Empirical Bayes shrinkage, and denoising of Poisson and heteroskedastic Gaussian signals (?)

1 Abstract

2 Introduction

Shrinkage and sparsity play a key role in many areas of modern statistics, including, for example, high-dimensional regression [21], covariance or precision matrix estimation [2], multiple testing [**?**] and signal denoising [7, 8]. One attractive way to achieve shrinkage and sparsity is via Bayesian or Empirical Bayes (EB) methods (e.g. [**?**, **?**, **?**, 4, 13]). However, these methods are usually perceived to require context-specific implementations, and this overhead can limit their use in practice. Here we consider a flexible EB approach to shrinkage, which we call *adaptive shrinkage*, whose goal is to provide a generic shrinkage method that could be useful for a range of applications. We show how this single shrinkage method can produce effective results for several denoising problems, including smoothing Gaussian means in the presence of heteroskedastic variance, smoothing of Gaussian variances, and smoothing Poisson means. These are all settings that are relatively underserved by existing EB implementations, and indeed we are unaware of any existing EB implementation for smoothing either the mean or the variance in the heteroskedastic Gaussian case. Consistent with previous studies ([1]) we find these EB methods to be more accurate than commonly-used thresholding rules, and, in the Poisson case, competitive with a purpose-built EB method.

The *adaptive shrinkage* (`ash`) method is described in detail in a companion paper, [**?**], which applies `ash` to estimate false discovery rates (FDR) in multiple testing settings. Here we apply the same method to a different problem, signal denoising. In brief, `ash` aims to provide EB shrinkage estimates of quantities $\beta_1, \ldots, \beta_J$, given observed estimates of these quantities $\hat{\beta}_1, \ldots, \hat{\beta}_J$, and corresponding standard errors $\hat{s}_1, \ldots, \hat{s}_J$. In common with most EB methods `ash` assumes that the β_j come from some common underlying distribution, g, which is estimated from all the data (as $\hat{g}$ say), and uses the posterior mean, $E(\beta_j|\hat{\beta}_j, \hat{s}_j, \hat{g})$ as a shrinkage estimate of β_j. Key features that make `ash` attractive as a flexible and generic approach to shrinkage

include: i) its flexible semi-parametric modelling of g, under the constraint that it be uni-modal at 0 (and, optionally, symmetric); ii) its incorporation of the precision of each measurement through the standard errors $\hat{s}_j$, so that the amount of shrinkage adapts to the precision of each measurement; iii) its computational simplicity and speed: e.g. our `R` implementation typically takes a fraction of a second for $J = 1000$; iv) its computational stability, with, for example, minimal problems due to convergence to local optima [**?**].

The `ash` method makes assumptions, particularly conditional independence of observations, and normality of likelihood $p(\hat{\beta}_j|\beta_j, \hat{s}_j)$, that will often be violated in practice, and indeed are violated in our applications here. However, as we demonstrate later, it can nonetheless produce good shrinkage estimates. Essentially, we treat `ash` as a generic or "black box" shrinkage procedure, which inputs estimates and their corresponding standard errors and outputs shrinkage estimates, without worrying too much about the details. Of course, one must be careful about how far one takes this. And in more complex applications (e.g. large-scale regression and covariance estimation) it is unclear whether this strategy can be usefully applied. However, we believe our results here for signal denoising, together with those in [**?**] for FDR estimation, illustrate `ash`'s potential and flexibility.

Shrinkage methods are widely-used in signal denoising applications, because signal denoising can be accurately and conveniently achieved by shrinkage in a transformed (e.g. wavelet) domain [8]. Commonly-used shrinkage methods include both simple thresholding rules [5, 7, 8] and EB methods [4, 13]. Being an EB method, `ash` has much in common with these previous EB methods, but generalizes them in two ways: first, `ash` allows more flexibility in the underlying distribution g than the Laplace or spike-and-slab distributions used in [4, 13]; second `ash` allows for variations in precision in the transformed observations (e.g. wavelet coefficients). The latter property is particularly important for the Poisson and heteroskedastic Gaussian settings we consider here.

Software implementations of our methods are available in the R packages `ashr` (Adaptive SHrinkage in R) and `smash` (SMoothing by Adaptive SHrinkage), available from http://www.github.com/stephens999/ashr and http://www.github.com/stephenslab/smash respectively.

Move some of this material to the relevant place? or discussion?

The first problem relates to mean and variance estimation with Gaussian noise. Although the framework for homoskedastic Gaussian errors has been thoroughly developed in the wavelet literature [references], methods dealing with heteroskedastic errors are uncommon. Fan & Yao (1998) estimated the variance by smoothing the squared residuals using local polynomial smoothing, while Brown & Levine (2007) employed difference-based kernel estimators. Making use of the local adaptivity of wavelet methods, Cai & Wang (2008) improved upon previous variance estimation methods using a wavelet thresholding approach on first order differences. Here we present an approach that estimates both the mean and variance accurately by incorporating ASH into a novel wavelet denoising framework. [mention the lack of available

software?]

The second task is to denoise a Poisson distributed signal. This often occurs in the experimental sciences such as gamma-ray burst signals in astronomy [references], and more recently in genetics where high throughput sequencing data is of interest. This problem is interesting because there are many distinct ways to perform signal recovery. Variance stabilizing techniques together with normal approximations have been proposed by Donoho (1993) and Fryzlewicz & Nason (2001), by exploiting the mean-variance relationship of a Poisson distribution. Kolaczyk (1997, 1999a) derived thresholds achieving optimal asymptotic properties in the context of wavelet transformations, similar to the thresholds in the i.i.d. Gaussian case. However, variance stabilizing methods are computationally inefficient due to the presence of external cycle-spinning, and threshold-based methods may not result in satisfactory finite sample performance. Furthermore, both these methods are quite sensitive to the choice of the primary resolution level as our simulations will show. Multiscale analysis using recursive dyadic partitions within a Bayesian framework was later developed by Kolaczyk (1999b) to make use of a particular form of likelihood factorization, but a relatively inflexible conjugate prior was chosen. We will improve upon the prior (and likelihood) specification by using ASH as the main shrinkage procedure in a Bayesian framework similar to that of Kolaczyk (1999b).

3 Methods

3.1 Adaptive shrinkage

Here we briefly outline the adaptive shrinkage method; see [**?**] for full details. Adaptive shrinkage (`ash`) is an EB method for estimating quantities $\beta = (\beta_1, \ldots, \beta_n)$ from noisy estimates $\hat{\beta} = (\hat{\beta}_1, \ldots, \hat{\beta}_n)$ and their corresponding standard errors $\hat{s} = (\hat{s}_1, \ldots, \hat{s}_n)$. In its simpest form it assumes the hierarchical model

$$\beta_j \,|\, \hat{s}_j \sim g \tag{1}$$

$$\hat{\beta}_j \,|\, \beta_j, \hat{s}_j \sim N(\beta_j, \hat{s}_j^2), \tag{2}$$

where the distribution g is constrained to be unimodal and symmetric. This constraint on g can be flexibly achieved using a mixture of zero-centered normal distributions

$$g(\cdot) = \sum_{k=0}^{K} \pi_0 N(\cdot; 0, \sigma_k^2), \tag{3}$$

where the mixture weights $\pi_0, \ldots, \pi_K$ are non-negative and sum to 1, and $N(\cdot; \mu, \sigma^2)$ denotes the density of a normal distribution with mean μ and variance σ^2. A key idea, which substantially simplies inference, is to take $\sigma_0, \ldots, \sigma_K$ to be a fixed grid of values. ranging from very small (e.g. $\sigma_0 = 0$, in which case g includes a point mass at 0) to very large. Estimating g then boils down to estimating the mixture weights

π, which is done by maximum likelihood using a simple EM algorithm. Given an estimate $\hat{g}$ for g, the conditional distributions $p(\beta_j \mid \hat{\beta}, \hat{s}, \hat{g})$ are analytically tractable, and the posterior mean $E(\beta_j \mid \hat{\beta}, \hat{s}, \hat{g})$ provides a shrinkage point estimate for β_j.

[**?**] also introduces various embellishments that are implemented in the `ashr` package, including generalizing the normal likelihood to a t likelihood, and dropping the symmetric constraint on g by replacing the mixture of normals with a more flexible (though less smooth) mixture of uniforms. However, we do not use these embellishments here. CHECK: IS THIS TRUE. WHAT PARAMETERS DO WE USE? We also do not use the penalty term on the likelihood introduced in [**?**] for FDR applications, because it is unnecessary in this context, and has little effect on the shrinkage estimates.

3.2 Signal Denoising

A common generic signal denoising problem, sometimes known as "non-parametric regression", involves estimating a "spatially-structured" mean $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_T)$, from corresponding noisy observations $\boldsymbol{Y} = (Y_1, \ldots, Y_T)$, where $t = 1, \ldots, T$ indexes location in a one-dimensional space, such as time, or, as in our example later, location along the genome. By "spatially-structured" we mean that μ_t will often be similar to μ'_t for small $|t - t'|$, though we do not rule out occasional abrupt changes in μ. For convenience we assume that $T = 2^J$ for some integer J, as is common when using wavelet methods.

The most studied denoising problem is the homoskedastic Gaussian case; that is, the data Y_t have Gaussian noise and constant variance. Here we consider the more general case of Gaussian data with spatially-structured mean *and* spatially-structured (non-constant) variance. In some settings the changes in variance may themselves be of interest, and our methods provide explicit estimates for the variance as well as the mean. In addition we consider Poisson data (where the variance depends on the mean, so a spatially-structure mean implies spatially-structure variance). To build up to these more interesting cases we start with the simplest setting: Gaussian noise with known variance.

3.2.1 Gaussian noise; estimate mean (known variance)

Suppose the Y_t are independent noisy observations of the μ_t, with Gaussian noise of known variance σ_t^2. That is,

$$\boldsymbol{Y} = \boldsymbol{\mu} + \boldsymbol{\epsilon} \tag{4}$$

where $\boldsymbol{\epsilon} \sim N(0, D)$ with D the diagonal matrix with diagonal entries $(\sigma_1^2, \ldots, \sigma_T^2)$.

Wavelet denoising involves first applying a discrete wavelet transform to the data $\boldsymbol{Y}$. This involves pre-multiplying $\boldsymbol{Y}$ by an orthogonal $n \times n$ matrix W that depends on the orthonormal wavelet basis chosen. Pre-multiplying (4) by W yields

$$W\boldsymbol{Y} = W\boldsymbol{\mu} + W\boldsymbol{\epsilon} \tag{5}$$

which, using ˜ to denote the wavelet transform, we write

$$\tilde{\boldsymbol{Y}} = \tilde{\boldsymbol{\mu}} + \tilde{\boldsymbol{\epsilon}}, \tag{6}$$

where $\tilde{\boldsymbol{\epsilon}} \sim N(0, WDW')$.

A key feature of the wavelet tranform is that if $\boldsymbol{\mu}$ is spatially structured then many elements of $\tilde{\boldsymbol{\mu}} = W\boldsymbol{\mu}$ will tend to be close to zero, and vice versa. In other words, smoothing in the data domain corresponds to shrinking in the wavelet domain. Thus, one can obtain a spatially structured estimate of mean $\boldsymbol{\mu}$ by first using shrinkage methods to estimate $\tilde{\boldsymbol{\mu}}$ and then reversing the wavelet transform.

Here we use the adaptive shrinkage method to obtain shrinkage estimates for $\tilde{\boldsymbol{\mu}}$. Focussing on the marginals of (6), we have

$$\tilde{Y}_j | \tilde{\mu}_j, s_j^2 \sim N(\tilde{\mu}_j, \omega_j^2). \tag{7}$$

where

$$\omega_j^2 := \sum_{t=1}^{T} \sigma_t^2 W_{jt}^2, \tag{8}$$

are the diagonal elements of WDW'. Thus, applying ash with $\hat{\beta}_j = \tilde{Y}_j$ and $\hat{s}_j = \omega_j$ yields shrinkage estimates for the $\tilde{\mu}_j$. (In practice it is important to group the wavelet-transformed observations j by their resolution level before shrinking; see note below.) Of course, by focusing on the marginals we are ignoring any correlations among the $\tilde{Y}_j$, and this is the primary simplification here. (We are not alone in making this simplification; see also [20] for example.)

The above outlines the basic strategy, but there are some important additional implementational details:

- Rather than use a single wavelet transforrm, we use the "non-decimated" wavelet transform, which treats the observations as coming from a circle, rather than a line, and averages results over all T possible rotations of the data. Although not always necessary, this is a standard trick to reduce artifacts that can occur near discontinuities in the underlying signal (see eg. [5]). COMMENT: WHat about reflection before doing this? DO OTHER EB APPROACHES CONSIDER THE NDWT? ALSO, POINT FOR DISCUSSION: NDWT INCREASES DEPENDENCE, SO MAKE INDEPENDENCE ASSUMPTION WORSE, BUT MUCH BETTER TO DO IT THAN NOT TO DO IT!
- The non-decimated wavelet transform yields T wavelet coefficients (transformed values of $\boldsymbol{Y}$) at each of $J = \log_2(T)$ resolution levels. We apply ash separately to the wavelet coefficients at each resolution level, so that a different distribution g for the $(\tilde{\mu}_j)$ is estimated for each resolution. This is the usual way that EB approaches are applied in this context (e.g. [**?**]) and indeed is crucial because the underlying distribution g will vary with resolution (because smoothness of $\boldsymbol{\mu}$ will vary with resolution).

- Although we have presented the wavelet transform as a matrix multiplication, which is an $o(T^2)$ operation, in practice both the wavelet transform and the inverse tranform are implemented using Mallat's pyramid algorithm, taking only $T\log(T)$ operations. Do we need $o(T\log T)$ or $O()$? What about the TI table? Is there a different name for the algorithm for the reverse transform?

3.2.2 Estimating spatially structured variance (known mean)

Although we assumed variances to be known above, the problem of variance estimation is itself non-trivial. A common approach is to assume that the variance is constant ($\sigma_t = \sigma$), and that changes in the adjacent means $\mu_t - \mu_{t+1}$ are negligible, so

$$\hat{\sigma} = (1/T)\sum_{t=1}^{T}(Y_t - Y_{t-1})^2 \tag{9}$$

(with $Y_0 := Y_T$) provides an unbiased estimate for σ. ...CHECK WITH TOM Here we make the more flexible assumption that the variance function is spatially structured, and use wavelet shrinkage to estimate it.

Assume initially that the mean μ_t is known, and define

$$Z_t^2 = (Y_t - \mu_t)_t^2 \tag{10}$$

to be the "observations" for the unknown variance function. Note that $\mathbb{E}(Z_t^2) = \sigma_t^2$, and so we have a mean estimation problem. To tackle this we use the mean estimation procedure above (effectively treating the wavelet-transformed values WZ_t^2 as Gaussian when really they are linear combinations of χ^2 random variables). To apply this procedure we need the variance of Z_t^2, $\mathbb{V}(Z_t^2)$, which is unknown. Here we use $\frac{2}{3}Z_t^4$ as an approximately (CHECK) unbiased estimator for $\mathbb{V}(Z_t^2)$. (This follows from the distributional result that, when $Z^2 \sim \sigma^2\chi^2$, then $\mathbb{E}(Z^4) \approx 3\sigma^4$, and $\mathbb{V}(Z^2) \approx 2\sigma^4$.)

Despite the approximations being made here, we have found this procedure to work well in practice in most cases, perhaps with a tendancy to oversmooth quickly-varying variance functions. (IS THIS A FAIR SUMMARY? DO WE REALLY KNOW IT IS OVERSMOOTHING? IS IT POSSIBLE TO DO BETTER?)

This approach is similar in spirit to [3], but they use first order differences instead of the squared residuals, removing the need to jointly estimate the mean and variance. WHY DID WE DO THIS? DID IT PERFORM BETTER? How do they deal with the unknown variance?

One way around this issue is to employ a procedure that jointly shrinks the coefficients $\boldsymbol{\gamma}$ and their variance estimates (see JASH). LET'S TALK MORE ABOUT THAT...

3.2.3 Estimating spatially-structured mean and variance

Having specified procedures for estimating the means with variances known, and the variances with means known, we iterate these procedures to deal with the (typical)

case where both are unknown. We initialize the algorithm by estimating the variance vector $\boldsymbol{\sigma}^2$ using

$$\hat{\sigma}_t^2 = \frac{1}{2}\left((Y_t - Y_{t-1})^2 + (Y_t - Y_{t+1})^2\right) \tag{11}$$

where $Y_0 \equiv Y_n$ and $Y_{T+1} \equiv Y_1$ (equivalent to putting the observations on a circle). IS THIS as in (eg Cai & Wang (2008)).? ALSO IS THIS CORRECT? I REMOVED A SUM OVER T THAT TOM HAD. ALSO, DID WE TRY USING (**??**))? Then we iterate:

1 Estimate $\boldsymbol{\mu}$ as if $\boldsymbol{\sigma}^2$ is known (with the value obtained from the previous step).
2 Estimate $\boldsymbol{\sigma}^2$ as if $\boldsymbol{\mu}$ is known (with the value obtained by the previous step); return to 1.

We cannot guarantee that this procedure will converge in general. In our simulations we found that two iterations of steps 1-2 yielded accurate results (so the full procedure consists of initialize + Steps 1-2-1-2).

3.3 Smoothing Poisson data

Now assume that each Y_t has a Poisson distribution with mean μ_t. To do denoising here we apply adaptive shrinkage to the Poisson multiscale models from [17] and [22], which are an analogue of wavelet methods for Poisson data.

To explain the idea, first recall the following elementary distributional result: if Y_1, Y_2 are independent, with $Y_j \sim \text{Poi}(\mu_j)$ then

$$Y_1 + Y_2 \sim \text{Poi}(\mu_1 + \mu_2) \tag{12}$$
$$Y_1|(Y_1 + Y_2) \sim \text{Bin}(Y_1 + Y_2, \mu_1/(\mu_1 + \mu_2)). \tag{13}$$

To extend this to $T = 4$, introduce the notation $v_{i:j}$ to denote, for any vector v, the sum $\sum_{t=i}^{t=j} v_t$. Then

$$Y_{1:4} \sim \text{Poi}(\mu_{1:4}) \tag{14}$$
$$Y_{1:2}|Y_{1:4} \sim \text{Bin}(Y_{1:4}, \mu_{1:2}/\mu_{1:4}) \tag{15}$$
$$Y_1|Y_{1:2} \sim \text{Bin}(Y_{1:2}, \mu_1/\mu_{1:2}) \tag{16}$$
$$Y_3|Y_{3:4} \sim \text{Bin}(Y_{3:4}, \mu_3/\mu_{3:4}). \tag{17}$$

Together these models are exactly equivalent to $Y_j \sim \text{Poi}(\mu_j)$, and they decompose the overall distribution $Y_1, \ldots, Y_4$ into parts involving aspects of the data at increasing resolution: (14) represents the coarsest resolution (the sum of all data points), whereas (16) and (17) represent the finest resolution, with (15) in between. Further, this representation suggests a reparameterization, from $\boldsymbol{\mu} = (\mu_1, \mu_2, \mu_3, \mu_4)$ to $\mu_{1:4}$ plus the binomial parameters $\boldsymbol{p} = (\mu_{1:2}/\mu_{1:4}, \mu_1/\mu_{1:2}, \mu_3/\mu_{3:4})$, where p_1 controls lower-resolution changes in the mean vector μ and p_2, p_3 control higher resolution changes.

This idea extends naturally to $T = 2^J$ for any J, reparameterizing $\boldsymbol{\mu}$ into its sum $\mu_{1:T}$ and a vector $\boldsymbol{p}$ of $T-1$ binomial probabilities that capture features of $\boldsymbol{\mu}$ at different resolutions. This can be thought of as an analogue of the wavelet transform of $\boldsymbol{\mu}$ for Poisson data.

Note that, in this reparameterization, $p_j = 0.5 \, \forall j$ corresponds to the case of a constant mean vector, and values of p_j far from 0.5 correspond to large changes in μ (at some scale). Thus estimating a spatially-structured $\boldsymbol{\mu}$ can be achieved by shrinkage estimation of $\boldsymbol{p}$, with shrinkage towards $p_j = 0.5$. Both [17] and [22] use purpose-built Bayesian models to achieve this shrinkage, by introducing a prior distribution on elements of $\boldsymbol{p}$ that is a mixture of a point mass at 0.5 (creating shrinkage toward 0.5) and a Beta distribution. Here we take a different approach, reparameterizing $\alpha_j = \log(p_j/(1-p_j))$, and then using adaptive shrinkage to shrink α_j towards 0.

To apply adaptive shrinkage we need an estimate $\hat{\alpha}_j$ and corresponding standard error $\hat{s}_j$ for each j. This boils down to estimating a log-odds ratio, and its standard error, which is a well-studied problem (e.g. [12]). The main challenge is in dealing satisfactorily with cases where the mle for α_j is infinite. Our choice of estimates, based on results from [12], are described in Appendix **??**.

The output of adaptive shrinkage is a posterior distribution on each α_j. The simplest approach to obtain estimates of $\boldsymbol{\mu}$ would be to estimate α_j by its posterior mean, and then reverse the reparameterization. The resulting estimate of μ_t would be the exponential of the posterior mean for $\log(\mu_t)$ (because each $\log_(\mu_t)$ is a linear combination of α_j). [IS THIS TRUE?] An alternative is to estimate the posterior mean using the Delta method; see Appendix B.

4 Results

4.1 Illustration

Idea of adaptive shrinkage.

By estimating the uni-modal distribution from the data, ash uses the data to determine how sparse $\tilde{\boldsymbol{\mu}}$ should be, and therefore how spatially structured $\boldsymbol{\mu}$ should be. Thus the degree of spatial smoothing is adaptive to the dataset: more smoothing for datasets that appear consistent with a smooth signal.

4.2 Simulations

We now seek to validate the advantages of our method, named "SMASH", via simulation studies. To thoroughly investigate the performance of SMASH, we consider the Gaussian and Poisson cases separately.

For the Gaussian case, we focus our attention mostly on mean estimation. Different test functions, sample sizes, signal-to-noise ratios (SNR) and variance functions were considered, to ensure that SMASH behaves well in a variety of settings. However, only a small portion of the results will be shown here to highlight key findings

from the rather extensive simulation study. Full results from the study can be found in the Supplementary Materials.

We first present a simple illustration of our method as a signal denoising technique. Figure 1 demonstrates that SMASH is able to better capture the true signal than TI-thresholding (with variance estimated by running median absolute deviation (RMAD); see [11]) on a simple simulated signal.

To demonstrate the robustness of SMASH, we now consider only homoskedastic errors. Due to the large number of methods available in the literature for this setting, we chose only a subset. The methods we considered in our simulation are discussed in detail in [1], and in particular Translation Invariant (TI) thresholding by [5] performed the best for most of the test signals, where performance is measured by mean squared error (MSE). We also considered the popular Empirical Bayes shrinkage procedure by [13], called Ebayesthresh. Figure 2 compares the mean integrated squared errors (MISEs) of TI-thresholding and Ebayesthresh to SMASH with two different options for the "Spikes" mean function, with a signal to noise ratio of 3 and sample size 1024. The default option in SMASH estimates the mean without any homoskedastic assumptions on the variance function, while the second options estimates the mean given the true variance function. We can clearly see that the performance of SMASH is not impacted by the lack of homoskedastic assumptions imposed on the variance function: it outperforms two of the most accurate wavelet denoising algorithms for i.i.d. Gaussian noise even when the i.i.d. assumption holds, and performs nearly as well as when the variance function is known, which is the best-case scenario for any denoising technique. These results also extend to other mean functions, SNRs and sample sizes (see Supplementary Materials for more details).

Having shown that SMASH performs well in the homoskedastic case without any explicit assumptions on the Gaussian noise, we now demonstrate its performance gain when the errors are heteroskedastic. For the sake of presentation, figure 3 highlights only the results from SMASH, TI-thresh (with various choices of variance estimation) and EbayesThresh for just two sets of test functions: the "Spikes" mean function with the "Clipped Blocks" variance function and the "Corner" mean function with "Doppler" variance function, both with SNRs of 3 and sample sizes of 1024. Nevertheless, similar results hold in general for different mean and variance functions, SNRs, and sample sizes (see Supplementary Materials for more details).

Several facts are immediately clear from figure 3:

1. SMASH does better than all TI-thresh variants, including the case when the true variance is provided, demonstrating the excellent finite sample performance of SMASH.
2. using the variance estimate from SMASH as an input to TI-thresh improves the accuracy of TI-thresh substantially, compared with using RMAD to estimate the variance as in [11]
3. accounting for heteroskedasticity allows SMASH to vastly outperform methods which assume homoskedastic errors.

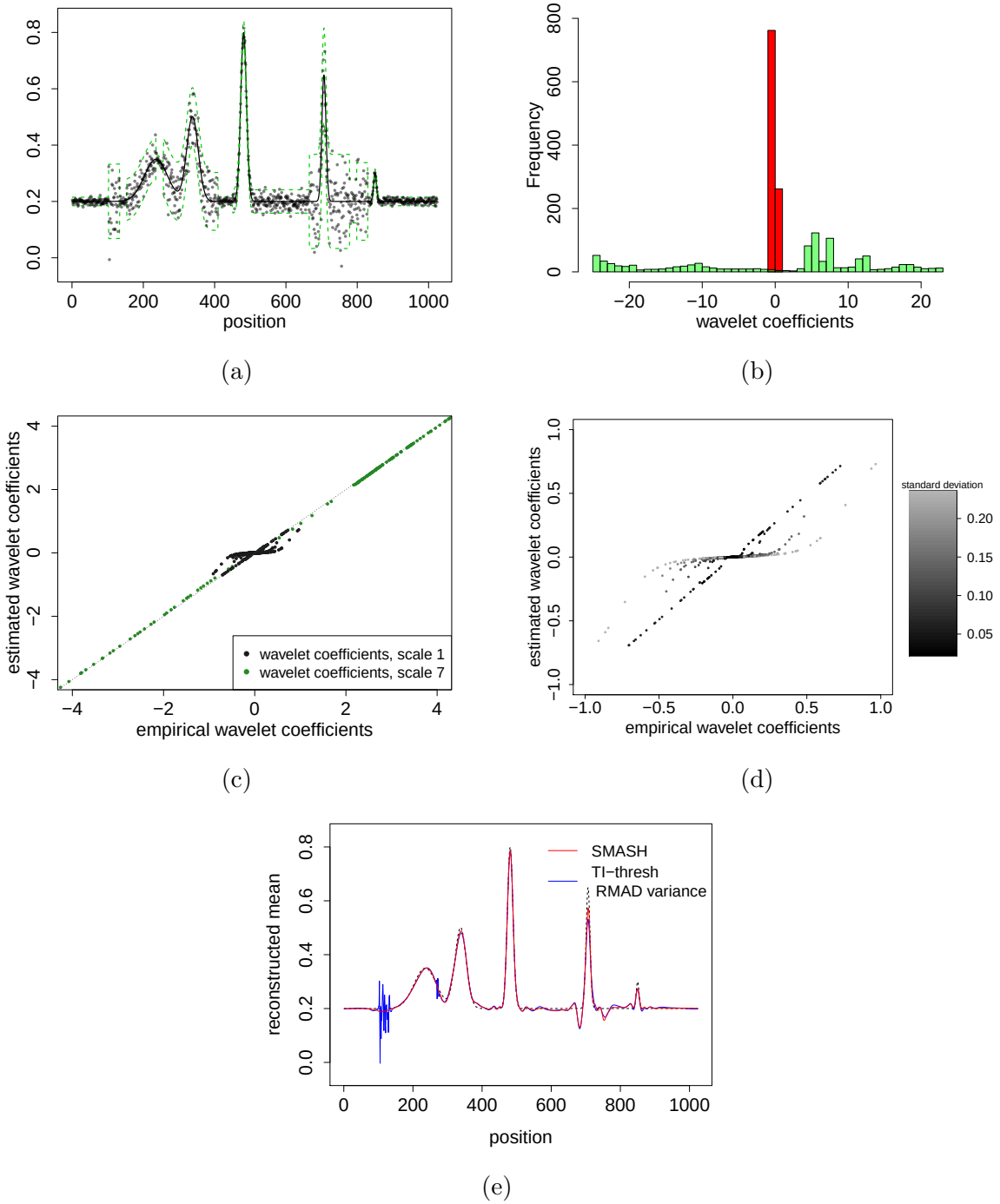


Figure 1: Simple illustration of the advantages of SMASH over TI-thresholding under heteroskedastic Gaussian errors. Top left plot shows the mean function, top right plot shows the variance function, bottom left plot shows a sample dataset, and bottom right plot shows the estimated mean functions from SMASH and TI-thresh against the true mean function.

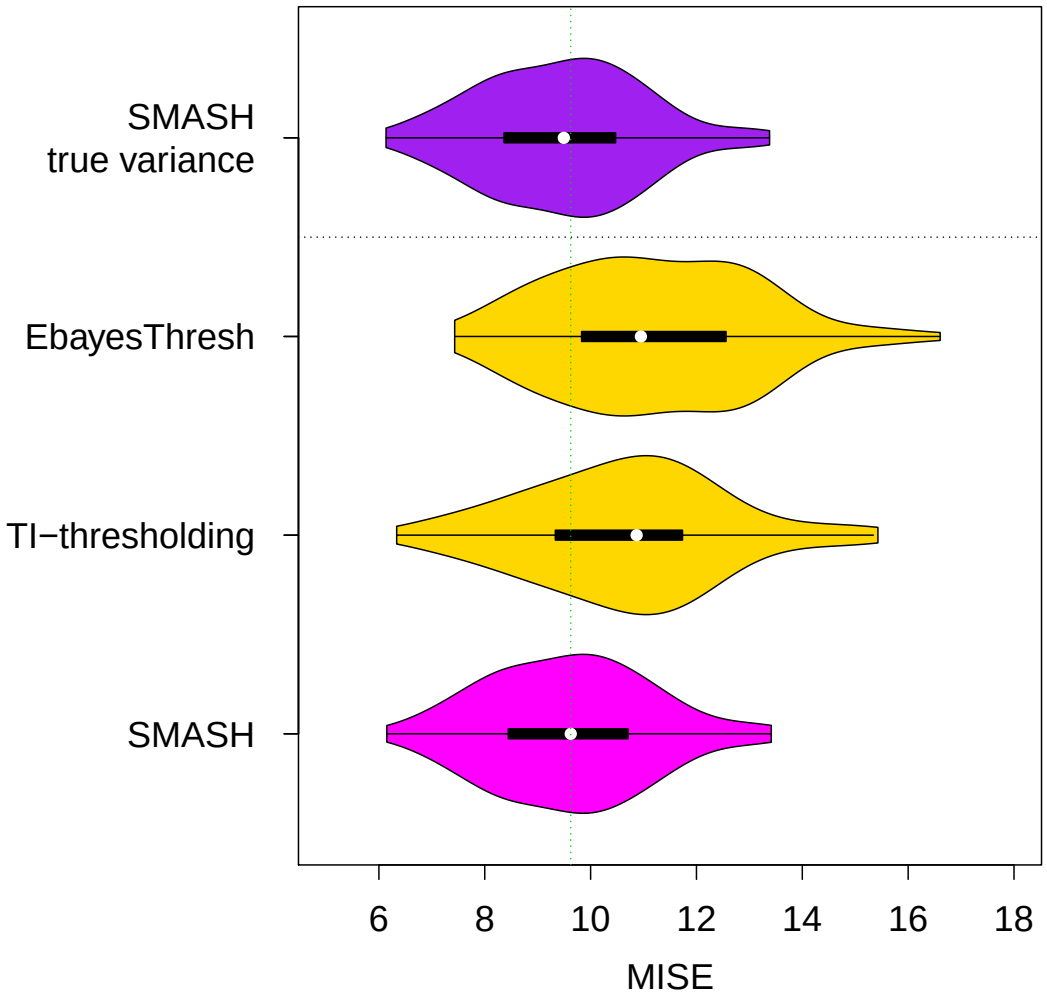
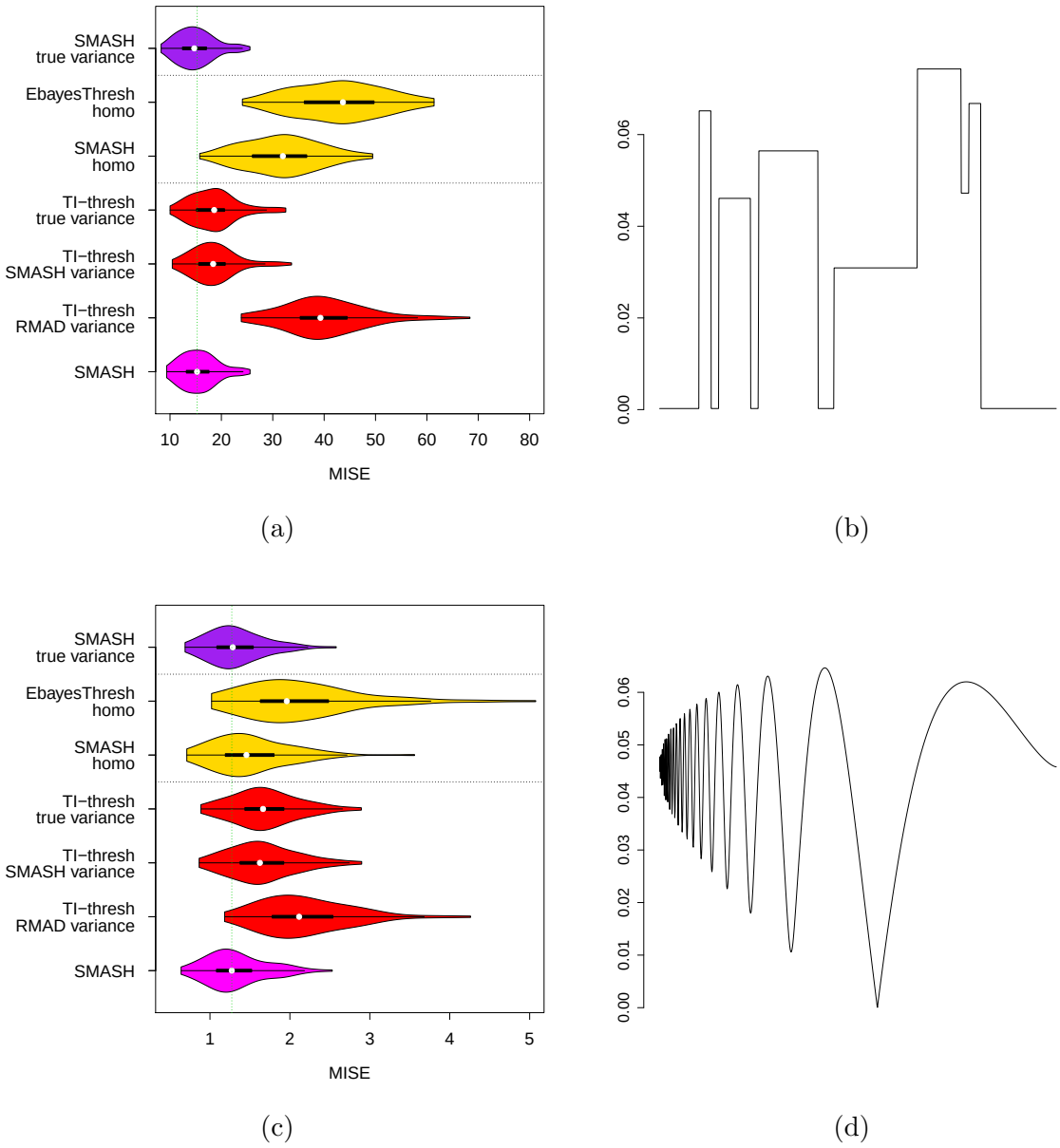


Figure 2: Violin plots of MISEs for various methods in denoising data with homoskedastic Gaussian errors for the "Spikes" mean function. Smaller MISE implies better performance; dashed green line indicates the median MISE for SMASH. a) demonstrates the performance gain of SMASH over two popular and accurate denoising methods: TI-thresholding and Ebayesthresh, and b) shows that SMASH with variance estimated can perform nearly as well as when it is given the true variance.



(a) (b)

(c) (d)

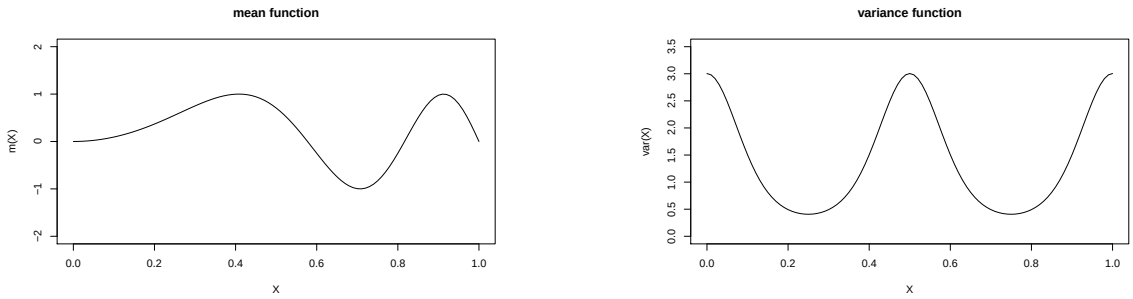
Figure 3: Violin plots of MISEs for various methods in denoising data with homoskedastic Gaussian errors for two sets of mean-variance functions: "Spikes" mean function with "Clipped Blocks" variance function, and "Corner" mean function with "Doppler" variance function. Smaller MISE implies better performance; dashed green line indicates the median MISE for SMASH. a) and c) demonstrates the performance gain of SMASH over TI-thresh with different variance options: variance estimated by RMAD, variance estimated by SMASH, and the true variance. b) and d) plots the associated.

4. empirical Bayes methods are more robust to violations of homoskedasticity compared with TI-thresholding.

It should also be noted that providing the true variance to SMASH does not result in substantial gains compared to the case when SMASH estimates its own variance function for most of the test functions (results shown in the supplementary materials). Hence, we can reasonably assume that SMASH does a good job of variance estimation, although this claim relies heavily on the variance function at hand. For example, the Bumps variance function (see appendix ?) is extremely difficult to estimate, and SMASH will perform much better when provided with the true variance function.

Since our method also provides variance estimates, one could potentially perform similar assessments for different variance functions as with mean functions. In this particular study, we compare our method against the only joint mean and variance estimation procedure for which software is easily available. Specifically, we consider the Mean Field Variational Bayes (MFVB) methodology developed by [15] for heteroskedastic Gaussian regression. Here we note that the MFVB approach is not well suited for dealing with standard test functions in the wavelet literature, as it is based on penalized splines (simulation studies were conducted to confirm this observation; results not shown). Hence, we chose the smooth mean and standard deviation functions (A) from [15] as our test functions, plotted in Figure 4 and denoted by $m(\cdot)$ and $sd(\cdot)$ respectively. We then considered two different simulation scenarios:

1. $n = 500$ independent (X_i, Y_i) pairs were generated. The X_i's were distributed as Uniform(0,1), and the Y_i's were distributed as $N(m(x_i), sd(x_i))$. The performance of a given method is measured by the standard MSE evaluated at 201 equally spaced points on (X_{min}, X_{max}) for both the mean and the standard deviation functions.
2. $n = 1024$ independent (X_i, Y_i) pairs were generated. The X_i's were deterministic and equally spaced on (0,1), while the Y_i's were distributed as $N(m(x_i), sd(x_i))$. The performance of a given method is measured by the MSE evaluated at the 1024 X_i's for both the mean and the standard deviation functions.



(a) (b)

Figure 4: Mean (a) and variance (b) functions used in the simulation study comparing SMASH against MFVB.

In the first scenario, the number of data points is not a power of two, nor are the point equally spaced. To deal with these complications, we adapted and modified the standard symmetric extension procedure commonly used in wavelet settings. We first mirrored the data about the right edge and extract the first $2^{\lfloor \log_2(2n) \rfloor}$ sample points. This ensures that the number of data points in the new "dataset" is a power of two, and the mean curve would be continuous at the right edge. To further ensure that the input to the Gaussian denoising method is periodic, we then reflected the new dataset about the right edge and used this as the final input. To obtain the original mean and variance functions, we extracted the first n points from the outputs (mean and variance) of our denoising technique. Since the data points are not evenly spaced, we took the simplest approach and applied our method treating the observations as if they were evenly spaced. This approach is not only intuitively appealing, but can also be considered a formal treatment of unequally spaced data in traditional wavelet settings (see eg. [19]). Evaluation of MSE at the 201 equally spaced points is then based on simple linear interpolation between the estimated points. Tables 1 display the MSEs over 100 independent runs for each scenario.

	Scenario 1		Scenario 2	
	mean	sd	mean	sd
MFVB	0.0330	0.0199	0.0172	0.0085
SMASH	0.0334	0.0187	0.0158	0.0065

Table 1: MSEs of MFVB and SMASH for two simulation scenarios

Note that wavelets in general are poorly suited for dealing with the setup in Scenario 1; not only are the number of data points not a power of two, they are also not equally spaced. Also, linear interpolation between sample points was used in computing the MSE, further impacting the accuracy of SMASH. At the same time, spline-based methods such as MFVB are well suited to dealing with smooth mean and variance functions such as those used in the simulations, whereas wavelet methods can better deal with spatial inhomogeneity which are present in functions such as those presented in Figure 2. Despite all these limitations, our method performs comparably to MFVB in terms of mean estimation for the simulation scheme presented in Scenario 1, and has a lower MSE in terms of variance estimation. For Scenario 2, our method outperforms MFVB in both mean and variance estimation.

Thus far, we have demonstrated that our method does a good job of mean and variance estimation for a variety of situations in the Gaussian case. As such, we now turn our attention to the Poisson case. For this simulation study, we considered different test functions and Poisson intensities for a given sample size of $n = 1024$ as in [22] and [10], over 100 independent runs. Figure 5 compares the MISE of SMASH with Haar-Fisz ([10]) and BMSM ([14]), which are two popular and accurate Poisson denoising techniques, for the "Bursts" function with (min,max) intensities of (0.01,3) and (1/8,8). The low intensities are of primary interest in genomic applications, and so we focus our attention to these. Complete results, including a (min,max) inten-

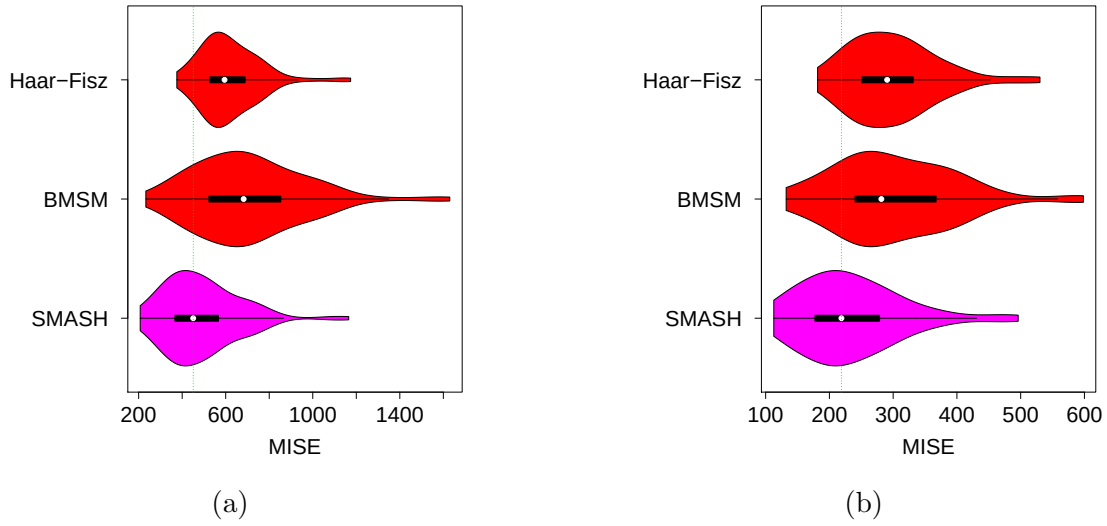


Figure 5: Violin plots of MISEs for various methods. a) corresponds to a (min,max) intensity of (0.01,3), and b) corresponds to a (min,max) intensity of (1/8,8)

sity of (1/128, 128), are included in appendix where?. From figure 5, it is clear that SMASH outperforms both methods, and the claim holds true in general. As an exception, Haar-Fisz outperforms both SMASH and BMSM for certain test functions with a (min,max) intensity of (1/128,128) when the asmyptotic variance of 1 is assumed in the Gaussian wavelet thresholding stage of the Haar-Fisz algorithm, but underperforms when the variance is estimated from the data instead. Besides the somewhat strong assumption of unit variance, the inconsistent performance of Haar-Fisz could also be attributed to the choice of the primary resolution level used, which can substantially affect the performance of Haar-Fisz in general. Here we analyzed primary resolution levels of 4, 5, 6 and 7. On the other hand, SMASH outperforms BMSM consistently, even though our method is based on the same likelihood factorization used in the latter. We can thus conclude that the choice of ASH as the shrinkage procedure (with the necessary likelihood approximations) is superior to that used in BMSM. More importantly, using this formulation in SMASH also makes it easily extensible to multiple samples in the context of a (generalized) linear model, something we will briefly describe in the Discussion section.

In terms of computation, we note that SMASH is much more efficient than Haar-Fisz with external cyclespinning, as the latter method needs to be rerun for each shift of the data. However, a direct comparison between SMASH and BMSM is uninformative, as they are coded in different programming environments. Nevertheless, the similarities between the two methods imply that they should behave identically in this aspect, barring differences in actual implementation.

Overall, these simulation studies demonstrate the ability of our method to accurately recover the mean functions for both the Gaussian and Poisson cases, and highlight the flexibility and adaptiveness of the shrinkage procedure ASH. Although

we have considered an extensive range of scenarios here, including different SNRs, sample sizes, variance functions, test functions and mean intensities where applicable, the performance of our method on real data has yet to be determined. In the next section, we will apply our method to two example datasets that have been discussed in previous work, and comment on the resulting estimates.

5 Application to real datasets

5.1 Three-month Treasury Bill Yields

In this section we apply the Gaussian and Poisson denoising techniques to one example dataset each. For the Gaussian case, we looked at yields of secondary market rates from three-month Treasure bills, which were recorded weekly on Fridays. These rates were quoted on a discount basis and annualized using a 360-day year of bank interest. To match the analysis in [9], we used 1735 weekly observations spanning Jan. 5 1962 to Mar. 31 1995. The data are plotted in Figure 6a. Similar to [9], we fit an autoregressive model of order 5 (AR(5)) to the data and obtained the following:

$$T_t = 1.228T_{t-1} - 0.234T_{t-2} + 0.028T_{t-3} + 0.039T_{t-4} - 0.066T_{t-5} + Y_t \tag{18}$$

where $T_t, t = 1, ..., 1735$ is the time series for the yields, and Y_t are the residuals from fitting the model. Figure 6b shows the plot of Y_t against $X_t \equiv T_{t-1}$. Our goal is to estimate the mean function defined by $E(Y_t|X_t = x)$ as well as the variance function $V(Y_t|X_t = x)$. Note that standard wavelet techniques are not designed for such types of data, where 1) repeated observations are present and 2) the number of data points is not a power of two and the points are unevenly spaced. To tackle the first issue, we use the median of the repeated observations at their respective sample points (see eg [6]). Next, we applied the procedure described in the Simulations section when comparing our method against MFVB, which was a modified version of symmetric extension. This deals with the second complication. The estimated mean and variance functions are given in Figures 6b and 6d respectively.

Except for possible boundary effects, our mean and conditional variance estimates are similar to those of [9]. Unfortunately, these boundary effects are difficult to deal with for non-periodic functions in the wavelet case, and our usage of symmetric extension with the Haar basis is just one possible solution. Better alternatives have been suggested by eg. Su et al. (2013), but is beyond the scope of discussion in this paper. Similar to the analysis in [9], we found the correlation coefficient between the logarithm of x_t and the logarithm $\hat{V}^{1/2}(Y_t|x_t)$ to be 0.949, which further supports the structural volatility model suggested by Andersen and Lund (source?):

$$Var^{1/2}(Y_t|x_t) = \alpha x_t^{\beta} \tag{19}$$

By performing least squares regression of $\log(\hat{V}^{1/2}(Y_t|x_t))$ on $\log(x_t)$, we have that $\hat{\alpha} = 0.0106$ and $\hat{\beta} = 1.429$, which are similar to the values reported in [9].

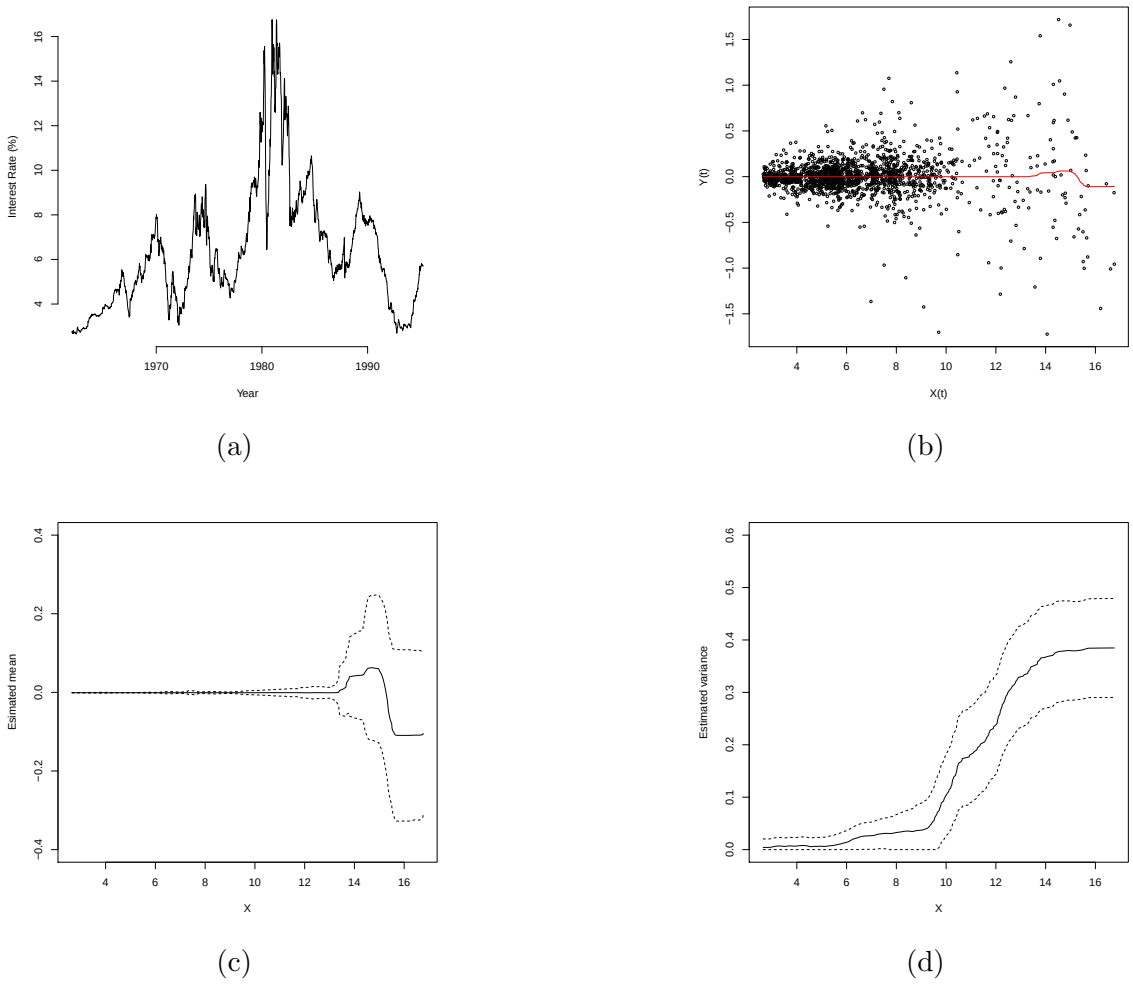


Figure 6: Analysis of yields from three-month Treasury Bills. (a) Raw data of interest rates as a function of time. (b) Residuals Y_t from fitting an AR(5) model to the data against $X_t \equiv T_{t-1}$. The red curve is the estimated mean curve from SMASH. (c) Plot of a zoomed-in version of the estimated mean curve, with approximate 95% credible bands. (d) The estimated conditional variance curve, with approximate 95% credible bands.

5.2 ChIP-Seq Data

Here we apply our Poisson de-noising procedure to next generation sequencing data, commonly seen in the field of genomics. Specifically, we chose an example dataset from the ENCODE (**Enc**yclopedia **O**f **D**NA **E**lements) project launched by the National Human Genome Research Institute. This dataset contains reads from chromatin immunoprecipitation sequencing (ChIP-seq) measuring transcription factor binding in two different cell types, with two samples for each cell type. Due to the massive size of the data, we selected a representative portion of the reads of length 2^{15} from chromosome 1. One goal of analyzing ChIP-seq data is to discover regions where transcription factors are likely to bind to DNA, thereby allowing us to better understand the mechanisms underlying gene regulation. These binding regions are often reflected in the data as "peaks", where more counts are present than background noise. Our method allows us to identify these peaks by looking at the estimated intensity function. To ensure that our method performs sensibly, we pooled the reads for the GM12878 cell line and ran our method as well as a popular peak calling procedure MACS on the selected region. The results are shown in Figure 7.

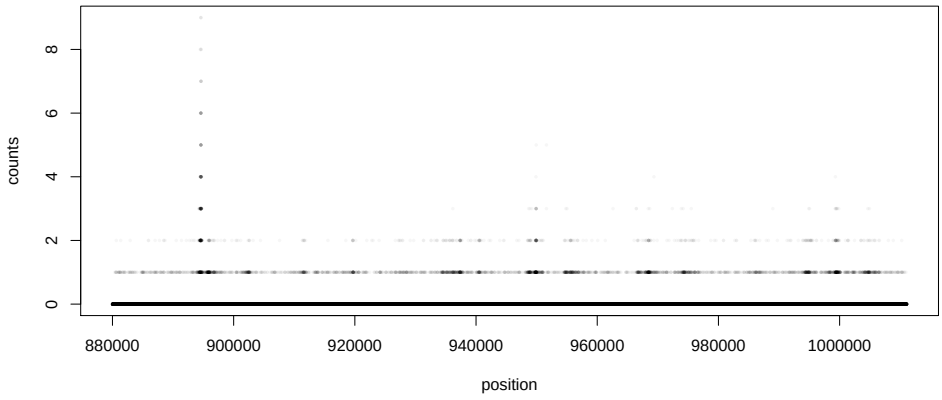
From Figure 7b it is clear that SMASH can recognize the peaks detected by MACS. Rather than calling peaks based on certain thresholds however, our method allows users to determine the relative strength and width of each peak, which provides a more comprehensive summary of the sequencing reads. At the same time, SMASH also provides the posterior variances for the intensity estimates, allowing for the option of calling peaks based on thresholds if desired.

6 Discussion

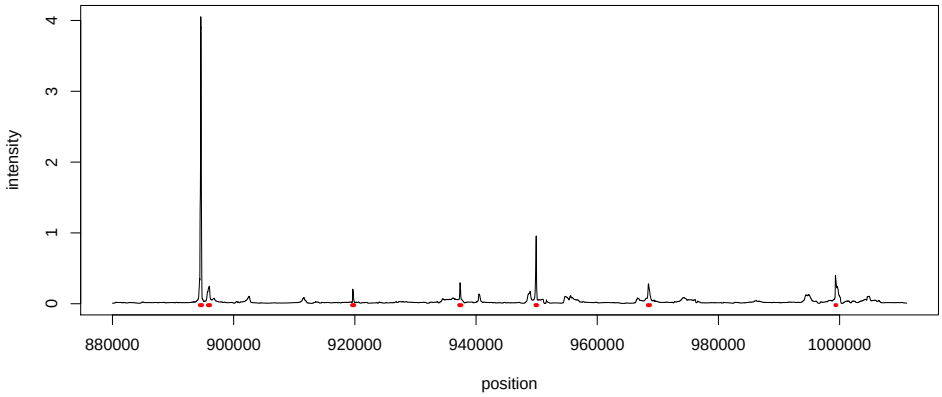
In this paper we have briefly introduced the adaptive shrinkage method ASH; while it was originally developed in the setting of FDR control for multiple comparisons, we have illustrated its usage as part of two wavelet denoising techniques. Both applications discussed in this paper relax the standard assumption of i.i.d. Gaussian noise, and are thus challenging tasks. Through these applications we are able to demonstrate the flexibility and accuracy of the shrinkage method, revealing its potential in many other applications.

In both the aforementioned applications, our software allows users to easily obtain point estimates for the mean function as well as their approximate posterior variances as a measure of uncertainty. In addition, the variance function can also be estimated, which would provide frequentist confidence intervals for other forms of mean estimation. To the best of our knowledge, there is no readily available software in the wavelet literature that implements joint mean and variance estimation. Simulations have also confirmed that our method is relatively robust to simple forms of autocorrelation between the errors (details needed). In the case of Poisson regression, we improved upon the conjugate Beta priors in [14] by using ASH as a shrinkage procedure, which allows for more flexibility and precision. In both the applications, one further advantage of both our methods is that there is no tuning parameter other than the type of wavelet basis used. On the other hand, the primary resolution level in almost all of the other wavelet-based methods actually affects their performance in varying degrees, depending on the underlying mean and/or variance function. Hence, our fully adaptive procedure allows users to easily apply it to any given dataset, depending on the type of noise.

We have also demonstrated through numerical studies that our methods mostly outperform their respective counterparts from the standard wavelet literature, in terms of pointwise accuracy (MSE in this case). Furthermore, the simplicity of the approximated Gaussian likelihoods as well as the conjugacy of the mixture Gaussian priors imply that our methods are computationally fast, since the posteriors can be computed analytically. In the Gaussian case, simulation results demonstrated that our method is competitive with standard wavelet methods in the case of i.i.d. errors (without explicitly assuming thus), whilst maintaining superior accuracy when heteroskedastic errors are present. Unfortunately, the lack of readily available software (except for MFVB, as described in [15]) for variance estimation made it difficult to



(a)



(b)

Figure 7: Analysis of reads taken from the GM12878 cell line, spanning positions 880001 to 1011072 on chromosome 1. (a) Raw sequencing counts. (b) Estimated intensity function from SMASH (black solid line) and location of peaks called by MACS (red markers beneath the estimated intensity).

assess the performance of our method in that context. On the other hand, we were able to compare our method to some of the more popular denoising techniques in the Poisson case. Specifically, we have improved upon the conjugate Beta priors used in conjunction with the binomial likelihoods [14]) by using ASH as the shrinkage procedure, which allows for more flexibility and accuracy. This is particularly evident when the mean intensity is low, as is common in many high-throughput genomic sequencing datasets. Our method is also much faster and comparable in accuracy to the popular Haar-Fisz algorithm. Unfortunately, we were not able to directly compare the computational efficiency of our method to many other methods due to differences in the programming software involved.

Although we have only focused on one-dimensional univariate denoising here, our methods can be extended to various scenarios. In the one-dimensional domain, our methods could be used in conjunction with multiple samples, otherwise known as regression analysis of functional data (see [16]). Instead of dealing with a vector of observations, we perform regression analysis on a matrix of observations, each row of which encapsulates a sample with temporally or spatially structured data points. While [16] proposed a way to solve a generic regression model, they implicitly assumed the same variance structure for each sample in the same group or category. Our work in the Gaussian case potentially allows for differing variance structures amongst all the samples, thereby relaxing their assumptions. In the simplest case, we could obtain spatially structured differences between groups by including a single covariate that categorizes each sample. In particular, the Poisson model is extremely useful for discovering regions in sequencing reads where structured differences are present between say, various cell lines, as per our sequencing example in the previous section. The Gaussian model could potentially be used in...(??).

With some work, our methods could also be extended to higher dimensions, where a wider range of applications is possible. For example, we could attempt a straight extension to the two dimensional case for both the Gaussian and the Poisson cases as described in [18]. However, recent research in image denoising problems has shown that smooth curves present in many images such as photographs might render wavelet transformations undesirable. We could thus incorporate ASH into other types of transformations such as curvelets, which would be a potential direction for future work.

7 Reference

Appendix A

Variance estimation for Gaussian denoising

With $\boldsymbol{Z}$ as defined in (10), we apply the wavelet transform W to $\boldsymbol{Z}^2$, and obtain the wavelet coefficients $\boldsymbol{\delta} = W\boldsymbol{Z}^2$. Note that $\mathbb{E}(\boldsymbol{\delta}) = (\boldsymbol{\gamma})$, where $\boldsymbol{\gamma} = W\boldsymbol{\sigma}^2$. As with (**??**), we treat the likelihood for $\boldsymbol{\gamma}$ as if it were independent, resulting in

$$L(\boldsymbol{\gamma}|\boldsymbol{\delta}) = \prod_{j=0}^{J} \prod_{k=0}^{T-1} P(\delta_{jk}|\gamma_{jk}) \tag{20}$$

However, the likelihoods $L(\gamma_{jk}|\delta_{jk})$ are not normal, and have no simple closed form expressions. As such, we approximate the likelihood by a normal likelihood through matching the moments of a normal distribution to the distribution $P(\delta_{jk}|\gamma_{jk})$ i.e.

$$P(\delta_{jk}|\gamma_{jk}) \approx N(\gamma_{jk}, \hat{\mathbb{V}}(\delta_{jk})) \tag{21}$$

so that

$$L(\gamma_{jk}|\delta_{jk}) \approx \phi(\delta_{jk}; \gamma_{jk}, \mathbb{V}(\delta_{jk})) \tag{22}$$

where ϕ is the normal density function, and $\mathbb{V}(\delta_{jk})$ is the variance of the detail coefficients. Since these variances are unknown, we estimate them from the data and then proceed to treat them as known. More specifically, since $Z_t \sim N(0, \sigma_t^2)$, we have that

$$\begin{aligned} & \mathbb{E}(Z_t^4) \approx 3\sigma_t^4 \\ \Rightarrow \quad & \mathbb{V}(Z_t^2) \approx 2\sigma_t^4 \end{aligned} \tag{23}$$

and so we simply use $\frac{2}{3}Z_t^4$ as an unbiased estimator for $\mathbb{V}(Z_t^2)$. It then follows that $\hat{\mathbb{V}}(\delta_{jk})$ is given by $\sum_{l=1}^{n} \frac{2}{3} Z_l^4 W_{jk,l}^2$, and is unbiased for $\mathbb{V}(\delta_{jk})$. These will be the inputs to ASH, which then produces shrunk estimates in the form of posterior means for the corresponding parameters. Although this works well in most cases, there are variance functions for which the above procedure tends to overshrink the detail coefficients at the finer levels. This is likely because the distribution of the wavelet coefficients are extremely skewed, especially when the true coefficients are large (at coarser levels the distributions are much less skewed since we are dealing a linear combination of a large number of data points). One way around this issue is to employ a procedure that jointly shrinks the coefficients $\boldsymbol{\gamma}$ and their variance estimates (see JASH). The final estimate of the variance function is obtained from the posterior means via the average basis inverse across all the shifts.

Appendix B

Poisson denoising

First summarize the data in a recursive manner:

$$Y_{Jk} \equiv Y_k \tag{24}$$

for $k = 1, ..., n$, and

$$Y_{jk} = Y_{j+1,2k} + Y_{j+1,2k+1} \tag{25}$$

for resolution $j = 0, ..., J-1$ and location $k = 0, ..., 2^j - 1$. Hence, we are summing more blocks of observations as we move to coarser levels.

This recursive scheme leads to:

$$Y_{jk} = \sum_{l=k2^{J-j}+1}^{(k+1)2^{J-j}} Y_l \tag{26}$$

for $j = 0, ..., J$ and $k = 0, ..., 2^j - 1$.

Further define the following:

$$\lambda_{Jk} \equiv \lambda_k \tag{27}$$

for $k = 1, ..., n$, and

$$\lambda_{jk} = \lambda_{j+1,2k} + \lambda_{j+1,2k+1} \tag{28}$$

for $j = 0, ..., J-1$ and $k = 0, ..., 2^j - 1$. Furthermore, define

$$\alpha_{jk} = \log(\lambda_{j+1,2k}) - \log(\lambda_{j+1,2k+1}) \tag{29}$$

$$\tag{30}$$

for $s = 0, ..., J-1$ and $l = 0, ..., 2^j - 1$. The α's defined this way is extremely similar to the (true) Haar wavelet coefficients, which forms the basis of our approach. Using this recursive representation, we can see that the likelihood for $\boldsymbol{\alpha}$ factorizes into a product of likelihoods, where $\boldsymbol{\alpha}$ is the vector of all the α_{sl}'s. To be specific, we have

$$\begin{aligned} L(\boldsymbol{\alpha}|\mathbf{Y}) &= P(\mathbf{Y}|\boldsymbol{\alpha}) && (31)\\ &= P(Y_{0,0}|\lambda_{0,0}) \prod_{j=0}^{J-1} \prod_{k=0}^{2^j-1} P(Y_{j+1,2k}|Y_{j,k}, \alpha_{j,k}) && (32)\\ &= L(\lambda_{0,0}|Y_{0,0}) \prod_{j=0}^{J-1} \prod_{k=0}^{2^j-1} L(\alpha_{j,k}|Y_{j+1,2k}, Y_{j,k}) && (33) \end{aligned}$$

where the factorization is due to the recursive definition above. Note that $Y_{00}|\lambda_{00} \sim \text{Pois}(\lambda_{00})$. For any given j, k, Y_{jk} is a sum of two independent Poisson random variables, and is itself a Poisson random variable. Hence

$$Y_{j+1,2k}|Y_{jk}, \alpha_{jk} \sim \text{Bin}(Y_{jk}, \frac{1}{1+e^{-\alpha_{jk}}} \equiv \frac{\lambda_{j+1,2k}}{\lambda_{jk}})$$

B.1 Estimates and standard errors for α_j

Each α_j is a ratio of the form $\log(\mu_{a:b}/\mu_{c:d})$ whose maximum likelihood estimate (mle) is $\log(Y_{a:b}/Y_{c:d})$. The main challenge here is that the mle is not well behaved when either the numerator or denominator of $Y_{a:b}/Y_{c:d}$ is 0. To deal with this, when either is 0 we use Tukey's modification [12]. Specifically, letting S denote $Y_{a:b}$ and F denote $Y_{c:d}$ (corresponding to thinking of these as successes and failures in a binomial experiment, given $Y_{a:b} + Y_{c:d}$), we use

$$\hat{\alpha} = \begin{cases} \log\{(S+0.5)/(F+0.5)\} - 0.5 & S = 0 \\ \log\{S/F\} & S = 1, 2, ..., N-1 \\ \log\{(S+0.5)/(F+0.5)\} + 0.5 & S = N \end{cases} \tag{34}$$

$$se(\hat{\alpha}) = \sqrt{V^*(\hat{\alpha}) - \frac{1}{2}\{V_3(\hat{\alpha})\}^2 \left\{V_3(\hat{\alpha}) - \frac{4}{N}\right\}} \tag{35}$$

where

$$V_3(\hat{\alpha}) = \frac{N+1}{N}\left(\frac{1}{S+1} + \frac{1}{F+1}\right) \quad S = 0, ..., N \tag{36}$$

$$V^*(\hat{\alpha}) = V_3(\hat{\alpha})\left\{1 - \frac{2}{N} + \frac{V_3(\hat{\alpha})}{2}\right\} \tag{37}$$

The square of the standard error in (35) corresponds to V^{**} from p. 182 of [12], and is chosen because it is less biased for the true variance of $\hat{\alpha}$ (when N is small) as compared to the asymptotic variance of the MLE (see [12]). The other two variance estimators from [12], V_1^{++} and V^{++}, were also considered in simulations and gave similar results, but V^{**} was chosen for its simple form.

B.2 Signal reconstruction

Given the posterior means and variances of the α's from ASH, the first step to reconstructing the signal is to find the posterior means of $p_{jk} := \frac{\lambda_{j+1,2k}}{\lambda_{jk}}$ and $q_{jk} := \frac{\lambda_{j+1,2k+1}}{\lambda_{jk}}$ (for $j = 0, ..., J-1$ and $k = 0, ..., 2^j - 1$). Specifically, for each j and k, we wish to find

$$E(p_{jk}) \equiv E\left(\frac{e^{\alpha_{jk}}}{1+e^{\alpha_{jk}}}\right) \tag{38}$$

$$E(q_{jk}) \equiv E\left(\frac{e^{-\alpha_{jk}}}{1+e^{-\alpha_{jk}}}\right) \tag{39}$$

Given that we already have the posterior expectations and variances for α_{jk}, we can approximate (38)-(39) using the Delta method. First, define

$$ff(x) = \frac{e^x}{1+e^x} \tag{40}$$

and consider the Taylor expansion of $ff(x)$ about $ff(E(x))$:

$$ff(x) \approx ff(E(x)) + ff'(E(x))(x - E(x)) + \frac{ff''(E(x))}{2}(x - E(x))^2 \tag{41}$$

where

$$ff'(x) = \frac{e^x}{(1+e^x)^2} \tag{42}$$

$$ff''(x) = \frac{e^x(1-e^x)}{(1+e^x)^3} \tag{43}$$

It is easy to see that

$$E(p_{jk}) \approx ff(E(\alpha_{jk})) + \frac{ff''(E(\alpha_{jk}))}{2} Var(\alpha_{jk}) \tag{44}$$

$$E(q_{jk}) \approx ff(-E(\alpha_{jk})) + \frac{ff''(-E(\alpha_{jk}))}{2} Var(\alpha_{jk}) \tag{45}$$

noting that we have already computed $E(\alpha)$ and $Var(\alpha)$.

Finally, we can easily back-transform to construct an estimated signal, by noting that we can express λ_t as a product of the p's and q's for any $i = 1, 2, ..., n$. Specifically, let $\{c_1, ..., c_J\}$ be the binary representation of $i - 1$, and $d_m = \sum_{j=1}^{m} c_j 2^{m-j}$ for $j = 1, ..., J-1$. We then have

$$\lambda_k = \lambda_{00} p_{00}^{1-c_1} p_{1,d_1}^{1-c_2} ... p_{J-1,d_{J-1}}^{1-c_J} q_{00}^{c_1} q_{1,d_1}^{c_2} ... q_{J-1,d_{J-1}}^{c_J} \tag{46}$$

where we usually estimate λ_{00} by $\sum_l Y_l$ (see Kolaczyk (1999)). Using the independence of the p's and q's from different scales, we have:

$$\begin{aligned} E(\lambda_t) = \lambda_{00} E(p_{00})^{1-c_1} E(p_{1,d_1})^{1-c_2} ... E(p_{J-1,d_{J-1}})^{1-c_J} \\ E(q_{00})^{c_1} E(q_{1,d_1})^{c_2} ... E(q_{J-1,d_{J-1}})^{c_J} \end{aligned} \tag{47}$$

As an additional step, we can also construct a credible band around the signal using the posterior variances for inference purposes. From (46) we have the following:

$$\begin{aligned} E(\lambda_t^2) = \lambda_{00}^2 E(p_{00}^2)^{1-c_1} E(p_{1,d_1}^2)^{1-c_2} ... E(p_{J-1,d_{J-1}}^2)^{1-c_J} \\ E(q_{00}^2)^{c_1} E(q_{1,d_1}^2)^{c_2} ... E(q_{J-1,d_{J-1}}^2)^{c_J} \end{aligned} \tag{48}$$

To compute the terms in (48), we again make use of the Delta method (with $ff(x) = (\frac{e^x}{1+e^x})^2$) to obtain:

$$\begin{aligned} E(p_{jk}^2) \approx \left(ff(E(\alpha_{jk})) + \tfrac{ff''(E(\alpha_{jk}))}{2} Var(\alpha_{jk}) \right)^2 + \\ \{ff'(E(\alpha_{jk}))\}^2 Var(\alpha_{jk}) \end{aligned} \tag{49}$$

$$\begin{aligned} E(q_{jk}^2) \approx \left(ff(-E(\alpha_{jk})) + \tfrac{ff''(-E(\alpha_{jk}))}{2} Var(\alpha_{jk}) \right)^2 + \\ \{ff'(E(-\alpha_{jk}))\}^2 Var(\alpha_{jk}) \end{aligned} \tag{50}$$

Finally we combine (47) and (48) to find $Var(\lambda_k)$, which allows us to construct credible intervals.

Note here that for the reconstructed signal to possess the property of shift invariance (see Coifman & Donoho (1995)), the α's are extracted from a so-called translation invariant (TI) table (see [5], and [**?**]) rather than as described above. The idea remains the same however, and we can simply think of the extra α's as being defined similarly as the original α's, albeit from a shifted version of the original data points. To be more specific, the TI table contains the α_{jk} for all circulant shifts of the signal. Here we define the t-th shift of the signal $\boldsymbol{Y}$, denoted by $\boldsymbol{Y}^{(t)}$, to be created from $\boldsymbol{Y}$ itself by moving the first $n-t$ elements of $\boldsymbol{Y}$ t positions to the right and then putting the last t elements of $\boldsymbol{Y}$ in the first t locations. Using this table, we are essentially computing the posterior expectations in (47)-(48) by averaging over all posterior expectations for every shift of the original signal ie.

$$\frac{1}{n}\sum_{t=1}^{n} E(\hat{\lambda}_k^{(t)}) \tag{51}$$

which is an approximation to the true quantity we wish to compute, given by

$$E(\hat{\lambda}_k) = \sum_{t=1}^{n} E(\hat{\lambda}_k^{(t)}) P(t\text{-th shift}) \tag{52}$$

References

[1] Anestis Antoniadis, Jérémie Bigot, and Theofanis Sapatinas. Wavelet Estimators in Nonparametric Regression: A Comparative Simulation Study. *Journal of Statistical Software*, 6(6), 2001. 1, 9

[2] Peter J. Bickel and Elizaveta Levina. Covariance regularization by thresholding. *The Annals of Statistics*, 36(6):2577–2604, December 2008. 1

[3] Tony T. Cai and Lie Wang. Adaptive variance function estimation in heteroscedastic nonparametric regression. *The Annals of Statistics*, 36(5):2025–2054, October 2008. 6

[4] Merlise Clyde and Edward I. George. Flexible empirical Bayes estimation for wavelets. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 62(4):681–698, January 2000. 1, 2

[5] R. R. Coifman and D. L. Donoho. Translation-invariant de-noising, 1995. 2, 5, 9, 26

[6] V. Delouille, J. Simoens, and R. von Sachs. Smooth Design-Adapted Wavelets for Nonparametric Stochastic Regression. *Journal of the American Statistical Association*, 99(467):643–658, September 2004. 16

[7] David L. Donoho and Iain M. Johnstone. Adapting to Unknown Smoothness via Wavelet Shrinkage. *Journal of the American Statistical Association*, 90(432):1200–1224, December 1995. 1, 2

[8] David L. Donoho and Jain M. Johnstone. Ideal spatial adaptation by wavelet shrinkage. *Biometrika*, 81(3):425–455, September 1994. 1, 2

[9] Jianqing Fan and Qiwei Yao. Efficient Estimation of Conditional Variance Functions in Stochastic Regression. *Biometrika*, 85(3):645–660, 1998. 16

[10] Piotr Fryzlewicz and Guy P. Nason. A Haar-Fisz Algorithm for Poisson Intensity Estimation. *Journal of Computational and Graphical Statistics*, 13(3):621–638, September 2004. 14

[11] Hong-ye Gao. Wavelet Shrinkage Estimates For Heteroscedastic Regression Models. 1997. 9

[12] J. J. Gart and J. R. Zweifel. On the bias of various estimators of the logit and its variance with application to quantal bioassay. *Biometrika*, 54(1):181–187, June 1967. 8, 24

[13] Iain M. Johnstone and Bernard W. Silverman. Empirical Bayes selection of wavelet thresholds. *The Annals of Statistics*, 33(4):1700–1752, August 2005. 1, 2, 9

[14] Eric D. Kolaczyk. Bayesian Multiscale Models for Poisson Processes. *Journal of the American Statistical Association*, 94(447):920–933, September 1999. 14, 19, 21

[15] Marianne Menictas and Matt P. Wand. Variational Inference for Heteroscedastic Semiparametric Regression. *Aust. N. Z. J. Stat.*, 57(1):119–138, March 2015. 13, 19

[16] Jeffrey S. Morris and Raymond J. Carroll. Wavelet-based functional mixed models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(2):179–199, April 2006. 21

[17] Robert D. Nowak and Eric D. Kolaczyk. A statistical multiscale framework for Poisson inverse problems. *Information Theory, IEEE Transactions on*, 46(5):1811–1825, August 2000. 7, 8

[18] RobertD Nowak. Multiscale Hidden Markov Models for Bayesian Image Analysis. In Peter Müller and Brani Vidakovic, editors, *Bayesian Inference in Wavelet-Based Models*, volume 141 of *Lecture Notes in Statistics*, pages 243–265. Springer New York, 1999. 21

[19] Sylvain Sardy, Donald B. Percival, Andrew G. Bruce, Hong-Ye Gao, and Werner Stuetzle. Wavelet shrinkage for unequally spaced data. *Statistics and Computing*, 9(1):65–75, April 1999. 14

[20] Bernard W. Silverman. Wavelets in Statistics: Beyond the Standard Assumptions. *Philosophical Transactions: Mathematical, Physical and Engineering Sciences*, 357(1760):2459–2473, 1999. 5

[21] Robert Tibshirani. Regression Shrinkage and Selection via the Lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288, 1996. 1

[22] Klaus E. Timmermann and Robert D. Nowak. Multiscale modeling and estimation of Poisson processes with application to photon-limited imaging. *Information Theory, IEEE Transactions on*, 45(3):846–862, April 1999. 7, 8, 14