MA916 - MASDOC RSG

Research Study Group

Brain Imaging

Phase I - Problem Formulation

Don Praveen Amarasinghe, Andrew Lam, Pravin Madhavan

February 13, 2011

Contents

1 Introduction

Brain imaging is an important tool in the diagnosis and treatment of neurological ailments. Neuroscientists employ scanners to build a picture of the "activity" within a patient's brain. These images can then be used to determine the functions of various regions of the brain, or detect the presence of abnormalities (such as cancerous cells). In this section, we would like to highlight the workings of functional magnetic resonance imaging (fMRI) and electro-encephalography (EEG) and discuss their limitations.

1.1 fMRI

fMRI detects activation in brain regions using the different magnetic properties of protons in oxygenated and deoxygenated haemoglobin, to analyse blood flow and oxgenation levels in regions of the brain. The MRI machine emits an energy pulse inside a magentic field. Protons in Hameoglobin molecules will then abosrb energy corresponding to a particular frequency, which will change its spin and orientation in the external magnetic field – In deoxygenated haemoglobin, protons will want to align themselves to an external magnetic field, wherease in oxygenated haemoglobin, protons are more likely to go against the field. A short time afterwards, the pulse is switched off and these protons will then emit a photon signal corresponding to the energy absorbed earlier. These signals are measured simultaneously and their spatial coordinates are sorted based on their frequency. Volume elements on the image, called *voxels*, are then highlighted corresponding to the intensity of the photons detected at that point, which is itself linked to the levels of oxygenated and deoxygenated blood present there. A voxel that is comparatively brighter would indicate more oxygenated blood in that region of the brain, indicating comparatively more activity in that region [**1**].

fMRI experiments work by first taking a control scan, where the patient is at rest, and then taking another scan while the patient responds to a particular stimulus or stimuli. For example, the patient may be asked to press a button when they see a particular image on a screen inside the scanner. The scans are then subtracted from one another, in order to determine the regions of the brain most prominent in responding to the stimulus within the test scan.

1.2 EEG

EEG measures brain activity through the monitoring of neural electrical signals generated by brain cells whilst the patient is performing tasks in response to a stimulus. Numerous electrodes are placed on the scalp of the patient, each of which detects neural activity through the gain or loss of electrons. This electron flow is caused by the movement of ions along the axons of nerve cells in the brain, corresponding to brain activity. Voltages between electrodes can then be used to determine the areas with most neural activity [**2**].

1.3 Limitations

There are several limitations which need to be addresed.

- fMRI measures only the changes in magnetic field, indirectly measuring levels of oxygenated /deoxygenated blood – no direct detection of brain activity is made [**1**]. EEG is comparatively better in this respect as it is designed to detect neural activity directly.

- EEG lacks spatial resolution – that is, there is no clear method to precisely pinpoint the location of neural activity. Even with a setup of 64 – 128 electrodes on the patient's scalp, the technicians cannot infer the location of a signal's source.

- In fMRI, it is hard for the subject to stay perfectly still during the scan and a control experiment is difficult to implement as the brain is never completely at rest [**3**].

- Detector noise always features in fMRI and EEG scans. The latter also suffers from bias, caused by the muscles in the head region have a higher amplitude than the EEG signal [**4**]. In extreme cases the EEG signal might be lost in the noise generated by other electrical activity from other parts of the body [**2**].

- There is an implicit delay of several seconds with signals detected using fMRI due to coupling with blood flow. This can result in a spatial spread of millimeters [**3**].

2 Research proposal

Given the technical difficulties with fMRI and EEG, we propose to look at a sequence of brain images taken in time to trace the brain activity linked to a response to some fixed stimulus: the first objective would be to filter out the noise from the image taken at the first time-point. Once this is done, on the assumption that the activity traces a path in the brain over time, the denoised data can be used to evolve the observed signal in time and thus determine the path. We propose that these steps can be implemented using mathematical and statistical tools.

2.1 Model

The original data is composed of noisy surfaces defined on the square domain $[-1, 1] \times [-1, 1]$. The toy model considers a 2D function with rotational symmetry, given by

$$\phi(x, y) = \exp(-\beta((x - c_1)^2 + (y - c_2)^2)) \tag{2.1}$$

where β is some strictly positive parameter. A plot of this function for $\beta = 20$ and $\mathbf{c} = (0, 0)$ is shown in Figure 1, which has been generated using MATLAB.

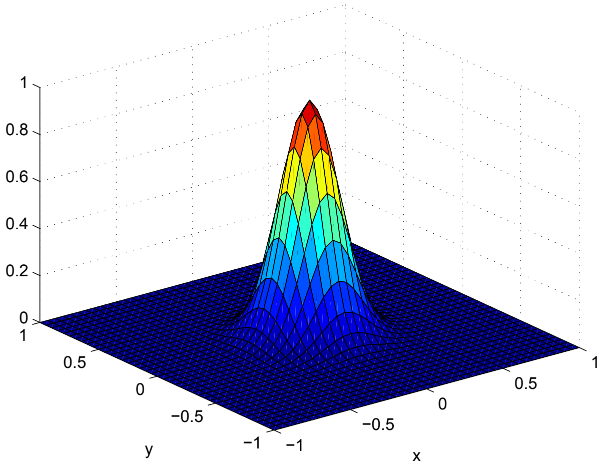

Figure 1: Plot of ϕ for $\beta = 20$ and $\mathbf{c} = (0, 0)$.

We now add noise to this function by drawing independent samples from a normal distribution with mean 0 and variance small enough that the noise isn't of the same order as the signal itself, and adding it to (2.1). Figure 2 shows the plot of this noisy function for the same parameters as before.

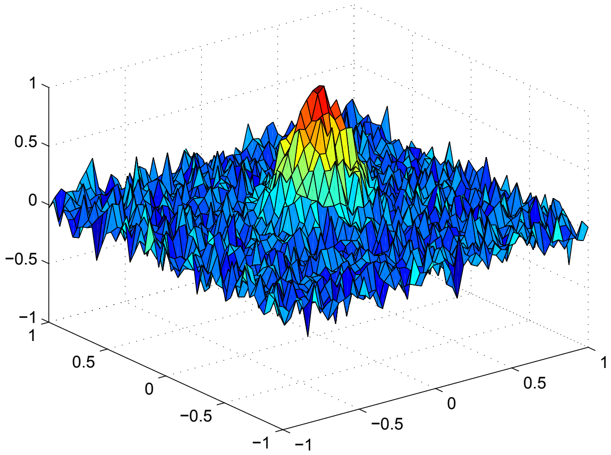

Figure 2: Plot of noisy signal for $\beta = 20$ and $\mathbf{c} = (0, 0)$.

We improve our model further by making the parameters β and $\mathbf{c}$ of (2.1) noisy. The parameter β regulates the signal in a non-trivial way: clearly the value of β that we chose in the previous plot ($\beta = 20$) makes the signal stand out from the noise, but choosing much larger values of β would result in the signal being indistinguishable from the noise.

Now we choose β to be a stochastic process taking two distinct values: one, with low probability, which drowns the signal in the noise for a short period of time and the other, with high probability, in which the signal can be, *a priori*, distinguished from the noise. Such a binary process incorporates some key limitations of medical scanners discussed in the introduction, namely the disappearance of the signal for a short period of time. Furthermore, we choose $\mathbf{c} = (c_1, c_2)$ to follow a path of the form

$$c_2 = c_1^3 + u \tag{2.2}$$

where c_1 moves from -1 to 1 and $u \sim \text{Unif}([-0.1, 0.1])$. The reason for choosing such an expression is to try and capture the non-linear structure of brain processes. Typically, regions of the brain activated by a stimulus need not lie on a path with simple geometry. Whilst equation (2.2) does not fully reflect the complexity of such structures, it still provides a model that captures some of the non-linearity.

A more realistic model for the signal evolution would be to consider the motion of the signal as a random walk. That is, treating **c** as a pair of Markov random variables, as we believe that the location of the signal at a future time would only depend on its current location.

2.2 Proposed Method of Noise Removal

One of the issues with data from fMRI is the noise that is inherent within the images obtained. Whilst there are many methods outlined in the literature about ways to filter out this noise, we would like to focus on a method based upon Bayesian methodology - that of approximate Bayes factors. The aim is to apply an image segmentation technique to reduce the number of colours (or shades of grey) used in the image, where a Bayes factor approximation called *PLIC* will determine this number. The result should be an image with cleanly defined boundaries, making it easier to distinguish between various structures.

The idea behind Bayes factors is that, unlike in frequentist hypothesis testing, you can compare the weight of evidence for a hypothesis or model as well as that against it [**5**]. The factors are calculated as ratios between the conditional probability distributions of the observed data, **D**, i.e.

$$B_{i,j} = \frac{\mathbb{P}(\mathbf{D}|H_i)}{\mathbb{P}(\mathbf{D}|H_j)} \tag{2.3}$$

is the Bayes factor comparing H_i and H_j, where

$$\mathbb{P}(\mathbf{D}|H_k) = \int \mathbb{P}(\mathbf{D}|\theta_k, H_k)\pi(\theta_k|H_k)d\theta_k \tag{2.4}$$

with θ_k the parameter under H_k with prior distribution $\pi(\theta_k|H_k)$. With a few exceptions, this integral cannot be calculated explicitly. So we need to appeal to approximations to Bayes factors instead. In the case of [**6**], they use a crtierion called the Penalised Pseudolikelihood Criterion as the approximation – more details of how this criterion is derived are given in the appendices.

In [**6**], an example using a PET (Positron Emission Tomography) scan of a dog's lung is used as the test image. The hypotheses being tested are models corresponding to the number of shades of grey (i.e. the greyscale) to use in the final image. Some preliminary image segmentation techniques (such as the Iterated Conditional Modes algorithm to establish initial parameter estimates) need to be implemented before the PLIC can be calculated. Only models with between 2 and 6 image segments (colours or grey shades) are considered, as we only want a few colours to be used. The model with the highest PLIC value is chosen, and the final image developed using the selected number of segments.

A word of warning: This concept of "removing noise" might appear to be somewhat of a miracle! It is important to remember that there is always some form of stochastic error inherent in this process – we aren't regenerating the true data without the noise; we are only finding an *estimate* of the true data. Naturally, the variation in the result depends upon the model being used and the data being analysed. An interesting problem would be to characterise this variation mathematically, but this shall be left as a potential extension of the problem being set.

2.3 Proposed Method of Signal Tracking

We have already mentioned the problems of low spatial resolution in the data obtained in EEG scans, and the sensitivity of EEG scanners to electrical activity generated by other sources, such as external magnetic fields. [**7**] and [**8**] describe methods of applying Kalman filters to reduce the effects of these problems, and we will briefly sketch their ideas below.

In [**8**], Morbidi et al. propose a Kalman filter approach to remove artifacts on an EEG trace. Specifically, they propose applying a Kalman filter to a linear system arising from two models – one for the EEG signal and one for artifact generation. The end result is the complete removal of the induced artifacts in the EEG recording, whilst preserving the integrity of the EEG signal.

While [**8**] uses the Kalman filter to remove artifacts on the EEG trace, [**7**] uses the Kalman filter to enhance existing data spikes for epileptiform activity in the EEG. Epileptic seizures are characterised by high amplitude synchronised periodic waveforms that reflect abnormal discharge of a large group of neurons [**7**]. On the EEG data, these seizures come in the form of isolated spikes, sharp waves and spike-and-wave complexes that are distingushable from background activity. [**7**] aims to enhance any signal that vaguely resembles a spike, and a Kalman filter is applied to cancel the background activity and noise from the EEG signal.

We would also like to highlight a paper that utilise Kalman filters to detect activation regions for fMRI data. In [**9**], the authors attempt to fit a general linear model on the fMRI data using an extended Kalman filter [1]. We will sketch the main idea of the paper below.

Let $\mathbf{y} = [\mathbf{y_1}, \ldots, \mathbf{y_n}]^\mathbf{T}$ denote the time course vector associated to a particular voxel in the fMRI image sequence, where $1, \ldots, n$ denote the acquisition times. The general linear model is

$$y = \mathbf{X}\beta + \epsilon, \tag{2.5}$$

where the design matrix $\mathbf{X} = (\mathbf{x_1^T}, \ldots, \mathbf{x_p^T})$ is a $(n \times p)$ matrix with the regressors $x_1, \ldots, x_p$ as columns. β is the unknown $(p \times 1)$ vector of regression coefficients and ϵ denotes the noise, assumed to be a autoregressive Gaussian process

$$\epsilon_t = a\epsilon_{t-1} + n_t, \tag{2.6}$$

where a is the autocorrelation parameter and n_t is white noise.

Let $\mathbf{b} = (\beta, a)$ be the augmented state vector and S_t be the posterior variance-covariance matrix of $\mathbf{b}$ given the information available at time t. The Extended Kalman filter performs successive estimates $(\hat{\mathbf{b}}, \mathbf{\Sigma_t})$ on the pair $(\mathbf{b}, \mathbf{S_t})$ at each time step. To detect whether an effect is present, the brain's responses to different stimuli are compared and estimates for the variance, as well as the statistical significance of the effect, are calculated using t-tests.

Let c denote a contrast vector of parameters that make up the desired effect. For example, to test activation in the first condition versus a baseline, the contrast vector is

$$c = (1, 0, 0, \ldots 0). \tag{2.7}$$

To compare the difference between the first and second conditions, the contrast vector is

$$c = (1, -1, 0, \ldots 0). \tag{2.8}$$

The effect is now defined via

$$c\beta = \beta_1 - \beta_2 \tag{2.9}$$

[1] See appendix for details on the Kalman Filter and Extended Kalman Filter.

and the estimate of the variance of the effect is $\mathrm{Var}(c\beta)$. Next, the following T-statistic is used to test whether or not there is any evidence for the effect being studied.

$$t = \frac{\beta_1 - \beta_2}{\sqrt{Var(c\beta)}}. \tag{2.10}$$

The approach in [**9**] is similar – for a given contrast vector c, the main aim is to identify the voxels that demonstrate a constrast effect $c\beta$. The authors considered the Mahalanobis transform [2] of the contrasted effect

$$T_i = (c\Sigma_i c^T)^{-1/2} c\beta, \tag{2.11}$$

and the probability that the contrasted effect is positive can be derived from there.

2.4 Evaluation metrics

One of the key assumptions to apply the Kalman filter is that the noise is Gaussian. However, this may not necessarily be the case. One important thing to know is how this would affect the outcome of your analysis. For example, if the real data is generated with noise that has a non-standard distribution, and we apply the Kalman filter as if the noise was Gaussian, would our analysis reflect closely what is going on? A simpler case would be when the noise takes a standard distribution which is not Gaussian (such as a Chi-squared distribution). Again, we would need to consider the validity in applying our chosen techniques under this assumption.

In order to compare results derived from different models, we need a metric to evaluate this difference. One possibility would be to consider the matrix norm of the differences between the corresponding matrices derived from different models for the error. This would allow us to know how sensitive the chosen techniques are to various error distributions.

Such an evaluation metric is easily computable, but does not take into account the inherent stochasticity of the denoised data matrices. An evaluation metric that does so compared the resulting signal paths rather than the data matrices. More precisely, we generate a number of signal trajectory paths at every timepoint and take their average. This is done for each model we choose for the error, and so the corresponding averaged paths can then be compared *statistically*. Various statistical metrics that compare such paths can be found in [**10**].

[2] The Mahalanobis transformation is used to decorrelate two random variables.

2.5 Extensions

2.5.1 Multimodal data

The signal surface used in the toy model is a unimodal distribution that evolves along a simple path. A more realistic model would be to have a multimodal distribution for the signal surface – for example, cases where more than one area of the brain is active in a response to a stimulus. Such a model mirrors the situation where the true activation signal is hidden among a number of false-positives (generated either by noise or by temporal delay). However the complexity of the problem has increased, the signal surface has become more rough and the path of the signal is harder to track.

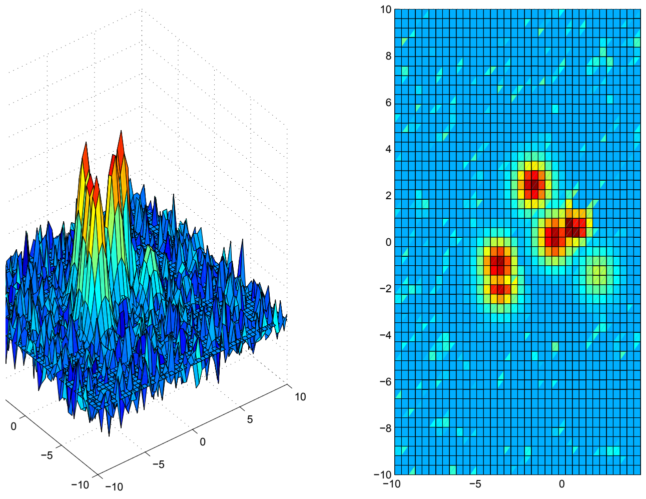

Figure 3: An example of several false positives surrounding the true signal

A starting point to tackle this extension is to differentiate between the “true” signal and false-positives that arise from the detection process. These surfaces bare some similarities to random fields, hinting at the use of random field theory to analyse these images [**11**]. In random field theory, one assigns a threshold to these surfaces and looks at the number of clusters that are left after thresholding. From this, hypothesis testing may aid in locating the true activation region.

2.5.2 Delayed detection region

We propose an extension problem involving delayed detection. For example two brain regions might simultaneously activate in response to a stimulus, but due to temporal bias, the detector might pick up one signal first and the other signal might appear later in time. One issue would be to set a threshold so that the delayed detected region does not get rejected as a false positive. This setting is similar to the multimodal extension mentioned above, and so random field theory could also be of use to solve this problem.

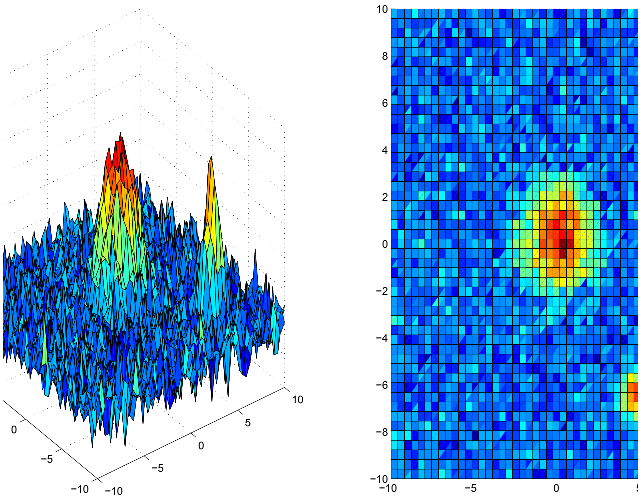

Figure 4: An example of the delayed appearance of a smaller second region

2.6 Action Plan

- Generate the noisy data using the MATLAB code provided in the Appendix. Perform multiple runs and experiment with different parameter values to get a feel for how this toy model behaves. [**1 week**]

- Read up on the various mathematical and statistical tools discussed in the report, namely approximate Bayes factors and Kalman filters, as well as any other tools which you believe would be useful to remove noise and/or track the signal(s) in a sequence of time indexed images generated by the aforementioned code. Make use of the various resources mentioned in the bibliography. [**3 weeks**]

- Implement your chosen techniques and apply these on dummy data to test them out. Once this is done, the implementation should be adapted to work on the noisy data generated in the first step. [**4 weeks**]

- Analyse the resulting data to estimate the path of the signal, and compare this with the actual data before noise was added to it, at each time-step, to determine how successful your choice of techniques was. You may also wish to consider applying different noise distributions to your data, and applying evaluation metrics to assess the sensitivity of your techniques to these models. [**3 weeks**]

- If you have time, consider applying the work you have done to the extension problems.

A Appendix - Model Implementation Details

A.1 MATLAB Code for Data Creation

```
%%%%%%%%%%%%%%%%%%%%%%Creates the noisy surface.%%%%%%%%%%%%%%%%%%%%%%%%%%
gridRes = 50; % Grid resolution.
timeRes = 20; % Time resolution.

x = linspace(-1,1,gridRes);
beta = zeros(1,timeRes);
y=x;
[X,Y] = meshgrid(x,y);

% Random walk for centre c.
c1 = linspace(-1,1,timeRes);
c2 = c1.^3 + unifrnd(-.1,.1,1,timeRes);

% Binary process for beta.
betaSwap = unifrnd(0,1,timeRes);
for i = 1:timeRes
    if (betaSwap(i) < 0.1)
        beta(i) = 10000;
    else
        beta(i) = 20;
    end
end

for i = 1:timeRes
matrix = exp(-beta(i)*((X-c1(i)).^2 + (Y-c2(i)).^2)) + 0.1*randn(size(X));
surf(X,Y,matrix);
axis([-1 1 -1 1 -1 1])
% Uncomment the two lines below to see signal in action!
%drawnow
%pause(0.1)
end

%%%%%%Outputs values of matrix at last timestep in file data1.txt.%%%%%%%%%
file = fopen('data1.txt','w');
for i=1:gridRes
    for j=1:gridRes
fprintf(file,'%4.8f\t',matrix(i,j));
```

```
    end
    fprintf(file,'\n');
end
fclose(file);
```

A.2 Noise Removal via Approximate Bayes Factors

A.2.1 Exact & Approximate Bayes Factors

In statistical inference, there are two major methodologies for analysis of data - the frequentist approach and the Bayesian approach. The frequentist (or classical) approach takes the parameter, θ say, to be established as a fixed value, and the data that we gain from samples, $\mathbf{D}$ say, allows us to make guesses of the value of θ. This is typically done in a hypothesis testing framework. Suppose we have a null hypothesis,

$$H_0 : \theta \in \Theta_0 \subset \Theta$$

which we want to test against an alternative hypothesis,

$$H_1 : \theta \in \Theta \backslash \Theta_0$$

where Θ is the parameter space. The usual method of hypothesis testing involves a Likelihood Ratio Test Statistic, given by

$$S_{LR}(\mathbf{D}) = \frac{sup_{\Theta_0} L(\theta; \mathbf{D})}{sup_{\Theta} L(\theta; \mathbf{D})} \tag{A.1}$$

We choose a significance level, or *size*, α for the hypothesis test, and determine a suitable critical region, C, for which $\mathbb{P}(S_{LR}(\mathbf{D}) \in C) = \alpha$.

The main problem with this approach is that, whilst it will update our knowledge of θ when we get enough evidence to reject our null hypothesis, if we are able to provide more data samples, then each hypothesis test ignores any information that could have been gained from previous data sets tested. In a similar vein, we would like to pool the data we have and determine the strength of the evidence which supports the null hypothesis as well as that against it [**5**]. Furthermore, if we have more than one hypothesis for the value of θ, then this approach will be very difficult to implement. This is where the Bayesian approach comes into play.

In the Bayesian framework, we think of our parameter, θ as being a random variable and our data $\mathbf{D}$ as being fixed, and we update our knowledge of the probability distribution of θ via a likelihood based upon the $\mathbf{D}$. When we are testing hypotheses, rather than looking at parameters, we look at the hypotheses themselves and the probabilities of each hypothesis being true given the data at hand – i.e. $\mathbb{P}(H_k|\mathbf{D})$ for $k = 0, 1$ in the simple case. Now observe that, by Bayes' theorem

$$\mathbb{P}(H_k|\mathbf{D}) = \frac{\mathbb{P}(\mathbf{D}|H_k)\mathbb{P}(H_k)}{\mathbb{P}(\mathbf{D}|H_0)\mathbb{P}(H_0) + \mathbb{P}(\mathbf{D}|H_1)\mathbb{P}(H_1)} \tag{A.2}$$

with $k = 0, 1$. We then get

$$\frac{\mathbb{P}(H_0|\mathbf{D})}{\mathbb{P}(H_1|\mathbf{D})} = \frac{\mathbb{P}(\mathbf{D}|H_0)}{\mathbb{P}(\mathbf{D}|H_1)}\frac{\mathbb{P}(H_0)}{\mathbb{P}(H_1)} \tag{A.3}$$

where

$$\mathbb{P}(\mathbf{D}|H_k) = \int \mathbb{P}(\mathbf{D}|\theta_k, H_k)\pi(\theta_k|H_k)d\theta_k \tag{A.4}$$

with θ_k the parameter under H_k with prior distribution $\pi(\theta_k|H_k)$. We define

$$B_{0,1} = \frac{\mathbb{P}(\mathbf{D}|H_0)}{\mathbb{P}(\mathbf{D}|H_1)} \tag{A.5}$$

to be the *Bayes factor* comparing H_0 and H_1. The general case of the Bayes factor comparing H_i and H_j is defined in the same way.

The major problem in trying to calculate Bayes factors is calculating the probabilities in the ratio using integral (A.4). There are some simple cases where this integral is tractable, but more often, other methods need to be sought. Details of various methods that could be used can be found in [**5**, §4], including MCMC algorithms, such as a Metropolis-Hastings sampler to simulate from the posterior, Monte Carlo methods, including importance sampling, and asymptotic methods. We focus on the last of these, looking at the such as Bayes Information Criterion which is based upon Schwarz criterion [**5**, §4.1.3]. Of the methods described in the paper, the Schwarz criterion method is depicted as being the easiest to apply. Indeed, in [**6**], this method is used to clear up noise from a PET scan of a dog's lung.

One of the issues with integral (A.4) is having to include prior densities

$\pi(\theta_k|H_k)$. The Schwarz criterion, described in (A.6) below, provides a way out of this problem.

$$S = \log \mathbb{P}(\mathbf{D}|\hat{\theta}_0, H_0) - \log \mathbb{P}(\mathbf{D}|\hat{\theta}_1, H_1) - \frac{1}{2}(d_1 - d_2)\log\,(n) \tag{A.6}$$

We have taken n to be the sample size of the data set $\mathbf{D}$, d_k as the dimension of parameter θ_k (so, if $\theta_k = (\mu_k, \sigma_k^2)$, then $d_k = 2$), and $\hat{\theta}_k$ is the maximum likelihood estimator under H_k. It can be shown that as $n \to \infty$

$$\frac{S - \log B_{0,1}}{\log B_{0,1}} \to 0 \tag{A.7}$$

even though the error of $\exp(S)$ as an approximation to $B_{0,1}$ is of order $O(1)$. So whilst it might not be a particularly accurate approximation, for large samples, it will give a good idea as to the relative strength of the hypotheses based on the data given. Typically, a variant of the Schwarz criterion with similar levels of performance, called the *Bayes Information Criterion* or *BIC*, given below in (A.8), is used instead.

$$BIC = 2\,\log \mathbb{P}(\mathbf{D}|\hat{\theta}_k, H_k) - d_k\log\,(n) \tag{A.8}$$

In fact, as described in [**5**] and [**6**], $BIC \approx 2\,\log \mathbb{P}(\mathbf{D}|H_k)$.

A.2.2 Image Segmentation & Bayes Factor Approximations

*The notes given here are a summary of the approach adopted in [**6**, §3], where it used to clean up a PET scan of a dog's lungs. There is also mention of the method in [**12**, §5]*

Take any position, i, on the image. Suppose that the pixel can take an integer value, Y_i, between 1 and K, representing the shade of grey displayed at the point. Suppose also that the true state at position i is represented by X_i and this determines the distribution of Y_i. We look at the whole image as being a hidden Markov random field, and apply the Potts model given in [**6**, §2.1]. This gives rise to a conditional distribution, given by

$$\mathbb{P}(X_i = m|N(X_i), \phi) = \frac{\exp(\phi U(N(X_i), m))}{\sum_k \exp(\phi U(N(X_k), m))} \tag{A.9}$$

where

- $N(X_i)$ = Set of pixels neighbouring pixel i

- $U(N(x_i), k)$ = Number of neighbouring pixels taking value k. Note, if k is replaced with X_i, this becomes the number of neighbouring pixels which take the same value as X_i does.
- ϕ represents how closely related values at neighbouring pixels are. This is a fallout of the Potts model, given by

  $$\mathbb{P}(X) \propto \exp\left(\phi \sum_{i\sim j} \mathbb{I}_{X_i,X_j}\right). \tag{A.10}$$

  Thus, if $\phi = 0$, then the X_i are independent of one another, whereas postive/negative values would mean that pixels would have similar/dissimilar values of X_i to their neighbours.

Let M_K represent the model which uses K different shades of grey – we will treat these models as our hypotheses. This model will have parameter set, θ_K, consisting of parameters for the conditional distribution of Y_i given X_i. We could try to use our Bayes Information Criterion method here to determine which model (i.e. number of colours) is appropriate. However, to calculate the *BIC*, we would need to calculate the $\mathbb{P}(\mathbf{D}|\hat{\theta}_k, H_k)$ term – here, this is equivalent to calculating $\mathbb{P}(\mathbf{Y}|M_K)$, the likelihood, given by

$$\mathbb{P}(\mathbf{Y}|M_K) = \sum_{\mathbf{x}} \mathbb{P}(Y|\mathbf{X} = \mathbf{x}, M_K)\mathbb{P}(\mathbf{X} = \mathbf{x}|M_K). \tag{A.11}$$

This requires a summation over a huge number of different values, representing the different combination of values over the pixels. The sheer size of this calculation means that a new approach is required. In [**6**, §3.3], the authors propose a new criterion to approximate Bayes factors between two models, called the *Penalised Pseudolikelihood Criterion* of *PLIC*.

In order to implement the *PLIC*, we first need to mention an algorithm called *Iterated Conditional Modes* (ICM), described in [**13**, §2.5], which is used in many image reconstruction techniques. This algorithm takes an initial estimate for X_i, which could be generated, as explained in [**12**], using an *Expectation-Maximisation* (EM) algorithm, described in [**14**], or otherwise. It then updates this value using the details of the image (specifically the image excluding the pixel being studied) under the Markov random field model. Specifically, the update maximises $\mathbb{P}(x_i|\mathbf{Y}, \mathbf{X} \setminus \{x_i\})$. Let the estimate generated by this algorithm be denoted $\hat{x_i}$, and the corresponding collection of all these estimates $\hat{\mathbf{X}}$. For notational simplicity, we define $\mathbf{X}_{-i} := \mathbf{X} \setminus \{x_i\}$. The algorithm also provides estimates of the parameters for the distribution of $\mathbb{P}(Y_i|X_i)$ and ϕ.

The idea of *PLIC* is to avoid calculating the likelihood by proposing an alternative pseudo-likelihood. This pseudo-likelihood is based upon configurations which are close to the estimates generated by the ICM, so we are now summing over a much smaller number of values. The single pixel likelihood is given by

$$\mathbb{P}(Y_i|\hat{X}_{-i}, K) = \sum_{j=1}^{K} \mathbb{P}(Y_i|\mathbf{X}_i = j)\mathbb{P}(\mathbf{X}_i = j|N(\hat{\mathbf{X}}_i)) \tag{A.12}$$

where the first term on the RHS is just the conditional distribution of Y_i given X_i, and the second term is caluclated using equation (A.9). Combining these together gives an image pseudo-likelihood

$$\mathbb{P}_{\hat{X}}(\mathbf{Y}|K) = \prod_i \sum_{j=1}^{K} \mathbb{P}(Y_i|X_i = j)\mathbb{P}(X_i = j|N(\hat{X}_i), \hat{\phi}) \tag{A.13}$$

where $\hat{\phi}$ is the estimate of ϕ from the ICM algorithm. We can now derive the *PLIC* by replacing the $\mathbb{P}(\mathbf{Y}|M_K)$ term in the *BIC* with $\mathbb{P}_{\hat{\mathbf{X}}}(\mathbf{Y}|K)$ to get

$$PLIC(K) = 2 \log \mathbb{P}_{\hat{\mathbf{X}}}(\mathbf{Y}|K) - d_k\log{(n)} \tag{A.14}$$

Now, rather than calculating the *PLIC* for all values of K, we can start at $K = 1$, and keep working until we hit a local maximum. The idea is that we want to have a relatively low value for K but that it should be the best small value of K.

There is an alternative criterion to the *PLIC*, called the *Marginal Mixture Information Criterion* (*MMIC*) which is much faster to compute than the PLIC. However, this is seen as being a heuristic guideline rather than a good indicator of the number of grey scales we would like to use. This is because of the likelihood defined in the expression being dependent upon the independence of the Y_i, which doesn't occur typically in images. Further details on this can be found in [**6**, §3.3].

A.3 Signal Tracking

A.3.1 State space models

When we measure any sort of signal, it will typically be contaminated by noise. So the actual observation is given by

$$observation = signal + noise. \tag{A.15}$$

We assume that the signal is a linear combination of a set of variables, called *state variables*, which give rise to a state vector at time t. Denote the observation at time t by X_t and the state vector (of size $m \times 1$) at time t by θ_t. Then (A.15) becomes

$$X_t = h_t^T \theta_t + n_t \tag{A.16}$$

where the $(m \times 1)$ column vector h_t is assumed to be a known vector of constants, and n_t denotes the observation error. The state vector θ_t cannot usually be observed directly. Thus we will typically want to use to observations on X_t to make inferences about θ_t.

Although θ_t may not be directly observable, it is often assumed that we know how it changes through time - this behaviour is given by the updating equation (A.17)

$$\theta_t = G_t \theta_{t-1} + w_t \tag{A.17}$$

where the $(m \times m)$ matrix G_t is assumed to be known, and w_t denotes a $(m \times 1)$ vector of errors. The errors in both (A.16) & (A.17) are generally assumed to be uncorrelated with each other at all time periods. We may further assume that n_t is $N(0, \sigma_n^2)$ while w_t is a multivariate normal with zero mean vector and a known variance-covariance matrix Q.

A.3.2 Kalman filter

We want to estimate the signal in the presence of noise – in other words, we want to estimate the $(m \times 1)$ state vector θ_t, which usually cannot be observed directly. The Kalman filter provides a general method for doing this. It consists of a set of equations that allow us to update the estimate of θ_t when a new observation becomes available.

Suppose we have observed a univariate time series up to time $(t-1)$, and that $\hat{\theta}_{t-1}$ is the 'best' estimator for θ_{t-1} based on information up to this time – here, 'best' is defined as the minimum mean square error estimator. Furthermore, suppose that we can evaluate the $(m \times n)$ variance-covariance matrix of $\hat{\theta}_{t-1}$, which we denote by P_{t-1}.

The first stage – the prediction stage – forecasts θ_t using data up to time $(t-1)$. We denote the resulting estimator by $\hat{\theta}_{t|t-1}$. In this setting, consider the equation (A.17) again. Since we do not know w_t at time $t-1$, our first guess at an estimator for θ_t would be

$$\hat{\theta}_{t|t-1} = G_t \hat{\theta}_{t-1} \tag{A.18}$$

with variance-covariance matrix

$$P_{t|t-1} = G_t P_{t-1} G_t^T + Q. \tag{A.19}$$

When the new observation, X_t, at time t has been observed, the estimator for θ_t can be modified to take into account this extra information. At time $(t-1)$, the best forecast of X_t is given by $h_t^T \hat{\theta}_{t|t-1}$ so that the prediction error is given by

$$e_t = X_t - h_t^T \hat{\theta}_{t|t-1}. \tag{A.20}$$

This quantity can be used to update the estimate of θ_t and of its variance-covariance matrix. It can be shown (see [**15**] and [**16**]) that the best way to do this is by means of the following equations:

$$\begin{aligned} \hat{\theta}_t &= \hat{\theta}_{t|t-1} + K_t e_t \\ P_t &= P_{t|t-1} - K_t h_t^T P_{t|t-1}, \end{aligned} \tag{A.21}$$

where $K_t = P_{t|t-1} h_t / [h_t^T P_{t|t-1} h_t + \sigma_n^2]$ is called the *Kalman gain matrix.* (A.21) constitutes the second updating stage of the Kalman filter and are called the *updating equations.*

In order to initialize the Kalman filter, we need estimates of θ_t and P_t at the start of the time series. This can be done by *a priori* guesswork, relying on the fact that the Kalman filter will rapidly update these quantities so that the initial choices become dominated by the data. We will discuss the derivation of the update equations later on, but we refer the reader to [**15**] and [**16**] for a full analysis of these equations.

A.3.3 Extended Kalman Filter for non-linear problems

Suppose that the state space model is now non-linear:

$$\theta_t = f(\theta_{t-1}) + w_t \tag{A.22}$$

where f is the non-linear system transition function and w_t is the zero mean Gaussian process noise, $w_t \sim N(0, Q)$. The observation at time $t+1$ is given by

$$X_t = h(\theta_t) + n_t, \tag{A.23}$$

where h is the observation function and n_t is the zero mean Gaussian observation noise $n_t \sim N(0, R)$.

Suppose the initial state θ_0 follows a known Gaussian distribution $\theta_0 \sim N(\hat{\theta}_0, P_0)$ and the distribution of the state at time t is

$$\theta_t \sim N(\hat{\theta}_t, P_t), \tag{A.24}$$

then θ_{t+1} at time $t+1$ follows

$$\theta_{t+1} \sim N(\hat{\theta}_{t+1}, P_{t+1}) \tag{A.25}$$

where $\hat{\theta}_{t+1}$ and P_{t+1} can be computed using the Extended Kalman Filter formula [**17**], which is derived as follows.

Let ∇f_θ be the Jacobian of the function f with respect to θ, and evaluated at $\hat{\theta}_t$. Then the predict process is

$$\begin{aligned} \hat{\theta}_{t+1|t} &= f(\hat{\theta}_t) \\ P_{t+1|t} &= \nabla f_\theta P_t \nabla f_\theta^T + Q \end{aligned} \tag{A.26}$$

and the update process

$$\begin{aligned} \hat{\theta}_{t+1} &= \hat{\theta}_{t+1|t} + K(X_{t+1} - h(\hat{\theta}_{t+1|t})) \\ P_{t+1} &= P_{t+1|t} - K(\nabla h P_{t+1|t} \nabla h^T + R)K^T \\ K &= P_{t+1|t} \nabla h^T (\nabla h P_{t+1|t} \nabla h^T + R)^{-1} \end{aligned} \tag{A.27}$$

where ∇h is the Jacobian of h evaluated at $\hat{\theta}_{t+1|t}$.

A.3.4 Particle filters

Consider a model with a state equation

$$\theta_t = f_t(\theta_{t-1}, w_t) \tag{A.28}$$

and an observation equation

$$X_t = h_t(\theta_t, n_t) \tag{A.29}$$

where θ_t is the state at time t, f_t is a, potentially, non-linear function and w_t is noise. Similarly, h_t could also be a non-linear function, taking the state θ_t and observation noise n_t at time t to generate the observation X_t.

If the data is modelled by a Gaussian state space model, then one can apply the Kalman filter or the Extended Kalman filter to compute the evolution of the posterior distributions [**18**]. However, the Taylor expansions of non-linear functions do not capture the dynamics of the non-linear models, hence the filter can diverge. In addition, the Kalman filter relies on the assumption that the noise is Gaussian, which can be too restrictive an assumption in many instances.

We can use *particle filters* to estimate the state of a non-linear dynamic system sequentially in time. Estimating from a multi-modal, multivariate, non-Gaussian probability distribution, p, can be difficult. One could generate samples from p using the importance sampling method. We replace p with an importance function, f, which is easier to sample from and has greater support than p. Then one could estimate the integral

$$I(g) = \int g(x)p(x)dx = \frac{\int \frac{g(x)p(x)}{f(x)} f(x)dx}{\int \frac{p(x)}{f(x)} f(x)dx} \tag{A.30}$$

with

$$I_{IS}(g) = \frac{\frac{1}{N}\sum_{i=1}^{N} \frac{p(x_i)}{f(x_i)} g(x_i)}{\frac{1}{N}\sum_{i=1}^{N} \frac{p(x_i)}{f(x_i)}} = \sum_{i=1}^{N} g(x_i)w(x_i) \tag{A.31}$$

where $\{x_i\}_{i=1}^N$ is a sample from f and $\{w(x_i)\}_{i=1}^N$ are the normalised weights. The particle filter algorithm includes an extra step to discard the particles with low importance weights, and multiply the particles having high importance weights [**18**]. After this step, all the weights are then renormalised and we have a sample from the distribution p.

A.3.5 Derivation of the Kalman Filter

Recall the form of our observation:

$$X_t = H^T\theta_t + n_t \tag{A.32}$$

where

- X_t is the $(n \times 1)$ observation/measurement vector at time t
- H is the $(n \times m)$ noiseless connection matrix between the state vector and the measurement vector, and is assumed to be stationary over time
- θ_t is the $(m \times 1)$ state vector
- n_t is the $(n \times 1)$ associated measurement error which is assumed to have known covariance.

The overall objective is to estimate θ_t. The error is defined as the difference between the estimate $\hat{\theta}_t$ and θ_t itself.

Assume that we want to know the value of a state variable within a process of the form:

$$\theta_t = G\theta_{t-1} + w_t \tag{A.33}$$

where θ_t is the state vector at time t, G is the $(m \times m)$ state transition matrix of the process from the state at time $t-1$ to the state at time t, and is assumed stationary over time, w_t is the associated $(m \times 1)$ noise process vector with known variance. This is assumed to have zero cross correlation with n_t.

The covariances of the two noise models are assumed stationary over time and are given by

$$Q = \mathbb{E}\,[w_t w_t^T], \quad R = \mathbb{E}\,[n_t n_t^T] \tag{A.34}$$

We will use the mean squared error (MSE) function $\epsilon(t) = \mathbb{E}\,[e_t^2]$ with

$$P_t = \mathbb{E}\,[e_t e_t^T] \tag{A.35}$$

as the error covariance matrix at time t. Then,

$$P_t = \mathbb{E}\,[e_t e_t^T] = \mathbb{E}\,[(\theta_t - \hat{\theta}_t)(\theta_t - \hat{\theta}_t)^T] \tag{A.36}$$

Assuming the prior estimate of $\hat{\theta}_t$ is $\hat{\theta}_{t|t-1}$ it is possible to write an update equation for the new estimate, combining the old estimate with measurement data. We have

$$\hat{\theta}_t = \hat{\theta}_{t|t-1} + K_t(X_t - H\hat{\theta}_{t|t-1}) \tag{A.37}$$

where K_t is the *Kalman gain*, which will be derived below. The term $X_t - H\hat{\theta}_{t|t-1} =: i_t$ is known as the *innovation* or *measurement residual.* Substituting (A.32) gives

$$\hat{\theta}_t = \hat{\theta}_{t|t-1} + K_t(H\theta_t + n_t - H\hat{\theta}_{t|t-1}) \tag{A.38}$$

Substituting this into (A.36) then yields

$$P_t = \mathbb{E}\,(\Gamma\Gamma^T) \quad \text{where} \quad \Gamma = (I - K_t H)(\theta_t - \hat{\theta}_{t|t-1}) - K_t n_t \tag{A.39}$$

Recall that $\theta_t - \hat{\theta}_{t|t-1}$ is the error of the prior estimate. This is uncorrelated with the measurement noise and therefore the expectation may be re-written as

$$\begin{aligned} P_t &= (I - K_t H)\mathbb{E}([(\theta_t - \hat{\theta}_{t|t-1})(\theta_t - \hat{\theta}_{t|t-1})]^T)(I - K_t H)^T \\ &\quad + K_t \mathbb{E}[n_t n_t^T] K_t^T \end{aligned} \tag{A.40}$$

Hence

$$P_t = (I - K_t H) P_{t|t-1} (1 - K_t H)^T + K_t R K_t^T \tag{A.41}$$

where $P_{t|t-1}$ is the prior estimate of P_t. The above equation is the error covariance update equation. Recall that the main diagonal of the covariance matrix is the variance of the errors, so the trace of P_t will give the sum of the mean squared errors. We can thus minimise the mean squared error by minimising the trace of P_t.

The trace of P_t is first differentiated with respect to K_t and the result set to zero in order to find the conditions of this minimum. First let us expand P_t.

$$P_t = P_{t|t-1} - K_t H P_{t|t-1} - P_{t|t-1} H^T K_t^T + K_t (H P_{t|t-1} H^T + R) K_t^T \tag{A.42}$$

Let $T[A]$ denote the trace of a matrix A. Recall that the trace of a matrix is equal to the trace of its transpose, so we can write

$$T[P_t] = T[P_{t|t-1}] - 2T[K_t H P_{t|t-1}] + T[K_t (H P_{t|t-1} H^T + R) K_t^T] \tag{A.43}$$

Differentiating with respect to K_t gives

$$\frac{dT[P_t]}{dK_t} = -2(H P_{t|t-1})^T + 2K_t (H P_{t|t-1} H^T + R). \tag{A.44}$$

So setting the above to zero and solving for K_t gives

$$K_t = P_{t|t-1} H^T (H P_{t|t-1} H^T + R)^{-1} \tag{A.45}$$

which is the *Kalman gain equation.* Substituting this into (A.42) yields

$$\begin{aligned} P_t &= P_{t|t-1} - K_t H P_{t|t-1} - P_{t|t-1} H^T K_t^T + K_t (H P_{t|t-1} H^T + R) K_t^T \\ &= P_{t|t-1} - K_t H P_{t|t-1} + A \end{aligned} \tag{A.46}$$

where

$$\begin{aligned} A &= -P_{t|t-1} H^T K_t^T + K_t (H P_{t|t-1} H^T + R) K_t^T \\ &= -P_{t|t-1} H^T (P_{t|t-1} H^T (H P_{t|t-1} H^T + R)^{-1})^T \\ &\quad + P_{t|t-1} H^T (H P_{t|t-1} H^T + R)^{-1} H P_{t|t-1} \\ &= 0 \end{aligned} \tag{A.47}$$

So the update equation for the error covariance matrix is

$$P_t = (I - K_t H) P_{t|t-1} \tag{A.48}$$

Collecting all of the equations for developing an estimate for θ_t, we have

$$\begin{aligned} P_t &= (I - K_t H) P_{t|t-1} && \text{(A.49a)} \\ K_t &= P_{t|t-1} H^T (H P_{t|t-1} H^T + R)^{-1} && \text{(A.49b)} \\ \hat{\theta}_t &= \hat{\theta}_{t|t-1} + K_t (X_t - H \hat{\theta}_{t|t-1}). && \text{(A.49c)} \end{aligned}$$

The state projection is achieved using

$$\hat{\theta}_{t+1|t} = G \hat{\theta}_t. \tag{A.50}$$

To complete the recursion, it is necessary to find an equation which projects the error covariance matrix into the next time interval $t+1$. This is achieved by first forming an expression for the prior error.

$$\begin{aligned} e_{t+1} &= \theta_{t+1} - \hat{\theta}_{t+1|t} \\ &= (G\theta_t + w_t) - G_t \hat{\theta}_t \\ &= Ge_t + w_t. \end{aligned} \tag{A.51}$$

Then,

$$P^-_{t+1} = \mathbb{E}\,[e^-_{t+1}(e^-_{t+1})^T] = \mathbb{E}\,[(G_t e_t + w_t)(G_t e_t + w_t)^T]. \tag{A.52}$$

Note that e_t and w_t have zero cross-correlation, because the noise w_t actually accumulates between t and $t+1$, whereas the error e_t is the error up to time t. Therefore,

$$\begin{aligned} P_{t+1|t} &= \mathbb{E}\,[(Ge_t + w_t)(Ge_t + w_t)^T] \\ &= \mathbb{E}\,[Ge_t(Ge_t)^T] + \mathbb{E}\,[w_t w_t^T] \\ &= GP_tG^T + Q, \end{aligned} \tag{A.53}$$

which completes the recursive filter.

To summarise the updating equations are:

$$\begin{aligned} \hat{\theta}_{t+1|t} &= G\hat{\theta}_t \\ P_{t+1|t} &= GP_tG^T + Q \\ P_t &= (I - K_t H) P_{t|t-1} \\ K_t &= P_{t|t-1} H^T (H P_{t|t-1} H^T + R)^{-1} \\ \hat{\theta}_t &= \hat{\theta}_{t|t-1} + K_t (X_t - H \hat{\theta}_{t|t-1}). \end{aligned} \tag{A.54}$$

References

[1] Edgar A. DeYoe, Peter Bandettini, Jay Neitz, David Miller, and Paula Winans. Functional Magnetic Resonance Imaging (fMRI) of the Human Brain. *Journal of Neuroscience Methods*, 54(2):171 – 187, 1994. Imaging Techniques in Neurobiology.

[2] Petra Ritter and Arno Villringer. Simultaneous EEG-fMRI. *Neuroscience & Biobehavioral Reviews*, 30(6):823–838, 2006. Methodological and Conceptual Advances in the Study of Brain-Behavior Dynamics: A Multivariate Lifespan Perspective.

[3] Roberto Cabeza and Lars Nyberg. Imaging Cognition II: An Empirical Review of 275 PET and fMRI Studies. *Journal of Cognitive Neuroscience*, 12(1):1–47, January 2000.

[4] Hallez et al. Muscle and eye movement artifact removal prior to EEG source localization. *Conference Proceedings of the International Conference of IEEE Engineering in Medicine and Biology Society*, 1:1002–1005, 2006.

[5] Robert E. Kass and Adrian E. Raftery. Bayes Factors. *Journal of the American Statistical Association*, 90(430):773–795, June 1995.

[6] Derek C. Stanford and Adrian E. Raftery. Approximate Bayes Factors for Image Segmentation: The Pseudolikelihood Information Criterion (PLIC). *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(11):1517–1520, November 2002.

[7] V.P. Oikonomou, A.T. Tzallas, and D.I. Fotiadis. A Kalman filter based methodology for EEG spike enhancedment. *Computer methods and pograms in biomedicine*, 85(2):101 – 108, Feburary 2007.

[8] F. Morbidi, A. Garulli, D. Prattichizzo, C. Rizzo, and S. Rossi. A Kalman filter approach to remove TMS-induced artifacts from EEG recordings. In *Proceedings of the European Control Conference*, July 2007.

[9] A. Roche, P.-J. Lahaye, and J.-B. Poline. Incremental Activation Detection in fMRI Series using Kalman Filtering. In *Biomedical Imaging: Nano to Macro, 2004. IEEE International Symposium on*, pages 376–379 Vol 1, April 2004.

[**10**] Chris Needham and Roger Boyle. Performance Evaluation Metrics and Statistics for Positional Tracker Evaluation. In James Crowley, Justus Piater, Markus Vincze, and Lucas Paletta, editors, *Computer Vision Systems*, volume 2626 of *Lecture Notes in Computer Science*, pages 278–289. Springer Berlin / Heidelberg, 2003. The University of Leeds School of Computing Leeds LS2 9JT UK.

[**11**] Thomas Nichols. Random Field Theory for Brain Imaging: Foundation & Extensions, October 2010. MASDOC RSG Talk.

[**12**] Derek C. Stanford. *Fast Automatic Unsupervised Image Segmentation and Curve Detection in Spatial Point Patterns.* PhD thesis, University of Washington, 1999.

[**13**] Julian Besag. On the Statistical Analysis of Dirty Pictures. *Journal of the Royal Statistical Society, Series B (Methodological)*, 48(3):259–302, 1986.

[**14**] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum Likelihood from Incomplete Data via the EM. *Journal of the Royal Statistical Society, Series B (Methodological)*, 39(1):1–38, 1977.

[**15**] Chris Chatfield. *The Analysis of Time Series: An Introduction.* Chapman and Hall/CRC, 6 edition, July 2003.

[**16**] Tony Lacey. The Kalman Filter. Tutorial – From the notes of the CS7322 course at Georgia Tech – Notes taken from the TINA Algortihms' Guide by N. Thacker, Electronic Systems Group, University of Sheffield.

[**17**] Shoudong Huang. Understanding Extended Kalman Filter – Part III: Extended Kalman Filter, April 2010. Notes appear to be from a course based at the University of Technology Sydney.

[**18**] Mauro Costagli and Ercan Engin Kuruoglu. Image Separation using Particle Filters. *Digital Signal Processing*, 17(5):935–946, September 2007. Special Issue on Bayesian Source Separation.