Voxelwise Inference using Convolution Random Fields

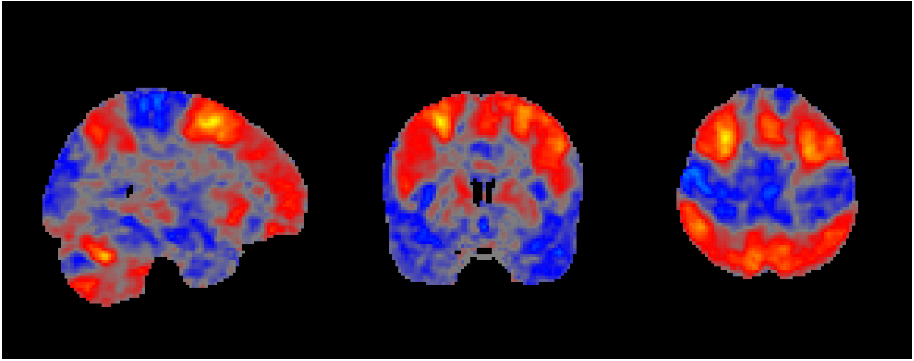
Samuel Davenport

University of Oxford

May 7, 2020

Brain Imaging

Bain Imaging



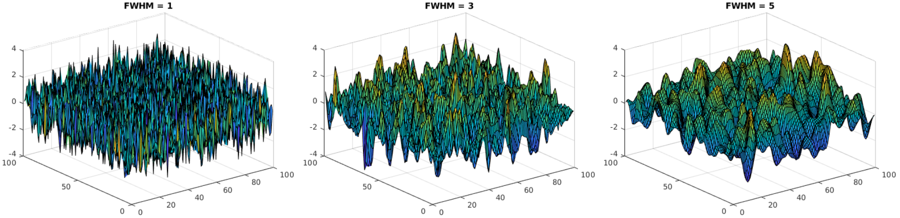
Random Fields and Test Statistics

Definition

Given $D \in \mathbb{N}$ and $S \subset \mathbb{R}^D$, define an D-dimensional random field X to be a random function

$$X : T \longrightarrow \mathbb{R}$$

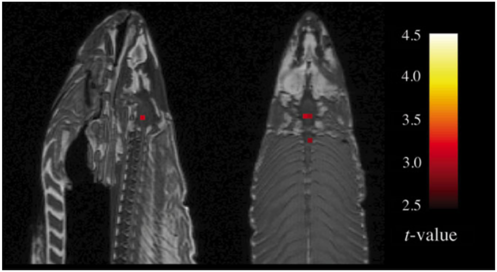
we say that X is a Gaussian random field if for all $k \in \mathbb{N}$, given $(t_1, \ldots, t_k) \in S$, $(X(t_1), \ldots, X(t_k))$ has a non-degenerate Gaussian distribution.



Multiple Testing

- Let $V \subset S$ be the set of voxels
- Take $X(v)$ to be our test statistic at each $v \in V \subset T$ so we reject the null hypothesis that there is no activity at v if $\mathbb{P}(X(v) > u) < \alpha$.

There are typically large number (around 200000) of voxels. In particular taking $\alpha = 0.05$ will mean around 10000 false discoveries! One amusing paper tried scanning a dead salmon to see what would be without without multiple testing correction.



Voxelwise Inference; Controlling the FWER

Definition

Suppose that $V_0 \subset V$ is the set of voxels that are null. Then we define the FWER (family wise error rate) to be the probability of at least one false discovery. I.e.

$$\mathbb{P}\left(\max_{v \in V} X(v) > u\right)$$

and we seek to control this at a level α.

Note that for a fine enough lattice V,

$$\mathbb{P}(\max_{v \in V} X(v) > u) \approx \mathbb{P}(\max_{t \in T} X(t) > u).$$

Cluster Failure

In 2016 (?, ?) showed that a widely used method for controlling false positive rates: Random field Theory didn't work in practice. Called into question the results of at least 3000 papers.

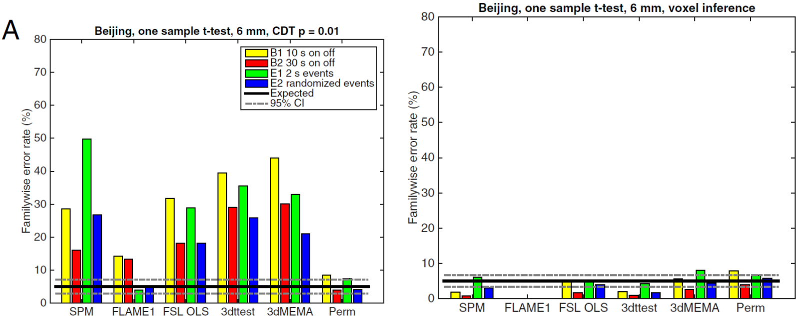


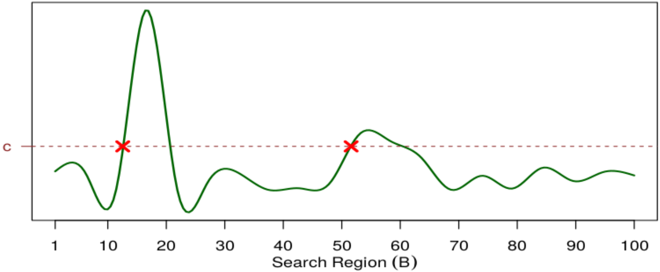
Figure 1: Clusterwise (left) has inflated false positive whereas voxelwise inference (right) is often conservative

Voxelwise RFT

Let M_u be the number of local maxima of X above u then assuming that X is twice differentiable,

$$\mathbb{P}\left(\sup_{t\in T} X(t) > u\right) = \mathbb{P}(M_u \geq 1) \leq \mathbb{E}[M_u].$$

because X exceeds u if and only if there is at least one local maxima above u. This is best seen by looking at a picture



Lipshitz Killing Curvatures

In order to evaluate this we need the following theorem ((?, ?), (?, ?)).

Theorem

Let X be a Gaussian random field on a compact domain S with constant variance σ^2, then under certain regularity conditions,

$$\liminf_{u\rightarrow\infty} -u^2 \log\left|\mathbb{P}\left(\sup_{t\in S} X(t) > u\right) - \sum_{d=0}^{D} \mathcal{L}_d \rho_d(u)\right| \geq \frac{1}{2}\left(1 + \frac{1}{2\sigma^2}\right)$$

where $\mathcal{L}_0, \dots, \mathcal{L}_D$ are constants and $\rho_d : \mathbb{R} \rightarrow \mathbb{R}$ are known functions.

$\mathcal{L}_0, \dots, \mathcal{L}_D$ are known as the Lipshitz Killing Curvatures. In particular there must exists a $\tilde{u}$ such that for all $u \geq \tilde{u}$,

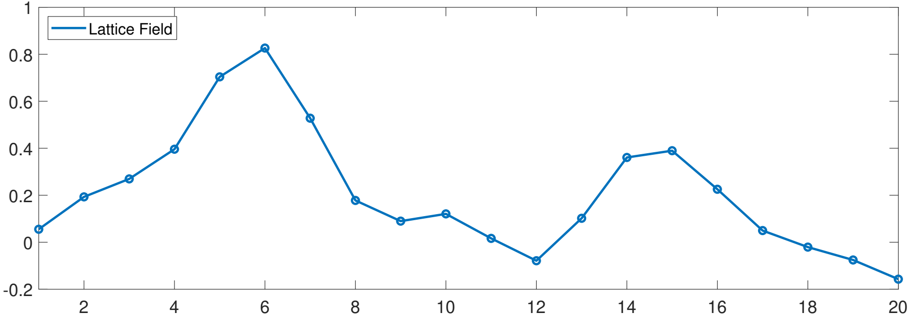
$$\left|\mathbb{P}\left(\sup_{t\in S} X(t) > u\right) - \sum_{d=0}^{D} \mathcal{L}_d \rho_d(u)\right| \leq e^{-\left(\frac{1}{2}\left(1+\frac{1}{2\sigma^2}\right)u^2\right)}.$$

Smoothing

In fMRI smoothing is done in order to increase the signal to noise ratio. To understand how this works, let $X(l)$ be random at every point l of a lattice L. Then smoothing X with a kernel K gives

$$Y(v) = \sum_{l \in L} K(v-l)X(l)$$

at every voxel $v \in L$.

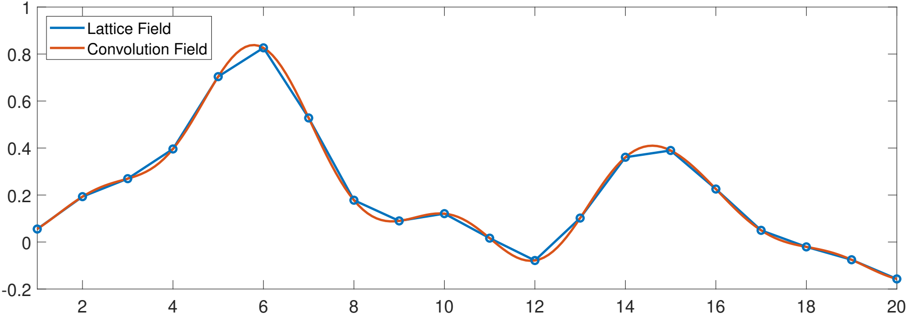


Convolution Random Fields

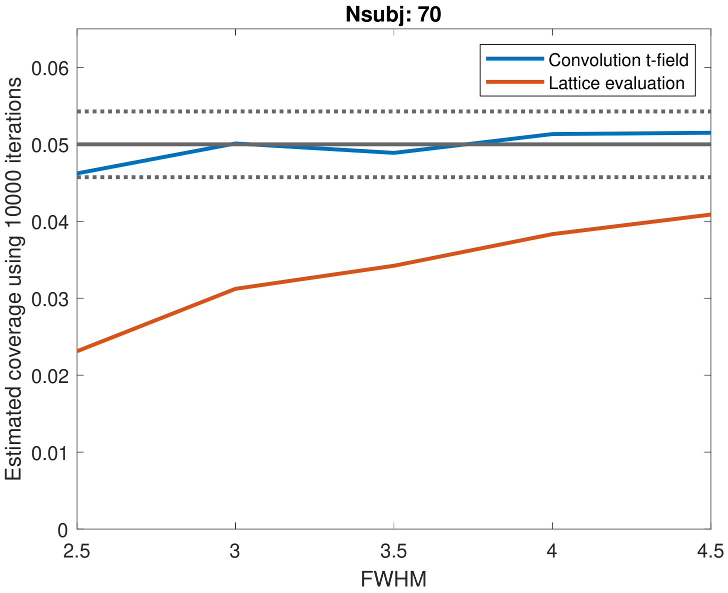
Definition

Given random data X on a lattice $L \subset \mathbb{R}^D$ for $s \in \mathbb{R}^D$ and some kernel K, define the convolution field to be $Y : \mathbb{R}^D \to \mathbb{R}$,

$$Y(s) = (K \star X)(s) = \sum_{l \in L} K(s-l)X(l).$$



2D voxelwise inference



Conclusion

- Existing software only has LKC implementations under stationarity but the framework is more general.
- Using convolution fields exactly controls the FWER at the right level.
- It's also much faster than existing non-parametric methods that are currently widely used but are slow and inefficient.
- Software available to run LKC estimation under non-stationary and to generate convolutoin fields is available at sjdavenport.github.io/software.

Bibliography