1 Introduction

Functional Magnetic Resonance Imaging (FMRI) is a powerful tool in the analysis of neural activity. Despite its rather limited temporal resolution, FMRI is still the best way of measuring neural activity deep in the brain. Whereas other methods of analyzing neural signals can be invasive or difficult to acquire, FMRI is relatively quick and cheap, and its analysis is straight forward. Because of these benefits, FMRI continues to be crucial to the study of human cognition. Despite the fact that FMRI is so widely used, the choice of analysis tools is still relatively limited. Every widely available analysis tool is based on parametric modeling, which requires prior knowledge of the noise distribution. Indeed, the linear methods usually applied are known not to be robust to non-gaussian, non-white noise, which is why so many pre-processing steps are necessary. These limitations cast a long shadow over any experiment that claims to be able to reject the null hypothesis, no matter the P-value. While obviously there is a correlation between the BOLD signal and actual activity, its possible the current parametric tests are over-estimating the actual amount of activity. Its also possible that the current methods underestimate actual activity. In this paper we propose a robust model-based approach to activation detection that makes no assumptions about the underlying noise model.

fact, typical activation detection does little to account for variations in the shape of the signal, but instead is a direct T-test between the measured signal and the expected shape of the signal. Because of these limitations, multiple comparison tests are imperative (ref fish), and active regions must be very large to be detected with any confidence. This paper will introduce a more robust method of analyzing FMRI signals for activation, by means of estimating the BOLD parameters.

1.1 FMRI

Before going further, it is necessary to discuss Functional MRI, and what it measures. All MRI works by exciting nuclear spin away from a steady state controlled by a large electromagnet. Different nuclei respond to RF impulses based on their natural frequency of oscillation. Therefore contrast in MRI is based on the aggregate nuclear composition of a given region. Thus, regions with more oxygen will respond differently from regions with proportionately less oxygen. This exact difference is what causes the changes over time in the MR signal in functional MRI. Unfortunately, it takes a significant amount of time for nuclear spin to return to equilibrium. Thus instead of waiting, a type of imaging called Echo Planar Imaging (EPI) is used, wherein a single excitation is used, and then as quickly as possible every voxel (3 Dimensional Pixel) in a particular region is read. Because "reading" a particular region requires the switching on of a gradient magnetic field, it takes a decent amount of time to perform measurements. Problematically, the signal from the original excitation drops exponentially, thus the signal to noise ratio can be very low even at modest resolutions. Eventually the signal becomes unreadable, and a new excitation must be applied. As more measurements are made, and thus as the spatial resolution increases, the rate at which entire volumes can be acquired drops. Typically volumes can be acquired every two to three seconds, at resolutions of about 64 by 64 by 32, but their signal to noise ratio is often poor. Other imaging modalities such as EEG and MEG are capable of much faster measurements, but are unable to isolate measurement locations so precisely as FMRI. As with all biological systems, noise is much higher than in human designed systems, and yet these biological systems are capable of performing amazing feats.

1.2 BOLD Physiology

The physiology that leads to the BOLD model has been well studied over the past decade and a half. While some uncertainty remains (for instance in relation to the BOLD post-stimulus undershoot ¡papers relating to back and forth on post-stimulus undershoot¿) generally the model fits reality very well. As with every piece of tissue in the body, the brain requires oxygen to extract energy from glucose in the blood. This process of removing oxygen from red blood cells sets into motion a chain of events that locally alters the composition of the blood. Inactive neurons can be thought of as a slingshot cocked and ready to fire. As soon as a signal moves up the axon and causes the neuron to fire (and thus changes the state of a membrane at that location), ions quickly move across the altered membrane to compensate for a high charge imbalance between the interior and exterior of the cell. This process, similar to allowing a stretched rubber band to contract leaves the system at a lower energy state. This cannot last though, because the neuron needs to be ready to fire again rather quickly. To return the cell to a ready position the membrane becomes impermeable to the ions again, and then begins pumping ions back into the cell. This process takes a large amount of energy, whereas the actual firing takes very little energy. Thus, after firing, glucose is burned, removing oxygen from the blood and thus causing a dip in the ratio of oxygenated hemoglobin to de-oxygenated hemoglobin. This is the first potentially measurable effect that MRI can see, although it usually lasts fewer than 2 seconds making it difficult for FMRI to catch. As a result of the decreased amount of local oxygen, the capillaries compensate by increasing blood flow to that region. Because, of the quickly increased blood flow into local capillaries, the blood volume increases in addition to the local oxygen content. This leads to a Windkessel effect which further lags the normalization of oxygen content. The effect of all this is an overshoot in the oxygen content above the initial level. After the work of recharging is done, there may be a prolonged

undershoot lasting as much as 90 seconds [Yacoub et al.(2006)Yacoub, Ugurbil, and Harel], though the reason for this is debated [Chen and Pike(2009)].

A large number of models have been proposed for the BOLD signal with varying amounts of complexity. The simplest model is the so called "Balloon Model" which first proposed the windkessel effect as a factor the BOLD response. The model we will use is the model proposed by Buxton Et. Al. and later used in Riera et. al. [Riera et al.(2004)Riera, Watanabe, Kazuki, Naoki, Aubert, Ozaki The model has four state variables, s, f, v, q, representing flow inducing signal, cerebral blood flow, cerebral blood volume and deoxyhemaglobin to hemaglobin ratio, respectively. The state variables change over time, given by the state evolution equations:

$$\dot{s} = \epsilon u - \tag{1}$$

$$\dot{f} = s \tag{2}$$

$$\dot{v} = \tag{3}$$

$$\dot{q} = \tag{4}$$

Additional papers have added such things as ....

All these effects are generally accepted as the cause of the BOLD signal, but FMRI doesn't detect this happening in one neuron, but rather as the aggregate over millions of cells. Though local neurons act "together" (i.e. around the same time), the density of neurons, the density of capillaries, and slight differences in activation across a particular voxel can all lead to signal attenuation or noise. A particularly insidious type of noise present in FMRI is a low frequncey drift, characterized by a Weiner process. Though not present in all regions, it is prevalent enough to cause problems [Tanabe et al.(2002)Tanabe, Miller, Tregellas, Freedman, and Meyer]. It is still not clear where exactly this noise comes although it is possible it is the result of magnets heating up, or some distortion in magnetic fields. It is clear that this drift signal is not due to a true physiological effects however, given its presence in cadavers and phantoms[Smith et al.(1999)Smith, Lewis, Ruttimann, Ye, Sinnwell, Yang, Duyn, and Frank].

2 Current Techniques

2.1 Basic Statistical Parametric Mapping

The most basic method of analyzing FMRI data is through a basic T-test between "resting state" and "active state" samples. This is done by taking the average and variance of the inactive period, and the active period separately then treating them both as gaussian distributions. If they are in fact Gaussian distributions, then a basic t-test will give the likelihood that the samples came from the same distribution (the p-value). Of course, this test is fraught with problems; even if the drift mentioned earlier has been removed, there is little reason to believe that the noise is Gaussian, or even stable. Additionally, even if the noise were Gaussian, a t-test with a p-value of .05 over 5000 or more samples is on average going to generate $.05 * 5000$ false positives. To compensate for this, bonferoni correction, also known as multiple comparison tests are performed; essentially p-values are divided by the number of independent tests being run. This, however, leads to extremely low p-values, so low that it would be impossible for any biological system to satisfy. To compensate, a Gaussian kernel is applied to the image, thus reducing variance (and thus separating the active and inactive distributions) as well as decreasing the effective number of voxels. Since t-tests are now no longer being applied to n ¡I need to define n¿ independent voxels, the factor by which the p-value must be divided by can be decreased. ¡Do I need to mathematically define all this?¿ The derivation and application of random field theory, and its use can be found in various papers [Worsley et al.(2004)Worsley, Taylor, Tomaiuolo, and Lerch].

2.2 General Linear Model

The most used form of FMRI analysis is still based on Statistical Parametric Mapping, but is able to account for several different levels or types of stimulus. By adding a General Linear Model to the analysis, the output signal timeseries (what FMRI detects) is regressed over the weighted sum of the various confound's timeseries. The equation for a general linear model is then

$$Y(t) = X(t)\beta + \epsilon(t) \tag{5}$$

where $Y(t)$ is the smoothed or detrended timeseries of measurements, $X(t)$ is a row vector of stimuli, β is a column vector of weights, and ϵ is the error. Thus for every time, the measurement is assumed to be a weighted sum of the inputs plus some error. The calculation of β then is merely a gradient descent search to minimize the mean squared error.

¡Image of GLM¿

Of course, a square wave input is not going to result in a square wave in the activation of brain regions. Thus, various methods are used to smooth $X(t)$ through time, and bandlimit the input. The best technique is convolving the stimulus input with a hemodynamic response function, which mimicks the basic shape of BOLD activation, including a delay due to rise time and fall time. This hemodynamic signal is static however, so every region of the brain gets the same design matrices (X(t)), although the weights of various stimulus or confounds are allowed to vary.

¡Image of Hemodynamic Response Function¿

Ultimately, activation due to a particular stimuli is decided by the β value corresponding to that stimuli's column of $X(t)$. ¡Need to check this¿ The null hypothesis as to whether the outcome was random is then based on a t-test of the $\epsilon(t)$ timeseries.

2.3 Whats wrong with these techniques

There are a few problems with the techniques mentioned in the previous sections. First, they essentially ignore prior knowledge about the system. Although the most advanced form of the general linear model includes a "Hemodynamic Response Function," that hemodynamic response function is static across every region of the brain. It is well known that capillary beds are not uniform and so blood perfusion cannot possible be static across the brain. Thus, if extra information were available a-priori, that information could not be incorporated without modifications to the General Linear Model. Similarly heart rate could not be added either. It would obviously be advantageous to have true physiological parameters as entry points for these various other model parameters. The physiological models for the BOLD signal are quite good and based on realistic physics. While the exact connection between a stimulus and the flow inducing signal is not precisely known, model fits are actually quite good [Deneux and Faugeras(2006)]. Regardless, being based on some real physiological parameter would allow for the establishment of reasonable priors and decrease model variance without breaking a sweat. Of course, using real parameters has the additional bonus of potentially providing information about physical pathologies. It is quite possible that physical properties such as decreased compliance of blood vessels could indicate a neurological condition that is not easily seen in a T1 or T2 map. In essence, this could make FMRI a much more useful clinical tool than it is now. The other problem with linear models is that they are a linear fit to a nonlinear signal. It is not uncommon for data to be thrown out in FMRI studies because no significant activation has been seen. However, if, for whatever reason, the BOLD response was acting more nonlinear than in other patients it would be completely possible for SPM to miss that activation.

¡Image with two different αs¿

¡image comparing the results of 10% changes in various signals¿

Secondly, these methods are still based on t-tests, which notoriously lack robustness to non-Gaussian noise. While different techniques exist for imposing Gaussianity, those techniques are incapable of discriminating noise from signal. There is no way to know how much signal is removed by various smoothing techniques, or even if entire regions have been smoothed into oblivion. Instead of extensively filtering data to remove noise, the analysis method itself must be robust a wide range of noise, which is why we propose here the use of particle filters.

3 Proposed Approach

3.1 Goal

The ultimate goal of this project is to provide a new set of tools for analyzing FMRI data. Whereas SPM techniques have been highly successful at finding macroscopic regions of activation, linear modeling can carry significant bias error due to lack of model flexibility. While adding parameters can significantly increase error due to model variance, this effect is mitigated by the fact that we plan to use a model that is based on first principals. The purpose of this paper is thus to evaluate the potential of using a particle filter along with the BOLD model to derive physical parameters. In so doing, we hope to be able to show that neuronal efficacy, ϵ is a suitable variable for estimating voxel activation from a standard FMRI image. We also hope to show that estimated posterior distribution of the parameters, derived from the particle filter, is able to provide an accurate measure of the confidence interval.

3.2 Introduction to Particle Filters

Particle filters, a type of Sequential Monte Carlo (SMC) methods are a powerful way of estimating the posterior probability distribution of a set of parameters give a timeseries of measurements. Unlike Markov Chain Monte Carlo estimation, Sequential Monte-Carlo methods are designed to be used with parameters that vary with time. Unlike variations of the Kalman filter,

particle filters do not make the assumption that noise is Gaussian. Thus particle filters are often the best solution to bayesian tracking for non-linear, non-gaussian systems.

3.2.1 Model

The idea of the particle filter is to start with a wide mixture PDF of possible parameter sets, and then, as measurements come in, to weight more heavily parameter sets that tend to give good estimations of the measurements. The reliance on an initial mixture PDF can introduce bias; however, this effect can be minimized by alterring the initial weights in the mixture pdf. Of course every gradient descent must choose starting points and it is often quite easy to establish a reasonable range of parameters, especially when the model being used has a physical meaning. Suppose a set or stream of measurements are given, $\{y(t), t = 1, 2, 3, ...T\}$, where T is permitted to go to infinity. Then the goal is to find the parameters, $\hat{\theta}$, and underlying state time series, $\hat{x}[0:T]$ that minimize the difference between $\hat{y}[0:T]$ and $y[0:T]$. In our case, we will assume that we know the form of the model, which is based on first principals, and that there is some true θ and a true time-series of underlying state variable, $x[0:T]$ that drives $y[0:T]$. Assuming a model form such as we do here reduces model variance, potentially at the cost of increased bias (or systematic) error. We will assume a basic state space model:

$$\dot{x}(t) = f(t, x(t), u(t), \theta, \nu_x) \tag{6}$$

$$y(t) = g(t, x(t), u(t), \theta, \nu_y) \tag{7}$$

Where $x(t)$ is a vector of state variables, θ is a vector of system constants, $u(t)$ is a stimulus, $y(t)$ an observation, and ν_x and ν_y are random variates. Obviously any one of these could be a vector, so for instance $u(t)$ could encode multiple types of stimuli.

Although not generally necessary for particle filters, we will make a few assumptions based on the particular type of systems faced in biological processes. First, the systems are assumed to be time invariant. This assumption is based on the idea that if you froze the system for Δt seconds, when unfrozen the system would continue as if nothing happend. Few biological systems are predictible enough for them to be summarized by a time varying function. Although the heart may seem like an obvious exception, period between heartbeats vary often enough that prediction would necessate another state-space model. In short, we assume no parameters are time varying, because not enough information exists to describe any of theme in that way. Luckily particle filters are capable of dealing with non-white, non-Gaussian noise, so unanticipated influence may be re-factored as noise. Secondly we assume that input cannot directly influence the output, which in the case of the BOLD signal is a good assumption. Third, we will assume noise is additive, and that ν_x may be projected into a weiner, or other summing process that is additive with g and ν_y, which will be named ν_d. Finally, $x(t)$ will encapsulate θ, the unknown model constants, which means that the vector $\dot{x}$ will always have members that are 0. The results of these assumptions are a simplified version of the state space equations:

$$\dot{x}(t) = f(x(t), u(t)) \tag{8}$$

$$y(t) = g(x(t)) + \nu_y + \nu_d \tag{9}$$

Because ν_d is something akin to and additive Weiner process $y[0:T]$, it will include low frequency noise. ν_y on the other hand will cause i.i.d. noise in $y[0:T]$. For some of the tests, I will use de-trending methods to reduce the effects of ν_d, the remainder of which will be re-factored into ν_y. Both ν_d and ν_y have biological and non-biological sources. MR can lead to both types of noise, as demonstrated in [Smith et al.(1999)Smith, Lewis, Ruttimann, Ye, Sinnwell, Yang, Duyn, and Frank]. Meanwhile changes in metabolism, heart rate, or other biochemical intervention could all lead to either ν_d or ν_y.

3.2.2 Prior

The goal of the particle filter is to evolve a probably distribution $Pr(\hat{x}(T)|u[0:T], y[0:T])$, that asymptotically approaches the probability distribution $Pr(x(T)|u[0:T])$. Considering that y contains measurement noise and noise in x can drive changes in y, it is clear that $Pr(x(t)|u[0:T])$ is not a single true value but a true posterior. To begin with, the particle filter starts with a proposal distribution, and N_p particles need to be drawn from that distribution, $\alpha(x)$:

$$\{\hat{Pr}x_i(0), w_i] : x_i(0) \sim \alpha(x), w_i = \frac{1}{N_p}, i \in \{1, 2, ..., N_p\}\} \tag{10}$$

Where N_p is the number of particles or points used to describe the prior using a Mixture PDF.

$$\hat{Pr}(x(0) = \hat{x}) = \sum_{i=1}^{N_p} w_i \delta(\hat{x} - x_i(0)) dx \tag{11}$$

Where $\delta(x - x_0)$ is 1 if and only if $x = x_0$ (the Kronecker delta function).

If a true prior is preferred, then the weights should all be $1/N_p$, and since x_i was drawn from the prior, this will be an approximation of the prior distribution. If a relatively flat prior is preferred, then each particle's weight could be divided by the density, $\alpha(x_i)$, which creates a flat prior with support points in the region of $\alpha(x)$. Either way, $\alpha(x)$ should be much broader than the true posterior, $Pr(x(0))$, since the choice of support points is crucial to the convergence of any sampling importance algorithm. For the BOLD signal all the parameters have been studied and have relatively well known mean and variance, so a prior could be very helpful. We ran simulations for both normalized and un-normalized priors, although we believe in cases such as this, where a good prior exists, it should be used. For strictly positive parameters (members of x) we used a gamma distribution, whereas for parameters that could be negative, we used a Gaussian distribution. In both cases standard deviations twice that found in previous studies were used.

Note that all the probabilities implicitly depend on $u[0:T]$, so those terms will be left off for simplicity. Once the probability, $\hat{Pr}(x(T)|x[0:T-1], y[0:T-1])$ has been found (initially this is just Mixture approximating the prior since no measurements are available and no previous probabilities are available), its possible to approximate the probability for short times between times when measurement is available, by shifting the probability according the progression of the state equations. This is only an approximate, since integrating ν_d should increase uncertainty as time without a measurement passes.

$$\hat{Pr}(x(T + \Delta t)) \approx \sum_{i=1}^{N_p} w_i \delta \left(x - (x_i(T) + \int_T^{T+\Delta} \dot{x}_i(t) dt) \right) \tag{12}$$

3.2.3 Weighting

When a measurement becomes available it is incorporated into the probability. This process of incorporating new data is called sequential importance sampling, and eventually causes the probability to converge. The weight is defined as

$$w_i(T) \propto \frac{\hat{Pr}(x_i[0:T]|y[0:T])}{q(x_i[0:T]|y[0:T])} \tag{13}$$

where q is called an *importance density*, meaning it decides where the support points for $x(T)$ are located. To remove the bias due to the location of the support points, we divide by $q(x_i[0:T]|y[0:T])$. By dividing by the posterior density of the support points (particles), the effect of the particle distribution may be removed from the posterior density. As a result the weight is dependent solely based on $\hat{Pr}(x_i[0:T]|y[0:T])$, the probability of the i^{th} particle's measurements being different from $y[0:T]$ due to noise alone. An example of an importance density would be drawing a large number of points from the standard normal, $N(0,1)$ and then weighting each point, l by $1/\beta(l), \beta \sim N(0,1)$. Of course if there is a far off peak in the posterior that q does not allocate support points in, there will be a quantization error, and that part of the density can't be modeled. This is why it is absolutely necessary that q covers $\hat{Pr}(x_i[0:T]|y[0:T])$.

$q(x_i[0:T]|y[0:T])$ may be simplified by assuming that $y(T)$ doesn't contain any information about $x(T-1)$, which is more practical since knowledge of future measurements is impractical.

$$\begin{aligned} q(x[0:T]|y[0:T]) &= q(x(T)|x[0:T-1], y[0:T]) q(x[0:T-1]|y[0:T]) \\ &= q(x(T)|x[0:T-1], y[0:T]) q(x[0:T-1]|y[0:T-1]) \\ &= q(x(T)|x(T-1), y[0:T]) q(x[0:T-1]|y[0:T-1]) \end{aligned} \tag{14}$$

In this paper we will use $q(x_i(T)|x_i(T-1), y[0:T]) = \hat{Pr}(x_i(T)|x_i(T-1))$, based on the Markov assumption, and the belief that the state space model is able to approximate the true state. This means that prior to re-weighting particles, the particles will be distributed the same as the previous time but moved forward according to the integration of $f(x(t), u(t))$.

In addition to $q(x_i(T)|x_i[0:T-1], y[0:T])$, the weight is also based on $Pr(x_i[0:K]|y[0:K])$, which may be broken up as follows.

$$\begin{aligned}
\hat{Pr}(x[0:T]|y[0:T]) &= \frac{\hat{Pr}(y[0:T], x[0:T])}{\hat{Pr}(y[0:T])} \\
&= \frac{\hat{Pr}(y(T), x[0:T]|y[0:T-1])\cancel{\hat{Pr}(y[0:T-1])}}{\hat{Pr}(y(T)|y[0:T-1])\cancel{\hat{Pr}(y[0:T-1])}} \\
&= \frac{\hat{Pr}(y(T)|x[0:T], y[0:T-1])\hat{Pr}(x[0:T]|y[0:T-1])}{\hat{Pr}(y(T)|y[0:T-1])} \\
&= \frac{\hat{Pr}(y(T)|x[0:T], y[0:T-1])\hat{Pr}(x(T)|x[0:T-1], y[0:T-1])\hat{Pr}(x[0:T-1]|y[0:T-1])}{\hat{Pr}(y(T)|y[0:T-1])}
\end{aligned} \tag{15}$$

Using the assumption that $y(t)$ is fully constrained by $x(t)$ (25), and that $x(t)$ is fully constrained by $x(t-1)$ (8), we are able to make the reasonably good assumptions that:

$$\hat{Pr}(y(T)|x[0:T], y[0:T-1]) = \hat{Pr}(y(T)|x(T)) \tag{16}$$

$$\hat{Pr}(x(T)|x[0:T], y[0:T-1]) = \hat{Pr}(x(T)|x(T-1)) \tag{17}$$

Additionally, for the particle filter $y(T)$ and $y[0:T-1]$ are given, and therefore constant across all particles. Thus $\hat{Pr}(x[0:T]|y[0:T])$ may be simplified to:

$$\begin{aligned}
\hat{Pr}(x[0:T]|y[0:T]) &= \frac{\hat{Pr}(y(T)|x[0:T], y[0:T-1])\hat{Pr}(x(T)|x[0:T-1], y[0:T-1])\hat{Pr}(x[0:T-1]|y[0:T-1])}{\hat{Pr}(y(T)|y[0:T-1])} \\
&= \frac{\hat{Pr}(y(T)|x(T))\hat{Pr}(x(T)|x(T-1))\hat{Pr}(x[0:T-1]|y[0:T-1])}{\hat{Pr}(y(T)|y[0:T-1])} \\
&\propto \hat{Pr}(y(T)|x(T))\hat{Pr}(x(T)|x(T-1))\hat{Pr}(x[0:T-1]|y[0:T-1])
\end{aligned} \tag{18}$$

Plugging these simplifications into (13) leads to:

$$\begin{aligned}
w_i(T) &\propto \frac{\hat{Pr}(y(T)|x(T))\cancel{\hat{Pr}(x(T)|x(T-1))}\hat{Pr}(x[0:T-1]|y[0:T-1])}{\cancel{\hat{Pr}(x_i(T)|x_i(T-1))}q(x[0:T-1]|y[0:T-1])} \\
&\propto w_i(T-1)\hat{Pr}(y(T)|x(T))
\end{aligned} \tag{19}$$

Thus, by making the following relatively weak assumptions, evolving a posterior density is easy and requires almost no knowledge of noise distribution.

1. $f(t, x(t), u(t)) = f(x(t), u(t))$ and $g(t, x(t), u(t)) = g(x(t))$ provide a sufficiently flexible model to encapsulate the true time series.
2. $E[\nu_d] = 0$ and $E[\nu_y] = 0$, and $\nu_x = d\nu_d$, ν_y are stationary
3. The PDF $q(x_i(0))$ (the prior) fully covers $Pr(x_i(0))$
4. Markov Assumption: $Pr(x(T)|x[0:T]) = Pr(x(T)|x(T-1))$
5. $q(x[0:T-1]|y[0:T]) = q(x[0:T-1]|y[0:T-1])$

3.2.4 Basic Particle Filter Algorithm

From the definition of w_i, the algorithm sequential importance sampling (SIS) is relatively simple.

Initialize N_p Particles: $\{x_i(0), w_i(0) : x_i(0) \sim \alpha(x), w_i(0) = \frac{1}{N_p}, i \in \{1, 2, ..., N_p\}\}$
$T = \{\text{Set of Measurement Times}\}$
for t in T **do**
  for i in N_p **do**
    $x_i(t) = x_i(t-1) + \int_{t-1}^{t} f(x(\tau), u(\tau))d\tau$
    $w_i(t) = w_i(t-1)\hat{Pr}(y(t)|x(t))$
  end for
end for
At $t + \Delta t$, $t \in T$, $\hat{Pr}(x(t + \Delta t)) \approx \sum_{i=1}^{N_p} w_i(t)\delta\left(x - (x_i(t) + \int_t^{t+\Delta t} f(x(\tau), u(\tau))d\tau)\right)$

The result is then a discrete approximation of the posterior distribution.

3.2.5 Resampling

As a consequence of the wide prior distribution (required for a proper discretization of a continuous distribution), there will be many particles with insignificant weights. While this does help describe the tails of the distribution very well, it means that only a small portion of the computation will be spent describing the most probable region. Ideally every particle would equally decrease the entropy of the distribution, thus the lower the variance of the weights, the more efficiently the discrete distribution is in describing the continuous distribution. A common measure of "Particle Degeneracy" is the effective number of particles, described in (Bergman "Navigation and Tracking Applications", 1999, J S Liu and R Chen "Sequential Monte Carlo Methods for Dynamical Systems", 1998), which is based on the "true weight" of each particle. Of course the true weight is unknown, so a heuristic approximating N_{eff} is used:

$$\hat{N}_{eff} \approx \frac{N_p}{\sum_{i=1}^{N_p} w_i^2} \tag{20}$$

Any quick run of a particle filter will reveal that unless the prior is particularly accurate, N_{eff} drops precipitously. To alleviate this problem a common technique known as resampling must be applied. The idea of re-sampling is to draw from the approximate posterior, thus generating a replica of the posterior with a support more suited to the distribution. Thus, if weights are all set to $1/N_p$, and N_p points are drawn from the posterior,

$$\hat{\chi}_j \sim \left(\sum_{i=1}^{N_p} w_i(t)\delta(x - x_i(t))\right), j \in \{1, ..., N_p\} \tag{21}$$

then $\hat{\chi} \sim \hat{x}$ should hold. Unfortunately, this isn't necessarily the truth: since the support is still limited to the original particles, the number of unique particles can only go down. This effect, often dubbed "particle impoverishment" can result in excessive quantization errors in the final distribution. However, there is a solution. Instead of sampling from the discrete distribution, a smoothing kernel is applied, and $\hat{\chi}_j$ are drawn from that distribution. Because the distribution is continuous, there is no way for a collapse of the particles to occur. The question then, is how to decide on the smoothing kernel. Often times the easiest way to sample from the continous distribution is to break the re-sampling down into two steps. First a member of the discrete distribution is randomly selected based on the weights, and then based on the smoothing a nearby state variable is selected. The process of the selection will be defined as:

$$\chi_i = x_i + h\sigma\epsilon \tag{22}$$

Where h is the bandwidth, σ is the standard deviation such that $\sigma\sigma^T = cov(x)$ and ϵ is drawn from the chosen kernel. It has been proven that when all the elements of the mixture have the same weight, as is the case after basic resampling, the kernel that minimizes the MSE between the estimated and true posterior is the Epanechnikov Kernel (cite Improving Regularised Particle Filters, C Musso, N Oudjane and F LeGrand).

$$K = \begin{cases} \frac{n_x+2}{2c_{n_x}}(1 - \|x\|^2) & if\ \|x\| < 1 \\ 0 & otherwise \end{cases} \tag{23}$$

If the noise is assumed to be Gaussian then it is possible to further optimize. Thus we let h be defines as:

$$h = [N_s 8c_{n_x}^{-1}(n_x + 4)(2\sqrt{\pi})^{n_x}]^{\frac{1}{n_x+4}} \tag{24}$$

and although it is very possible the underlying noise is non-gaussian, the Gaussian may work, but sub-optimally. It has been proposed that (Monte Carlo Approximations for General State-Space Models, markus Hurzeler and Hans R. Kunsch) if the underlying distribution is non-Gaussian, then using this bandwidth will oversmooth. In reality over smoothing should not be too great an issue because the smoothing is only being applied to find new particles. If the distribution is over smoothed then the algorithm may not converge as rapidly; however, because the bandwidth is still based on particle variance, which will decay as particles are ruled out, it is still able to converge. In fact over smoothing is preferrable to under smoothing, since the latter would result in false negatives, but the previous only results in a slower decay of the variance. At the same time, as n_x, the number of dimensions in x, goes to infinity, the standard deviation based approximation becomes less effective (cite a Tutorial on Particle Filters for on-line non-linear non-gaussian bayesian tracking, sanjeev arulampalam, simon maskell, neil gordon...). Because of the high dimensionality of our system, and limited measurements, it is helpful to have a broader bandwidth to explore the distribution. Nevertheless, because of the potentially wide smoothing factor applied by regularized resampling, performing this step at every measurement would allow particles a great deal of mobility. This mobility is the enemy of convergence, which is why regularized resampling should only be done when $\hat{N}_{eff}$ drops very low (say less than 50). Other than the periodic regularized resampling then, the regularized particle filter is nearly identical to the basic sampling importance sampling filter (SIS).

Initialize N_p Particles: $\{x_i(0), w_i(0) : x_i(0) \sim \alpha(x), w_i(0) = \frac{1}{N_p}, i \in \{1, 2, ..., N_p\}\}$
$T = \{\text{Set of Measurement Times}\}$
for t in T **do**
  for i in N_p **do**
    $x_i(t) = x_i(t-1) + \int_{t-1}^{t} f(x(\tau), u(\tau))d\tau$
    $w_i(t) = w_i(t-1)\hat{Pr}(y(t)|x(t))$
  end for
  Calculate N_{eff} with (20)
  if $N_{eff} < N_R$ (recommend $N_R = min(50, .1N_p)$) **then**
    Calculate empirical σ
    $h = [N_s 8 c_{n_x}^{-1}(n_x + 4)(2\sqrt{\pi})^{n_x}]^{\frac{1}{n_x+4}}$
    Redraw particles using (stratified) basic resampling
    for i in N_p **do**
      Draw $\epsilon \sim K$
      $x_i = x_i + h\sigma\epsilon$
    end for
  end if
end for
At $t + \Delta t$, $t \in T$, $\hat{Pr}(x(t + \Delta t)) \approx \sum_{i=1}^{N_p} w_i(t)\delta\left(x - (x_i(t) + \int_t^{t+\Delta t} f(x(\tau), u(\tau))d\tau)\right)$

The ultimate effect of this regularized resampling is a convergence similar to simulated annealing or a genetic algorithm. Versions of x that are "fit" (give good measurements) spawn more children nearby which allow for more accurate estimation near points of high likelihood. As the variance of the estimated x's decrease, the radius in which children are spawned also decreases. Eventually the radius will approach the width of the underlying uncertainty, ν_x and ν_y.

3.3 Choosing $\hat{Pr}(y(T)|x(T))$

Choosing a representation of an unknown distribution is certainly tricky, and so the fact that $\hat{Pr}(y(T)|x(T)) = \nu_d + \nu_y$ means that there is a significant piece of the algorithm that is based primarily conjecture. Studies of the noise in FMRI typically attribute noise to a Gaussian random variable or an additive noise process with Gaussian steps.

3.3.1 Classical De-trending

The non-stationary aspect of a Weiner process as with ν_d is difficult to compensate for, and so various methods have been developed to compensate for it. [Tanabe et al.(2002)Tanabe, Miller, Tregellas, Freedman, and Meyer] and [Smith et al.(1999)Smith, Lewis, F have demonstrated that this component is prevalent, and may in fact be a characteristic of FMRI. In some studies, as many as half the voxels benefit from detrending, meaning that this is certainly a serious barrier to inference. All the existing methods are performed during the preprocessing stage, rather than as an integral part of analyzing the BOLD signal. There is no shortage of theories on the "best" method of detrending, however a head to head comparison, [Tanabe et al.(2002)Tanabe, Miller, Tregellas, Freedman, a

showed that in most cases subtracting off a spline works the best. The benefit of the spline versus wavelets, high pass filtering or other DC removal techniques is that the frequency response is not set. A spline is able to move quickly when the signal is moving quickly, and move more slowly when the signal moves more slowly. That said, the spline will still remove some amount of signal, just like all of these methods.

¡image of de-spline'd lines with "true" lines¿

3.3.2 Delta Based Inference

I also propose and test a different method of dealing with the so called "drift". Instead of comparing the direct output of the particle filter with the direct measurement, the algorith compares the change in signal over a single TR. In most signal processing cases this would foolish, but that is because the general assumption that all noise is high frequency is not the case here. In fact, every pipeline for the analysis of BOLD signal uses a high pass filter, but low poss filters are rarely applied, because it is a well known fact that most of the signal is in the high frequency range and most of the noise is actually in the low frequency range. The particle filter is an extremely robust method of inference, and so I would assert that the particle filter ought to be given as *raw* data as possible. While taking direct measurements without de-trending would give awful results, using the difference removes the DC component and turns a Weiner process into a Gaussian random variable.

$$\Delta y = y(t) - y(t-1) = g(x(t)) - g(x(t-1)) + \nu_y(t) - \nu_y(t-1) + \nu_d(t) - \nu(t-1) \tag{25}$$

Because ν_d is a Weiner process, then $\nu_d(t) - \nu_d(t-1)$ is simply a Gaussian step. If ν_d is some other additive process, the difference will still be one of a few stable distributions. If ν_y is i.i.d. then the resulting distribution will still be zero mean with a maximum variance of twice the original variance. All the assumptions made originally for the particle filter hold, and all of the parameters may be distringuished based on the step sizes, thus it is not unreasonable to attempt to match the string of step sizes rather than string of direct readings.

¡frequncy response graphs, highlighting noise frequency range and signal frequency range¿

3.3.3 Weighting Function

Because $\hat{Pr}(y(T)|x(T))$, what I will call the weighting funciton, is based on an unknown distribution, it is necessary to decide on a function that will approximate $\hat{Pr}(y(t)|x(T))$. Obviously the function, $\omega(y(t), f(x(t))$ needs to centered at zero and have a scale comparable to the signal levels. While a Gaussian function is the natural choice, we also wanted to try a distribution with wider tails, so that outliers don't completely destroy particle's weights. Therefore, we tried three weighting functions; Gaussian, exponential and the Cauchy distribution. The standard deviations (or scale) was set to $\sigma_y/5$, where σ_y is the variance of all $y[0:\infty]$. Of course since the particle filter requires a weighting function to run, this means that before the particle filter starts all the measurements have to be in. In cases where this is impossible, a heuristic based on a small sample may work just as well.

The shape of the weighting function is extremely important, because it essentially decides the rejection rate of particles. A very thin gaussian probability distribution function has nice properties, but thin tails. As a result, large outliers in the measurement vector could easily force all the particles to have near 0 weights, thus forcing the particle filter to converge improperly. On the other end of the spectrum, a cauchy PDF, has relatively fat tails, and may not weight central particles high enough, preventing the particles from converging at a reasonable rate. Exponetials have the benefit of having an extremely smooth drop to zero, and a slope of 1 at the origin. Having a non-zero slope at the origin is beneficial because it discriminates all the way up until the measured and predicted y are the same. The importance of the weighting function cannot be overstated, as this is the primary factor in deciding the rate at which the particle filter converges.

Obviously the optimal $\hat{Pr}(y(T)|x(T))$ is the true $Pr(y(T)|x(T))$; however since that is unknown, I tested multiple different distributions. As stated previous, because the exact scale of $Pr(y(T)|x(T))$ it not even known, the exponential distribution has the added benefit of having a negative slope all the way form zeo to infinity. Thus even if the scale is completely wrong, particles will still be well differentiated.

¡Q-Q plot of real fmri data with Gaussian, DC¿ ¡Q-Q plot of real fmri data with Gaussian, deltas¿

4 Methods

This paper describes two types of experiments; first we will cover simulations which have the benefit of a ground truth, then we will cover the methods used in the use of the particle filter on real data.

4.1 Preprocessing

As discussed in the section on de-trending, the normal pipeline for analyzing FMRI involves a great deal of preprocessing. In this paper we make an effort to minimize any type of preprocessing that will degrade the signal. After FMRI data has been acquired it is always necessary to modify the data in some way to make different runs comparable. Because FMRI signal levels are unit-less, at the very least it is necessary to convert the data into % difference from the baseline. This process removes no data from signal since it merely subtracting then dividing by a constant. This is the signal that was input into the delta based particle filter. Of course there are much more advanced ways of performing this task. The generally accepted standard is actually to use a high pass filter, although the cutoff frequency is application dependent and often applied haphazardly. The high pass filter thus removes the DC component of the signal, and some amount of the so called "drift". The problem with this method is that it is not adaptive to the input. Huge variations in drift frequencies can exist in a single time-series. Thus, a single cutoff frequency could miss a significant drift component, or it could remove *actual* signal, if the cutoff frequency is set too high. This is why, as I mentioned in the De-trending section, a spline based detrending method will generally give better results.

For simulated and real images (tests with multiple time-series), tests were also run with and without Gaussian filtering with sigma of were run, since it is standard practice to apply a Gaussian spatial filter to the images at each timestep. Obviously a spatial filter such as Gaussian filtering increased SNR but can also lead to less precision in the output maps.

4.2 Simulation

We performed two types of simulations. First, we simulated a single BOLD time-series based on a random chosen set of model parameters. This process was relatively straight forward given the state-space equations for the BOLD signal. After a "true" signal was generated, we then added a carrier level, since BOLD is typically measured as a % difference from the base level. Finally, we added Gaussian noise, and a Weiner Process to the clean signal. The variance of the Gaussian noise may be expressed in terms of the desired noise SNR, R as:

$$var(y_{noisy}) = var(y)/R \tag{26}$$

Since SNR doesn't have quite the same meaning for a Wiener process based noise, the variance of the Gaussian steps was set to be:

$$var(y_{noisy}) = var(y)/(4R) \tag{27}$$

Once this noisy simulated time series was generated, the exact same particle filter algorithm that would later be run on full sized images, was run on this single voxel image. We ran a series of tests to determine the convergence rate of the particle filter, the number of particles that were required, how weighting functions compared, how different de-trending methods compared with each other and, finally the variance of the result. By running the exact time-series with different noise realizations, it was possible to determine the model variance. As the reader may know, the error of an estimator may be calculated as:

$$MSE(\Theta) = Var(\Theta) + Bias(\Theta)^2 \tag{28}$$

The variance is an expression of how much the result would change for different noise realizations, whereas the bias is an expression of how well the model matches the true underlying model. In this case, because the same model is being used in the particle filter and underlying simulation, the bias is actually zero. Obviously when this is calculated using *real* data with an unknown underlying state space equation, there will be some amount of bias error, but assuming that the noise is similar to the noise used in these tests, the model variance will actually be about the same. Thus calculating the model variance is extremely helpful in calculating how well determined our model is, and how consistent it will be for real data. A single timeseries, as opposed to the thousands present in a real image, makes it easier to compare the output with the ground truth, with various parameters.

Second we used a modified version of the FSL tool POSSUM to generate an entire FMRI image from a parameter map. The parameter map was generated by creating a random image, smoothing it with a large Gaussian kernel, then thresholding the results. Finally connected regions were each given a set of parameters from a finite list of randomly chosen parameter sets. The result was a four dimensional (length x width x height x parameter) image with spatially varying parameters. Time-series of activation level was generated for each set of parameters, then activation levels were fed into POSSUM's function for generating frequency domain data. The patche for POSSUM will be made available. For each time-series in the simulate FMRI image, the final *static* parameters are saved into a parameter map. This parameter map may then be compared to the map used to generate the simulated data; additionally a new simulation using the calculated parameters may also be generated to test the functional difference between the two maps. This would give an absolute quantitative difference between the two parameter sets irrespective to parameter slopiness. So for instance, if V_0 is halved, ϵ doubling may very well give a similar result. In this case

the % difference between the parameters will be large in each case, but the functional difference between the parameters will not be great. This is obviously a bad situation, which is why we wanted to test for it.

4.3 Real Data

Finally, we also performed inference based on real FMRI data. The scanner we used...

The final result from calculating parameters with the real data was similar to that from the results from the POSSUM simulated data. The difference being that there was no ground truth the check it with.

5 Results

5.1 Single Time-Series Simulation

Graphs:

For {delta, DC/Spline}, {exponential, gaussian, cauchy}, {biased, unbiased initial}, {100, 500, 1000} particles

1. Comparison with a linear system with similar number of degrees of freedom
2. Ground truth vs. Estimated signal during particle filter run
3. Ground truth vs. Estimated signal with final parameter set
4. True Parameters vs. Final Parameter Sets
5. Variance of final parameters when faced with same ground truth, different noise
6. Variance of final parameters when faced with same ground truth, different noise
7. MSE of (a new timeseries based on X(t) vs. ground truth) for all t
8. Estimator Variance based on different noise runs
9. Final Particle Distribution

5.2 Simulated Volume

1. Parameter Map
2. Error map of parameters
3. Histogram of %errors between parameters
4. Activation Map based on a single region with high ϵ, compared with linear

5.3 FMRI Data

....

image comparing epsilon-map with GLM activation map

6 Conclusion

References

[Chen and Pike(2009)] J. J. Chen and G. B. Pike. Origins of the BOLD post-stimulus undershoot. *NEUROIMAGE*, 46(3): 559–568, JUL 1 2009. ISSN 1053-8119. doi: 10.1016/j.neuroimage.2009.03.015.

[Deneux and Faugeras(2006)] T. Deneux and O. Faugeras. Using nonlinear models in fmri data analysis: Model selection and activation detection. *NeuroImage*, 32(4):1669 – 1689, 2006. ISSN 1053-8119. doi: DOI: 10.1016/j.neuroimage.2006.03.006. URL `http://www.sciencedirect.com/science/article/B6WNP-4KDBM3X-1/2/8a437832d609f25f1d5c969c5e6b7414`.

[Riera et al.(2004)Riera, Watanabe, Kazuki, Naoki, Aubert, Ozaki, and Kawashima] J. Riera, J. Watanabe, T. Kazuki, M. Naoki, E. Aubert, T. Ozaki, and R. Kawashima. A state-space model of the hemodynamic approach: nonlinear filtering of BOLD signals. *NEUROIMAGE*, 21(2):547–567, FEB 2004. ISSN 1053-8119. doi: 10.1016/j.neuroimaging.2003.09.052.

[Smith et al.(1999)Smith, Lewis, Ruttimann, Ye, Sinnwell, Yang, Duyn, and Frank] A. Smith, B. Lewis, U. Ruttimann, F. Ye, T. Sinnwell, Y. Yang, J. Duyn, and J. Frank. Investigation of low frequency drift in fMRI signal. *NEUROIMAGE*, 9(5): 526–533, MAY 1999. ISSN 1053-8119.

[Tanabe et al.(2002)Tanabe, Miller, Tregellas, Freedman, and Meyer] J. Tanabe, D. Miller, J. Tregellas, R. Freedman, and F. Meyer. Comparison of detrending methods for optimal fMRI preprocessing. *NEUROIMAGE*, 15(4):902–907, APR 2002. ISSN 1053-8119. doi: 10.1006/nimg.2002.1053.

[Worsley et al.(2004)Worsley, Taylor, Tomaiuolo, and Lerch] K. J. Worsley, J. E. Taylor, F. Tomaiuolo, and J. Lerch. Unified univariate and multivariate random field theory. *NeuroImage*, 23(Supplement 1):S189 – S195, 2004. ISSN 1053-8119. doi: DOI: 10.1016/j.neuroimage.2004.07.026. URL `http://www.sciencedirect.com/science/article/B6WNP-4D9D8DJ-5/2/52aca4e3b540819e6d11f04f124272e0`. Mathematics in Brain Imaging.

[Yacoub et al.(2006)Yacoub, Ugurbil, and Harel] E. Yacoub, K. Ugurbil, and N. Harel. The spatial dependence of the post-stimulus undershoot as revealed by high-resolution BOLD- and CBV-weighted fMRI. *JOURNAL OF CEREBRAL BLOOD FLOW AND METABOLISM*, 26(5):634–644, MAY 2006. ISSN 0271-678X. doi: 10.1038/sj.jcbfm.9600239.