

Multiscale Variance Stabilization via Maximum Likelihood

G. P. Nason*,
School of Mathematics,
University of Bristol,
United Kingdom

April 19, 2013

Abstract

This article proposes maximum likelihood approaches for multiscale variance stabilization (MVS) transforms for independently and identically distributed data. For two MVS transforms we present new unified theoretical results on their Jacobians which are a key component of the likelihood. The new results provide (i) a deeper theoretical understanding of the transforms and (ii) the ability to compute the likelihood in linear time compared to previous numerical methods which required quadratic effort. The new MVS transforms are shown empirically to compare favourably to the well-known Box-Cox transform and almost dominate it. We show how the new transforms, like Box-Cox, can facilitate simpler models by an example involving the famous wool data.

KEYWORDS: VARIANCE STABILIZATION, HAAR-FISZ, HAAR WAVELET, MULTISCALE BOX-COX

1 Introduction

Variance stabilization through transformation is a popular and commonly performed technique in statistics. For example, analysts routinely apply log and/or square-root transformations to draw data towards homoscedasticity and/or normality. Atkinson (1985) provides an excellent and comprehensive exposition of stabilization methods. For some contemporary problems variance stabilization is important and essential. For example, in astronomical image processing, where variance stabilization methods are used in combination with other methods to “recover important structures of various morphologies in (very) low-count images” and demonstrate that such techniques are “competitive relative to many existing denoising methods”, Zhang et al. (2008).

Recently, Fryzlewicz (2003) introduced the Haar-Fisz transform for the variance stabilization of a Poisson intensity sequence, see also Fryzlewicz and Nason (2004). Informally, the Haar-Fisz transform works by first applying a Haar wavelet transform to the data sequence, dividing the mother coefficients by the square root of the father coefficients and then inverting the transform. The original Haar-Fisz transform

*g.p.nason@bristol.ac.uk

took a Poisson intensity sequence and turned it into a near-Gaussian sequence with near-constant variance with the intensity information encoded into the mean of the transformed sequence. The Haar-Fisz transform was the first multiscale variance stabilization transform and is an example of a non-diagonal transform. By comparison applying standard log or square root transforms to a sequence X_1, \dots, X_n to obtain a transformed sequence Y_1, \dots, Y_n is diagonal because one, and only one, X_i is used to form each Y_i . Simulation studies in Fryzlewicz and Nason (2004) showed the power of Haar-Fisz with a performance that “was nearly always better than that of the current state-of-the-art in terms of accuracy and speed”.

The Haar-Fisz method has since been extended and applied to a number of different situations, see Section 2.2.2 for more details. Also, new multiscale variance stabilization techniques have appeared, notably Zhang et al. (2008): a version of which we call the ‘multiscale Box-Cox’ here. The current article contributes the following:

1. until now multiscale variance stabilization has been concerned with function estimation. For example, for Haar-Fisz for Poisson the models are essentially $X_t \sim \text{Pois}\{\lambda(t)\}$ where $\lambda(t)$ is some function and the ordering of the X_t matters. This article considers the case where the $\{X_t\}$ are independent and identically distributed (iid) according to some unknown distribution. This is the setting, for the seminal Box and Cox (1964) paper and the setting for many problems in applied statistics.
2. most previous work in this area focuses on variance stabilization and Gaussianization is often a by-product, see Stuart and Ord (1994). Here, we use maximum likelihood (ML) to explicitly ‘Gaussianize’ a sequence, again paralleling the Box and Cox (1964) method.
3. Bailey (2008) and Nason and Bailey (2008) used ML but *only* with function-sequence data and computed completely by numerical means. In particular, the Jacobian computation used the `fjac` algorithm from Press et al. (1992) which costs $\mathcal{O}(n^2)$ for *each* likelihood optimization step for a data set of size n . Our main result below derives analytical formulae for the Jacobians which permits a faster $\mathcal{O}(n)$ computation. Our new result is general applying to a version of Haar-Fisz and the different, more recent, method of Zhang et al. (2008). As well as producing practical benefits this new theory sheds considerable theoretical light on the likelihood and how it should be interpreted.

The article is laid out as follows. Section 2 reviews the Box-Cox transform introducing our version of the Haar-Fisz and multiscale Box-Cox transforms in sections 2.2 and 2.3, respectively. Section 3 examines the likelihood for each transforms paying careful attention to the Jacobians. The Box-Cox Jacobian is reviewed in section 3.1 and new theory that derives the analytical Jacobians for Haar-Fisz and multiscale Box-Cox follows in sections 3.2–3.4. Section 4 compares the transforms via simulation and we present an analysis of the famous wool data using Box-Cox and the new transforms.

2 Transformations

2.1 Box-Cox Transform

Let $\mathbf{X} = \{X_1, \dots, X_n\}$ be an independent and identically distributed data set of size n , where $X_i \geq 0$, $\mathbb{E}X_i = \mu < \infty$ and $\text{var } X_i = \sigma^2 < \infty$ for $i = 1, \dots, n$. Box and Cox (1964) introduced the following parametric transform of the data:

$$Y_{\text{BC},i}^{(\lambda)} = \begin{cases} (X_i^\lambda - 1)/\lambda & \text{for } \lambda \neq 0 \\ \log X_i & \text{for } \lambda = 0. \end{cases} \quad (1)$$

2.2 The Haar-Fisz Transform

The Haar-Fisz variance stabilizing transform bolts the Fisz transform, which pulls a pair of random variables towards the Gaussian, onto the discrete Haar wavelet transform. We review both of these components next.

2.2.1 The Discrete Haar Wavelet Transform

The multiscale variance stabilization techniques defined in the next two sections are obtained by modifying the Haar wavelet transform. For further details on Haar wavelets see Vidakovic (1999) or Nason (2008) or many of the general books on wavelets such as Daubechies (1992), Burrus et al. (1997), or Mallat (1998).

Given data X_1, \dots, X_n where $n = 2^J$ for some integer $J > 1$ we define the Haar wavelet transform as follows. Set initial coefficients $c_{J,i-1} = X_i$ for $i = 1, \dots, n$. Then perform the recursive operation:

$$c_{j-1,i} = (c_{j,2i} + c_{j,2i+1})/2, \quad (2)$$

and

$$d_{j-1,i} = (c_{j,2i} - c_{j,2i+1})/2, \quad (3)$$

for $j = J, \dots, 1$ and $i = 0, \dots, 2^j - 1$. One often writes the coefficients at a given scale level, j , as a vector. Hence, $\mathbf{d}_j = (d_{j,0}, \dots, d_{j,2^j-1})$ and $\mathbf{c}_j = (c_{j,0}, \dots, c_{j,2^j-1})$. The discrete Haar wavelet transform of \mathbf{X} is the collection $\mathbf{d} = (\mathbf{c}_0, \mathbf{d}_0, \mathbf{d}_1, \mathbf{d}_2, \dots, \mathbf{d}_{J-1})$. It is convenient to use the notation \mathcal{H} to denote the Haar wavelet transform, so $\mathbf{d} = \mathcal{H}(\mathbf{X})$ where $\mathbf{X} = (X_1, \dots, X_n)$.

The $\{d_{j,i}\}$ are known as mother wavelet coefficients and the $\{c_{j,i}\}$ are known as the father wavelet or scaling coefficients. We have chosen a particular normalization for the Haar wavelet transform here. Formulae (2) and (3) use $2^{-1}(1, -1)$ whereas other expositions use $2^{-1/2}(1, -1)$ or $(1, -1)$ in other applications.

The transform can easily be inverted and the original data can be recovered from \mathbf{d} by reversing the recursive steps (2) and (3):

$$c_{j,2i} = c_{j-1,i} + d_{j-1,i}, \quad (4)$$

and

$$c_{j,2i+1} = c_{j-1,i} - d_{j-1,i}, \quad (5)$$

again for $j = 1, \dots, J$ and $i = 0, \dots, 2^j - 1$.

Note, for the purposes of exposition we assume $n = 2^J$ but the Haar transform can be applied for any $n \in \mathbb{N}$ by computing every possible Haar wavelet coefficient.

2.2.2 Haar-Fisz Transform

The Haar-Fisz transform was introduced in Fryzlewicz (2003) as a method for variance stabilization of data P_1, \dots, P_n distributed $P_i \sim \text{Pois}(\lambda_i)$ for some sequence of intensities λ_i , see also Fryzlewicz and Nason (2004). Here λ_i was taken to be a sequence of samples from some function $\lambda(x)$ with given smoothness properties. The Haar-Fisz method was subsequently adapted to χ^2 -like data for local spectral estimation in Fryzlewicz and Nason (2006) and for spectral estimation for stationary time series in Fryzlewicz et al. (2008). The Haar-Fisz method was extended in a different direction by Fryzlewicz and Delouille (2005) and Fryzlewicz et al. (2007) by creating the ‘data-driven Haar-Fisz’ transformation which addressed problems where the mean-variance function, $h(\mu)$, is not known and has to be estimated from the data. For example, for Poisson data $h(\mu) = \mu$ and for χ^2 -like $h(\mu) \propto \mu^2$. In an example, Fryzlewicz et al. (2007) estimated a two linear piece h for GOES satellite X-ray flux data’s mean-variance function and postulated two mean-variance regimes linked to the autoranging electronics within the sensor. Further empirical evidence concerning the effectiveness of data-driven Haar-Fisz was demonstrated in Motakis et al. (2006) for variance stabilization of microarray data which also handled replicates and Nason and Bailey (2008) on estimation of conflict intensity. A detailed analysis of data-driven Haar-Fisz appears in Fryzlewicz (2008), theoretical work demonstrating asymptotic normality for the inhomogeneous Poisson case appears in Schmidt and Xu (2008), and a generalization concerning wavelets/filters other than Haar was achieved by Jansen (2006).

The original Haar-Fisz method, for Poisson data, uses the following result by Fisz (1955):

Theorem 1 (Fisz) *Let $P_i \sim \text{Pois}(\lambda_i)$ for $i = 1, 2$ and X_1, X_2 independent. Define the function $\xi : \mathbb{R}^2 \rightarrow \mathbb{R}$ by*

$$\xi(X_1, X_2) = \begin{cases} 0 & \text{if } X_1 = X_2 = 0, \\ (X_1 - X_2)/(X_1 + X_2)^{1/2} & \text{else.} \end{cases} \quad (6)$$

If $(\lambda_1, \lambda_2) \rightarrow (\infty, \infty)$ and $\lambda_1/\lambda_2 \rightarrow 1$ then $\xi(X_1, X_2) - \xi(\lambda_1, \lambda_2) \xrightarrow{D} N(0, 1)$.

The theorem shows that, under the right conditions, the quantity $\xi(X_1, X_2)$ is approximately Gaussian with a constant variance. The reader will note that $X_1 - X_2$ and $X_1 + X_2$ are the Haar mother and father wavelet coefficients of X_1, X_2 . The innovation of the Haar-Fisz transform was to realize that one could replace the Haar mother coefficient in the Haar wavelet transform of section 2.2.1 by $\xi(X_1, X_2)$ and to do this recursively throughout the whole transform. Consequently, the wavelet tableaux now has *all* mother coefficients approximately Gaussian. This new tableaux is just a Haar wavelet transform containing these new coefficients which can be inverted. Since the transform is orthogonal the inverted transform will be approximately Gaussian with near-constant variance. This is the Haar-Fisz transform, for Poisson data, and is fully invertible, just by reversing the steps.

To summarize, given data \mathbf{X} , the steps in the Haar-Fisz transform for Poisson data are:

1. Apply the Haar wavelet transform to the data: $\mathbf{d} = \mathcal{H}(\mathbf{X})$.

2. Replace the mother wavelet Haar coefficients, $d_{j,k}$ by the Fisz-transformed equivalents $f_{j,k} = d_{j,k}/c_{j,k}^{1/2}$ to form $\mathbf{f} = (\mathbf{f}_0, \mathbf{f}_1, \dots, \mathbf{f}_{J-1})$.
3. Invert the new wavelet coefficients to obtain the final transformed sequence $\mathbf{Y} = \mathcal{H}^{-1}(\mathbf{f})$.

In what follows the data X_1, \dots, X_n are assumed iid, but with no particular underlying parametric distribution in mind. The Haar-Fisz transform we introduce here lies somewhere between the fixed parametric assumptions in Fryzlewicz and Nason (2004, 2006) (Poisson and χ^2) and the more general data-driven mean-variance relationships found in Fryzlewicz and Delouille (2005); Fryzlewicz et al. (2007); Motakis et al. (2006) and Fryzlewicz (2008). Here, the assumption, as far as Haar-Fisz is concerned is $h(\mu) \propto \mu^{2\lambda}$. The appropriate value of λ for Poisson, then, is $\lambda = 1/2$ and for χ^2 we would have $\lambda = 1$. Our version of Haar-Fisz is conceptually similar to the single-parameter Box-Cox transform, but obviously Haar-Fisz is multiscale.

2.2.3 General form of our Haar-Fisz transform

We modify the general formula for the Haar-Fisz transform for Poisson data that appears in (Fryzlewicz, 2003, page 164) by adding a more general power transformation parameter λ as follows.

Let $\mathbf{X} = (X_1, \dots, X_n)$ for $n = 2^J$ be the vector of interest. Introduce the family of Haar wavelet vectors $\{\psi^{j,k}\}$, where $j = 0, 1, \dots, J-1$ is the scale parameter ($J-1$ is fine scale, 0 is coarsest) and $k = l2^{J-j}$, $l = 0, 1, \dots, j$ is the location parameter. The components of $\psi^{j,k}$ will be denoted by $\psi_i^{j,k}$ for $i = 0, \dots, n-1$. We define:

$$\psi_i^{j,k} = 2^{j-J} \begin{cases} 0 & \text{for } i < k, \\ 1 & \text{for } k \leq i < k + 2^{J-j-1}, \\ -1 & \text{for } k + 2^{J-j-1} \leq i < k + 2^{J-j}, \\ 0 & \text{for } k + 2^{J-j} \leq i. \end{cases} \quad (7)$$

Similarly, we introduce the family of Haar scaling vectors $\{\phi^{j,k}\}$, whose components will be denoted by $\phi_i^{j,k}$ (the range of j, k and i remains unchanged). We define

$$\phi_i^{j,k} = 2^{j-J} \begin{cases} 0 & \text{for } i < k, \\ 1 & \text{for } k \leq i < k + 2^{J-j}, \\ 0 & \text{for } k + 2^{J-j} \leq i. \end{cases} \quad (8)$$

This definition of discrete Haar wavelets is similar to that of Nason et al. (2000). The difference is that we “pad” the wavelet vectors with zeroes on both sides so that they all have length n , and do not normalize them.

Further, let $\langle \cdot, \cdot \rangle$ denote the inner product of two vectors, and let the binary representation of the integer i be $\mathbf{b}^J(i) = (b_0^J(i), b_1^J(i), \dots, b_{J-1}^J(i))$, where $i < 2^J$.

The formula for the i th element of the Haar-Fisz transformed vector of \mathbf{X} , with parameter λ , is

$$U_i = \frac{\langle \phi^{0,0}, \mathbf{X} \rangle}{n} + \sum_{j=0}^{J-1} (-1)^{b_j^J(i)} c_{j,J,i}(\mathbf{X}), \quad (9)$$

where

$$c_{j,J,i}(\mathbf{X}, \lambda) = \begin{cases} \frac{\langle \psi^j, [i/2^{J-j}]2^{J-j}, \mathbf{X} \rangle}{\langle \phi^j, [i/2^{J-j}]2^{J-j}, \mathbf{X} \rangle^\lambda} & \text{if } \langle \phi^j, [i/2^{J-j}]2^{J-j}, \mathbf{X} \rangle > 0, \\ 0 & \text{otherwise.} \end{cases} \quad (10)$$

The only difference between (10) and Fryzlewicz (2003) is that we use a general λ whereas a fixed $\lambda = 1/2$ was used in Fryzlewicz (2003) to be used for Poisson distributed data. This is akin to the difference between the Anscombe (1948) transform and the Box-Cox transform with parameter λ .

We now adapt the general formula (10) into a slightly different form. Our adaption is useful for two purposes: (i) the new form conveniently encapsulates the next variance stabilization technique, described in section 2.3, (ii) the new form facilitates the establishment of an interesting new result concerning the Jacobians of both transforms.

Define the function $F_{\text{HF}} : [0, \infty)^3 \rightarrow \mathbb{R}$ by

$$F_{\text{HF}}(a, b, \lambda) = a(a + b)^{-\lambda}. \quad (11)$$

Then all of the terms in the sum in (9) can be represented by the difference of two F_{HF} terms computed on the first and second half of the partial sum involved in that term. An example, for the case $n = 4$, should make this clear.

Suppose the data set is (X_1, X_2, X_3, X_4) . Then let

$$Y_{\text{HF},1}^{(\lambda)} = \frac{1}{4} \sum_{i=1}^4 X_i + \frac{X_1 + X_2 - X_3 - X_4}{4^{1-\lambda}(X_1 + X_2 + X_3 + X_4)^\lambda} + \frac{X_1 - X_2}{2^{1-\lambda}(X_1 + X_2)^\lambda} \quad (12)$$

$$= \bar{X} + 4^{\lambda-1} F_{\text{HF}}(X_1 + X_2, X_3 + X_4, \lambda) - 4^{\lambda-1} F_{\text{HF}}(X_3 + X_4, X_1 + X_2, \lambda) + 2^{\lambda-1} F_{\text{HF}}(X_1, X_2, \lambda) - 2^{\lambda-1} F_{\text{HF}}(X_2, X_1, \lambda). \quad (13)$$

The other three components of the Haar-Fisz transform are

$$Y_{\text{HF},2}^{(\lambda)} = \bar{X} + \frac{X_1 + X_2 - X_3 - X_4}{4^{1-\lambda}(X_1 + X_2 + X_3 + X_4)^\lambda} - \frac{X_1 - X_2}{2^{1-\lambda}(X_1 + X_2)^\lambda} \quad (14)$$

$$Y_{\text{HF},3}^{(\lambda)} = \bar{X} - \frac{X_1 + X_2 - X_3 - X_4}{4^{1-\lambda}(X_1 + X_2 + X_3 + X_4)^\lambda} + \frac{X_3 - X_4}{2^{1-\lambda}(X_3 + X_4)^\lambda} \quad (15)$$

$$Y_{\text{HF},4}^{(\lambda)} = \bar{X} - \frac{X_1 + X_2 - X_3 - X_4}{4^{1-\lambda}(X_1 + X_2 + X_3 + X_4)^\lambda} - \frac{X_3 - X_4}{2^{1-\lambda}(X_3 + X_4)^\lambda}, \quad (16)$$

all of which can be put into a form similar to (13).

2.3 The Multiscale Box-Cox Transform

More recently, Zhang et al. (2008) introduced a simple and elegant new variance stabilization technique that involved combining the discrete Haar wavelet transform with the well known Anscombe (1948) transform. Donoho (1993) first proposed denoising Poisson-distributed signals using wavelets by first applying Anscombe's transform, which results in approximately variance stabilized Gaussian data, and then using regular wavelet shrinkage for Gaussian data. Anscombe's transform, given by $\mathcal{A}(X_i) = \sqrt{X_i + 3/8}$, is essentially equivalent to preprocessing one's data with the Box-Cox transform with $\lambda = 1/2$

Zhang et al. (2008) begin by forming the Haar *father* wavelet coefficients by recursively applying formula (2) as normal. Then Anscombe’s transform is applied to all of the father coefficients. Then, those Anscombe-transformed coefficients are used to form Haar *mother* wavelet coefficients. In other words, formula (3) becomes

$$d_i^{j-1} = \mathcal{A}(c_{2i-1}^j) - \mathcal{A}(c_{2i}^j). \quad (17)$$

The beauty of their idea is that if the X_i are iid Poisson distributed then clearly so are all the c_k^j (as they are merely sums of independent Poissons). Hence, all the $\mathcal{A}(c_k^j)$ are approximately Gaussian with the the same variance. Inversion of the new d_k^j Haar wavelet tableaux results in an approximately variance-stabilized Gaussian sequence. Both the Haar-Fisz transform and the transform introduced by Zhang et al. (2008) are similar in that they both produce stabilized Gaussian Haar wavelet coefficients which can then be inverted to provide variance stabilized data.

We generalize Zhang et al. (2008) by replacing Anscombe in (17) by the Box-Cox transform resulting in the *multiscale Box-Cox transform*. Define the function $F_{BC} : [0, \infty) \times \mathbb{R} \rightarrow \mathbb{R}$ by

$$F_{BC}(a, \lambda) = \begin{cases} (a^\lambda - 1)/\lambda & \text{for } \lambda \neq 0, \\ \log a & \text{for } \lambda = 0. \end{cases} \quad (18)$$

For example, again for the case $n = 4$, we obtain:

$$\begin{aligned} Y_{\text{MB},1}^{(\lambda)} &= \bar{X} + F_{BC}\{(X_1 + X_2)/2, \lambda\} - F_{BC}\{(X_3 + X_4)/2, \lambda\} \\ &+ F_{BC}(X_1, \lambda) - F_{BC}(X_2, \lambda), \end{aligned} \quad (19)$$

similar formulae apply for $Y_{\text{MB},2}^{(\lambda)}$, $Y_{\text{MB},3}^{(\lambda)}$ and $Y_{\text{MB},4}^{(\lambda)}$. The full formula for $n = 8$ is presented as (31) in Appendix B. A general formula for the multiscale Box-Cox transform can be obtained by replacing $c_{j,J,n}$ in (10) by

$$\begin{aligned} f_{j,J,i}(\mathbf{X}, \lambda) &= F\{(\langle \psi^{j,\lfloor i/2^{J-j} \rfloor 2^{J-j}}, \mathbf{X} \rangle + \langle \phi^{J,\lfloor i/2^{J-j} \rfloor 2^{J-j}}, \mathbf{X} \rangle)/2, \lambda\} \\ &- F\{(\langle \psi^{j,\lfloor i/2^{J-j} \rfloor 2^{J-j}}, \mathbf{X} \rangle - \langle \phi^{J,\lfloor i/2^{J-j} \rfloor 2^{J-j}}, \mathbf{X} \rangle)/2, \lambda\}, \\ &= F\{(d_{j,J,i} + c_{j,J,i})/2, \lambda\} - F\{(d_{j,J,i} - c_{j,J,i})/2, \lambda\}, \end{aligned} \quad (20)$$

where $c_{j,J,i}$ is as in (10), $d_{j,J,i}$ is the associated mother coefficient, $i = 0, \dots, n - 1$ and $F = F_{\text{BC}}$.

3 Likelihood

Section 2 defined the three transformations that we are considering in this paper: the Box-Cox, the Haar-Fisz and the multiscale Box-Cox. In practice, for any given set of data all of the transforms require the parameter λ to be chosen in some way. The seminal paper of Box and Cox (1964) introduced and studied a range of approaches to parameter estimation, including maximum likelihood and Bayesian methods, and these have become widely used across many fields. For brevity and focus we concentrate on exploration of the maximum likelihood approach here. However, a Bayesian analysis would be perfectly possible and desirable in many contexts.

The likelihood approach is explained in Atkinson (1985) whose clear approach we follow. The aim of the likelihood approach is to choose λ that maximizes the Gaussian likelihood of the transformed observations but expressed as a function of the *original observations*, i.e.

$$(2\pi\sigma^2)^{-n/2} \exp\{-(Y^{(\lambda)} - X)^T(Y^{(\lambda)} - X)/2\sigma^2\}J, \quad (21)$$

where $Y^{(\lambda)}$, X are the vectorized versions of $Y_i^{(\lambda)}$ and X_i and where J is the Jacobian of the transformation, i.e.

$$J = \prod_{i=1}^n \left| \frac{\partial Y_i^{(\lambda)}}{\partial X_i} \right|. \quad (22)$$

It is important to write the likelihood in terms of the original observations, which necessitates the use of the Jacobian, to enable likelihoods from the same transformation, but with different λ , to be compared, and also to compare likelihoods from different transformations.

When the observations, X_i , are considered to be part of some model, e.g. $\mathbb{E}X = W\beta$, then the transformation approach needs to estimate both the transformation parameter, λ , the parameters of interest in the model, β and maybe σ^2 , which has to be estimated but might not be of direct interest. As described by Atkinson (1985) this can be achieved by a two-stage approach where the parameters (σ^2, β) are estimated in the normal way conditioned on λ to obtain the profile log-likelihood:

$$L_{\max}(\lambda) = -(n/2) \log \hat{\sigma}^2(\lambda) + \log J, \quad (23)$$

where $\hat{\sigma}^2 = n^{-1}Y^{(\lambda),T}(I - H)Y^{(\lambda)}$ is the usual maximum likelihood estimate of σ^2 and H is the usual hat matrix. Conveniently, this single approach will work with all of our transformations above. However, the exact likelihood and the form of Jacobian is different in each case. We shall address these details next. However, before we do it is worth noting that, as usual, the maximum likelihood framework provides important, useful and interesting information. For example, the asymptotic distribution of the maximizing parameter, confidence intervals for the parameter and convergence results.

3.1 Box-Cox likelihood

The Jacobian for the Box-Cox transform has a simple form due to the simplicity of the functional form of F_{BC} and the important fact that the Box-Cox transform is diagonal. In other words, the transformed value $Y_i^{(\lambda)}$ only depends on X_i and none of the other X_j for $j \neq i$. So,

$$\partial Y_i^{(\lambda)} / \partial X_i = \partial F_{\text{BC}}(X_i, \lambda) / \partial X_i = X_i^{\lambda-1}. \quad (24)$$

Hence, the (log) Jacobian is

$$\log J_{\text{BC}} = (\lambda - 1) \sum_{i=1}^n \log X_i. \quad (25)$$

Observe that the log of the Jacobian is essentially the log of the geometric mean of the data, X . Combining (23) with (25) one can see that the problem is one of penalized

likelihood: the aim is to reduce the sample variance $\hat{\sigma}^2$ tensioned against an increasing (or decreasing if $\lambda < 1$) geometric mean of the data.

It seems that the penalized likelihood interpretation for the basic problem has not been emphasized in quite this way before, although it has appeared in more complex situations: such as using Box-Cox transformed curves in the estimation of reference centile curves in Cole and Green (1992). The penalized likelihood interpretation becomes increasingly useful and interesting when one considers the Jacobians of the Haar-Fisz and Multiscale Box-Cox transforms below.

3.2 Likelihood and Jacobian for multiscale transforms

Analytical Jacobians for the Haar-Fisz and Multiscale Box-Cox transforms (=“the multiscale transforms”) are considerably more difficult to establish because the transforms are not diagonal. Below we establish a general result for the multiscale transform Jacobians which makes use of the fact that both Haar-Fisz and Multiscale Box-Cox arise from the same general form in (9).

Definition 1 Define the general multiscale variance stabilization transform $Y^{(\lambda)}$ of X by the general sum given in (9) with the new $f_{j,J,n}$ given in (20) with $F = F_{HF}$ for Haar-Fisz and $F = F_{BC}$ for the Multiscale Box-Cox transform.

Theorem 2 Define \mathcal{T} to be the set of all unique possible terms in the general sum in (9) for all $i = 0, \dots, n - 1$. The Jacobian of the multiscale variance stabilization transform given in Definition 1 is given by

$$J(F, \lambda) = 2^{n-1} \prod_{j \in \mathcal{T}} F\{(d_{j,J,n} + c_{j,J,n})/2, \lambda\} + F\{(d_{j,J,n} - c_{j,J,n})/2, \lambda\}. \quad (26)$$

The proof of Theorem 2 appears in Appendix A. The proof is constructive and makes heavy use of the dyadic structure in the Jacobian which arises because of the binary/dyadic construction of the general formula in (9). An example of the Jacobian for the Multiscale Box-Cox transform is given in Appendix B.

The theoretical Jacobians below have been checked for correctness by comparing to the result of a numerical Jacobian procedure adapted from the `fdjac` routine from Press et al. (1992, Page 388).

3.3 Haar-Fisz Jacobian and the Haar-Fisz geometric mean

Let $c_{j,k}$ be the father Haar wavelet coefficients defined in (2). Then the general Jacobian (26) for the Haar-Fisz transform on choosing $F = F_{HF}$ further simplifies to

$$J_{HF}(\lambda) = J(F_{HF}, \lambda) = 2^{n-1} \prod_{j=0}^{J-1} \prod_{k=0}^{2^j-1} c_{j,k}^{-\lambda}. \quad (27)$$

Remarkably, the Jacobian of the rather complicated non-diagonal Haar-Fisz transform is merely the the product of the father wavelet coefficients raised to the power of $-\lambda$.

Just as the Jacobian for the regular Box-Cox transform (for $\lambda = 1$) is proportional to the geometric mean of the data, the quantity $J(F_{HF}, 1)$ is proportional to the (reciprocal of the) geometric mean of the father wavelet coefficients of the data.

Table 1: Number of times particular stabilization method achieves maximum likelihood out of 100 trials.

Distribution	Box-Cox	Haar-Fisz	Multiscale BC
Poisson	0	67	33
Log-Normal	1	6	93
Folded Normal	0	11	89
χ^2	0	0	100
Geometric	0	6	94

3.4 Multiscale Box-Cox Jacobian and its geometric mean

The Multiscale Box-Cox transform's Jacobian also simplifies on choosing $F = F_{\text{BC}}$ in (26) to give

$$J(F_{\text{BC}}, \lambda) = 2^{n-1} \prod_{j=1}^J \prod_{k=0}^{2^{j-1}-1} (c_{j,2k}^{\lambda-1} + c_{j,2k+1}^{\lambda-1}). \quad (28)$$

For example, if $n = 4$ then

$$J(F_{\text{BC}}, \lambda) = 8\{(X_1 + X_2)^{\lambda-1} + (X_3 + X_4)^{\lambda-1}\}(X_1^{\lambda-1} + X_2^{\lambda-1})(X_3^{\lambda-1} + X_4^{\lambda-1}). \quad (29)$$

As in the previous section, for the special value of $\lambda = 2$, since the $c_{j,k}$ are arithmetic means of parts of the data set at different scales and locations, the sum of $c_{j,2k} + c_{j,2k+1}$ in (28) is proportional to yet another arithmetic mean, and the Jacobian $J(F_{\text{BC}}, 2)$ is related to a geometric mean of all of those. Indeed, setting $\lambda = 2$ and using (2) we obtain

$$J(F_{\text{BC}}, 2) = 2^{n-1} \prod_{j=1}^J \prod_{k=0}^{2^{j-1}-1} c_{j-1,k} = 2^{n-1} \prod_{j=0}^{J-1} \prod_{k=0}^{2^j-1} c_{j,k}. \quad (30)$$

Hence, the associated measure of location for the multiscale Box-Cox transform is the same as for the Haar-Fisz transform.

4 Example and Simulation

4.1 Comparison of Stabilization Methods

Simulation is used to compare the transforms. Each run draws $n = 64$ iid observations from the following distributions: (a) $X_i \sim \text{Poisson}(3) + 1$; (b) $X_i \sim \exp(Z_i)$, where $Z_i \sim N(1, 1)$, lognormal; (c) $X_i \sim |Z_i|$, where $Z_i \sim N(1, 1)$, folded normal; (d) χ_1^2 ; (e) $X_i \sim \text{Geometric}(0.2) + 1$. Then the three transforms are applied to each set of observations, their likelihood maximized and the 'winner' identified. The overall competition consists of 100 runs: Table 1 counts the number of winners for each transform for each distribution. The multiscale Box-Cox transform performs the best for all distributions apart from the Poisson, where the Haar-Fisz transform works best.

Table 2: Partial ANOVA table for first-order and complete second-order effects. Columns y , $\log y$, y^{-1} are $\times 10^{-2}$, column Haar-Fisz, MBC are $\times 10^5$. SS=Sum of squares, MS=Mean square.

	DF	Response, y				
		y	$\log y$	y^{-1}	Haar-Fisz	MBC
1st Order SS	3	147 485	22 411	129 220	117.9	3126.8
2nd Order SS	6	42 243	154	32 010	14.7	296.8
2nd Order MS		7 040	26	5 335	2.5	49.5
Residual SS	17	12 567	639	7 540	36.7	321.3
Residual MS		739	38	443	2.2	18.9
F		9.52	0.68	12.0	1.14	2.62

4.2 Wool Data Example

We analyze the famous ‘wool’ data which was presented in *A Textile Experiment using a Single Replicate of a 3^3 Design* from Box and Cox (1964) and further analyzed in Atkinson (1985) whose results we reproduce here in Table 2. The experiment investigated the behaviour of worsted yarn under cycles of repeated loading with the following variables y : the number of cycles to failure, y , obtained in a single replicate of a 3^3 experiment under the following factors x_1 : length of test specimen (250, 300, 350)mm, x_2 : amplitude of loading cycle (8, 9, 10)mm and x_3 : load (40, 45, 50)g. Table 2 shows an ANOVA table adapted and extended from the one in Atkinson (1985). The essential question raised by Atkinson (1985) is whether a second-order or, simpler, first order model is required.

The F -statistics of various transformed models can be seen in the last row of Table 2. In the untransformed model (the y column) $F = 9.52$ for the second-order terms: this is significant compared to, e.g., $F_{6,17}(0.95) = 2.70$. However, if one analyzes the transformed $\log y$ data adding the second-order terms gives $F = 0.68$ which is not significant. Hence, considerable model simplicity is obtained by modelling the log-transformed data. Table 2 also shows that the corresponding value for the reciprocal y^{-1} transform is $F = 12.0$ which indicates that second-order terms are still required.

The statistics for the Haar-Fisz and Multiscale Box-Cox transforms are $F = 1.14$ and $F = 2.62$ both are not significant at the 5% level. So, we could use either of these transforms to fit the simpler model that did not require second-order terms like the log-transformed data. The multiscale transforms above are defined for dyadic data where $n = 2^J$ and here $n = 27$. To use the multiscale transforms we extended the data sequence to length $n = 32$ by repeating the first and last values three and two times respectively. In principle, it would be possible to define the multiscale algorithms for arbitrary n by computing *all possible* Haar wavelet coefficients.

Figure 1 shows the log-likelihood curves as a function of the parameter for each of variance stabilization transforms. The maximizing parameter value for Box-Cox is $\hat{\lambda} = -0.06$ and given the associated confidence interval we can agree with earlier analyzes that probably $\lambda = 0$ is suitable which translates into the log transform. For multiscale Box-Cox $\hat{\lambda} = 0.26$ and, given the confidence interval, the value of $\lambda = \frac{1}{4}$ might well be suitable. For the Haar-Fisz transform it is less clear. Here $\hat{\lambda} = 1.71$,

the confidence interval might permit $\lambda = 2$ in this case. In each case the likelihood is nicely quadratic.

5 Conclusions

We introduced new variants of the Haar-Fisz and multiscale Box-Cox transforms for variance stabilization and Gaussianization of data via maximum likelihood. We presented new theory that provides exact analytical formulae for the Jacobians of these new transforms: this sheds light on the analytical nature of the transforms as well as providing significant computational benefits (from quadratic to linear). Simulation results showed that our new transforms performed better than Box-Cox giving larger likelihoods although at the expense maybe of a more complex non-diagonal transform. We applied our new methods to the famous wool data which showed the new transforms shared the ability of the Box-Cox transform to enable simpler models to be appropriately fitted.

Acknowledgements

This work was partially supported by EPSRC grants EP/D005221/1, EP/K020951/1 and The Energy Programme. The Energy Programme is an RCUK cross-council initiative led by EPSRC and contributed to by ESRC, NERC, BBSRC and STFC. All the computations in this paper were carried out in R, R Development Core Team (2009).

A Proof of Theorem 2

Our proof begins by considering the final terms of the general sum for the multiscale transforms, essentially (9) with the $f_{j,J,n}$ coefficient given by (13) and (20) for Haar-Fisz and Multiscale Box-Cox respectively. Let us temporarily concentrate on the Multiscale Box-Cox transform. The last terms in the sum, for each successive i are $a_1 = +\{f_{\text{BC}}(X_1) - f_{\text{BC}}(X_2)\}$, $a_2 = -\{f_{\text{BC}}(X_1) - f_{\text{BC}}(X_2)\}$ (for $i = 1, 2$), then $a_3 = +\{f_{\text{BC}}(X_3) - f_{\text{BC}}(X_4)\}$, $a_4 = -\{f_{\text{BC}}(X_3) - f_{\text{BC}}(X_4)\}$ (for $i = 3, 4$), up to $a_{n-1} = +\{f_{\text{BC}}(X_{n-1}) - f_{\text{BC}}(X_n)\}$, $a_n = -\{f_{\text{BC}}(X_{n-1}) - f_{\text{BC}}(X_n)\}$ (for $i = n - 1, n$). It is important to note that $a_1 = -a_2$, $a_3 = -a_4$, and so on.

To construct the Jacobian all of the above n terms are each to be differentiated by ∂X_j for $j = 1, \dots, n$. The differentiation of a_1, a_2 by X_j is only non-zero for $j = 1, 2$, resulting in $a_{1,1}$ and $a_{2,2}$, and similarly for all the other terms. Hence, in the Jacobian the term involving the differential of a_1, a_2 only appears in two rows and as $a_1 = -a_2$ this term in that row is of opposite sign in the two rows. The same thing happens for all of the other pairs a_{2i-1}, a_{2i} , but the terms only appear at the places where their differential with respect to X_j is nonzero. Precisely the same pattern occurs with the Haar-Fisz transform, although each differential term is a fraction with a denominator containing the $X_1 + X_2$ sum.

A similar pattern emerges for the coarser scale coefficients, but with more repeated rows. For example, for the next most ‘coarsest’ terms in the general sum we have, for successive i : $b_1 = +\{f_{\text{BC}}(\sum_1^2 X_k) - f_{\text{BC}}(\sum_3^4 X_k)\}$, $b_2 = b_1$, $b_3 = -b_1$, $b_4 = -b_1$

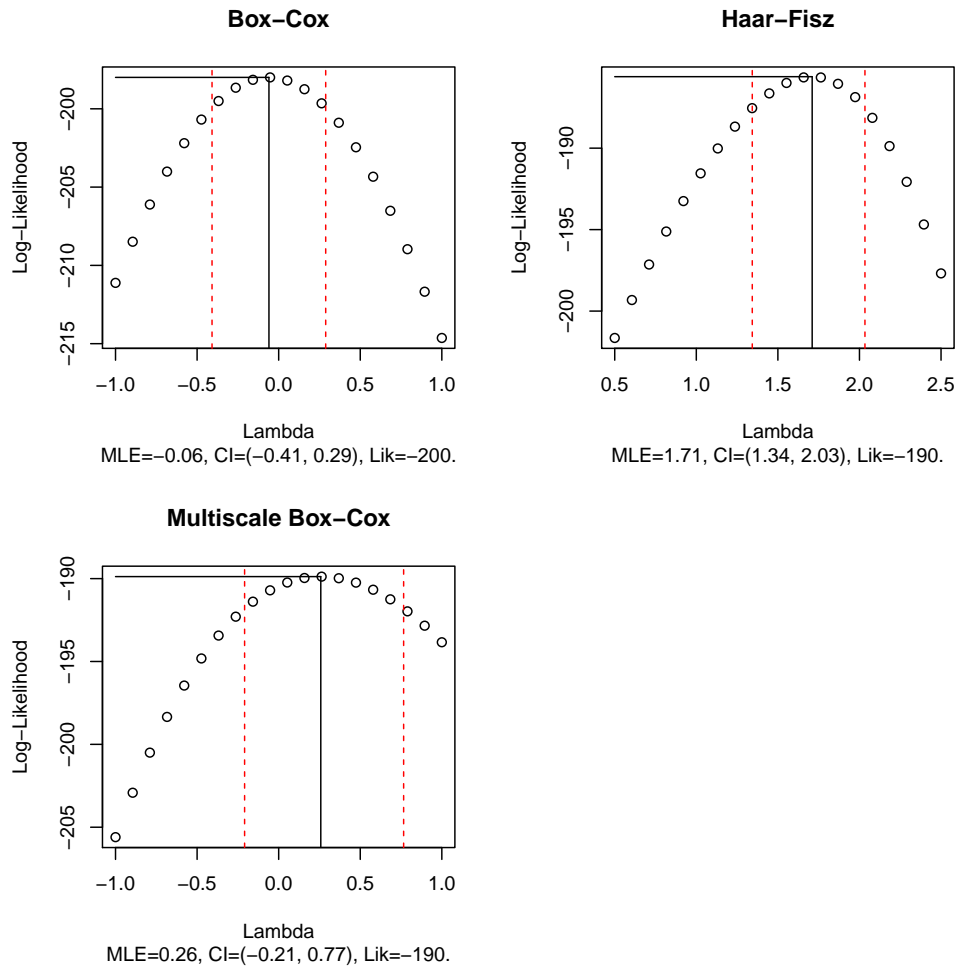


Figure 1: Log-likelihood curves for each of the variance stabilization transforms for the wool data. Solid vertical line indicates maximizing parameter value with 95% confidence intervals depicted by dashed lines. Sub-title shows the maximum likelihood estimator for transform parameters with 95% confidence interval and the maximizing likelihood value.

and then $b_5 = +\{f_{\text{BC}}(\sum_5^6 X_k) - f_{\text{BC}}(\sum_7^8 X_k)\}$, $b_6 = b_5$, $b_7 = -b_5$, $b_8 = -b_5$ and so on up to $b_{n-3} = +\{f_{\text{BC}}(\sum_{n-3}^{n-2} X_k) - f_{\text{BC}}(\sum_{n-1}^n X_k)\}$, $b_{n-2} = b_{n-3}$, $b_{n-1} = -b_{n-3}$, $b_n = -b_{n-3}$.

These terms enter into the Jacobian and are each differentiated by ∂X_j for $j = 1, \dots, n$. The differentiation of b_1, \dots, b_4 by X_j are only non-zero for $j = 1, \dots, 4$, and similarly for other terms. So, these terms, after differentiation occur in blocks of four rows each. The first two of the four are the same, and the second two are the negative of those).

Similar patterns emerge for the coefficients of coarser scales still (i.e. the next would occur in blocks of 8, with four rows of terms the same, and the next four the negative of those, and so on.). The first term in the general sum is the ‘‘sum of all the coefficients’’ which when differentiated gives a constant which is added to every term in the Jacobian.

Hence, we end up with a Jacobian with a great deal of structure. The highly-organized structure suggests a pattern of row and column operations that can considerably simplify the Jacobian. Given the dyadic block nature of the structure described above, the pattern of operations follows a binary pattern. Let R_k denote the k th row, and C_k denote the k th column. The notation $x \rightarrow y$ means row (or column) x replaces row (or column) y . The operations are:

1. Perform $R_{2k} + R_{2k-1} \rightarrow R_{2k-1}$ for $k = 1, \dots, 2^{J-1}$. This cancels out the finest scale terms in every odd row, i.e. the a_2 neutralizes the a_1 in the first row. This operation also doubles the values of all of the other terms in the odd rows (as the even row contains the same information at the coarser scales, it is just the finest scale information that differs on successive rows). Hence, a factor of 2 can be extracted from all the odd rows. Taken for all the rows a scale factor of 2^{J-1} can be extracted.
2. Then perform $R_{2k} - R_{2k-1} \rightarrow R_{2k}$ for $k = 1, \dots, 2^{J-1}$. All of the information at scales coarser than the finest cancels out. Each of the even rows contains only two non-zero columns which contain $(-a_{1,1} \ a_{2,2})$, $(-a_{3,3} \ a_{4,4})$ and so on up until the last row which contains all zeroes followed by $(-a_{n-1} \ a_n)$.
3. Then perform $R_{4k-1} + R_{4k-3} \rightarrow R_{4k-3}$ for $k = 1, \dots, 2^{J-2}$. This cancels out the next coarser scale information at rows R_{4k-3} similar to step 1 for the finer scales. Similarly, we can extract a factor of 2^{J-2} at this point, making the total extracted factor 2^{2J-3} . Then perform $R_{4k-1} - R_{4k-3} \rightarrow R_{4k-1}$ as in step 2. This results in 2^{J-2} rows at R_{4k-1} which are all zeroes apart from blocks of four coefficients corresponding to $(-b_{1,1} \ -b_{1,1} \ b_{2,2} \ b_{2,2})$ and so on (actually, the second $b_{1,1}$ is formally $b_{1,2}$ but this is equal to $b_{1,1}$, and so on).
4. Then perform $R_{8k-1} + R_{8k-7} \rightarrow R_{8k-7}$ for $k = 1, \dots, 2^{J-3}$, this cancels out the next coarser scale and enables another factor of 2^{J-3} to be extracted resulting in a total extracted factor of 2^{3J-6} . Then perform $R_{8k-1} - R_{8k-7} \rightarrow R_{8k-1}$ as in the previous steps. This results in 2^{J-3} rows at R_{8k-1} which are all zeroes apart from blocks of eight coefficients.
5. These steps should be continued until row R_{n-1} is reached and processed. The extracted factor at this stage is $2^{J-1} 2^{J-2} \dots 2 = 2^{2^J-1} = 2^{n-1}$.

6. Steps 1. to 5. are now applied to the *columns* of the Jacobian, but because of the abundance of zeroes no doubling of values of rows occur and now extra factors of two are extracted. This results in a Jacobian where (i) the top row consists of a single 1 followed by $n - 1$ zeros; (ii) the diagonal of the Jacobian consists of the top-left 1 just mentioned and each of the terms in formula in (26): one for each scale and location apart from the very coarsest (hence n of them); (iii) a sparse arrangement of off-diagonal elements.
7. We now expand the determinant using Laplace's method by successively pivoting on the diagonal elements only (and that is why they end up in the product in (26)). The ordering is carried out from finer scale to coarser scale coefficients. Expand the determinant in the following order (i) pivot on (1,1), which has the single entry of 1 and a row of zeroes; (ii) then pivot successively on the elements which have the finest scale $f_{j,J,k}$ coefficients in the diagonal. These are all in columns with all other entries zero and there are 2^{J-1} of them, half of the columns. This will result in a Jacobian with all the $f_{j,J,n}$ coefficients at the next coarsest scale, and these too will now be in columns consisting of entirely of zeroes apart from the diagonal entry; (iii) pivot on the next coarsest scale elements, and so on.

B Example of Jacobian result

We follow the steps of the constructive proof presented in Appendix A for the case $n = 2^J = 8$, $J = 3$ for the Multiscale Box-Cox transform where the data is X_1, \dots, X_8 . The Multiscale Box-Cox transform for $i = 1, \dots, n$ and parameter λ is

$$\begin{aligned}
Y_1 &= \Sigma X_i + \{f_{\text{BC}}(\Sigma_1^4 X_i, \lambda) - f_{\text{BC}}(\Sigma_5^8 X_i, \lambda)\} \\
&\quad + \{f_{\text{BC}}(\Sigma_1^2 X_i, \lambda) - f_{\text{BC}}(\Sigma_3^4 X_i, \lambda)\} + \{f_{\text{BC}}(X_1, \lambda) - f_{\text{BC}}(X_2, \lambda)\}, \\
Y_2 &= \Sigma X_i + \{f_{\text{BC}}(\Sigma_1^4 X_i, \lambda) - f_{\text{BC}}(\Sigma_5^8 X_i, \lambda)\} \\
&\quad + \{f_{\text{BC}}(\Sigma_1^2 X_i, \lambda) - f_{\text{BC}}(\Sigma_3^4 X_i, \lambda)\} - \{f_{\text{BC}}(X_1, \lambda) - f_{\text{BC}}(X_2, \lambda)\}, \\
Y_3 &= \Sigma X_i + \{f_{\text{BC}}(\Sigma_1^4 X_i, \lambda) - f_{\text{BC}}(\Sigma_5^8 X_i, \lambda)\} \\
&\quad - \{f_{\text{BC}}(\Sigma_1^2 X_i, \lambda) - f_{\text{BC}}(\Sigma_3^4 X_i, \lambda)\} + \{f_{\text{BC}}(X_3, \lambda) - f_{\text{BC}}(X_4, \lambda)\}, \\
Y_4 &= \Sigma X_i + \{f_{\text{BC}}(\Sigma_1^4 X_i, \lambda) - f_{\text{BC}}(\Sigma_5^8 X_i, \lambda)\} \\
&\quad - \{f_{\text{BC}}(\Sigma_1^2 X_i, \lambda) - f_{\text{BC}}(\Sigma_3^4 X_i, \lambda)\} - \{f_{\text{BC}}(X_3, \lambda) - f_{\text{BC}}(X_4, \lambda)\}, \\
Y_5 &= \Sigma X_i - \{f_{\text{BC}}(\Sigma_1^4 X_i, \lambda) - f_{\text{BC}}(\Sigma_5^8 X_i, \lambda)\} \\
&\quad + \{f_{\text{BC}}(\Sigma_5^6 X_i, \lambda) - f_{\text{BC}}(\Sigma_7^8 X_i, \lambda)\} + \{f_{\text{BC}}(X_5, \lambda) - f_{\text{BC}}(X_6, \lambda)\}, \\
Y_6 &= \Sigma X_i - \{f_{\text{BC}}(\Sigma_1^4 X_i, \lambda) - f_{\text{BC}}(\Sigma_5^8 X_i, \lambda)\} \\
&\quad + \{f_{\text{BC}}(\Sigma_5^6 X_i, \lambda) - f_{\text{BC}}(\Sigma_7^8 X_i, \lambda)\} - \{f_{\text{BC}}(X_5, \lambda) - f_{\text{BC}}(X_6, \lambda)\}, \\
Y_7 &= \Sigma X_i - \{f_{\text{BC}}(\Sigma_1^4 X_i, \lambda) - f_{\text{BC}}(\Sigma_5^8 X_i, \lambda)\} \\
&\quad - \{f_{\text{BC}}(\Sigma_5^6 X_i, \lambda) - f_{\text{BC}}(\Sigma_7^8 X_i, \lambda)\} + \{f_{\text{BC}}(X_7, \lambda) - f_{\text{BC}}(X_8, \lambda)\}, \\
Y_8 &= \Sigma X_i - \{f_{\text{BC}}(\Sigma_1^4 X_i, \lambda) - f_{\text{BC}}(\Sigma_5^8 X_i, \lambda)\} \\
&\quad - \{f_{\text{BC}}(\Sigma_5^6 X_i, \lambda) - f_{\text{BC}}(\Sigma_7^8 X_i, \lambda)\} - \{f_{\text{BC}}(X_7, \lambda) - f_{\text{BC}}(X_8, \lambda)\}, \quad (31)
\end{aligned}$$

Note, the coefficients have a slightly different normalization as given in (19). We will explain the difference this makes at the end of the proof.

For the Jacobian we need to differentiate each Y_i by X_j for both $i, j = 1, \dots, n$. For example,

$$\frac{\partial Y_1}{\partial X_1} = 1 + A + C + G, \quad (32)$$

where $A = (\sum_1^4 X_i)^{\lambda-1}$, $C = (\sum_1^2 X_i)^{\lambda-1}$ and $G = X_1^{\lambda-1}$. Define $B = (\sum_4^8 X_i)^{\lambda-1}$, $D = (\sum_3^4 X_i)^{\lambda-1}$, $E = (\sum_5^6 X_i)^{\lambda-1}$, $F = (\sum_7^8 X_i)^{\lambda-1}$, $H = X_2^{\lambda-1}$, $I = X_3^{\lambda-1}$, $J = X_4^{\lambda-1}$, $K = X_5^{\lambda-1}$, $L = X_6^{\lambda-1}$, $M = X_7^{\lambda-1}$ and $N = X_8^{\lambda-1}$. The Jaco-

bian is

$$\begin{array}{c}
 \begin{array}{cccccccc}
 1+A+C+G & 1+A+C-H & 1+A-D & 1+A-D & 1-B & 1-B & 1-B & 1-B \\
 1+A+C-G & 1+A+C+H & 1+A-D & 1+A-D & 1-B & 1-B & 1-B & 1-B \\
 1+A-C & 1+A-C & 1+A+D+I & 1+A+D-J & 1-B & 1-B & 1-B & 1-B \\
 1+A-C & 1+A-C & 1+A+D-I & 1+A+D+J & 1-B & 1-B & 1-B & 1-B \\
 1-A & 1-A & 1-A & 1-A & 1+B+E+K & 1+B+E-L & 1+B-F & 1+B-F \\
 1-A & 1-A & 1-A & 1-A & 1+B+E-K & 1+B+E+L & 1+B-F & 1+B-F \\
 1-A & 1-A & 1-A & 1-A & 1+B-E & 1+B-E & 1+B+F+M & 1+B+F-N \\
 1-A & 1-A & 1-A & 1-A & 1+B-E & 1+B-E & 1+B+F-M & 1+B+F+N
 \end{array} \\
 \hline
 \end{array}$$

After executing steps 1 and 2 of the proof gives

$$\begin{array}{c}
 \begin{array}{cccccccc}
 1+A+C & 1+A+C & 1+A-D & 1+A-D & 1-B & 1-B & 1-B & 1-B \\
 -G & H & 0 & 0 & 0 & 0 & 0 & 0 \\
 1+A-C & 1+A-C & 1+A+D & 1+A+D & 1-B & 1-B & 1-B & 1-B \\
 0 & 0 & -I & J & 0 & 0 & 0 & 0 \\
 1-A & 1-A & 1-A & 1-A & 1+B+E & 1+B+E & 1+B-F & 1+B-F \\
 0 & 0 & 0 & 0 & -K & L & 0 & 0 \\
 1-A & 1-A & 1-A & 1-A & 1+B-E & 1+B-E & 1+B+F & 1+B+F \\
 0 & 0 & 0 & 0 & 0 & 0 & -M & N
 \end{array} \\
 \hline
 2^4
 \end{array}$$

Then execute step 3 to obtain:

$$\begin{array}{c}
 \begin{array}{cccccccc}
 1+A & 1+A & 1+A & 1+A & 1-B & 1-B & 1-B & 1-B \\
 -G & H & 0 & 0 & 0 & 0 & 0 & 0 \\
 -C & -C & D & D & 0 & 0 & 0 & 0 \\
 0 & 0 & -I & J & 0 & 0 & 0 & 0 \\
 1-A & 1-A & 1-A & 1-A & 1+B & 1+B & 1+B & 1+B \\
 0 & 0 & 0 & 0 & -K & L & 0 & 0 \\
 0 & 0 & 0 & 0 & -E & F & F & F \\
 0 & 0 & 0 & 0 & 0 & 0 & -M & N
 \end{array} \\
 \hline
 2^4 2^2
 \end{array}$$

Then the next step (4/5), followed by the first set of column operations gives:

$$2^4 2^2 2 \left| \begin{array}{cccccccc} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ -G & H & 0 & 0 & 0 & 0 & 0 & 0 \\ -C & -C & D & D & 0 & 0 & 0 & 0 \\ 0 & 0 & -I & J & 0 & 0 & 0 & 0 \\ -A & -A & -A & -A & B & B & B & B \\ 0 & 0 & 0 & 0 & -K & L & 0 & 0 \\ 0 & 0 & 0 & 0 & -E & -E & F & F \\ 0 & 0 & 0 & 0 & 0 & 0 & -M & N \end{array} \right| \sim 2^7 \left| \begin{array}{cccccccc} 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ -G & H+G & 0 & 0 & 0 & 0 & 0 & 0 \\ -C & 0 & D & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -I & J+I & 0 & 0 & 0 & 0 \\ -A & 0 & -A & 0 & B & 0 & B & 0 \\ 0 & 0 & 0 & 0 & -K & L+K & 0 & 0 \\ 0 & 0 & 0 & 0 & -E & 0 & F & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -M & N+M \end{array} \right|$$

Then perform the next scale column operation (3rd subtract 1st, and 7th subtract 5th) gives

$$2^7 \left| \begin{array}{cccccccc} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ -G & H+G & G & 0 & 0 & 0 & 0 & 0 \\ -C & 0 & D+C & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -I & J+I & 0 & 0 & 0 & 0 \\ -A & 0 & 0 & 0 & B & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -K & L+K & K & 0 \\ 0 & 0 & 0 & 0 & -E & 0 & F+E & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -M & N+M \end{array} \right|$$

and then the 5th subtract the first gives:

$$2^7 \left| \begin{array}{cccccccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -G & H+G & G & 0 & G & 0 & 0 & 0 \\ -C & 0 & D+C & 0 & C & 0 & 0 & 0 \\ 0 & 0 & -I & J+I & 0 & 0 & 0 & 0 \\ -A & 0 & 0 & 0 & A+B & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -K & L+K & K & 0 \\ 0 & 0 & 0 & 0 & -E & 0 & F+E & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -M & N+M \end{array} \right|$$

Now we use Laplace's method to obtain the determinant, pivoting first on the top left element as described in step 7.

$$2^7 \left| \begin{array}{cccccccc} H+G & G & 0 & G & 0 & 0 & 0 & 0 \\ 0 & D+C & 0 & C & 0 & 0 & 0 & 0 \\ 0 & -I & J+I & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & A+B & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -K & L+K & K & 0 & 0 \\ 0 & 0 & 0 & -E & 0 & F+E & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -M & N+M & 0 \end{array} \right|$$

Now pivot on the finest scale diagonals, $N+M$, $L+K$, $J+I$ and $H+G$:

$$2^7(N+M) \left| \begin{array}{ccccccc} H+G & G & 0 & G & 0 & 0 & 0 \\ 0 & D+C & 0 & C & 0 & 0 & 0 \\ 0 & -I & J+I & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & A+B & 0 & 0 & 0 \\ 0 & 0 & 0 & -K & L+K & K & 0 \\ 0 & 0 & 0 & -E & 0 & F+E & 0 \end{array} \right|$$

Then:

$$2^7(N+M)(L+K) \left| \begin{array}{cccccc} H+G & G & 0 & G & 0 & 0 \\ 0 & D+C & 0 & C & 0 & 0 \\ 0 & -I & J+I & 0 & 0 & 0 \\ 0 & 0 & 0 & A+B & 0 & 0 \\ 0 & 0 & 0 & -E & F+E & 0 \end{array} \right|$$

Then:

$$2^7(N + M)(L + K)(J + I) \begin{vmatrix} H + G & G & G & 0 \\ 0 & D + C & C & 0 \\ 0 & 0 & A + B & 0 \\ 0 & 0 & -E & F + E \end{vmatrix}$$

Then:

$$2^7(N + M)(L + K)(J + I)(H + G) \begin{vmatrix} D + C & C & 0 \\ 0 & A + B & 0 \\ 0 & -E & F + E \end{vmatrix}$$

Then the next finest: $D + C$ and/or $A + B$ gives

$$2^7(N + M)(L + K)(J + I)(H + G)(D + C) \begin{vmatrix} A + B & 0 \\ -E & F + E \end{vmatrix}$$

which finally gives the result as $2^7(A + B)(C + D)(E + F)(G + H)(I + J)(K + L)(M + N)$ as required. Now, the different normalization mentioned above means that A, B is 2^2 bigger than it should be, and C, D, E and F also are all twice as big.

References

- Anscombe, F. J. (1948) The transformation of Poisson, binomial and negative-binomial data., *Biometrika*, **35**, 246–254.
- Atkinson, A. C. (1985) *Plots, Transformations and Regression*, Clarendon Press, Oxford.
- Bailey, D. (2008) *Data mining of Early Day Motions and multiscale variance stabilisation of count data.*, Ph.D. thesis, University of Bristol, U.K.
- Box, G. E. P. and Cox, D. R. (1964) An analysis of transformations, *J. R. Statist. Soc. B*, **26**, 211–252.
- Burrus, C. S., Gopinath, R. A., and Guo, H. (1997) *Introduction to Wavelets and Wavelet Transforms: A Primer*, Prentice Hall, Upper Saddle River, NJ.
- Cole, T. J. and Green, P. J. (1992) Smoothing reference centile curves: the LMS method and penalized likelihood, *Stat. Med.*, **11**, 1305–1319.
- Daubechies, I. (1992) *Ten Lectures on Wavelets*, SIAM, Philadelphia.
- Donoho, D. L. (1993) Nonlinear wavelet methods of recovery for signals, densities, and spectra from indirect and noisy data, in *Proceedings of Symposia in Applied Mathematics*, volume 47, American Mathematical Society, Providence: RI.
- Fisz, M. (1955) The limiting distribution of a function of two independent random variables and its statistical application, *Colloquium Mathematicum*, **3**, 138–146.
- Fryzlewicz, P. (2008) Data-driven wavelet-Fisz methodology for nonparametric function estimation, *Elec. J. Stat.*, **2**, 863–896.

- Fryzlewicz, P. and Delouille, V. (2005) A data-driven Haar-Fisz transform for multi-scale variance stabilization, in *Proceedings of the 13th IEEE Workshop on Statistical Signal Processing, Bordeaux*, pp. 539–544.
- Fryzlewicz, P. and Nason, G. P. (2004) A Haar-Fisz algorithm for Poisson intensity estimation, *J. Comp. Graph. Stat.*, **13**, 621–638.
- Fryzlewicz, P. and Nason, G. P. (2006) Haar-Fisz estimation of evolutionary wavelet spectra, *J. R. Statist. Soc. B*, **68**, 611–634.
- Fryzlewicz, P., Delouille, V., and Nason, G. P. (2007) GOES-8 X-ray sensor variance stabilization using the multiscale data-driven Haar-Fisz transform., *J. R. Statist. Soc. C*, **56**, 99–116.
- Fryzlewicz, P., Nason, G. P., and von Sachs, R. (2008) A wavelet-Fisz approach to spectrum estimation, *J. Time Ser. Anal.*, **29**, 868–880.
- Fryzlewicz, P. Z. (2003) *Wavelet Techniques for Time Series and Poisson Data*, Ph.D. thesis, University of Bristol, U.K.
- Jansen, M. (2006) Multiscale Poisson data smoothing, *J. R. Statist. Soc. B*, **68**, 27–48.
- Mallat, S. G. (1998) *A Wavelet Tour of Signal Processing*, Academic Press, San Diego.
- Motakis, E. S., Nason, G. P., Fryzlewicz, P., and Rutter, G. A. (2006) Variance stabilization and normalization for one-color microarray data using a data-driven multi-scale approach, *Bioinformatics*, **22**, 2547–2553.
- Nason, G. P. (2008) *Wavelet Methods in Statistics with R*, Springer, New York.
- Nason, G. P. and Bailey, D. (2008) Estimating the intensity of conflict in Iraq, *J. R. Statist. Soc. A*, **171**, 899–914.
- Nason, G. P., von Sachs, R., and Kroisandt, G. (2000) Wavelet processes and adaptive estimation of the evolutionary wavelet spectrum, *J. R. Statist. Soc. B*, **62**, 271–292.
- Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P. (1992) *Numerical Recipes in C, the Art of Scientific Computing*, Cambridge University Press, Cambridge.
- R Development Core Team (2009) *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, ISBN 3-900051-07-0.
- Schmidt, T. and Xu, L. (2008) Some limit results on the Haar-Fisz transform for inhomogeneous poisson signals, *J. Anal. App.*, **27**, 483–497.
- Stuart, A. and Ord, J. K. (1994) *Kendall's Advanced Theory of Statistics: Distribution Theory*, volume 1, Arnold, London.
- Vidakovic, B. (1999) *Statistical Modeling by Wavelets*, Wiley, New York.
- Zhang, B., Fadili, M. J., and Starck, J.-L. (2008) Wavelets, ridgelets and curvelets for Poisson noise removal, *IEEE Trans. Im. Proc.*, **17**, 1093–1108.