Gaussian Process Classifiers (GPC)

Here we provide a brief overview of the Gaussian Process (GP) prediction models that summarises material presented in more detail elsewhere (Kuss and Rasmussen, 2005: Rasmussen and Williams, 2006).

We assume a set of training data D={**X,y**}, where **X** is an m x d matrix consisting of input vectors $\mathbf{x}_i$ (i.e. m training samples with d features each) and **y** is a column vector of target variables for classification where $y_i \in \{+1, -1\}$. Training samples are indexed by i **=**1,…,m. The goal is to identify some function from the training data that allows us to accurately predict a new target y* from a new data sample **x***. For binary GPCs, as those employed here, predictions take the form of class probabilities p(y*=1**|x*, D**). In GPC, prediction proceeds by placing a GP prior over an unconstrained latent function and computing its posterior distribution. In this case, we do not observe this function directly, but instead use a sigmoidal response function to map it to the unit interval. While there are several possibilities for this function, here we use the cumulative Gaussian density Φ (x) (or probit likelihood) which can be computed as

$$\sigma(x) = \Phi(x) = \int_{-\infty}^{x} N(u|0,1)du$$

The response function thus converts an unbounded regression problem into a classification problem where output is constrained to the unit interval, ensuring a valid probabilistic interpretation.

For binary classification, we can write each likelihood term as $p(y_i|f_i)=\Phi(y_i f_i)$ (owing to the symmetry of the probit likelihood) and rewrite Bayes rule as:

$$p(\mathbf{f}|D,\boldsymbol{\theta}) = \frac{p(\mathbf{f}|\mathbf{X},\boldsymbol{\theta})p(\mathbf{y}|\mathbf{f})}{p(D|\boldsymbol{\theta})} = \frac{N(\mathbf{f}|\mathbf{0},\mathbf{K})}{p(D|\boldsymbol{\theta})}\prod_{i=1}^{m}\sigma(y_i f_i) \quad \text{(S1)}$$

Here, $\mathbf{f} = (f_1,\ldots,f_m)^T$ collects the latent function values at training points, N(**f|0,K**) describes the prior over the latent function (i.e. the covariance function), $p(D|\boldsymbol{\theta})$ is the marginal likelihood or model evidence and we have factorised the likelihood over training samples

(because the class labels are independent given the latent function). In this work, we use a simple inner product covariance function. Making GPC predictions is a two-step process. First, we compute the distribution of the latent variable at the test point, and then we compute its expectation to produce a probabilistic prediction. As opposed to point predictions produced by Support Vector Machines, class probabilities are derived from integrating over the entire distribution for the latent function at the test data point.

Exact inference for GP classification is not analytically tractable, but the posterior and marginal likelihood in equation S1 can both be approximated by Gaussians. The approximate posterior can be written as $q(\mathrm{f}|D,\boldsymbol{\theta}) = N(\boldsymbol{\mu},\boldsymbol{\Sigma})$, where the approximate parameters $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are computed using the Expectation Propagation algorithm (Minka, 2001; Rasmussen and Williams, 2006). Once the approximate posterior has been computed, it can then be used to compute: (1) the marginal likelihood and (2) the approximate posterior for the test case. Following Kuss and Rasmussen (2005), the latter can be computed by:

$$
\begin{aligned}
q(f^*|D,\mathbf{x}^*,\boldsymbol{\theta}) &= N(f^*|\mu^*,{\sigma^2}^*) \\
\mu^* &= \mathbf{k}^{*\mathrm{T}}\mathbf{K}^{-1}\boldsymbol{\mu} \\
{\sigma^2}^* &= k(\mathbf{x}^*,\mathbf{x}^*) - \mathbf{k}^{*\mathrm{T}}(\mathbf{K}^{-1} - \mathbf{K}^{-1}\boldsymbol{\Sigma}\mathbf{K}^{-1})\mathbf{k}^*
\end{aligned}
$$

Finally, predictions are made by computing the posterior expectation of the latent function at the test point:

$$
\begin{aligned}
p(y_* = 1|D,\mathbf{x}^*) &= \int \Phi(f^*)q(f^*|D,\boldsymbol{\theta},\mathbf{x}^*)df^* \\
&= \Phi\left(\frac{\mu^*}{\sqrt{1+{\sigma^2}^*}}\right)
\end{aligned}
\qquad \text{(S2)}
$$

Training a GP model refers to finding the best functional form and optimising any free hyperparameters for the covariance function, which is commonly done by maximising the logarithm of the marginal likelihood. The marginal likelihood measures the total probability of the data given the model hyperparameters and has the attractive property that it constitutes

a trade-off between good fit to the data and a penalty for model complexity, so that simpler models are favoured. We refer the reader to other sources for a detailed treatment of training GPC models (Rasmussen and Williams, 2006; Marquand et al., 2010). Note that the PROBID toolbox used in this study employs the Gaussian processes for machine learning toolbox (www.gaussianprocess.org/gpml) for all GPC inference.

Discrimination Maps

Discrimination mapping for GPC is described in detail in Marquand et al., (2010), but briefly, for linear covariance functions, such as those employed here, it is possible to construct a spatial representation of the discrimination function, which can be computed by:

$$\hat{\mathbf{w}} = \mathbf{X}^{\mathrm{T}}\mathbf{K}^{-1}\mathbf{y}$$

The quantity $\hat{\mathbf{w}}$ is referred to as the maximum a posteriori (MAP) estimate of the GPC weight vector, and is the best point estimate of the GPC decision function (i.e. the mode of the Gaussian approximation in voxel space).

Here, we aim to provide an intuitive interpretation of the maps and consider a simplified version of the GPC decision function, given by $p(y = \text{class } 1|\mathbf{x}, \hat{\mathbf{w}}) = \sigma(\mathbf{x}^{\mathrm{T}}\hat{\mathbf{w}})$, where y is the class label of a test subject (in this paper, y>0.5, corresponds to class 1 or bipolar disorder, otherwise class 2 or control), **x** is the feature vector containing gray or white matter voxels for the test subject, $\hat{\mathbf{w}}$ is the MAP estimate of the GPC weight vector (i.e. the discrimination map) and σ is a sigmoid function that maps the values to the interval [0,1].[1]

During the training phase any GPC hyperparameters are computed by maximizing the logarithm of the marginal likelihood as described above. During the test phase, for classifying a new example we first multiplied each voxel by its corresponding coefficient in the weight vector. After that we add all multiplied values and pass the sum through a sigmoid

[1] Note that this is the MAP approximation for GPC and accurately reflects its behaviour in that it is guaranteed to give predictions with the same class label as the true GPC predictive probability (see Rasmussen and Williams, 2006).

function in order to obtain an output between 0 and 1 (which are predictive probabilities). This process is illustrated graphically in figure 1 (main text).

References:

Kuss M, Rasmussen CE (2005) Assessing approximate inference for binary Gaussian process classification. Journal of Machine Learning Research 6:1679-1704.

Marquand A, Howard M, Brammer M, Chu C, Coen S, Mourao-Miranda J (2010) Quantitative prediction of subjective pain intensity from whole-brain fMRI data using Gaussian processes. NeuroImage 49:2178-2189.

Minka T (2001) A Family of Algorithms for Approximate Bayesian Inference (PhD Thesis). In. Massachusetts: MIT press.

Rasmussen C, Williams CKI (2006) Gaussian Processes for Machine Learning. Cambridge, Massachusetts: The MIT Press.