A data-driven Haar-Fisz transform for multiscale variance stabilization

Piotr Fryzlewicz
Department of Mathematics, Imperial College, London, UK

Véronique Delouille
Royal Observatory of Belgium, Brussels, Belgium

Guy P. Nason
Department of Mathematics, University of Bristol, Bristol, UK

Summary. Transforming data so that its variance is stable and its distribution is taken closer to the Gaussian is the aim of many techniques (e.g. Anscombe, Box-Cox). Recently, new techniques based on the Haar-Fisz transform have been introduced that use a multiscale method to transform and stabilize data with a known mean-variance relationship, e.g. Poisson or Chi-square. In many practical cases, the variance of the data can be assumed to increase with the mean, but other characteristics of the distribution are unknown. We introduce a method, the data-driven Haar-Fisz transform (DDHFT), which uses Haar-Fisz but also estimates the mean-variance relationship. For known noise distributions, the DDHFT is shown to be competitive with the fixed Haar-Fisz methods. We show how the DDHFT effectively denoises a solar flux time series obtained from the XRS X-ray sensor on GOES8 satellite where other existing methods fail.

Keywords: variance stabilization, Gaussianisation, Haar-Fisz.

1. Introduction

In non-parametric regression, we are often faced with the problem of estimating a one-dimensional function \( f : [0, 1] \mapsto \mathbb{R} \) from noisy observations \( X_i \) taken on an equispaced grid:

\[
X_i = f\left(\frac{i}{n}\right) + \varepsilon_i, \quad i = 1, \ldots, n,
\]

where \( \varepsilon_i \)'s are random variables with \( \mathbb{E}(\varepsilon_i) = 0 \). Various subclasses of the problem can be identified, depending on the joint distribution of \( (\varepsilon_i)_{i=1}^n \) and on the smoothness of \( f \). In particular, in cases where \( f \) possesses irregular features (such as discontinuities or spikes), several authors advocate the use of nonlinear estimators based on wavelet shrinkage: see e.g. Donoho and Johnstone (1994), Nason (1996), Johnstone and Silverman (2005b), to name but a few. Many of these estimators combine excellent finite-sample performance and (near-)optimal MISE behaviour over a variety of smoothness classes of \( f \).

However, several of those wavelet-based methods rely on the assumption that \( (\varepsilon_i)_{i=1}^n \) is a sequence of independent variables distributed as \( N(0, \sigma^2) \). In practical situations, this assumption is often violated in that the distribution of \( \varepsilon_i \) depends on the level of the underlying signal \( f(i/n) \). We now briefly describe two important examples of such situations.

Poisson intensity estimation. Suppose that a (possibly inhomogeneous) one-dimensional Poisson process is observed on the interval \([0, 1]\) and discretised into a vector \( X = (X_1, \ldots, X_n) \),
where $X_i$ is the number of events falling into the interval $[i/n, (i + 1)/n)$. Each $X_i$ can be thought of as coming from a Poisson distribution with an unknown parameter $\lambda_i$, which needs to be estimated. Note that in this case the “noise” $\varepsilon_i = X_i - \mathbb{E}(X_i) = X_i - \lambda_i$ is independent but not identically distributed.

**Periodogram smoothing.** In periodogram smoothing, observations $(X_i)_{i=1}^n$ are asymptotically independent and, neglecting the edge effects, asymptotically distributed as

$$X_i \sim \frac{1}{2} f(\omega_i) \chi^2_2,$$

where $f : [0, \pi] \mapsto \mathbb{R}_+$ is the “nonnegative half” of the (continuous) spectral density to be estimated and $\omega_i$'s are the Fourier frequencies. In this case, the “noise” $\varepsilon_i = X_i - \mathbb{E}(X_i) \approx X_i - f(\omega_i)$ is asymptotically independent but not identically distributed.

One route to the construction of successful nonlinear wavelet-based denoising algorithms consists in tailoring the method so as to handle one particular type of noise. For Poisson data, recent examples of such techniques include the multiscale Bayesian methods of Kolaczyk (1999) and Timmermann and Nowak (1999). Some other recent techniques for estimating Poisson intensities are reviewed in Besbeas et al. (2004). Antoniadis and Sapatinas (2001) and Antoniadis et al. (2001) consider wavelet shrinkage for noise whose variance is (at most) a quadratic or cubic function of the mean, respectively. Several authors consider wavelet thresholding of $\chi^2$ data arising in various time series contexts: see Fryzlewicz (2003), Section 2.3.2, for a list of references. The major drawback of these techniques is that they effectively require a pre-estimate of the signal to be supplied to the estimation procedure.

A more “modular” approach consists in transforming the noisy signal into a signal contaminated with approximately Gaussian noise with constant variance, then applying one of the many denoising procedures designed for Gaussian noise (not necessarily a wavelet-based one), and then transforming back to obtain an estimate of the original signal. For Poisson data, Anscombe (1948) proposed a square-root transform which induces exact asymptotic normality and stabilizes the variance. No such transform exists for scaled $\chi^2$ data, although the log transform stabilizes the variance of scaled $\chi^2$ exactly, and the cube root transform (Wilson and Hilferty (1931)) brings the data close to Gaussian but has no variance-stabilizing effect. All these transforms are, up to constants, special cases of the Box-Cox transformation (Box and Cox (1964)). Ruppert (2001) is a recommendable review article on variance-stabilizing transforms.

In many cases, the underlying noise distribution is not known and the appropriate variance-stabilizing transformation has to be estimated from the data. Examples of such data-driven transformations include the AVAS technique of Tibshirani (1988), the ACE method of Breiman and Friedman (1985), as well as the procedure of Linton et al. (1997).

Fryzlewicz and Nason (2004) proposed a Haar-Fisz (HF) transform for Gaussianising and stabilizing the variance of sequences of Poisson counts. The HF transform is performed in linear computational time as a computationally straightforward modification of the Discrete Haar Transform (Mallat (1989)). Unlike the Anscombe transform, the HF is not diagonal and has a multiscale structure. The authors also proposed a Poisson intensity estimation algorithm based on the HF transform and showed that it outperformed other state-of-the-art methods. Jansen (2003) extended that idea to other families of wavelets. Fryzlewicz and Nason (2005) proposed a similar HF transform for $\chi^2$ data.

In this paper, we combine the above ideas and propose a fully automatic Haar-Fisz method for (approximately) Gaussianising and stabilizing the variance of sequences of nonnegative independent
variables whose variance is a non-decreasing (but otherwise unknown) function of the mean. We later call this non-decreasing function the *link* function, \( h \). To avoid a possible notational clash with Fryzlewicz and Nason (2004) and Fryzlewicz and Nason (2005), we call our automatic multiscale transform the *Data-Driven Haar-Fisz transform* (DDHFT).

The paper is organised as follows. Section 2 describes the general model setup and describes the type of data that our method is likely to be suitable for. Section 3 is a reminder of the discrete Haar transform which forms the basis of the Data-Driven Haar-Fisz transform with the link function \( h \) known (Section 4) and unknown (Section 5). Section 6 reports simulation results which show that the denoising algorithms based on the DDHFT is nearly as good as (and occasionally better than) those based on the regular Haar-Fisz transforms when applied to Poisson and \( \chi^2 \) noise. Section 7 shows how an important solar flux data set with a fascinating mean-variance relationship can be successfully denoised. Section 8 concludes.

2. General Model Setup

In this section, we describe the probabilistic structure of input vectors to the Data-Driven Haar-Fisz transform (DDHFT), and give examples of distributions which satisfy these requirements. Let \( X = (X_i)_{i=1}^n \) denote an input vector to the DDHFT. The following list specifies the generic distributional properties of \( X \).

(a) The length \( n \) of \( X \) must be a power of two. We denote \( J = \log_2(n) \).
(b) \( (X_i)_{i=1}^n \) must be a sequence of independent, nonnegative random variables with finite positive means \( \mu_i = \mathbb{E}(X_i) > 0 \) and finite positive variances \( \sigma_i^2 = \text{Var}(X_i) > 0 \).
(c) The variance \( \sigma_i^2 \) must be a non-decreasing function of the mean \( \mu_i \): \[ \sigma_i^2 = h(\mu_i), \tag{1} \]

where the function \( h \) is independent of \( i \).

Note that the condition (a) above can be relaxed: even if \( n \) is not a power of two, we can still implement the DDHFT using the lifting scheme proposed by Sweldens (1996, 1997). We now give two examples of well-known probability distributions which satisfy the above requirements.

**Poisson.** Let \( X_i \sim \text{Pois}(\lambda_i) \). We have \( \mu_i = \lambda_i \) and \( \sigma_i^2 = \lambda_i \), which gives \( h(x) = x \).

**Gamma.** Let \( X_i \sim \text{Gamma}(\lambda_i, \beta_i) \). We have \( \mu_i = \beta_i/\lambda_i \) and \( \sigma_i^2 = \beta_i/\lambda_i^2 \). This yields two natural possibilities:

(a) \( h(x) = x^2/\beta_i \), hence \( \beta_i \) must be independent of \( i \). As an example, consider \( X_i = \alpha_i (Z_1^2 + \ldots + Z_m^2) \) where \( \alpha_i > 0 \) and \( Z_j \) i.i.d. \( N(0, 1) \). Then

\[ X_i \sim \text{Gamma}(1/(2\alpha_i), m/2) \]

and, for a fixed \( m, \beta_i \) is independent of \( i \) whereas \( \lambda_i \) is allowed to vary along the signal.

(b) \( h(x) = x/\lambda_i \), hence \( \lambda_i \) must be independent of \( i \). Consider now

\[ X_i = \alpha (Z_1^2 + \ldots + Z_m^2) \]

where \( \alpha > 0 \) and \( Z_j \) i.i.d. \( N(0, 1) \). Then \( X_i \sim \text{Gamma}(1/(2\alpha), m_i/2) \) and, for a fixed \( \alpha, \lambda_i \) is independent of \( i \) while \( \beta_i \) is allowed to vary along the signal.
Our motivation for restricting the link function $h$ to be non-decreasing can be summarised as follows:

- As demonstrated above, for the two canonical examples of Poisson and $\chi^2$ data, the true $h$ is increasing.

- For a number of measuring devices (for example Charge Coupled Devices, which form the basis of modern image sensors and can be found in both domestic video cameras and advanced astronomical instruments, see e.g. Janesick (2001)), we can expect a larger noise variance for observations $X_i$ which are larger in value (i.e. have a larger mean).

- With a shape restriction of this form, it is natural to estimate $h$ via isotone regression, which is a fully automatic procedure, i.e. no smoothing parameters (such as bandwidth) need to be supplied by the user. See Section 5 for details of the estimation procedure.

In the above examples, the exact form of $h$ was known. However, in many practical situations it is unknown and needs to be estimated from the data. Sections 4 and 5 below describe the Haar-Fisz transform in the cases where $h$ is known and unknown, respectively.

### 3. The Haar transform

Before we move on to describe the Haar-Fisz transform and its data-driven version, we briefly explain the Discrete Haar Transform (DHT). The Haar-Fisz transform will arise as a computationally straightforward modification of the DHT. The DHT is a linear orthogonal transform $\mathbb{R}^n \to \mathbb{R}^n$ where $n = 2^J$. Given an input vector $X = (X_i)_{i=1}^n$, the DHT is performed as follows:

(a) Let $s^0_i = X_i$, for $i = 1, \ldots, n$.

(b) For each $j = J - 1, J - 2, \ldots, 0$, recursively form vectors $s^j$ and $d^j$:

\[
\begin{align*}
    s^j_k &= \frac{s^{j+1}_{2k-1} + s^{j+1}_{2k}}{2} \\
    d^j_k &= \frac{s^{j+1}_{2k-1} - s^{j+1}_{2k}}{2}
\end{align*}
\]

for $k = 1, \ldots, 2^j$.

(c) The operator $\mathcal{H}$, where $\mathcal{H}(X) = (s^0, d^0, \ldots, d^{J-1})$, defines the DHT.

The Inverse DHT is performed as follows:

(a) For each $j = 0, 1, \ldots, J - 1$, recursively form $s^{j+1}$:

\[
\begin{align*}
    s^{j+1}_{2k-1} &= s^j_k + d^j_k \\
    s^{j+1}_{2k} &= s^j_k - d^j_k
\end{align*}
\]

for $k = 1, \ldots, 2^j$.

(b) Set $X_i = s^J_i$, for $i = 1, \ldots, n$.

The elements of $s^j$ and $d^j$ have a simple interpretation: they can be thought of as smooth and detail (respectively) of the original vector $X$ at scale $2^j$. As an example, assume $n = 8$ and consider
the reconstruction of $X_1$:  
\[
X_1 = s_1^0 + d_1^0 = s_1^0 + d_1^1 + d_1^2 = s_1^0 + d_1^1 + d_1^2 = \frac{\sum_{i=1}^{s_1} X_i}{8} + \frac{\sum_{i=1}^{4} X_i - \sum_{i=5}^{8} X_i}{8} + \frac{X_1 + X_2 - X_3 - X_4}{4} + \frac{X_1 - X_2}{2}.
\]  
(2)

Noting the pattern of operations in the above formula will make it easier to understand the origin of the more complicated formulae for the (Data-Driven) Haar-Fisz transform which we shall describe later.

4. The Haar-Fisz transform with $h$ known

In this section, we introduce the Haar-Fisz (HF) transform: a multiscale algorithm for (approximately) stabilizing the variance of $X$ and bringing its distribution closer to normality. For the time being, we assume that the function $h$ (see formula (1)) is known. In Section 5, we propose a method for estimating it from the data.

The main idea of the HF transform is to decompose $X$ using the Discrete Haar Transform (DHT), then “Gaussianise” the coefficients $d_j^k$ and stabilize their variance, and then apply the Inverse DHT to obtain a vector which is closer to Gaussianity and has its variance approximately stabilized. We first describe the middle step: the “Gaussianisation” of $d_j^k$, As a motivating example, consider $d_1^{J-1}$ and $d_1^{J-2}$. Later, we will describe the complete algorithm in its generality.

We first turn to $d_1^{J-1}$. Recall that $d_1^{J-1} = (X_1 - X_2)/2$. To facilitate understanding, the reader is invited to think of a situation where $(\mu_i)_{i=1}^n$ is a piecewise constant sequence with a small number of jumps, and the entire distribution of $X_i$ depends only on $\mu_i$ (as is the case, for example, with Poisson-distributed signals). We now assume that the distributions of $X_1$ and $X_2$ are identical (which is indeed likely in the piecewise-constant setup described above). This implies that the distribution of $d_1^{J-1}$ is symmetric around zero. Our aim is to stabilize the variance of $d_1^{J-1}$ around $2^{-J} = 2^{J-1-J} = 1/2$. A natural way to achieve this is to divide $d_1^{J-1}$ by $2^{1/2}$ times its own standard deviation. We have

\[
\text{Var}(d_1^{J-1}) = 1/4 (\text{Var}(X_1) + \text{Var}(X_2)) = \sigma_1^2/2,
\]

which gives $2^{1/2} (\text{Var}(d_1^{J-1}))^{1/2} = \sigma_1 = h_1/\mu_1$. Of course in practice $\mu_1$ is unknown and we propose to estimate it locally by $\hat{\mu}_1 = (X_1 + X_2)/2 = s_1^{J-1}$. Thus, we obtain an approximately variance-stabilized coefficient $f_1^{J-1}$:

$$f_1^{J-1} = \frac{d_1^{J-1}}{h_1^{1/2} (s_1^{J-1})},$$

with the convention $0/0 = 0$. In a different context, Fisz (1955) showed that for some well-known distributions of $X_1, X_2$ (such as Poisson or Gamma), the random variable $f_1^{J-1}$ converges in distribution to $N(0, 1/2)$ in a certain asymptotic regime. On the other hand, if the distributions of $X_1$ and $X_2$ are distinct, the distribution of $f_1^{J-1}$ deviates from $N(0, 1/2)$ and the coefficient $f_1^{J-1}$ carries significant information, and not only pure “noise”.
The relation \( Y \sim P \). Fryzlewicz, V. Delouille and G.P. Nason introduce the algorithm for the Haar-Fisz transform where the function deviation. We have \( 2 \) identical, then using symmetry arguments it can readily be shown that \( f_1^{J-2} \) and \( f_1^{J-2} \) are exactly uncorrelated.

The asymptotic distribution of random variables of a form similar to \( f_k^J \) was studied by Fisz (1955): hence we label \( f_k^J \) the Fisz coefficients of \( X \). Motivated by the above discussion, we now introduce the algorithm for the Haar-Fisz transform where the function \( h \) is known. Given the input vector \( X \), the algorithm proceeds as follows:

(a) Let \( s_i^J = X_i \), for \( i = 1, \ldots, n \).

(b) For each \( j = J - 1, J - 2, \ldots, 0 \), recursively form vectors \( s^J \) and \( f^J \):

\[
\begin{align*}
s_k^J &= \frac{s_{2k-1}^J + s_{2k}^J}{2} \\
f_k^J &= \frac{s_{2k-1}^J - s_{2k}^J}{2h^{1/2} (s_k^J)}
\end{align*}
\]

for \( k = 1, \ldots, 2^J \), with the convention \( 0/0 = 0 \).

(c) For each \( j = 0, 1, \ldots, J - 1 \), recursively modify \( s^{j+1} \):

\[
\begin{align*}
s_{2k-1}^{j+1} &= s_k^J + f_k^J \\
s_{2k}^{j+1} &= s_k^J - f_k^J
\end{align*}
\]

for \( k = 1, \ldots, 2^J \).

(d) Set \( Y = s^J \).

The relation \( Y = F_h X \) defines a nonlinear, invertible operator \( F_h \) which we call the Haar-Fisz transform (of \( X \)) with link function \( h \).

As an example, we look again at the case \( n = 8 \) and consider \( Y_1 \):

\[
Y_1 = \frac{\sum_{i=1}^{8} X_i}{8} + \frac{\sum_{i=1}^{4} X_i - \sum_{i=5}^{8} X_i}{8h^{1/2} \left( \sum_{i=1}^{8} X_i \right)} + \frac{X_1 + X_2 - X_3 - X_4}{4h^{1/2} \left( \sum_{i=1}^{4} X_i \right)} + \frac{X_1 - X_2}{2h^{1/2} \left( \sum_{i=1}^{4} X_i \right)}
\]

(3)

To gain a better understanding of the above pattern the reader might find it useful to compare (3) to the "resolution of identity" in formula (2). Each term in (3) applies the suitable division which Gaussianises and variance-stabilizes each individual Haar coefficient, and then the final coefficient summation induces further Gaussianisation.
5. The Haar-Fisz transform with \( h \) unknown

As we already mentioned before, in practice \( \hat{h} \) is often unknown and needs to be estimated from the data. Since \( \sigma_i^2 = h(\mu_i) \), ideally we would wish to be able to estimate \( h \) by computing the empirical variances of \( X_1, X_2, \ldots \) at points \( \mu_1, \mu_2, \ldots \), respectively, and then smoothing the observations to obtain an estimate of \( h \). Suppose for the time being that \( \mu_i \)'s are known and, as an illustrative example, consider \( \mu_i = \mu_{i+1} \) (recall the piecewise constant setup evoked before). The empirical variance of \( X_i \) can be pre-estimated, for example, as

\[
\hat{\sigma}_i^2 = \frac{(X_i - X_{i+1})^2}{2}.
\]

It is easily seen that, given that \( \sigma_i^2 = \sigma_{i+1}^2 \), we have \( \mathbb{E}(\hat{\sigma}_i^2) = \sigma_i^2 \), so that on any piecewise constant stretch, our pre-estimate is exactly unbiased.

The above discussion motivates the following regression setup:

\[
\hat{\sigma}_i^2 = h(\mu_i) + \varepsilon_i,
\]

where \( \varepsilon_i = \hat{\sigma}_i^2 - \sigma_i^2 = (X_i - X_{i+1})^2/2 - \sigma_i^2 \) and “in most cases” \( \mathbb{E}(\varepsilon_i) = 0 \). Of course, in practice \( \mu_i \)'s are not known and, since we pre-estimate the variance of \( X_i \) using \( X_i \) and \( X_{i+1} \), we analogously pre-estimate \( \mu_i \) by

\[
\hat{\mu}_i = \frac{X_i + X_{i+1}}{2}.
\]

Note that for each \( k = 1, \ldots, 2^J - 1 \), we have \( \hat{\mu}_{2k-1} = s_k^{J-1} \) and \( \hat{\sigma}_{2k-1}^2 = 2(d_k^{J-1})^2 \), which leads us to our final regression setup

\[
2(d_k^{J-1})^2 = h(s_k^{J-1}) + \varepsilon_k.
\]

In other words, we estimate \( h \) from the finest-scale Haar smooth and detail coefficients of \( (X_i)_{i=1}^n \), where the smooth coefficients serve as pre-estimates of \( \mu_i \), and the squared detail coefficients serve as pre-estimates of \( \sigma_i^2 \).

Because we restrict \( \hat{h} \) to be a non-decreasing function of \( \mu \), we choose to estimate it from the regression problem (4) via least-squares isotone regression, using the automatic “pool-adjacent-violators” algorithm described in detail in Johnstone and Silverman (2005a), Section 6.3. The resulting estimate, denoted here by \( \hat{h} \), is a non-decreasing, piecewise constant function of \( \mu \).

The DDHFT is performed as in Section 4 except that \( h \) is replaced by \( \hat{h} \). In the following example, we exhibit the performance of our DDHFT for Poisson and \( \chi^2 \) noise.

- **Poisson noise.** The left column in Figure 1 shows results for a Poisson-contaminated signal.

  The top plot shows a simulated Poisson vector \( X_t \) whose underlying intensity is Donoho’s Blocks function sampled at 1024 equispaced points and scaled to have a minimum (maximum) of 3 (25). The plot underneath shows the estimate \( h(\mu) \) (solid line), estimated for \( X_t \) from the regression problem (4) via least-squares isotone regression. The dotted line is the true \( h(\mu) \) function: recall that for Poisson data, we have \( h(\mu) = \mu \). The next plot down the column shows the DDHFT of \( X_t \): the variance of the noise is now clearly well-stabilized. For comparison, the bottom plot shows the Anscombe square-root transform of \( X_t \), scaled to have the same vertical range as the DDHFT of \( X_t \). Visually, there is little to choose between the two, however one must remember that the Anscombe transform is specifically designed for Poisson noise whereas the DDHFT “does not know” the nature of the noise and needs to estimate some of its characteristics (namely, the link function \( h(\mu) \)) from the data. The analysis of the noise (not shown) of the DDHF-transformed vector reveals that no spurious autocorrelation is induced.
Fig. 1. Blocks function contaminated by Poisson (left) and $\chi^2$ (right) noise. The second row shows the link function $h$ (dotted line) and its estimate (continuous line). The third row shows the DDHFT of both signals. For comparison, the bottom plots show the Poisson data stabilized via the Anscombe transform (left) and the $\chi^2$ data stabilized via the log-transform (right). See text for more details.
• $\chi^2$ noise. The right column in Figure 1 shows results for independent scaled $\chi^2$ noise with two degrees of freedom (which is similar to the periodogram setting). The series is denoted by $Y_t$. The underlying mean of the data is again Donoho’s Blocks function sampled at 1024 equispaced points and scaled to have a minimum (maximum) of 6 (50). The plot underneath shows the estimate $h(\mu)$ (solid line) and the true $h(\mu)$: in this setting, $h(\mu) = \mu^2$. The next two plots show the DDHFT of $Y_t$ and the log transform of $Y_t$, scaled to have the same range. The noise in the DDHFT of $Y_t$ appears to be more symmetric, and the shape of the underlying signal is clearer. Again, the analysis of the noise (not shown) of the DDHF-transformed vector reveals that no spurious autocorrelation is present.

6. Signal denoising using the DDHFT

Fryzlewicz and Nason (2004) demonstrated how to use the Haar-Fisz transform for estimating the intensity of an inhomogeneous Poisson sequence (as described in Section 1). The general idea is to Haar-Fisz transform the “noisy” sequence, which produces a “signal+noise” representation with approximately constant variance and a distribution which is closer to Gaussian. A suitable denoiser can then be applied to estimate the signal and remove the noise. Generally speaking, since the (transformed) noise has reasonably constant variance and is “nearly Gaussian”, many simple smoothers perform very well. The final step is to take the smooth and perform the inverse Haar-Fisz transform to obtain the desired estimate. Such a “HF-denoise-inverse HF” strategy can also make use of the DDHFT.

In this section, we compare intensity estimation results for Poisson data using (a) the Haar-Fisz transform suitable for Poisson data (with $h(\mu) = \mu$), and (b) the DDHFT. We use exactly the same denoising methodology and error measure as Fryzlewicz and Nason (2004). In summary, Fryzlewicz and Nason (2004) repeated the simulation study of Timmermann and Nowak (1999), which created “noisy” data by taking one of the Doppler, Blocks, Bumps or Heavisine signals from Donoho and Johnstone (1994), scaling them to have (minimum, maximum) intensity of $(1/8, 8)$ or $(1/128, 128)$, and then sampling 1024 independent Poisson replicates with mean equal to the intensity values at $i/1024$ for $i = 0, \ldots, 1023$. Fryzlewicz and Nason (2004) demonstrated the superiority of the Haar-Fisz algorithm over existing methods.

Table 1 shows the (scaled) Mean Integrated Square Errors averaged over 100 simulated noisy signals. The DDHFT-based algorithm is hardly worse than the Haar-Fisz algorithm, even though the link function is being estimated by DDHFT.

A similar experiment was carried out for $\chi^2$ variation. Here the Timmermann and Nowak (1999) intensity functions were multiplied pointwise by independent $\chi^2$ random variables and denoised by both a fixed Haar-Fisz transform designed for $\chi^2$ data (with $h(\mu) = \mu^2$) and the DDHFT. Note that as the setup is now multiplicative (i.e. two noisy vectors corresponding to the two scalings are exactly multiples of each other), it is sufficient to investigate the performance for one intensity scaling. The results are also shown in Table 1. Overall, apart from the Bumps intensity, the DDHFT produces results which are up to 6% worse than the specific Haar-Fisz transform. However, for Bumps, the DDHFT does about 9% better.

7. Analysis of GOES8 XRS X-ray data

This section considers denoising of a time series obtained from the XRS X-ray sensor instrument aboard the GOES8 satellite, see Bornmann et al. (1996). This sensor provides warning that a solar-flare has occurred. This information is then used to predict a solar-terrestrial disturbance at the
Table 1. Scaled MISE for Poisson intensity estimation (Poisson) and local variance estimation for Gaussian data ($\chi^2$): comparing Haar-Fisz with Data-Driven Haar-Fisz. Smoothing used Daubechies’ least asymmetric wavelets with 10 vanishing moments, hard thresholding, universal threshold with variances computed by MAD and applied to scale level 3 and finer. All estimates are averaged over 50 circular shifts of the data.

<table>
<thead>
<tr>
<th>Intensity</th>
<th>Poisson Peak intensity=8</th>
<th>Poisson Peak intensity=128</th>
<th>$\chi^2$ Peak intensity=8</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>HFT</td>
<td>DDHFT</td>
<td>HFT</td>
</tr>
<tr>
<td>Doppler</td>
<td>94</td>
<td>97</td>
<td>12</td>
</tr>
<tr>
<td>Blocks</td>
<td>287</td>
<td>290</td>
<td>31</td>
</tr>
<tr>
<td>Heavisine</td>
<td>39</td>
<td>40</td>
<td>6</td>
</tr>
<tr>
<td>Bumps</td>
<td>1243</td>
<td>1423</td>
<td>144</td>
</tr>
</tbody>
</table>

Earth. Such a prediction is essential since the effects of large flares can be dramatic: they can disrupt communications, navigation systems, satellites and even knock out power grids.

The series that we choose to analyse are $2^{14} = 16384$ observations recorded from 10am to midnight on 9th February 2001. They represent measurements for the whole Sun X-ray fluxes in the 0.1–0.8nm wavelength band.

The time series, denoted by $X_t$, is plotted in the top plot of Figure 2. The series is visibly noisy although the variance of the noise is not large. It is of interest to solar physicists both to remove the noise before proceeding with their analyses, and to understand the noise structure itself. In the remaining part of this section, we explain why some traditional approaches fail in denoising the signal, and then advocate the use of our DDHFT for this purpose.

7.1. Traditional approaches

Wavelet smoothing of the raw data. As the underlying signal appears “spiky”, it is natural to attempt to denoise it by means of a wavelet-based method. Here, we make a preliminary attempt at denoising the signal using the translation-invariant version of the standard hard thresholding estimator with the universal threshold, as described in Donoho and Johnstone (1994) and Coifman and Donoho (1995). We use Daubechies’ (1992) Extremal Phase wavelet with two vanishing moments (DEP2): the signal roughly resembles the Bumps signal (Donoho and Johnstone (1994)) and the DEP2 wavelet is known to perform well on that signal, see e.g. Fryzlewicz (2005). The above denoising procedure will be referred to as UNIV below. The resulting estimate (displayed on part of the domain only, for clarity) is shown in the bottom plot of Figure 2.

In the region shown, the UNIV estimate is extremely noisy. The reason for this poor performance is that UNIV (like most classical wavelet denoising techniques) assumes that the variance of the noise is constant over the support of the signal. This certainly does not hold for $X_t$: from visual inspection, it is apparent that the level of the noise between the 21st and 22nd hour is higher than that between, say, the 18th and 19th hour. Some extensions of classical wavelet denoising to inhomogeneous variance have been developed, but they are still very preliminary: most of them only use the universal threshold, and require a pre-estimate of the noise level, where the question of the choice of the pre-estimation method, and its parameters, arises, see e.g. von Sachs and MacGibbon (2000). In contrast to this methodology, our approach based on the DDHFT is (a) completely automatic, i.e. no smoothing parameters need to be supplied by the user, and (b) modular, i.e. it can make use of any denoising procedure intended for homogeneous Gaussian data.
Fig. 2. Top: Solar X-ray flux $X_t$ recorded by GOES on February 9, 2001. Units are Wm$^{-2}$. Bottom: fragment of $X_t$ denoised by means of the UNIV procedure, see text for discussion.
Fig. 3. $X_t$ transformed via the AVAS (top) and ACE (bottom) variance-stabilizing transforms.
Classical variance stabilization techniques. To avoid having to pre-estimate the time-varying noise level (which is an inherently difficult task as it requires knowledge of the noise which the analyst obviously does not have), an alternative is to apply a “data-driven” (automatic) variance stabilization technique, such as the AVAS or ACE techniques mentioned in the Introduction. Both are implemented in the popular statistical package S-Plus as avas and ace, respectively. Figure 3 exhibits the performance of AVAS and ACE on $X_t$. It is clear that neither procedure does a good job here: the variance of the noise in the transformed signals still varies from one region to another.

7.2. Approach via DDHFT

In this section, we combine our DDHFT algorithm with a wavelet denoising method. First, we estimate the link function from the data set $\{X_t\}$. The top left plot of Figure 4 shows the pairs $(s_k^{d-1}, 2(d_k^{d-1})^2)$ plotted on a square-root scale for clarity, where $s_k^j$ and $d_k^j$ are the empirical Haar smooth (detail) coefficients of $X_t$ (see formula (4)). The top right plot shows the estimated function $\hat{h}(\mu)$, again plotted on a square-root scale. The sharp step around $\mu \approx 1.2 \times 10^{-6}$ seems to indicate (at least) two noise regimes: one with a lower variance for $\mu < 1.2 \times 10^{-6}$, and the other with a higher variance for $\mu > 1.2 \times 10^{-6}$. The middle left plot shows $F_{\hat{h}}X_t$: the DDHFT of $X_t$ computed using the estimated link function $\hat{h}$. Note that unlike in AVAS or ACE, the variance of the noise in $F_{\hat{h}}X_t$ appears to be constant over time.

This observation is confirmed when $F_{\hat{h}}X_t$ is denoised: the middle right plot shows the empirical residuals obtained from denoising $F_{\hat{h}}X_t$ by means of the UNIV procedure described in Section 7.1 (except that, for speed, the DWT-based, rather than the translation-invariant, version of UNIV was used). The variance of the empirical residuals appears to be constant over time. We denote the denoised version of $F_{\hat{h}}X_t$ by $\tilde{F}_{\hat{h}}X_t$.

The last step of the denoising procedure is to take the inverse DDHFT transform of $\tilde{F}_{\hat{h}}X_t$ to obtain the final estimate of $\hat{h}X_t$, denoted here by $F^{-1}_{\hat{h}}\tilde{F}_{\hat{h}}X_t$. The estimate $F^{-1}_{\hat{h}}\tilde{F}_{\hat{h}}X_t$ is shown in the bottom left plot of Figure 4: for comparison with the UNIV estimate from the bottom plot of Figure 2, only part of the time domain is displayed. The noise-free character of $F^{-1}_{\hat{h}}\tilde{F}_{\hat{h}}X_t$, compared to UNIV, is remarkable.

Finally, the bottom right plot on Figure 4 shows the residuals $X_t - F^{-1}_{\hat{h}}\tilde{F}_{\hat{h}}X_t$ (solid line) and the plot of the function $10^{-7}I(X_t < 1.2 \times 10^{-6})$, where $I(\cdot)$ is the indicator function (dotted line). The plot clearly confirms our earlier observation that (at least) two noise regimes are present in $X_t$: one with a lower variance for $\mu < 1.2 \times 10^{-6}$, and the other with a higher variance for $\mu > 1.2 \times 10^{-6}$. Also, the fact that the residuals oscillate around zero, shows the apparent lack of bias in $F^{-1}_{\hat{h}}\tilde{F}_{\hat{h}}X_t$.

8. Conclusions and future work

This article introduced the Data-Driven Haar-Fisz transform for approximately Gaussianising and stabilizing the variance of a sequence of nonnegative independent variables whose variance is a non-decreasing (but otherwise unknown) function of the mean. The DDHFT performs nearly as well as the fixed Haar-Fisz transforms for Poisson and $\chi^2$ noise (even though it does not know the mean-variance relationship). The method seems to work very well compared with existing methods (AVAS and ACE) and on important data such as that obtained through the GOES8 XRS X-ray sensor.

A great deal has been written in the statistics literature about the new paradigm of vast quantities of data collected by automatic systems. However, for many of these sensors (some of which work by collecting photons) the mean-variance relationship is usually not of the ‘signal+noise’ kind,
Fig. 4. DDHFT of GOES data $X_t$. Top plots show the scaled moduli of finest-scale Haar detail coefficients against the finest-scale Haar smooth coefficients (left) and the estimated link function $h$ (right). Middle plots: DDHFT of $X_t$ (left) and empirical residuals when the DDHFT of $X_t$ is denoised (right). Bottom plots: final estimate of $E(X_t)$ (left) and its residuals (right). Units for top plots ($x$ and $y$ axes), and bottom plots ($y$ axes) are W m$^{-2}$. See text for more details.
nor is it likely to be Gaussian nor have constant variance. Some important examples are to be found in astronomy (sensing over wide ranges of the electromagnetic spectrum), bioinformatics (in microarrays, for example), low-light vision in security and defence applications. In many of these cases it is likely that the use of the DDHFT to Gaussianise and stabilise variance will help improve and ease the analysis of these kinds of data.

Software implementing the DDHFT can be obtained upon request from the first author.

References


