negative Estep: Expectation given the model

For the negative e-step, we are using sampling alternating for $\mathbf{h}$ and $\mathbf{U}$, using the main equations. The expectations are then probabilities before the last sampling step. These give us the expectations $\langle . \rangle_p$ that we need for subsequent learning.

Gradient step for parameter estimation

Given the expectation of the hidden and latent variable for the positive and negative phase of the expectation.

$$\begin{aligned}\nabla_W &= \langle \mathbf{h}\rangle_q^T vec(\langle \mathbf{U}\rangle_q) - \langle \mathbf{h}\rangle_p^T vec(\langle \mathbf{U}\rangle_p)\\ \nabla_b &= \langle \mathbf{h}\rangle_q - \langle \mathbf{h}\rangle_p\\ \nabla_\eta &= \langle \mathbf{U}\rangle_q - \langle \mathbf{U}\rangle_p\end{aligned}$$

Convolutional multinomial probabilistic restricted Boltzmann machine (cmpRBM)

Another approach is to make both the hidden ($\mathbf{H}$) and the intermedicate ($\mathbf{U}$) nodes are multinomial version of a restricted Boltzmann machine. So with Q hidden nodes, H is the KxQ matrix with the one-hot encoded state of the hidden variables, and U is a KxP matrix of the one-hot encoded clustering. $\mathbf{W}$ is the QxP matrix of connectivity that connects the respective nodes.

The hidden variables is still a vector of binary latent variables

$$p(\mathbf{h}_j|\mathbf{U}) = \mathrm{softmax}(\mathbf{U}\mathbf{W}_{\mathrm{j,.}}^{\mathrm{T}}) \tag{24}$$

The probability of a brain location then is given by:

$$p(\mathbf{u}_i|\mathbf{h}) = \mathrm{softmax}(\mathbf{H}\mathbf{W}_{\mathrm{.,i}} + \boldsymbol{\eta}_{\mathrm{i}}). \tag{25}$$

Positive Estep: Expectation given the data

The advantage of a Boltzmann machine is that we can efficiently do inference and sampling in a blocked fashion. In the positive E-step, the expectation can be passed - and we can do one or more iteration between the $\mathbf{H}$ and the $\mathbf{U}$ layer.

We intialize the hidden layer with

$$\langle \mathbf{H}\rangle_q^{(0)} = \mathbf{0} \tag{26}$$

An then alternate:

$$\langle \mathbf{u}_i\rangle_q^{(t)} = \mathrm{softmax}(\langle \mathbf{H}\rangle^{(\mathrm{t})}\mathbf{W}_{\mathrm{.,i}} + \boldsymbol{\eta}_{\mathrm{i}} + \log \mathrm{p}(\mathbf{y}_{\mathrm{i}}|\mathbf{u}_{\mathrm{i}})) \tag{27}$$

$$\langle \mathbf{h}_j\rangle_q^{(t+1)} = \mathrm{softmax}(\langle \mathbf{U}\rangle_{\mathrm{q}}^{(\mathrm{t})}\mathbf{W}_{\mathrm{j,.}}^{\mathrm{T}}) \tag{28}$$

Negative Estep: Expectation given the model

For the negative e-step, nwe are using sampling alternating for $\mathbf{h}$ and $\mathbf{U}$, using the main equations. The expectations are then probabilities before the last sampling step. These give us the expectations $\langle . \rangle_p$ that we need for subsequent learning.

Gradient step for parameter estimation

The unnormalized log-probability of the model (negative Energy function) of the model is:

$$\log \tilde{p}(\mathbf{U},\mathbf{H}|\mathbf{Y}) = \sum_i \eta_i^T \mathbf{u}_i + \mathrm{tr}(\mathbf{H}\mathbf{W}\mathbf{U}^{\mathrm{T}}) \tag{29}$$

Given the expectation of the hidden and latent variable for the positive and negative phase of the expectation, the gradients are:

$$\begin{aligned}\nabla_W &= \langle \mathbf{H}\rangle_q^T \langle \mathbf{U}\rangle_q - \langle \mathbf{H}\rangle_p^T \langle \mathbf{U}\rangle_p\\ \nabla_\eta &= \langle \mathbf{U}\rangle_q - \langle \mathbf{U}\rangle_p\end{aligned}$$

Emission models

Given the Markov property, the emission models specify the log probability of the observed data as a function of $\mathbf{u}$.

$$\log p(\mathbf{Y}|\mathbf{U};\theta_E) = \sum_i \log p(\mathbf{y}_i|\mathbf{u}_i;\theta_E) \tag{30}$$

Furthermore, assuming that $\mathbf{u}_i$ is a one-hot encoded indicator variable (parcellation model), we can write the expected emission log-likelihood as:

$$\langle \log p(\mathbf{Y}|\mathbf{U};\theta_E)\rangle = \sum_i \sum_k \langle u_i^{(k)} \log p(\mathbf{y}_i|u_i = k;\theta_E)\rangle \tag{31}$$

In the E-step the emission model simply passes $p(\mathbf{y}_i|\mathbf{u}_i;\theta_E)$ as a message to the arrangement model. In the M-step, $q(\mathbf{u}_i) = \langle \mathbf{u}_i \rangle$ is passed back, and the emission model optimizes the above quantity in respect to θ_E.

Emission model 1: Multinomial

A simple (but instructive) emission model is that the observed data simpy has a multinomial distribution, like the latent variables $\mathbf{u}$. The coupling between the latent and the observed variable is stochastic, using a Potts model between the two nodes:

$$p(\mathbf{y_i}|\mathbf{u}_i;\theta_E) = \frac{\exp(\mathbf{y}_i^T \mathbf{u}_i w)}{(K-1)+\exp(w)} \tag{32}$$

The expected emission loglikelihood therefore is:

$$\mathcal{L}_E = \sum_i (\mathbf{y}_i^T \langle \mathbf{u}_i \rangle w - \log((K-1)+\exp(w)))$$

The derivative in respect to w then becomes:

$$\frac{\partial \mathcal{L}_E}{\partial w} = \sum_i^P \mathbf{y}_i^T \langle \mathbf{u}_i \rangle - P\frac{\exp(w)}{(K-1)+\exp(w)} \tag{33}$$

After setting the derivate to zero and solving for w, we obrain for the M-step:

$$\frac{\partial \mathcal{L}_E}{\partial w} = \sum_i^P \mathbf{y}_i^T \langle \mathbf{u}_i \rangle - P\frac{\exp(w)}{(K-1)+\exp(w)} \tag{34}$$

$$\begin{aligned}
\sum_i^P \mathbf{y}_i^T \langle \mathbf{u}_i \rangle &= P \frac{\exp(w)}{(K-1)+\exp(w)} \\
\sum_i^P \mathbf{y}_i^T \langle \mathbf{u}_i \rangle / P &= 1 - \frac{(K-1)}{(K-1)+\exp(w)} \\
1 - \sum_i^P \mathbf{y}_i^T \langle \mathbf{u}_i \rangle / P &= \frac{(K-1)}{(K-1)+\exp(w)} \\
\frac{(K-1)}{1 - \sum_i^P \mathbf{y}_i^T \langle \mathbf{u}_i \rangle / P} &= (K-1)+\exp(w) \\
1 - K + \frac{(K-1)}{1 - \sum_i^P \mathbf{y}_i^T \langle \mathbf{u}_i \rangle / P} &= \exp(w) \\
w = \log(1 - K + \frac{(K-1)}{1 - \sum_i^P \mathbf{y}_i^T \langle \mathbf{u}_i \rangle / P}&)
\end{aligned}$$

Emission model 2: Mixture of Gaussians

Under the Gaussian mixture model, we model the emissions as a Gaussian with a parcel-specific mean ($\mathbf{v}_k$), and with equal isotropic variance across parcels and observations:

$$p(\mathbf{y_i}|u^{(k)};\theta_E) = \frac{1}{(2\pi)^{N/2}(\sigma^2)^{N/2}} \exp\{-\frac{1}{2\sigma^2}(\mathrm{y_i} - \mathbf{X}\mathbf{v}_\mathrm{k})^\mathrm{T}(\mathrm{y_i} - \mathbf{X}\mathbf{v}_\mathrm{k})\} \tag{35}$$

The expected emission log-likelihood therefore is:

$$\begin{aligned}
\mathcal{L}_E &= \sum_i \sum_k \langle u_i^{(k)} \rangle_q [-\frac{N}{2}\log(2\pi) - \frac{N}{2}\log(\sigma^2) - \frac{1}{2\sigma^2}(\mathbf{y}_i - \mathbf{X}\mathbf{v}_k)^T(\mathbf{y}_i - \mathbf{X}\mathbf{v}_k)] \\
&= -\frac{PN}{2}\log(2\pi) - \frac{PN}{2}\log(\sigma^2) - \frac{1}{2\sigma^2}\sum_i \sum_k \langle u_i^{(k)} \rangle_q [(\mathbf{y}_i - \mathbf{X}\mathbf{v}_k)^T(\mathbf{y}_i - \mathbf{X}\mathbf{v}_k)]
\end{aligned}$$

Now, with the above expected emission log likelihood by hand, we can update the parameters $\theta_E = \{\mathbf{v}_1, \ldots, \mathbf{v}_K, \sigma^2\}$ in the M step.

1. Updating $\mathbf{v}_k$, we take derivative of *expected emission log likelihood* with respect to $\mathbf{v}_k$ and set it to 0:

$$\frac{\partial \mathcal{L}_E}{\partial \mathbf{v}_k} = \frac{1}{\sigma^2} \sum_i \langle u_i^{(k)} \rangle_q (\mathbf{y}_i - \mathbf{X}\mathbf{v}_k) = 0 \tag{36}$$

Thus, we get the updated $\mathbf{v}_k$ in current M step as,

$$\mathbf{v}_k^{(t)} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T \frac{\sum_i \langle u_i^{(k)} \rangle_q^{(t)} \mathbf{y}_i}{\sum_i \langle u_i^{(k)} \rangle_q^{(t)}} \tag{37}$$

2. Updating σ^2, we take derivative of *expected emission log likelihood* $\mathcal{L}(q, \theta)$ with respect to σ^2 and set it to 0:

$$\frac{\partial \mathcal{L}_E}{\partial \sigma^2} = -\frac{PN}{2\sigma^2} + \sum_i \sum_k \langle u_i^{(k)} \rangle_q [\frac{1}{2\sigma^4}(\mathbf{y}_i - \mathbf{X}\mathbf{v}_k^{(t)})^T(\mathbf{y}_i - \mathbf{X}\mathbf{v}_k^{(t)})] = 0 \tag{38}$$

Thus, we get the updated σ^2 for parcel k in the current M step as,

$$\sigma^{2(t)} = \frac{1}{PN} \sum_i \sum_k \langle u_i^{(k)} \rangle_q^{(t)} (\mathbf{y}_i - \mathbf{X}\mathbf{v}_k^{(t)})^T(\mathbf{y}_i - \mathbf{X}\mathbf{v}_k^{(t)}) \tag{39}$$

where P is the total number of voxels i.

The updated parameters $\theta_k^{(t)}$ from current $\mathbf{M}$-step will be passed to the next $\mathbf{E}$-step $(t+1)$ until convergence.

Emission model 3a: Mixture of Gaussians with Exponential signal strength

The emission model should depend on the type of data that is measured. A common application is that the data measured at location i are the task activation in N tasks, arranged in the $N \times 1$ data vector $\mathbf{y}_i$. The averaged expected response for each of the parcels is $\mathbf{v}_k$. One issue of the functional activation is that the signal-to-noise ratio (SNR) can be quite different across different participants, and voxels, with many voxels having relatively low SNR. We model this signal to noise for each brain location (and subject) as $s_i \sim exponential(\beta_s)$. Therefore the probability model for gamma is defined as:

$$p(s_i|\theta) = \beta e^{-\beta s_i} \tag{40}$$

Overall, the expected signal at each brain location is then

$$\mathrm{E}(\mathbf{y}_\mathrm{i}) = \mathbf{u}_\mathrm{i}^\mathrm{T}\mathbf{V}\mathrm{s}_\mathrm{i} \tag{41}$$

Finally, relative to the signal, we assume that the noise is distributed i.i.d Gaussian with:

$$\boldsymbol{\epsilon}_i \sim Normal(0, \mathbf{I}_K\theta_{\sigma s}) \tag{42}$$

Here, the proposal distribution $q(u_i^{(k)}, s_i|\mathbf{y}_i)$ is now a multivariate distribution across u_i and s_i. Thus, the *expected emission log likelihood* $\mathcal{L}_E(q,\theta)$ is defined as:

$$\begin{aligned}
\mathcal{L}_E &= \langle \sum_i \log p(\mathbf{y}_i, s_i|u_i;\theta_E)\rangle_q \\
&= \sum_i \sum_k \langle u_i^{(k)}[-\frac{N}{2}\log(2\pi) - \frac{N}{2}\log(\sigma^2) - \frac{1}{2\sigma^2}(\mathbf{y}_i - \mathbf{v_k}s_i)^T(\mathbf{y}_i - \mathbf{v_k}s_i)]\rangle_q \\
&+ \sum_i \sum_k \langle u_i^{(k)}[\log\beta - \beta s_i]\rangle_q \\
&= -\frac{NP}{2}\log(2\pi) - \frac{NP}{2}\log(\sigma^2) - \frac{1}{2\sigma^2}\sum_i \sum_k \langle u_i^{(k)}(\mathbf{y}_i - \mathbf{v_k}s_i)^T(\mathbf{y}_i - \mathbf{v_k}s_i)\rangle_q \\
&+ P\log\beta - \sum_i \sum_k \beta\langle u_i^{(k)}s_i\rangle_q \\
&= -\frac{NP}{2}\log(2\pi) - \frac{NP}{2}\log(\sigma^2) - \frac{1}{2\sigma^2}\sum_i \mathbf{y}_i^T\mathbf{y}_i - \frac{1}{2\sigma^2}\sum_i \sum_k(-2\mathbf{y}_i^T\mathbf{v}_k\langle u_i^{(k)}s_i\rangle_q + \mathbf{v}_k^T\mathbf{v}_k\langle u_i^{(k)}s_i^2\rangle_q) \\
&+ \log\beta - \beta\sum_i \sum_k \langle u_i^{(k)}s_i\rangle_q
\end{aligned}$$

Now, we can update the parameters θ of the Gaussians/Exponential mixture in the M step. The parameters of the gaussian mixture model are $\theta_E = \{\mathbf{v}_1, \dots, \sigma^2, \beta\}$.

1. We start with updating the $\mathbf{v}_k$ (Note: the updates only consider a single subject). We take the derivative of *expected emission log likelihood* $\mathcal{L}_E$ with respect to $\mathbf{v}_k$ and make it equals to 0 as following:

$$\frac{\partial\mathcal{L}_E}{\partial\mathbf{v}_k} = -\frac{1}{\sigma^2}\sum_i -\mathbf{y}_i^T\langle u_i^{(k)}s_i\rangle_q + \mathbf{v}_k^T\langle u_i^{(k)}s_i^2\rangle_q = 0 \tag{43}$$

Thus, we get the updated $\mathbf{v}_k$ in current M step as,

$$\mathbf{v}_k^{(t)} = \frac{\sum_i\langle u_i^{(k)}s_i\rangle_q^{(t)}\mathbf{y}_i}{\sum_i\langle u_i^{(k)}s_i^2\rangle_q^{(t)}} \tag{44}$$

2. Updating σ^2 , we take derivative of with respect to σ^2 and set it equals to 0 as following:

$$\frac{\partial \mathcal{L}_E}{\partial \sigma^2} = -\frac{NP}{2\sigma^2} + \frac{1}{2\sigma^4}\sum_i \mathbf{y}_i^T\mathbf{y}_i + \frac{1}{2\sigma^4}\sum_i\sum_k(-2\mathbf{y}_i^T\mathbf{v}_k\langle u_i^{(k)} s_i\rangle_q + \mathbf{v}_k^T\mathbf{v}_k\langle u_i^{(k)} s_i^2\rangle_q) = 0 \quad (45)$$

Thus, we get the updated σ^2 for parcel $\boldsymbol{k}$ in the current M step as,

$$\sigma^{2(t)} = \frac{1}{NP}(\sum_i \mathbf{y}_i^T\mathbf{y}_i + \sum_i\sum_k(-2\mathbf{y}_i^T\mathbf{v}_k\langle u_i^{(k)} s_i\rangle_q + \mathbf{v}_k^T\mathbf{v}_k\langle u_i^{(k)} s_i^2\rangle_q) \quad (46)$$

3. Updating β, we take derivative of $\mathcal{L}_E(q,\theta)$ with respect to β and set it equal to 0 as following:

$$\begin{aligned}\frac{\partial \mathcal{L}_E}{\partial \beta} &= \frac{\partial[P\log\beta - \beta\sum_i\sum_k\langle u_i^{(k)} s_i\rangle_q]}{\partial\beta}\\ &= \frac{P\alpha}{\beta} - \sum_i\sum_k\langle u_i^{(k)} s_i\rangle_q = 0\end{aligned}$$

Thus, we get the updated β_k in current M step as,

$$\beta_k^{(t)} = \frac{P}{\sum_i\sum_k\langle u_i^{(k)} s_i\rangle_q} \quad (47)$$

Emission model 3b: Mixture of Gaussians with Gamma signal strength

The emission model should depend on the type of data that is measured. A common application is that the data measured at location i are the task activation in N tasks, arranged in the $N \times 1$ data vector $\mathbf{y}_i$. The averaged expected response for each of the parcels is $\mathbf{v}_k$. One issue of the functional activation is that the signal-to-noise ratio (SNR) can be quite different across different participants, and voxels, with many voxels having relatively low SNR. We model this signal to noise for each brain location (and subject) as $s_i \sim Gamma(\theta_\alpha, \theta_{\beta s})$. Therefore the probability model for gamma is defined as:

$$p(s_i|\theta) = \frac{\beta^\alpha}{\Gamma(\alpha)}s_i^{\alpha-1}e^{-\beta s_i} \quad (48)$$

Overall, the expected signal at each brain location is then

$$\mathrm{E}(\mathbf{y}_\mathrm{i}) = \mathbf{u}_\mathrm{i}^\mathrm{T}\mathbf{V}\mathrm{s}_\mathrm{i} \quad (49)$$

Finally, relative to the signal, we assume that the noise is distributed i.i.d Gaussian with:

$$\boldsymbol{\epsilon}_i \sim Normal(0, \mathbf{I}_K\theta_{\sigma s}) \quad (50)$$

Here, the proposal distribution $q(u_i^{(k)}, s_i|\mathbf{y}_i)$ is now a multivariate distribution across u_i and s_i. Thus, the *expected emission log likelihood* $\mathcal{L}_E(q,\theta)$ is defined as:

$$
\begin{aligned}
\mathcal{L}_E &= \langle \sum_i \log p(\mathbf{y}_i, s_i | u_i; \theta_E) \rangle_q \\
&= \sum_i \sum_k \langle u_i^{(k)} [-\frac{N}{2} \log(2\pi) - \frac{N}{2} \log(\sigma^2) - \frac{1}{2\sigma^2} (\mathbf{y}_i - \mathbf{v_k} s_i)^T (\mathbf{y}_i - \mathbf{v_k} s_i)] \rangle_q \\
&+ \sum_i \sum_k \langle u_i^{(k)} [\alpha \log \beta - \log \Gamma(\alpha) + (\alpha - 1) \log s_i - \beta s_i] \rangle_q \\
&= -\frac{NP}{2} \log(2\pi) - \frac{NP}{2} \log(\sigma^2) - \frac{1}{2\sigma^2} \sum_i \sum_k \langle u_i^{(k)} (\mathbf{y}_i - \mathbf{v_k} s_i)^T (\mathbf{y}_i - \mathbf{v_k} s_i) \rangle_q \\
&+ P\alpha \log \beta - P \log \Gamma(\alpha) + \sum_i \sum_k \langle u_i^{(k)} (\alpha - 1) \log s_i - u_i^{(k)} \beta s_i \rangle_q \\
&= -\frac{NP}{2} \log(2\pi) - \frac{NP}{2} \log(\sigma^2) - \frac{1}{2\sigma^2} \sum_i \mathbf{y}_i^T \mathbf{y}_i - \frac{1}{2\sigma^2} \sum_i \sum_k (-2\mathbf{y}_i^T \mathbf{v}_k \langle u_i^{(k)} s_i \rangle_q + \mathbf{v}_k^T \mathbf{v}_k \langle u_i^{(k)} s_i^2 \rangle_q) \\
&+ P\alpha \log \beta - P \log \Gamma(\alpha) + (\alpha - 1) \sum_i \sum_k \langle u_i^{(k)} \log s_i \rangle_q - \beta \sum_i \sum_k \langle u_i^{(k)} s_i \rangle_q
\end{aligned}
$$

Now, we can update the parameters θ of the Gaussians/Gamma mixture in the M step. The parameters of the gaussian mixture model are $\theta_E = \{\mathbf{v}_1, \ldots, \sigma^2, \alpha, \beta\}$.

1. We start with updating the $\mathbf{v}_k$ (Note: the updates only consider a single subject). We take the derivative of *expected emission log likelihood* $\mathcal{L}_E$ with respect to $\mathbf{v}_k$ and make it equals to 0 as following:

$$
\frac{\partial \mathcal{L}_E}{\partial \mathbf{v}_k} = -\frac{1}{\sigma^2} \sum_i -\mathbf{y}_i^T \langle u_i^{(k)} s_i \rangle_q + \mathbf{v}_k^T \langle u_i^{(k)} s_i^2 \rangle_q = 0 \tag{51}
$$

Thus, we get the updated $\mathbf{v}_k$ in current M step as,

$$
\mathbf{v}_k^{(t)} = \frac{\sum_i \langle u_i^{(k)} s_i \rangle_q^{(t)} \mathbf{y}_i}{\sum_i \langle u_i^{(k)} s_i^2 \rangle_q^{(t)}} \tag{52}
$$

2. Updating σ^2 , we take derivative of with respect to σ^2 and set it equals to 0 as following:

$$
\frac{\partial \mathcal{L}_E}{\partial \sigma^2} = -\frac{NP}{2\sigma^2} + \frac{1}{2\sigma^4} \sum_i \mathbf{y}_i^T \mathbf{y}_i + \frac{1}{2\sigma^4} \sum_i \sum_k (-2\mathbf{y}_i^T \mathbf{v}_k \langle u_i^{(k)} s_i \rangle_q + \mathbf{v}_k^T \mathbf{v}_k \langle u_i^{(k)} s_i^2 \rangle_q) = 0 \tag{53}
$$

Thus, we get the updated σ^2 for parcel k in the current M step as,

$$
\sigma^{2(t)} = \frac{1}{NP} (\sum_i \mathbf{y}_i^T \mathbf{y}_i + \sum_i \sum_k (-2\mathbf{y}_i^T \mathbf{v}_k \langle u_i^{(k)} s_i \rangle_q + \mathbf{v}_k^T \mathbf{v}_k \langle u_i^{(k)} s_i^2 \rangle_q) \tag{54}
$$

3. Updating β, we take derivative of $\mathcal{L}_E(q, \theta)$ with respect to β and set it equal to 0 as following:

$$
\begin{aligned}
\frac{\partial \mathcal{L}_E}{\partial \beta} &= \frac{\partial [P\alpha \log \beta - \beta \sum_i \sum_k \langle u_i^{(k)} s_i \rangle_q]}{\partial \beta} \\
&= \frac{P\alpha}{\beta} - \sum_i \sum_k \langle u_i^{(k)} s_i \rangle_q = 0
\end{aligned}
$$

Thus, we get the updated β_k in current M step as,

$$
\beta_k^{(t)} = \frac{P\alpha_k^{(t)}}{\sum_i \sum_k \langle u_i^{(k)} s_i \rangle_q} \tag{55}
$$

4. Updating α_k is comparatively hard since we cannot derive closed-form, we take derivative of $\mathcal{L}_E(q, \theta)$ with respect to α and make it equals to 0 as following:

$$\begin{aligned}\frac{\partial \mathcal{L}_E}{\partial \alpha} &= \frac{\partial [P\alpha \log \beta - P \log \Gamma(\alpha) + (\alpha - 1) \sum_i \sum_k \langle u_i^{(k)} \log s_i \rangle_q]}{\partial \alpha} \\ &= P \log \beta - P\frac{\Gamma'(\alpha)}{\Gamma(\alpha)} + \sum_i \sum_k \langle u_i^{(k)} \log s_i \rangle_q = 0\end{aligned}$$

The term $\frac{\Gamma'(\alpha)}{\Gamma(\alpha)}$ in above equation is exactly the *digamma function* and we use $\digamma(\alpha)$ to represent. Also from (4), we know $\beta = \frac{P\alpha_k^{(t)}}{\sum_i \sum_k \langle u_i^{(k)} s_i \rangle_q}$ Thus, we get the updated α in current M step as,

$$\begin{aligned}\digamma(\alpha)^{(t)} &= \log \frac{P\alpha_k^{(t)}}{\sum_i \sum_k \langle u_i^{(k)} s_i \rangle_q} + \frac{1}{P} \sum_i \sum_k \log \langle u_i^{(k)} s_i \rangle_q \\ &= \log P\alpha^{(t)} - \log \sum_i \sum_k \langle u_i^{(k)} s_i \rangle_q + \frac{1}{P} \sum_i \sum_k \log \langle u_i^{(k)} s_i \rangle_q\end{aligned}$$

By applying "generalized Newton" approximation form, the updated α is as follows:

$$\alpha^{(t)} \approx \frac{0.5}{\log \sum_i \sum_k \langle u_i^{(k)} s_i \rangle_q - \frac{1}{P} \sum_i \sum_k \langle u_i^{(k)} \log s_i \rangle_q} \tag{56}$$

Note that $\log \sum_i \sum_k \langle u_i^{(k)} s_i \rangle_q \geqslant \frac{1}{P} \sum_i \sum_k \log \langle u_i^{(k)} s_i \rangle_q$ is given by Jensen's inequality.

Emission model 4: Mixture of Von-Mises Distributions

For a M-dimensional data $\mathbf{y}$ the probability density function of von Mises-Fisher distribution is defined as following,

$$V_M(\mathbf{y}|\mathbf{v}_k, \kappa) = C_M(\kappa) exp(\kappa \mathbf{v}_k^T \mathbf{y}) \tag{57}$$

where $\mathbf{v}_k$ denotes the mean direction for parcel k (a unit vector), $\mathbf{y}$ has unit length, κ indicates the concentration parameter ($\kappa \geqslant 0$), which is joint over all parcels. $C_M(\kappa) = \frac{\kappa^{M/2-1}}{(2\pi)^{M/2} I_{M/2-1}(\kappa)}$ is the normalization constant where $I_r(.)$ refers to the modified Bessel function of the r order. Thus, the *expected emission log-likelihood* of a mixture of K-classes von-Mises fisher distributions is defined as:

$$\begin{aligned}\mathcal{L}_E &= \langle \sum_i \log p(\mathbf{y}_i | \mathbf{u}_i; \theta_E) \rangle_q \\ &= P \log C_M(\kappa) + \sum_i \sum_k \langle u_i^{(k)} \rangle \kappa \mathbf{v}_k^T \mathbf{y}_i\end{aligned}$$

If the design has repeated measurements of the same M conditions, the user can specify this over the $N \times M$ design matrix X (N is number of observation, M is number of conditions). If we combine across the different repetitions, the resultant data would be $\mathbf{y} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\tilde{\mathbf{y}}$, and then normalized. However, we can also treat the different repetitions as independent observations, meaning that the resultant data is normalized to length 1 for each of the J independent partitions. The likelihood is then the sum over partitions and voxels :

$$\begin{aligned}\mathcal{L}_E &= PJ \log C_N(\kappa) + \sum_i^P \sum_j^J \sum_k^K \langle u_i^{(k)} \rangle \kappa \mathbf{v}_k^T \mathbf{y}_{i,j} \\ &= PJ \log C_N(\kappa) + \sum_i^P \sum_k^K \langle u_i^{(k)} \rangle \kappa \mathbf{v}_k^T \sum_j^J \mathbf{y}_{i,j}\end{aligned}$$

Effectively in the code, the user passes the unnormalized data, a design matrix, and a partition vector. We first compute $\mathbf{y} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\tilde{\mathbf{y}}$ for each partition and then normalize the resultant data in each partition. Finally, we sum the vectors across partitions. $\mathbf{y}_i = \sum_j \mathbf{y}_{i,j}$, and retain the number of observations for voxel i: J_i. The resultant summed vectors are not length 1 anymore, but will be fine as a sufficient statistics.

Now, we update the parameters θ of the von-Mises mixture in the M step by maximizing $\mathcal{L}_E$ in respect to the parameters in vn-Mises mixture $\theta_k = \{\mathbf{v}_k, \kappa\}$. (Note: the updates only consider a single subject).

1. Updating mean direction $\mathbf{v}_k$, we take derivative in respect to $\mathbf{v}_k$ and set it to 0. Then, we get the updated $\mathbf{v}_k$ in current M step in two options: **(A)** learn a common $\mathbf{v}_k$:

$$\mathbf{v}_k^{(t)} = \frac{\tilde{\mathbf{v}}_k}{r_k}, \qquad \text{where } \ \tilde{\mathbf{v}}_k = \sum_i \langle u_i^{(k)} \rangle_q \mathbf{y}_i; \ \ r_k = ||\tilde{\mathbf{v}}_k||$$

2. Updating concentration parameter κ is difficult in particularly for high dimensional problems since it involves inverting ratio of two Bessel functions. Here we use approximate solutions suggested in (Banerjee et al., 2005) and (Hornik et al., 2014 "movMF: An R Package for Fitting Mixtures of von Mises-Fisher Distributions"). If we have N independent observations $\mathbf{y}_i$, each with M dimensions, then we can **(A)** learn a common κ across classes:

$$\begin{gathered}\kappa^{(t)} \approx \frac{\overline{r}M - \overline{r}^3}{1 - \overline{r}^2} \\ \bar{r} = \frac{1}{N}\sum_k^K || \sum_i^N \langle u_i^{(k)} \rangle_q \mathbf{y}_i ||\end{gathered} \tag{58}$$

3.

Alternatively, we can **(B)** learn K-class specific kappa κ_k :

$$\begin{gathered}\kappa_k^{(t)} \approx \frac{\overline{r}_k M - \overline{r}_k^3}{1 - \overline{r}_k^2} \\ \bar{r}_k = \frac{|| \sum_i^N \langle u_i^{(k)} \rangle_q \mathbf{y}_i ||}{\sum_i^N \langle u_i^{(k)} \rangle_q}\end{gathered} \tag{59}$$

For our specific case, we want to integrate the evidence across $s = 1, \ldots, S$ subjects each with $i = 1, \ldots, P$ voxels. Each subject and voxel may have $J_{s,i}$ observations. Under this assumption, the estimates become:

$$\bar{r} = \frac{\sum_k^K || \sum_s^S \sum_i^P \langle u_{s,i}^{(k)} \rangle_q \mathbf{y}_{s,i} ||}{\sum_s^S \sum_i^P J_{s,i}} \tag{60}$$

and for class-specific κ :

$$\bar{r}_k = \frac{|| \sum_s^S \sum_i^P \langle u_{s,i}^{(k)} \rangle_q \mathbf{y}_{s,i} ||}{\sum_s^S \sum_i^P J_{s,i} \langle u_{s,i}^{(k)} \rangle_q} \tag{61}$$

Model Evaluation

After model fitting, we need a fair way to quantitatively compare different emission models between a Gaussian mixture model (GMM), a Gaussian Mixture with exponential signal strength (GMM_exp), and a directional model (VMF). Unfortunately, the three models are defined in different space: the GMM and GMM_exp are defined in $\mathbb{R}^N$ state space while the VMF is defined in $(N-1)$-hypersphere surface. Therefore, the traditional marginal log-likelihood based criterion (BIC, Bayes Factor) cannot provide a fair comparison, as the probability densities would cover different spaces. The main purpose of this section is trying to find evaluation criteria that would be suitable to compare model defined in different space.

Comparing the true $\mathbf{U}$ and the inferred $\hat{\mathbf{U}}$

Note that these criteria only have value for simulations, for which we have the true

1. the absolute error between $\mathbf{U}$ and $\hat{\mathbf{U}}$

the first evaluation criterion is to calculate the absolute error between the true parcellation $\mathbf{U}$ and the expected $\hat{\mathbf{U}}$ which inferred on the training data. It defined as,

$$\bar{U}_{error} = \frac{\sum_i |\mathbf{u_i} - \langle \mathbf{u}_i \rangle_q|}{P} \tag{62}$$

where the $\mathbf{u_i}$ represents the true cluster label of i and $\langle \mathbf{u}_i \rangle_q$ is the expected cluster label of brain location i under the expectation q. Both are multinomial encoded vectors. We can also replace the expectation with a the hard parcellation, again coded as a one-hot vector.

Note, this calculation of the mean absolute error is subject to the premutation of the parcellation, so that a loop over all possible permutations and find the minimum error is applied.

2. Normalized Mutual information (NMI) between $\mathbf{U}$ and $\hat{\mathbf{U}}$

the second criteria is the normalized mutual information which examine the actual amount of "mutual information" between two parcellations $\mathbf{U}$ and $\hat{\mathbf{U}}$. A NMI value closes to 0 indicate two parcellations are largely independent, while values close to 1 indicate significant agreement. It defined as:

$$NMI(\mathbf{U}, \hat{\mathbf{U}}) = \frac{2 \sum_{i=1}^{k_\mathbf{u}} \sum_{j=1}^{k_{\hat{\mathbf{u}}}} \frac{|\mathbf{u}=i| \cap |\hat{\mathbf{u}}=j|}{P} \log(P \frac{||\mathbf{u}=i| \cap |\hat{\mathbf{u}}=j||}{|\mathbf{u}=i| \cdot |\hat{\mathbf{u}}=j|})}{\sum_{i=1}^{k_\mathbf{u}} \frac{|\mathbf{u}=i|}{P} \log(\frac{|\mathbf{u}=i|}{P}) + \sum_{j=1}^{k_{\hat{\mathbf{u}}}} \frac{|\hat{\mathbf{u}}=j|}{P} \log(\frac{|\hat{\mathbf{u}}=j|}{P})} \tag{63}$$

where $k_\mathbf{u} = \{1, 2, 3, \dots, k\}$ and $k_{\hat{\mathbf{u}}} = \{1, 2, 3, \dots, k\}$ represents the cluster labels of $\mathbf{U}$ and $\hat{\mathbf{U}}$ respectively. The term$|\mathbf{u} = i|$ and $|\hat{\mathbf{u}} = j|$ are the number of brain locations that belongs to cluster $k_\mathbf{u} = i$ in parcellation $\mathbf{U}$ or to cluster $k_{\hat{\mathbf{u}}} = j$ in $\hat{\mathbf{U}}$, in other words, the terms $\frac{|\mathbf{u}=i|}{P}$ and $\frac{|\hat{\mathbf{u}}=j|}{P}$ represents the probability that a brain location picked at random from $\mathbf{U}$ falls into class $k_\mathbf{u} = i$, or from $\hat{\mathbf{U}}$ falls into class $k_{\hat{\mathbf{u}}} = j$.

Similarly, the $||\mathbf{u} = i| \cap |\hat{\mathbf{u}} = j||$ means the total number of a brain locations that both falls into classes $k_\mathbf{u} = i$ and $k_{\hat{\mathbf{u}}} = j$. Note, the NMI calculation would not suffer from the permutation.

3. Adjusted rand index (ARI) between $\mathbf{U}$ and $\hat{\mathbf{U}}$

the third one is the commonly used adjust rand index to test how similar the two given parcellations are. It defined as:

$$ARI(\mathbf{U}, \hat{\mathbf{U}}) = \frac{2 \times (M_{11}M_{00} - M_{10}M_{01})}{(M_{00} + M_{10})(M_{10} + M_{11}) + (M_{00} + M_{01})(M_{01} + M_{11})} \tag{64}$$

where M_{11} corresponds to the number of pairs that are assigned to the same parcel in both $\mathbf{U}$ and $\hat{\mathbf{U}}$, M_{00} corresponds to the number of pairs that are assigned to different clusters in both $\mathbf{U}$ and $\hat{\mathbf{U}}$, M_{10} corresponds to the number of pairs that are assigned to the same parcel in $\mathbf{U}$, but different parcels in $\hat{\mathbf{U}}$, and M_{01} corresponds to the number of pairs that are assigned to the same parcel in $\hat{\mathbf{U}}$, but different parcels in $\hat{\mathbf{U}}$.

Intuitively, M_{00} and M_{11} account for the agreement of parcellations, whereas M_{10} and M_{01} indicate their disagreement. Note, the ARI calculation would not suffer from the permutation.

Evaluation on independent test data ($\mathbf{Y}_{test}$)

1. Cosine error

the first evaluation criterion based on the difference between some observed activity profiles $\mathbf{Y}_{test}$ and the predicted mean directions from the model $\mathbf{v}_k$ given some expectation of which voxel belongs to what cluster $\langle \mathbf{u}_i \rangle$. One possibility is to use for each voxel the most likely predicted mean direction.

$$\bar{\epsilon}_{cosine} = \frac{1}{P} \sum_{i}^{P} (1 - {\mathbf{v}_{\underset{k}{\text{argmax}}}}^T \frac{\mathbf{y}_i}{||\mathbf{y}_i||}) \tag{65}$$

where $||\mathbf{y}_i||$ is the length of the data at brain location i, $\mathbf{v}_{\underset{k}{\text{argmax}}}$ represents the $\mathbf{v_k}$ with the maximum expectation for that voxel. We then compute the mean cosine distance across all P brain locations. We can also compute the *expected* mean cosine distance under the $q(\mathbf{u}_i)$ which defined as below:

$$\langle \bar{\epsilon}_{cosine} \rangle_q = \frac{1}{P} \sum_{i} \sum_{k} \hat{u}_i^{(k)} (1 - {\mathbf{v}_k}^T \frac{\mathbf{y}_i}{||\mathbf{y}_i||}) \tag{66}$$

where $\hat{u}_i^{(k)}$ is the inferred expectation on the training data using the fitted model.

2. the Adjusted Cosine error

A possible problem with the cosine error is that voxel that have very little signal count as much as voxel with a lot of signal. To address this, we can weight each error by the squared length of the data vector:

$$\bar{\epsilon}_{Acosine} = \frac{1}{\sum_i^P ||\mathbf{y}_i||^2} \sum_{i}^{P} (||\mathbf{y}_i||^2 - {\mathbf{v}_{\underset{k}{\text{argmax}}}}^T \mathbf{y}_i ||\mathbf{y}_i||) \tag{67}$$

where $||\mathbf{y}_i||$ is the length of the data at brain location i, $\mathbf{v}_{\underset{k}{\text{argmax}}}$ represents the $\mathbf{v_k}$ with the maximum expectation. We then compute the mean cosine distance across all P brain locations. Another option is to calculate the *expected* mean cosine distance under the $q(\mathbf{u}_i)$ which defined as below:

$$\langle \bar{\epsilon}_{Acosine} \rangle_q = \frac{1}{\sum_i^P ||\mathbf{y}_i||^2} \sum_{i} \sum_{k} \hat{u}_i^{(k)} (||\mathbf{y}_i||^2 - {\mathbf{v}_k}^T \mathbf{y}_i ||\mathbf{y}_i||) \tag{68}$$

where $\hat{u}_i^{(k)}$ is the inferred expectation on the training data using the fitted model.

Proof of the adjusted cosine distance is equivalent to $1 - R^2$

Weighting the error by the length of the vector effectively calculates squared error between $\mathbf{y}_i$ and the prediction scaled to the amplitude of the data ($\mathbf{v}_k ||\mathbf{y}_i||$). For simplicity, we use $\mathbf{v}_k$ to represent the most likely predicted mean direction $\mathbf{v}_{\underset{k}{\text{argmax}}}$ for each voxel in the following proof. $1 - R^2$ between $\mathbf{y}_i$ and the prediction scaled to the amplitude of the data ($\mathbf{v}_k ||\mathbf{y}_i||$) is defined as:

$$
\begin{aligned}
1 - R^2 &= \frac{RSS}{TSS} \\
&= \frac{1}{\sum_i ||\mathbf{y}_i||^2} \sum_i (\mathbf{y}_i - \mathbf{v}_k ||\mathbf{y}_i||)^2 \\
&= \frac{1}{\sum_i ||\mathbf{y}_i||^2} \sum_i [(\mathbf{y}_i - \mathbf{v}_k ||\mathbf{y}_i||)^T (\mathbf{y}_i - \mathbf{v}_k ||\mathbf{y}_i||)] \\
&= \frac{1}{\sum_i ||\mathbf{y}_i||^2} \sum_i (\mathbf{y}_i^T \mathbf{y}_i - 2\mathbf{y}_i^T \mathbf{v}_k ||\mathbf{y}_i|| + \mathbf{v}_k^T \mathbf{v}_k ||\mathbf{y}_i||^2) \\
&= \frac{1}{\sum_i ||\mathbf{y}_i||^2} \sum_i (||\mathbf{y}_i||^2 - 2\mathbf{y}_i^T \mathbf{v}_k ||\mathbf{y}_i|| + ||\mathbf{y}_i||^2) \\
&= \frac{2}{\sum_i ||\mathbf{y}_i||^2} \sum_i (||\mathbf{y}_i||^2 - \mathbf{y}_i^T \mathbf{v}_k ||\mathbf{y}_i||)
\end{aligned}
$$

By equation (67), we can see that $1 - R^2 = 2\bar{\epsilon}_{Acosine}$, and similarly we can easily proof below equation:

$$
\langle \bar{\epsilon}_{MSE} \rangle_q = \frac{1}{P} \sum_i \sum_k \hat{\mathbf{u}}_i^{(k)} (\mathbf{y}_i - \mathbf{v}_k ||\mathbf{y}_i||)^2 = 2 \langle \bar{\epsilon}_{Acosine} \rangle_q
$$

where $\hat{\mathbf{u}}_i^{(k)}$ is the inferred expectation on the training data using the fitted model.