COMPUTER SCIENCE
UNIVERSITY OF MARYLAND

Bayesian Nonparametrics Analysis of Spatial Temporal fMRI Signals

Philip Yang
Dept. Computer Science
University of Maryland
College Park
phi@cs.umd.edu

In any particular theory there is only as much real science as there is mathematics.
Immanuel Kant

This scholarly paper authored by Philip Yang will be used to partially fulfill the requirement of **MSc in Computer Science** (without thesis) at University of Maryland, College Park.

Yiannis Aloimonos, Professor
Department of Computer Science
Director, Computer Vision Laboratory
Institute for Advanced Computer Studies
Neural and Cognitive Science Program
University of Maryland, College Park
Email: yiannis@cs.umd.edu, Tel. 3014051743

1 Introduction

The human brain has approximately 86 neurons. In the scale of single neurons, the mechanism of active potential is well understood. Yet at the ensemble level, things could get quite complicated. The most I've heard from neuroscience practitioners is that we don't even where to look for.

We do not aim nor hope to understand the brain's operation mechanisms through fMRI. Rather, we show what **is** revealed of such mechanism by the fMRI scans - the *spatial and temporal patterns* of the tens of thousands of BOLD time series in a **disciplined** manner, and shed light on how to design further experiments to test our hypothesis on a finer scope.

But what amounts to a pattern? Many, like [3, Bullmore et al. 2012], proposed hypothesis based on the "patterns" extracted from fMRI scans. Pearson correlation of time series is blindly used to construct graph representations. Yet none has done a careful analysis of the implication of such measures. Similar cases are countless.

To us, the patterns that **differentiate two or more conditions** are a more reliable means to study the fMRI scans. These are the patterns to help us distinguish when

- Seeing a picture of a house versus a face
- Recognizing a concrete word versus an abstract concept
- Anticipating a potential threat or not

Traditionally such patterns are found with unsupervised learning. The researcher picks up a specific aspect of the data (spatial or temporal) and learn some specific decomposition

(PCA, ICA, etc.). He would then look into the learned representation, transform the data and perform statistical test of significance. If $p < 0.05$, he will start to find a neuroscience interpretation.

This process is not scalable with respect to the shear number of combinations. It is also unnecessarily tedious for exploration. Instead, we can consider generating many features from both a neuroscience perspective and the data, and perform supervised learning to retain those that **consistently** distinguish the conditions. This has also the advantage of revealing added strength of multiple different features. Furthermore, the significant of a highly generalizable model is **self-evident** - we do not have to lean on a specific statistical test with complicated set of assumptions that we can't easily verify.

With a small subset of feature selected, we can go ahead an look closely into why they are predictive. If we found a group regions (say, `V1`, `MT` and `PFC`) are significant, we should pull up their associated time series and study

- The temporal correlation structure and spectrum
- The variance among spatially contiguous voxels / vertices within the region

Fourier / Gabor transform, wavelet and Gaussian process are suitable for such task.

1.0.1 The Big Small Data

Our hands are tight with very small dataset. With a 20 people experiment, we have at best about 100 data points at all. Almost all the learning algorithms have fail (SVM, Random Forest, LASSO, etc.) on the raw voxel variables. Yet the size of each participant's collected data easily crosses the gigabyte boundary.

The Human Connectome Project (`HCP`) is an exception. How to leverage what we learn from a large dataset like `HCP` to smaller ones?

On a individual level, we have access to

- Cortical surface thickness, local area, curvature, myelin map, etc.
- Gene expression level (Allen brain atlas)

There are also many public accessible fMRI datasets and derivatives

- `OpenfMRI` (`https://openfmri.org`)
- The `NITRIC` datasets (`http://www.nitrc.org`)
- `OBHM` Hackathon (`http://ohbm-seattle.github.io`)
- `BrainMap` (`http://www.brainmap.org`)
- `Neurosynth` `http://neurosynth.org`

Can we leverage all of them to train more effective learning algorithms? Can we efficiently search the result from literature to corroborate our findings?

The available fMRI data pile up to tens of terabytes, consisting of a multitude of different formats and structures. How to effectively process them into multiple levels of representations to facilitate fast query and visualization is by no means an easy task. (It takes more than two thousand lines of code cross four different languages to transform the `HCP` dataset into the format for analysis and visualization).

2 FMRI, A Primer

Magnetic Resonance Imaging (MRI) is one of the the predominant means to obtaining information of the brain *in-vivo*.

In a typical neuro-imaging study, about 20 to 40 people participate by performing several experimental tasks in the MRI scanner. These tasks include

- watching a video clip
- listening to some auditory content
- reading text
- performing simple actions like squeezing hands / toes
- answering multiple choice questions
- receiving mild electrical shock

The external inputs to the experiment participant are termed *stimuli*. For a more specific list of tasks, check out `http://www.umiacs.umd.edu/~phi/hcp_dataset_slides.html`.

There are also cases where participants are asked to simply stay still and neither perform nor think of anything during the scan. These are called "**resting-state**" scans, which some believe to be able to reflect what the brain does when it is not concentrated on anything [2, Buckner et al. 2013].

For each participant, these data are collected

- "**Functional**" scan, consisting of several hundreds to more than a thousand "snapshot" of the brain during the experiment period. These "snapshots" are tuned to be sensitive to the blood oxygenation level, which to some degree reflexes the underlying activities of the neuronal ensemble. Each "snapshot" usually takes a several second to acquire (in our own experiments, 2.5s), although in some special cases (the *Human Connectome Project*), it takes "only" 0.72s. The blood oxygenation level decedent (BOLD) signals however do not have a very fine intrinsic time resolution, so that's about appropriate to acquire a good measurement. One usually have to make a trade-off between spatial and temporal resolutions.
- **Structural** scan of the brain (higher spatial resolution) which is used to align data from different participants to a common brain template so that we can perform group level study. The structual scan are also important in its own right, especially in the case when the goal is to study effects of diseases such as Alzheimer's or schizophrenia.
- Physiological data, including
  - respiratory / cardiac rate
  - head movements

  which are useful to remove noise in the functional data.
- Participants' responses to the task questions, if any.

The data go through preprocessing,

1. Removing noise related to high frequency physiological signals and scanner field bias.
2. Alignments
   (a) Align each "snapshots" of the functional scans into a common frame (the participant moves during the experiment).
   (b) Align the functional scans to the structural scan
3. Align and register the structural scan to a common brain template.

The human neo-cortex is a thin sheet of 6 layers. All the "functional" activities happens in these **gray matter** (referring to its postmortem coloring) whereas the **white matter** accounts for the physical wirings among the neurons. It is more popular these days to also extract gray matter sheet as a 3D triangle mesh.

So optionally during the preprocessing, we

- Extract the cortical surface from the structural scan
- Project the functional scan to this surface

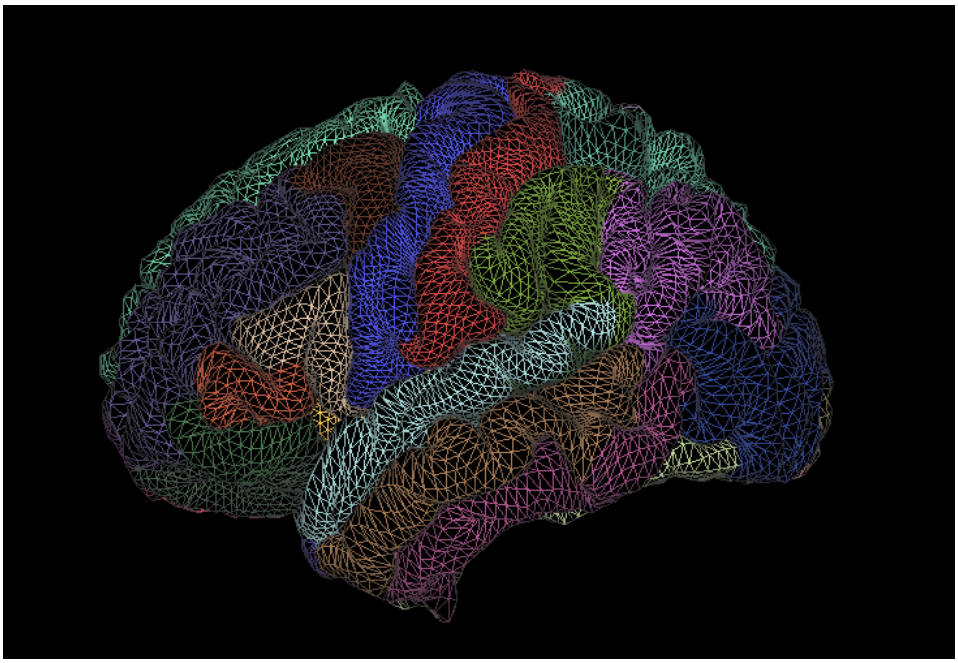

Figure 1: An example of the 3D cortical surface triangle mesh

- Align the register the triangle mesh to a common surface by cross-referencing the landmarks of gyri, sulci and accounting for folding patterns.

So in the end, we have in our hand, a dataset of signals / time series indexed either by their 3D voxel positions or the 3D vertex ID on the cortical surface. We know for each participant, when the stimuli show up (onset) and the duration, by which we can refer to the corresponding brain "responses" in the signals.

Typically one has as many as more than twenty thousand voxels or vertices. Each time series indexed by a single voxel / vertex, however, has only about several hundred (regularly sampled) time points of which only a rather small portion is related to the task. It is a daunting task to find meaningful patterns within these *large-p small-n* data.

3 Time Series and Signal Processing

3.1 Finding Groups of Coherent Temporal Patterns

When the participant perform an attention demanding task, we would expect multiple regions of the brain to be activated. Given that activated brain regions should generate signals with similar shapes, we should find a large amount of time series in our dataset to behave similarly during the task period.

In fMRI scan each individual has more than ten thousand voxel indexed time series. Finding groups of similar time series among them is a daunting task.

Community detection [4, Crossley et al. 2013] is a common technique in the computational neuroscience literature to detect consistent patterns across different regions of interest (ROI). A graph is generated over the set of ROIs where the edge weights is computed from some pairwise similarity functions. A *community* is defined as a *maximally similar* set of ROIs time series. Time series within the same community are highly *similar* than those in other communities. Algorithms such as [6, Girvan and Newman 2002] is used to find these communities.

Such methods are limited in that their usefulness rely heavily on the similarity function. It is difficult to come up with a sufficiently good way to measure the similarity of time series with rich dynamics regimes. Pearson correlation, the most commonly employed method, often fails to capture non-linearly correlations [1].

To study the directly the properties of the BOLD time series, people would usually apply variants of factor analysis methods such as principle component analysis (PCA) or independent component analysis (ICA). Yet this either try to find a *linear subspace* to minimize the reconstruction error (or to best explain the empirical variance) or to decompose the data into statistically independent factors.

Our goal is to effectively learn a model to discover groups of time series with coherent temporal patterns without imposing much structural assumptions *a priori*. A *generative* clustering method is suitable to exploit the *spatial locality* of fMRI data - BOLD signals are usually similar across adjacent voxels.

A Gaussian process [11, Rasmussen et al.2006] defines a prior distribution over the space of functions. Its finite dimensional marginal distribution is multivariate Gaussian, making it very flexible to be combined in a mixture model. As a mechanistic model, we can assume that the BOLD time series are generated from a mixture of Gaussian processes.

One common problem for mixture model is the need to determine the number of components K. Algorithms such as k-means will dutifully find K components even if the data disagree. Dirichlet process mixture allows us to learn the number of components from the data.

Dirichlet mixture of Gaussian processes have been successfully applied to study different dynamics regimes of time series [8, Meeds et al. 2006] [12, Rasmussen et al. 2002]. Similar to us, [14, Ross and Dy 2013] developed a nonparametric Dirichlet mixture of Gaussian processes for clinical studies.

The paper is organized as follows. We first give a brief review on Gaussian process and Dirichlet process in the first two sections. After that we describe our mixture model and algorithm settings. In the experiment section, we show the properties of the models learned via variational inference in a fMRI dataset.

3.1.1 Gaussian Process

A Gaussian process is a stochastic process with index set $\mathcal{X}$. Under this model, the data y is assumed to be generated from a latent function which follows Gaussian process with mean m and a covariance structure defined by a positive semi-definite kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$.

$$\begin{aligned} y &= f(x) + \epsilon \\ \epsilon &\sim \mathcal{N}(0, \delta_n^2) \\ f &\sim \mathrm{GP}(m(x), k(x, x')) \end{aligned}$$

Notice that the correlation of the functions are solely determined by their input values [11, Rasmussen 2006]. Gaussian process can be marginalized at a finite subset $\{x_1, \ldots, x_n\} \subset \mathcal{X}$, resulting in a multivariate normal distribution

$$\mathcal{N}(\mu, K) \textit{ where } \mu_i = m(x_i) \textit{ and } K_{i,j} = k(x_i, x_j)$$

Thus for a finite sample $y_1, \ldots, y_n$, sampled at such points, we can easily integrate out the latent function f and obtain the marginal log-likelihood as

$$\log p(y \mid X) = -\frac{1}{2}(y-\mu)^T (K + \delta_n^2 I)^{-1}(y-\mu) - \frac{1}{2}\log\left(\left|K + \delta_n^2 I\right|\right) - \frac{n}{2}\log 2\pi.$$

Since our prior and likelihood are both Gaussian distribution, the posterior is also Gaussian.

[1] DEFINITION NOT FOUND.

3.1.2 Dirichlet Process

Dirichlet process can be defined with respect to the conditional predictive distribution as follows. Given a finite sample $\eta_1, \eta_2, \ldots, \eta_n$, where η_i follows some distribution G_0,

$$\eta_i = \eta \mid \eta_1, \ldots, \eta_{i-1} \sim \frac{1}{i-1+\alpha} \sum_{j=1}^{i-1} \delta(\eta_j) + \frac{\alpha}{i-1+\alpha} G_0$$

Here G_0 is our GP prior. The right hand side is a valid probability measure since it is a convex combination of a finite number of probability measures. Notice that previously chosen values of η will have a non-zero probability to be chosen, even if the base measure G_0 is continuous. Thus we can use this concentration on discrete value property for clustering.

3.1.3 Dirichlet Process Mixture of Gaussian Processes

Equivalently, one can directly take the mixture of component perspective and introduce a latent cluster assignment variable. In this case, we first sample this categorical random variable through an infinite stick-breaking process [1, Blei and Jordan 2006] and sample the η for each mixture component from G_0 the base measure. This perspective allows us to derive an efficient mean-field *variational inference* algorithm to estimate the posterior [1, Blei and Jordan 2006].

In our case, each mixture component is a Gaussian process. Since all the time series are measured on the (approximately) same time point, we only have to work with the finite marginalized Gaussian processes. We use the Dirac delta kernel with a modulation factor $k(x, x') = h(x)\delta(\|x - x'\|)h(x')$. The finite marginalized version is simply a multivariate Gaussian with an anisotropic diagonal covariance matrix. The rationale is to not to assume any specific property of the time series and to allow heteroskedasticity in the time domain. We let G_0 to be an inverse Gamma distribution.

3.1.4 Experiments

We performed experiment on a single subject from the *Human Connectome Project*. The `tfMRI` data of the *left hemisphere* from a single run of the language task is taken. During the task, the subject is presented with auditory stimuli of either a *story* or a simple *math* problem. The subject is given a multiple choice problem related to the content after each stimulus is presented. The dataset consists of 29696 time series, each contains 316 time points.

We have performed experiments on both original and scaled data (normalized time series). In `fig-1` we show the number of activated components with respect to α. We performed *truncated stick breaking* with a multinomial distribution (listed as `sim`). In both experiments we set $\alpha = 1.23$.

1. Original Data The algorithm picked up 15 components, among which 10 are associated with more than 1000 time series. The rest 5 components shows clearly a pattern of random noise. The mean time series of the 10 non-noise components have basically the same shape, but different mean scale, as can be seen in `fig-2`.

   We choose the component whose time series has the largest mean scale. The areas mostly reside in this component are *superior frontal*, *rostral middle frontal*, *superior temporal* and *insula* as is listed in the *Deskikan-Killiany* cortical parcellation. All this are related to cognitive and auditory processing. The local maxima and minima in this time series aligns well with the onset of stimuli, as is seen in `fig-3`.
2. Scaled Data Although the model learned on the original data shows a globally salient temporal pattern, it is nonetheless affected by the absolute scale of each individual time series. To show the association of the temporal variations, we performed a second group of experimentation with each time series mean subtracted and variance scaled.

   We have trained the model over a range of α. The number of components grows as α gets larger, but eventually saturates at around 50.

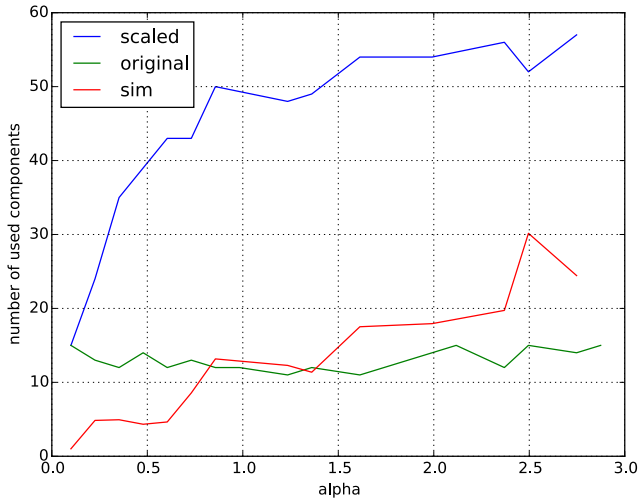

Figure 2: Number of activated components with respect to α.

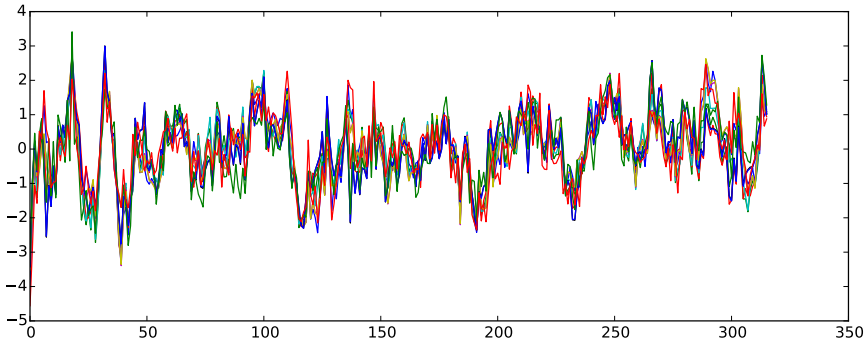

Figure 3: The 10 mean time series normalized, showing strong overlapping (best viewed in color).

We show in the appendix the plot of all component mean time series in this model. A few of the components have very similar mean time series, which is reflected in `fig-5`. A very small Pearson correlation based distance indicates high similarity.

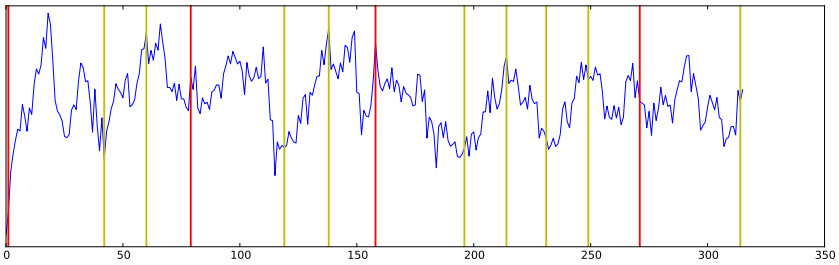

Figure 4: A mean time series chosen by the model. Yellow and red vertical lines stands for the onset of a math or story event respectively

As a result, one could optionally group the components among which the distances are tiny.

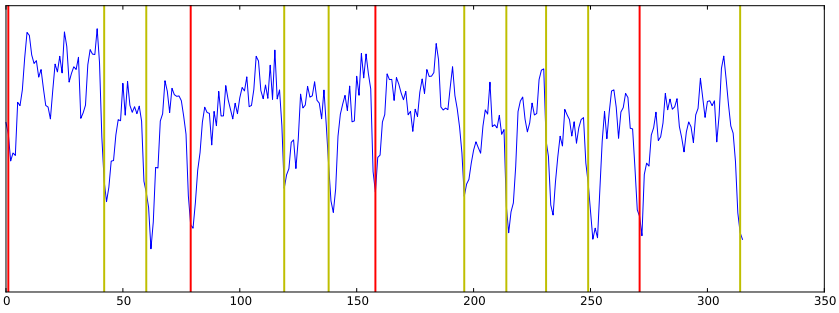

Figure 5: A component mean time series well aligned with stimuli onset

Similar to the experiment on original data, we find a component whose mean time series is very well aligned with the auditory stimuli onset. This component contains 354 elements, with a majority residing in *superior temporal* , *supramarginal* and *transverse temporal*, all of which are important functional components in language and auditory processing. The alignment with event onset and the component mean time series is shown in `fig-4`.

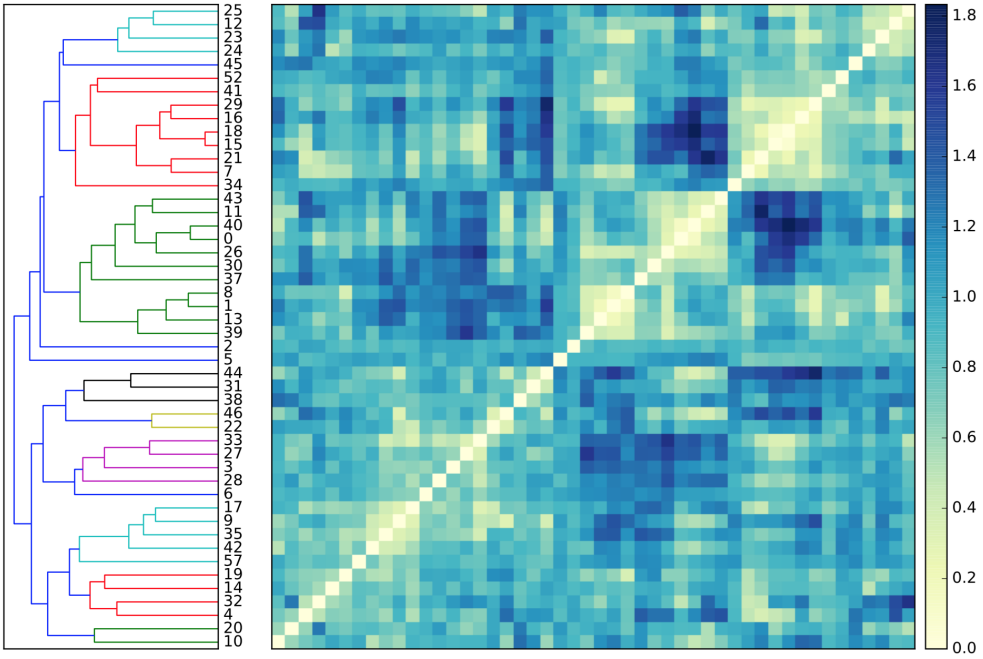

Figure 6: Pearson correlation distance matrix of the learned mean time series in scaled data at $\alpha = 1.23$.

3.1.5 Conclusion

We have proposed a Dirichlet process mixture of Gaussian processes model as an exploratory tool to study fMRI BOLD time series. We have demonstrated its ability to empirically finding interesting temporal patterns in a language related cognitive task. Further

study should exploit the intrinsic low frequency nature of BOLD signals and construct a more detailed model to encode both that and the high frequency noise. Faster variational inference algorithms such as [7, Hensman et al. 2012] and well optimized implementations could potentially speed up the computation.

3.2 Effective Gaussian Process Kernel Learning

What if we want to mixture model to pick up more complicated temporal patterns? Consider a more simplified case where we have a total of m time series generated from the same Gaussian process independently. How could we effectively learn the covariance kernel from these?

We observe that the finite sample marginal of a Gaussian process is a multivariate Gaussian distribution. Consider the Borel σ-algebra of the $\mathbb{R}^n$ where n is the number of samples in a time series. The theory of *Reproducing kernel Hilbert space* tells us that with a suitable kernel (here we use Weiestrass), we can construct a Hilbert space $\mathcal{H}$ where the kernel serves as a evaluation kernel (like Dirac delta for L_2). Furthermore, elements in $\mathcal{H}$ is dense in the space of bounded continuous random variables of the measurable space $(\mathbb{R}^n, \mathcal{B}(\mathbb{R}^n))$. To show that two distribution p, q in $(\mathbb{R}^n, \mathcal{B}(\mathbb{R}^n))$ are "equal", it suffices to show that their mean functional is in the bounded linear functional of $\mathcal{H}$ and that the expectation of all elements in $\mathcal{H}$ for p, q are equal. A metric termed *Maximum Mean Discrepancy* (MMD) computes the largest distance between p, q for elements in $\mathcal{H}$.

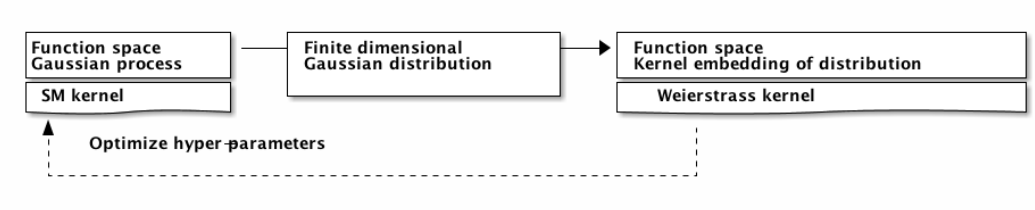

Figure 7: Flow chart of Gaussian process kernel learning

Let's consider the case in $\mathrm{MMD}(p, q)$ where the distribution p is only available through its samples. Denote the empirical distribution as $\hat{p}$. We can then rewrite the formula as

$$\mathrm{MMD}(\hat{p}, q) = \mathbb{E}_{s,s'}\left[k(s, s')\right] - 2\mathbb{E}_{s,t}\left[k(s, t)\right] + \mathbb{E}_{t,t'}\left[k(t, t')\right],$$

where $(s, s') \sim \hat{p} \times \hat{p}$, $(t, t') \sim q \times q$ and k is the Weierstrass kernel. We will take this quantity as a function of $q(\theta)$ and minimize it.

Notice that the first term is only related to the samples, which could be ignored for the optimization. The second term could be considered as interaction between the samples and $q(\theta)$. The last one is to be treated as a regularization term for the parameters. We can write it as a function of the parameters as

$$L_w(\theta, Y) = -\frac{2}{m}\sum_{i=1}^{m}\mathbb{E}_y\left[k(y_i, x)\right] + g(\theta).$$

Let f be the multivariate Gaussian density. According to the formulae derived in the last section,

$$\begin{aligned}
L_w(\theta, Y) &= -\frac{2}{m}\sum_{i=1}^{m}\mathbb{E}_y\left[k(y_i, x)\right] + g(\theta), \\
&= -\frac{2}{m}\sum_{i=1}^{m} W(f)(y_i) + \int_{\mathcal{X}} f(t)W(f)(t)\,dt \\
&= -\frac{2}{m}\sqrt{\frac{1}{(2\pi)^n\left|2I + K_y\right|}}\sum_{i=1}^{m}\exp\left(-\frac{1}{2}(y_i - \mu)^T(2I + K_y)^{-1}(y_i - \mu)\right) + \sqrt{\frac{1}{(4\pi)^n\left|I + K_y\right|}}.
\end{aligned}$$

In 1D, the formula could be written as follows.

$$L_w(\mu, \delta^2, Y) = \sqrt{\frac{1}{4\pi(1+\delta^2)}} - \frac{2}{m}\sqrt{\frac{1}{2\pi(2+\delta^2)}} \sum_{i=1}^{m} exp\left(-\frac{1}{2}(y_i - \mu)^2/(2+\delta^2)\right).$$

Notice that in the case of a normal distribution, the regularization term is independent of the mean. The figure shows the empirical L_w with respect to a sample size of 300 drawn from a standard normal distribution ($\mu = 0$ and $\delta = 1$). We manually set the mean $\mu = 0$ and tested a range of δ values shown in the horizontal axis. The loss function clearly achieved the minimum value at $\delta = 1$. Furthermore, for the standard normal distribution, we can expect the empirical estimation of the first term to be about 0.1995, by substituting $\delta = 1$ to the second term. With this we can estimate how close the distribution we sampled from is to the best possible normal distribution estimation by the theorems in the previous session.

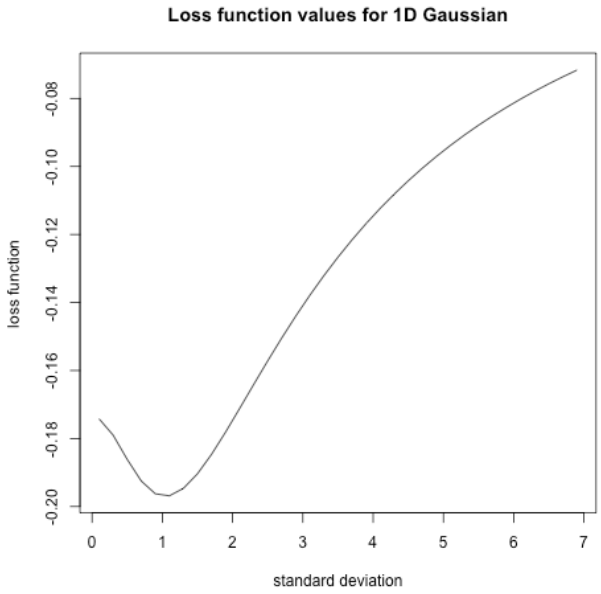

Figure 8: Loss function under standard normal distribution

Moreover, the plot indicates that the loss function is neither convex nor concave. Yet it is tempting to check its monotonicity before and after the minimum point. We could in general use Newton's method if, like in this case, the loss function has an closed form. It might also be useful to utilize that fact that Weiestrass transform can be approximated arbitrarily well with an analytical function (that is, functions given by complex

4 High Performance Cloud Computing

To build a system capable of learning complex patterns and relationships, we must be able to build and maintain robust and high performance computing systems.

Querying over hundreds of millions of time series is challenging. `HCPY` is a software package to help us navigate terabytes of neuro-imaging data in real time.

Design criteria

- Must run in any reasonable Linux distribution
- Allow data integrity checking

- Allow identification of problems in data (original or intermediary)
- Scalable to many compute nodes
- Elastic to node churns (sudden addition and removal of nodes)

The design principle is to separate these parts of system and try to maintain a relative low coupling among them. For the long running code, logging to text file is sufficient and necessary. It allows any program or human beings to monitor its progress. In addition, when the job is done or failed, the program should generate another file to indicate the status. A program could launch the job and use this indicator file to determine the exit status of the job. And that's all that is needed.

The working environment is as follows. We have a handful of servers managed by ourselves. The code could run there as long as the machine is online. There are several managed cloud computing infrastructures on campus, notably

- `CHIMERA`: our main dev-cluster, usually can acquire up to 4 nodes.
- `BISWIFT`: mostly available. No GPU.
- `DEEPTHOUGHT2`: best equipped, mostly occupied, but there is a scavenger queue for a quick job. A small number of `K20c` GPUs are available.

The dataset is physically stored in a NAS server running `FreeNAS`, a modified version of `FreeBSD`. The data access functionalities are abstracted with a python module (to provide "soft" access control).

We have to try to keep enough data in memory. The extracted data are kept in highly compressed format to save space. Uncompressing these files take a long time. Since each database client instance usually has to handle more of these files than the number of available cores, it is reasonable to spawn for each file a `subprocess` for decompression.

Once a query is submitted, the system is expected to pull all the data necessary for that query, further process them, and return the result (could still be big) into the client side.

Pulling data from remote site could be time consuming. We would like to first download all or part of the data into the local file. Since the remote site only stores the data, not any processing logic, further filtering has to be done locally. For the moment we assume that after this step all the data related to the query could be stored locally on the distributed compute nodes.

MPI is used as resource manager and database bootstrapping mechanism. This is because MPI and specifically the *Open-MPI* implementation is adaptive to most interconnected computers with some Linux distributions. A simple master server type of job queue is all that we need. During a large batch job such as preprocessing or information retrieval from the original data, problem is likely to occur.

Most queries request going through all the data. Yet not all of them require a sustained period of time. In this case we can scavenge the compute nodes in the spot job queues to accelerate the query.

The MPI processes must be able to communicate with the underlying processing workers. This is done through file system notification `inotify`. This will help reduce the coupling of the sub-systems and also make the intermediary result accessible through normal file operations from the shell.

5 Appendix

5.1 Fallacies

Computing the entropy and mutual information is not a matter of triviality.

Non-parametric - discretization Discretizing the continuous distribution is not quite a good choice. The number of "bins" grows exponentially, outpacing the number

of samples. We should always expect many bins without any observations. It has been shown by [10, Nemenman_NIPS_2002_Entropy-notes] that any choice of bins and the corresponding Dirichlet type prior will largely determine the entropy distribution when the sample size is not large enough. This effect can be shown in figure 9, where we obtained samples of different sizes from a normal distribution with $\mu = 5$ and $\delta = 7$. The methods are all quite consistent, yet largely biased towards $\log(K)$ where K is the number of bins used to discretize a finite support of $(-5+\mu, 5+\mu)$. This is analyzed in detail by [10, Nemenman_NIPS_2002_Entropy-notes].

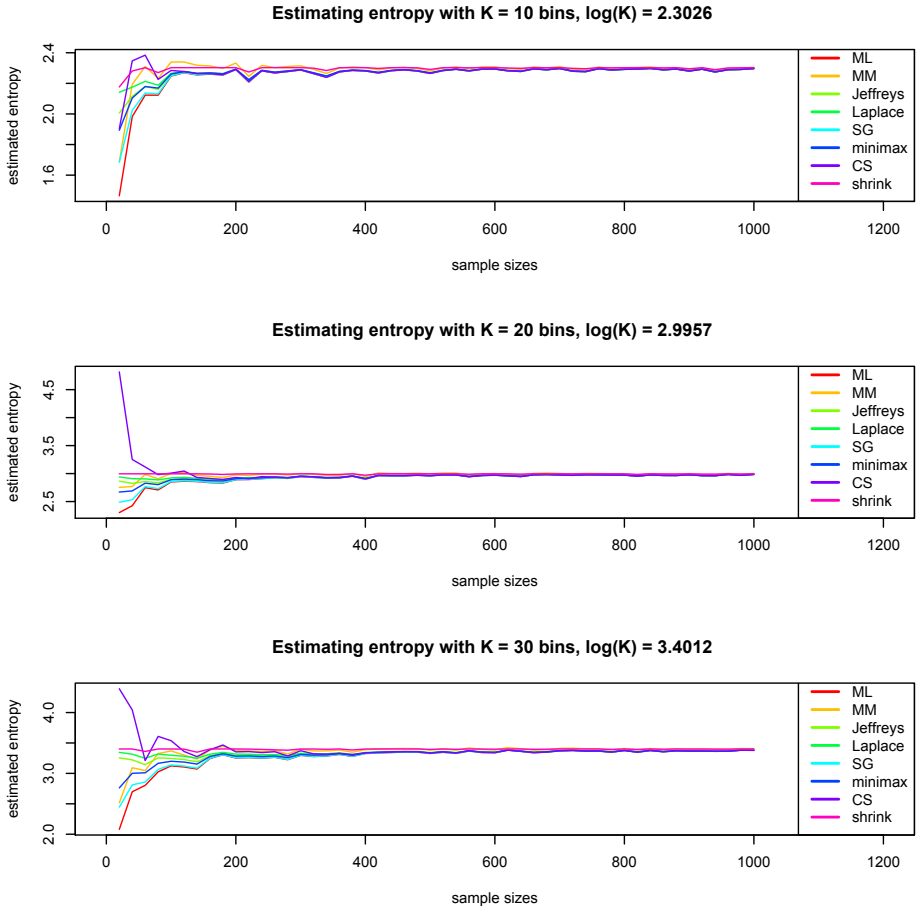

Figure 9: Estimating entropy of normal distribution

Non-parametric - density estimation The discrete method is very unlikely to produce satisfactory result, We consider taking an indirection by estimating the density function and then perform numerical integration.

Direct integration when the function is in $\mathbb{R}$ or $\mathbb{R}^2$ is efficient enough for practical concern. Yet for higher dimensions, the speed is unbearably slow (in many seconds). This is due to the fact that evaluating the kernel density at any point would result in n function value look-ups where n is the sample size. If the numerical integration algorithm (quadrature) requires m function evaluations, the total amount of time is then proportional to $O(mn)$. A more efficient algorithm [9, Morariu_2008_AOTFGS-notes] can be employed instead to bring down the cost to $O(m+n)$.

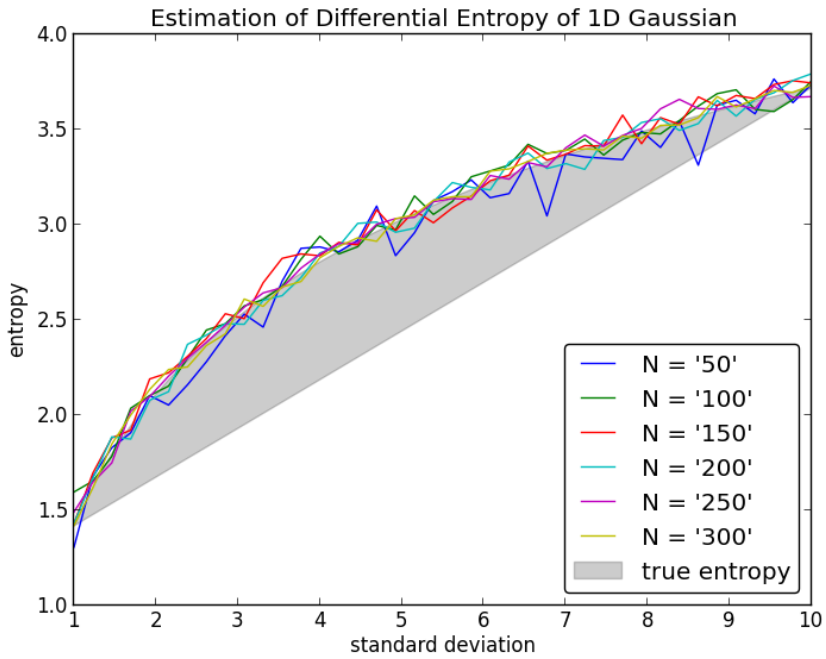

Figure 10: Empirical entropy estimation

As the dimension increases, the numerical integration becomes extremely time consuming. We consider using Markov Chain Monte Carlo [13, Rizzo_2007_SCR-bib] methods instead to approximate the integration. An overview of MCMC. Quasi Monte-Carlo (QMC) [5, Dick_2013_HITQCW-notes] method might also be an interesting way.

Non-parametric - k-Nearest-Neighbor We can use order statistics to derive a more reliable measure of entropy. This method is much less computationally intensive than those using density estimation. We tested the algorithm on a two dimensional normal distribution in figure 11. Each iteration contains an 1000 samples, repeated 30 times. The distribution has a covariance of this form $\left[\begin{smallmatrix} 1 & \rho \\ \rho & 1 \end{smallmatrix}\right]$. It can be seem from the figure that with only a small correlation, the difference in mutual information is not pronounced and thus not easy to detect. This is due to the fact that the mutual information for normal distribution is computed as $-\log(1-\rho^2)/2$.

5.2 Physics of MRI

A magnetic field is enacted. Longitudinal (parallel, Z) and transversal (perpendicular, XY) magnetization of nuclei. A radio frequency pulse is used to align the phase and 'tip over' the nuclei (hydrogen atom). The direction of the spinning nuclei is in a "circular" fashion. When the nuclei are aligned to the magnetic field, their circular frequency is called Larmor frequency.

Relaxation:

- Longitudinal: the restoration of net magnetization along the `Z` direction. The rate is characterized by the exponential distribution with T_1. T_1 is the time that 63% of the max longitudinal magnetization level is reached.
- Transversal: the loss of net magnetization in the `XY` plane due to lost of phase coherence (induced by the RF pulse). The rate is characterized by the exponential

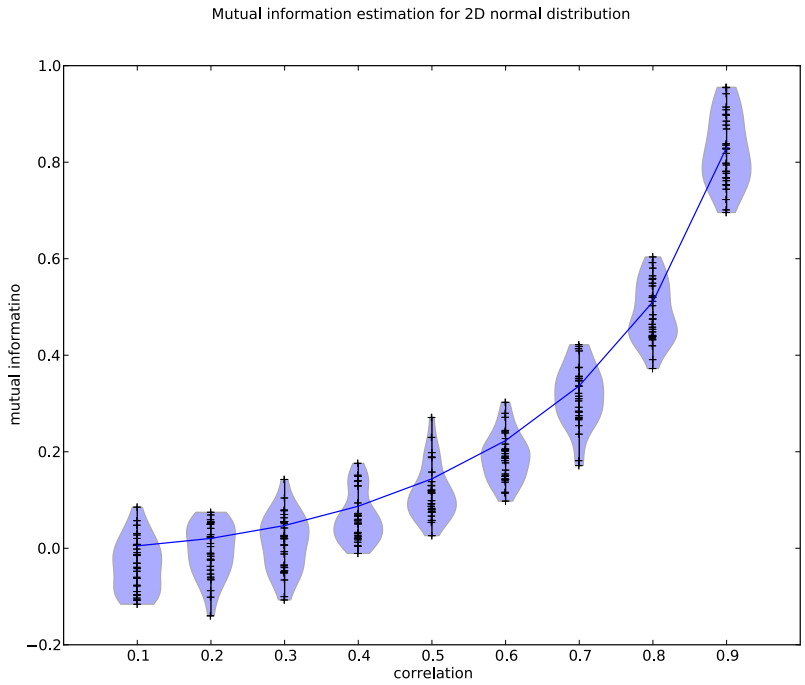

Figure 11: Empirical mutual information estimation with kNN

distribution with T_2. T_2 is the time till we have 37% of the magnetization in the transversal direction. Its values are from 40 to 200 ms depending on the tissue. This is usually ten time smaller than that of T_1.

- T_2* is the combined effect of T_2 and local inhomogeneities in the magnetic field.

Different structures has different relaxation time.

structure	T1	T2	T2*
white matter	600		
gray matter	1000		
CSF	3000		

The imaging technique uses the following parameters

- TR: how often we excite the nuclei.
- TE: how soon after the excitation do we begin to collect data.

Thus the BOLD signal measured is approximately

$$M_0(1 - e^{-TR/T_1})e^{-TE/T_2}.$$

TR \ TE	small	large
small	T1 weighted	garbage
large	protein density	T2 weighted

MRI acquired typically in *axial* slices, in a interlaced manner (odd number slices preceding even number ones). For a reference to the terminologies, check out the coursera video `https://class.coursera.org/fmri-001/lecture/`

6 References

[1] David M. Blei and Michael I. Jordan. "Variational inference for Dirichlet process mixtures". In: *Bayesian Analysis* 1.1 (Mar. 2006), pp. 121–143. DOI: 10.1214/06-BA104.

[2] Randy L Buckner, Fenna M Krienen, and BT Thomas Yeo. "Opportunities and limitations of intrinsic functional connectivity MRI". In: *Nature neuroscience* 16.7 (2013), pp. 832–837.

[3] Ed Bullmore and Olaf Sporns. "The economy of brain network organization". In: *Nature Reviews Neuroscience* 13.5 (2012), pp. 336–349.

[4] Nicolas A. Crossley et al. "Cognitive relevance of the community structure of the human brain functional coactivation network". In: *Proceedings of the National Academy of Sciences* 110.28 (2013), pp. 11583–11588. DOI: 10.1073/pnas.1220826110.

[5] Josef Dick, Frances Y. Kuo, and Ian H. Sloan. "High-dimensional integration: The quasi-Monte Carlo way". In: *Acta Numerica* 22 (May 2013), pp. 133–288. ISSN: 1474-0508. DOI: 10.1017/S0962492913000044.

[6] Michelle Girvan and Mark EJ Newman. "Community structure in social and biological networks". In: *Proceedings of the National Academy of Sciences* 99.12 (2002), pp. 7821–7826.

[7] James Hensman, Magnus Rattray, and Neil D. Lawrence. "Fast Variational Inference in the Conjugate Exponential Family". In: *Advances in Neural Information Processing Systems 25*. Ed. by F. Pereira et al. Curran Associates, Inc., 2012, pp. 2888–2896.

[8] Edward Meeds and Simon Osindero. "An alternative infinite mixture of Gaussian process experts". In: *Advances In Neural Information Processing Systems*. 2006, pp. 883–890.

[9] Vlad I Morariu et al. "Automatic online tuning for fast Gaussian summation." In: *NIPS*. 2008, pp. 1113–1120.

[10] Ilya Nemenman, Fariel Shafee, and William Bialek. "Entropy and inference, revisited". In: *Advances in neural information processing systems* 1 (2002), pp. 471–478.

[11] Carl Edward Rasmussen. *Gaussian processes for machine learning*. Adaptive Computation and Machine Learning. The MIT Press, 2006.

[12] Carl Edward Rasmussen and Zoubin Ghahramani. "Infinite mixtures of Gaussian process experts". In: *Advances in Neural Information Processing Systems 14*. The MIT Press, 2001, pp. 881–888.

[13] Maria L Rizzo. *Statistical Computing with R*. CRC Press, 2007.

[14] James Ross and Jennifer Dy. "Nonparametric Mixture of Gaussian Processes with Constraints". In: *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*. Ed. by Sanjoy Dasgupta and David Mcallester. Vol. 28. 3. JMLR Workshop and Conference Proceedings, May 2013, pp. 1346–1354.