Data Driven Inverse Problems
Lecture 2 : Introduction to Regularisation

Simon R. Arridge[1]

[1]Department of Computer Science, University College London, UK

Data Driven Inverse Problems
Autumn School
Sep 20^{th} – 22^{nd} 2023

Outline

Outline

- Regulsrisation by filtering
- Variational approach
- Bayesian approach
- Proximal operators
- Learned regularisation

Outline

Regularisation by filtering

Introduction

Recall the expression for the inverse of a linear operator from lecture 1 :

$$\mathrm{A}^{\dagger} g = \sum_i v_i \frac{\langle u_i, g \rangle}{w_i} \quad \Rightarrow \quad \mathrm{A}^{\dagger} = \sum_i \frac{v_i u_i^{\mathrm{T}}}{w_i} = \mathrm{V}\mathrm{W}^{\dagger}\mathrm{U}^{\mathrm{T}} \tag{1}$$

A first idea for regularisation comes from the idea of *filtering* the contribution of the singular vectors in the inverse. Rather than eq. (1) we construct a *regularised inverse*

$$\mathrm{A}_{\alpha}^{\dagger} = \mathrm{V}\mathrm{W}_{\alpha}^{\dagger}\mathrm{U}^{\mathrm{T}} = \sum_{i=1}^{R} v_i \frac{q_{\alpha}(w_i^2)}{w_i} u_i^{\mathrm{T}} \tag{2}$$

where the function $q_{\alpha}(w_i^2)$ is a filter on the SVD spectum.

Regularisation by filtering

Truncated SVD

An example of such an approach is Truncated SVD. Here the filter is

$$q_\alpha(w_i^2) = \begin{cases} 1, & \text{if } i < \alpha \\ 0, & \text{if } i \geq \alpha \end{cases} \tag{3}$$

The parameter α acts as a threshold, cutting off the contribution of higher order singular vectors. This is exactly analogous to the idea of a *low-pass filter* in signal processing, and the analysis of SVD filtering sometimes goes by the name of *Generalised Fourier Analysis*. The regularised inverse in this case is

$$\mathsf{A}_\alpha^\dagger = \sum_{i=1}^{\alpha} \frac{v_i u_i^{\mathrm{T}}}{w_i} \tag{4}$$

Rather than supply α as an index on the singular values (which requires them to be ordered in monotonically decreasing sequence), we could specify the threshold on the value of the singular component

$$q_\alpha(w_i^2) = \begin{cases} 1, & \text{if } w_i^2 > \alpha \\ 0, & \text{if } w_i^2 \leq \alpha \end{cases} \tag{5}$$

Regularisation by filtering

(Zero-Order) Tikhonov filtering

Rather than sharply cut off the higher frequencies, we can smoothly roll them off. The filter becomes

$$q_\alpha(w_i^2) = \frac{w_i^2}{w_i^2 + \alpha} \tag{6}$$

For this filter we can get a direct form for the regularised inverse :

$$\mathsf{A}_\alpha^\dagger = \left(\mathsf{A}^{\mathrm{T}}\mathsf{A} + \alpha\mathsf{I}\right)^{-1}\mathsf{A}^{\mathrm{T}} \tag{7}$$

To see this put

$$\left(\mathsf{A}^{\mathrm{T}}\mathsf{A} + \alpha\mathsf{I}\right) = \left(\mathsf{V}\mathsf{W}^2\mathsf{V}^{\mathrm{T}} + \alpha\mathsf{I}\right) = \mathsf{V}\left(\mathsf{W}^2 + \alpha\mathsf{I}\right)\mathsf{V}^{\mathrm{T}}$$

Which leads to

$$\left(\mathsf{A}^{\mathrm{T}}\mathsf{A} + \alpha\mathsf{I}\right)^{-1}\mathsf{A}^{\mathrm{T}} = \mathsf{V}\left(\mathsf{W}^2 + \alpha\mathsf{I}\right)^{-1}\mathsf{V}^{\mathrm{T}}\mathsf{V}\mathsf{W}\mathsf{U}^{\mathrm{T}} = \mathsf{V}\mathrm{diag}\left[\frac{\mathbf{w}}{\mathbf{w^2} + \alpha}\right]\mathsf{U}^{\mathrm{T}} \tag{8}$$

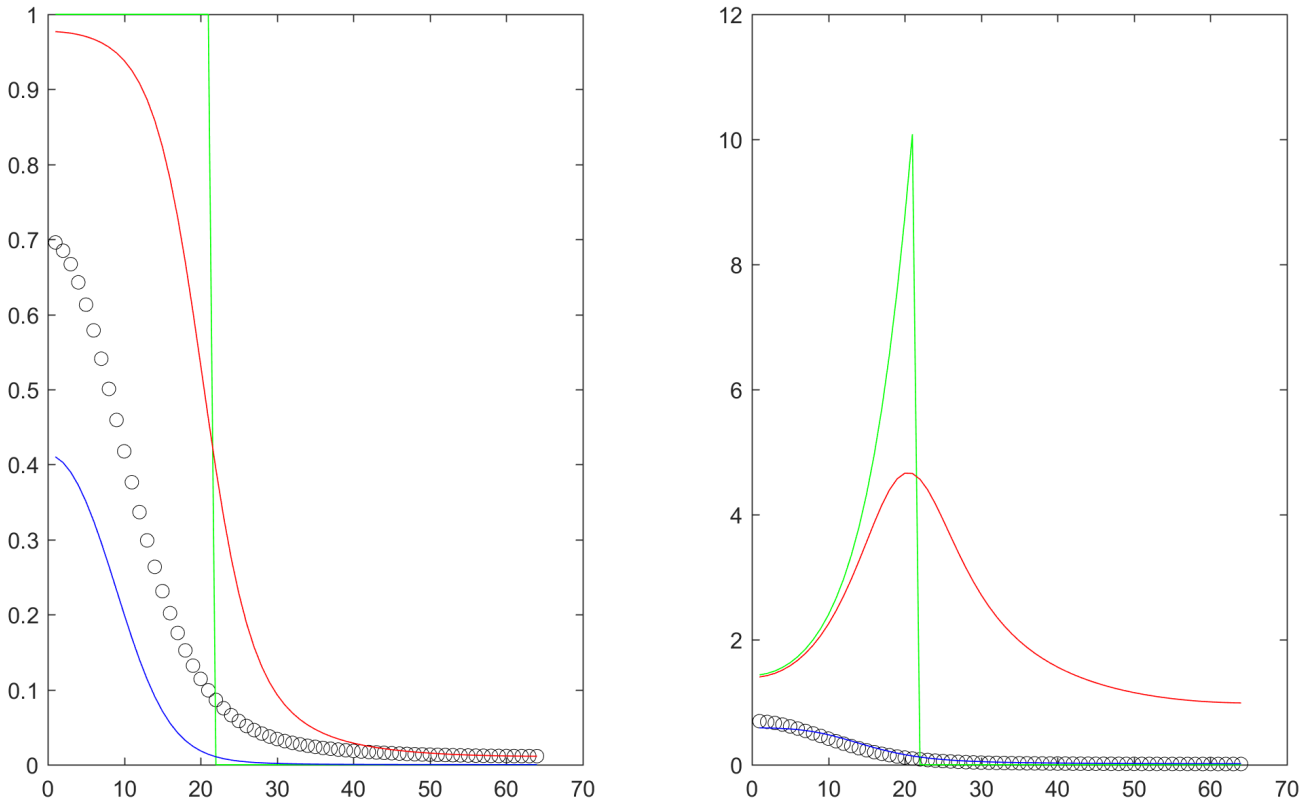
Zero-order Tikhonov regularisation is particularly simple considered as Fourier Domain filtering.

$$\hat{F}_\alpha(\mathbf{k}) = \hat{G}(k)\left(\frac{\hat{H}(\mathbf{k})}{\hat{H}(\mathbf{k})^2 + \alpha}\right)$$

In signal processing it is known as the *Wiener Filter*.

Regularisation by filtering

SV filters for Gaussian convolutoin

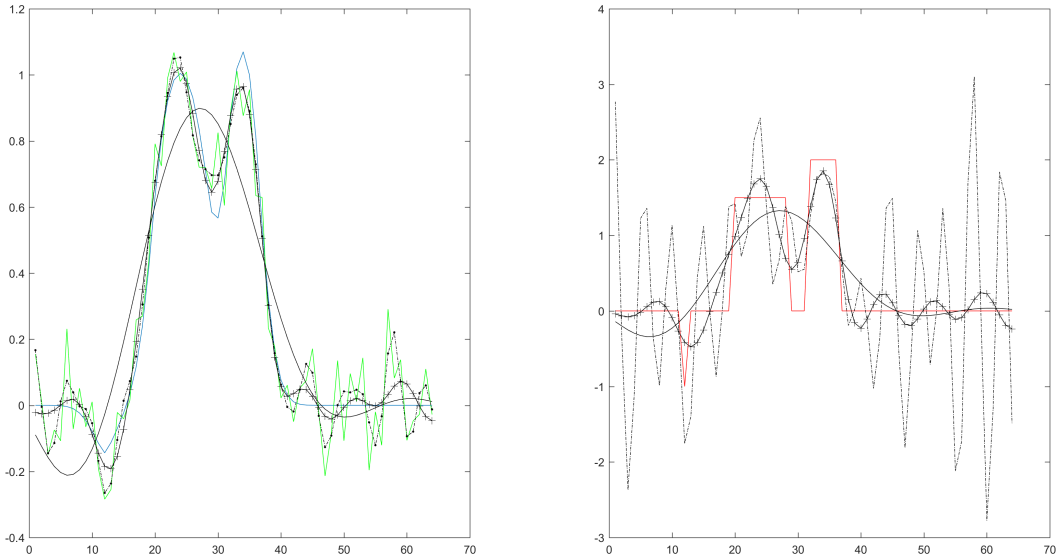


Left : filtering function $q_\alpha(w_i^2)$, Right : weighting $\frac{q_\alpha(w_i^2)}{w_i}$. Black : singular values; green line: the truncated SVD cutoff filter; zeroth-order Tikhonov with a small (Red) and large (Blue) value of α.

Regularisation by filtering

Example

The effect of truncated SVD regularisation with increasing threshold. The larger threshold includes more of the reconstructed function but at the expense of increasing noise.

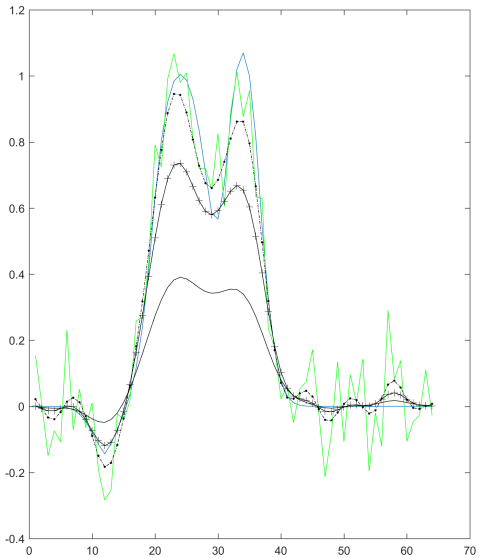


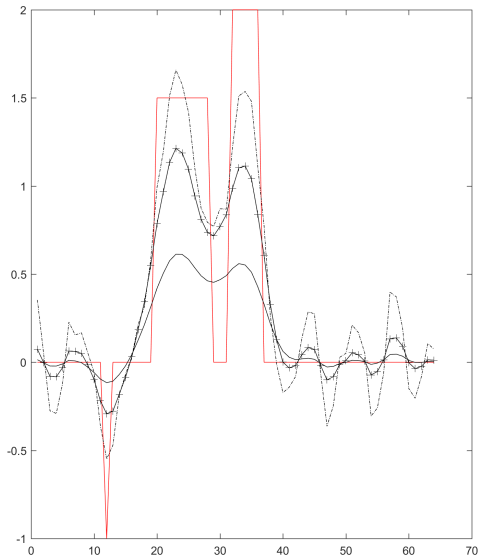
Left : The noiseless data (blue), and noisy data (green), and projection of the reconstructed solutions $g_\alpha = Af_\alpha^\dagger$. Right : True solution (Red) and reconstruction results $f_\alpha^\dagger$. Black : results with 4 singular values, Black + (16), dashed (32).

Regularisation by filtering

Example

The effect of Tikhonov regularisation with decreasing parameter α. The higher values give a strong smoothing effect; the smaller values recover more of the function, at the expense of increasing noise. Note that in addition to smoothing the higher regularisation *scales down* the amplitude of the solution.





Left : The noiseless data (blue), and noisy data (green), and projection of the reconstructed solutions $g_\alpha = Af_\alpha^\dagger$. Right : True solution (Red) and reconstruction results $f_\alpha^\dagger$. Black : high α, Black + (medium), dashed (low).

Outline

Look again at the solution with zero-order Tikhonov *filtering*

$$f_\alpha^\dagger = \mathsf{A}_\alpha^\dagger \tilde{g} = \left(\mathsf{A}^{\mathrm{T}}\mathsf{A} + \alpha \mathsf{I}\right)^{-1} \mathsf{A}^{\mathrm{T}} \tilde{g} \tag{9}$$

Look again at the solution with zero-order Tikhonov *filtering*

$$f_\alpha^\dagger = \mathsf{A}_\alpha^\dagger \tilde{g} = \left(\mathsf{A}^{\mathrm{T}}\mathsf{A} + \alpha \mathsf{I}\right)^{-1} \mathsf{A}^{\mathrm{T}} \tilde{g} \tag{9}$$

Comment: The right hand side of eq. 9 can be intepreted in two parts

$\mathsf{A}^{\mathrm{T}}\tilde{g}$ is a *back projection* : $Y \rightarrow X$ from data space to image space

$\left(\mathsf{A}^{\mathrm{T}}\mathsf{A} + \alpha \mathsf{I}\right)^{-1}$ is an *image filter* : $X \rightarrow X$ from image space to image space

Later we can think of generalising the imaging filtering step to other kinds of image processing operations

Look again at the solution with zero-order Tikhonov *filtering*

$$f_\alpha^\dagger = \mathrm{A}_\alpha^\dagger \tilde{g} = \left(\mathrm{A}^{\mathrm{T}}\mathrm{A} + \alpha \mathbf{I}\right)^{-1} \mathrm{A}^{\mathrm{T}} \tilde{g} \tag{9}$$

Comment: The right hand side of eq. 9 can be intepreted in two parts

$\mathrm{A}^{\mathrm{T}}\tilde{g}$ is a *back projection* : $Y \rightarrow X$ from data space to image space

$\left(\mathrm{A}^{\mathrm{T}}\mathrm{A} + \alpha \mathbf{I}\right)^{-1}$ is an *image filter* : $X \rightarrow X$ from image space to image space

Later we can think of generalising the imaging filtering step to other kinds of image processing operations
Eq. (9) is the same as solving the following *unconstrained* optimisation problem

$$f_\alpha^\dagger = \underset{f \in \mathbb{R}^n}{\arg\min} \left[\Phi = \frac{1}{2} ||\tilde{g} - \mathrm{A}\mathbf{f}||^2 + \alpha \frac{1}{2} ||f||^2 \right] \tag{10}$$

Eq. (10) is also the same as finding the *maximum* of a posterior probability

$$\mathbf{f}_{\alpha}^{\dagger} = \underset{\mathbf{f}\in\mathbb{R}^{n}}{\arg\max} \left[\mathrm{e}^{-\Phi} = \underbrace{\mathrm{e}^{-\frac{1}{2}||\tilde{g}-\mathbf{A}\mathbf{f}||^{2}}}_{\text{Likelihood}} \underbrace{\mathrm{e}^{-\alpha\frac{1}{2}||\mathbf{f}||^{2}}}_{\text{prior}} \right] \tag{11}$$

Eq. (10) is also the same as finding the *maximum* of a posterior probability

$$\mathbf{f}_{\alpha}^{\dagger} = \underset{\mathbf{f}\in\mathbb{R}^{n}}{\arg\max} \left[\mathrm{e}^{-\Phi} = \underbrace{\mathrm{e}^{-\frac{1}{2}||\tilde{g}-\mathbf{A}\mathbf{f}||^{2}}}_{\text{Likelihood}} \underbrace{\mathrm{e}^{-\alpha\frac{1}{2}||\mathbf{f}||^{2}}}_{\text{prior}} \right] \tag{11}$$

With this intepretation we can think more generally of non-Gaussian likelihood and prior and the corresponding optimisation problem

$$\mathbf{f}_{\alpha}^{\dagger} = \underset{\mathbf{f}\in\mathbb{R}^{n}}{\arg\min} \left[\Phi = \underbrace{\mathcal{D}(\tilde{g},\mathbf{A}\mathbf{f})}_{\text{Data fitting term}} + \underbrace{\alpha\Psi(f)}_{\text{Penalty term}} \right] \equiv \underset{\mathbf{f}\in\mathbb{R}^{n}}{\arg\max} \left[\mathrm{e}^{-\mathcal{D}(\tilde{g},\mathbf{A}\mathbf{f})} \mathrm{e}^{-\alpha\Psi(f)} \right] \tag{12}$$

Eq. (10) is also the same as finding the *maximum* of a posterior probability

$$\mathbf{f}_{\alpha}^{\dagger} = \underset{\mathbf{f}\in\mathbb{R}^n}{\arg\max} \left[\mathrm{e}^{-\Phi} = \underbrace{\mathrm{e}^{-\frac{1}{2}||\tilde{g}-\mathbf{A}\mathbf{f}||^2}}_{\text{Likelihood}} \underbrace{\mathrm{e}^{-\alpha\frac{1}{2}||\mathbf{f}||^2}}_{\text{prior}} \right] \tag{11}$$

With this intepretation we can think more generally of non-Gaussian likelihood and prior and the corresponding optimisation problem

$$\mathbf{f}_{\alpha}^{\dagger} = \underset{\mathbf{f}\in\mathbb{R}^n}{\arg\min} \left[\Phi = \underbrace{\mathcal{D}(\tilde{g}, \mathbf{A}\mathbf{f})}_{\text{Data fitting term}} + \underbrace{\alpha\Psi(f)}_{\text{Penalty term}} \right] \equiv \underset{\mathbf{f}\in\mathbb{R}^n}{\arg\max} \left[\mathrm{e}^{-\mathcal{D}(\tilde{g},\mathbf{A}\mathbf{f})} \mathrm{e}^{-\alpha\Psi(f)} \right] \tag{12}$$

Later in the course we'll look at more general, non-Gaussian priors.

Eq. (10) is also the same as finding the *maximum* of a posterior probability

$$\mathbf{f}_{\alpha}^{\dagger} = \underset{\mathbf{f}\in\mathbb{R}^n}{\arg\max} \left[\mathrm{e}^{-\Phi} = \underbrace{\mathrm{e}^{-\frac{1}{2}||\tilde{g}-\mathbf{A}\mathbf{f}||^2}}_{\text{Likelihood}} \underbrace{\mathrm{e}^{-\alpha\frac{1}{2}||\mathbf{f}||^2}}_{\text{prior}} \right] \tag{11}$$

With this intepretation we can think more generally of non-Gaussian likelihood and prior and the corresponding optimisation problem

$$\mathbf{f}_{\alpha}^{\dagger} = \underset{\mathbf{f}\in\mathbb{R}^n}{\arg\min} \left[\Phi = \underbrace{\mathcal{D}(\tilde{g}, \mathbf{A}\mathbf{f})}_{\text{Data fitting term}} + \underbrace{\alpha\Psi(f)}_{\text{Penalty term}} \right] \equiv \underset{\mathbf{f}\in\mathbb{R}^n}{\arg\max} \left[\mathrm{e}^{-\mathcal{D}(\tilde{g},\mathbf{A}\mathbf{f})} \mathrm{e}^{-\alpha\Psi(f)} \right] \tag{12}$$

Later in the course we'll look at more general, non-Gaussian priors.
For now we'll keep things simple with a generalisation of the prior that is still Gaussian.

Tikhonov refers in general to any *quadratic* functional

$$\Psi(\mathbf{f}) = \frac{1}{2}||\mathbf{f}||^2_{\Gamma} = \frac{1}{2}\mathbf{f}^{\mathrm{T}}\Gamma\mathbf{f} \tag{13}$$

where Γ is a matrix.

Tikhonov refers in general to any *quadratic* functional

$$\Psi(\mathbf{f}) = \frac{1}{2}||\mathbf{f}||^2_{\Gamma} = \frac{1}{2}\mathbf{f}^{\mathrm{T}}\Gamma\mathbf{f} \tag{13}$$

where Γ is a matrix. There are two ways that this is commonly used.

Tikhonov refers in general to any *quadratic* functional

$$\Psi(\mathbf{f}) = \frac{1}{2}||\mathbf{f}||_{\Gamma}^2 = \frac{1}{2}\mathbf{f}^{\mathrm{T}}\Gamma\mathbf{f} \tag{13}$$

where Γ is a matrix. There are two ways that this is commonly used. The first is to induce *correlation* in the elements of the solution. This is clear from the Bayesian interpretation of the regularisation functional as the negative log of a prior probability density on the solution:

$$\Psi(\mathbf{f}) = -\log \pi(\mathbf{f}) \quad \Leftrightarrow \quad \pi(\mathbf{f}) = \exp\left[-\frac{1}{2}\mathbf{f}^{\mathrm{T}}\Gamma\mathbf{f}\right] \tag{14}$$

where the expression on the right is a multivariate Gaussian distribution with covariance $C = \Gamma^{-1}$.

Tikhonov refers in general to any *quadratic* functional

$$\Psi(\mathbf{f}) = \frac{1}{2}||\mathbf{f}||^2_{\Gamma} = \frac{1}{2}\mathbf{f}^{\mathrm{T}}\Gamma\mathbf{f} \tag{13}$$

where Γ is a matrix. There are two ways that this is commonly used. The first is to induce *correlation* in the elements of the solution. This is clear from the Bayesian interpretation of the regularisation functional as the negative log of a prior probability density on the solution:

$$\Psi(\mathbf{f}) = -\log \pi(\mathbf{f}) \quad \Leftrightarrow \quad \pi(\mathbf{f}) = \exp\left[-\frac{1}{2}\mathbf{f}^{\mathrm{T}}\Gamma\mathbf{f}\right] \tag{14}$$

where the expression on the right is a multivariate Gaussian distribution with covariance $C = \Gamma^{-1}$.
The corresponding optimisation problem is solved by

$$\mathbf{f}^{\dagger}_{\alpha,\Gamma} = \mathrm{A}^{\dagger}_{\alpha,\Gamma}\tilde{\mathbf{g}} = \left(\mathrm{A}^{\mathrm{T}}\mathrm{A} + \alpha\Gamma\right)^{-1}\mathrm{A}^{\mathrm{T}}\tilde{\mathbf{g}} \tag{15}$$

Another option is to penalise *derivatives* of the image.

Another option is to penalise *derivatives* of the image. Staying in 1D we would have

$$\Psi(\mathbf{f}) = \frac{1}{2}\left\|\frac{\mathrm{d}\mathbf{f}}{\mathrm{d}x}\right\|^2 \tag{16}$$

Another option is to penalise *derivatives* of the image. Staying in 1D we would have

$$\Psi(\mathbf{f}) = \frac{1}{2}\left\|\frac{\mathrm{d}\mathbf{f}}{\mathrm{d}x}\right\|^2 \tag{16}$$

In the discrete setting, the derivative can be implemented as a **finite difference matrix** giving

$$\Psi(\mathbf{f}) = \frac{1}{2}||\mathrm{D}\mathbf{f}||^2 = \frac{1}{2}\mathbf{f}^{\mathrm{T}}\mathrm{D}^{\mathrm{T}}\mathrm{D}\mathbf{f} = \frac{1}{2}\mathbf{f}^{\mathrm{T}}\Gamma\mathbf{f} \tag{17}$$

where $\Gamma = \mathrm{D}^{\mathrm{T}}\mathrm{D}$ is the discrete second order derivative.

Another option is to penalise *derivatives* of the image. Staying in 1D we would have

$$\Psi(\mathbf{f}) = \frac{1}{2}\left\|\frac{\mathrm{d}\mathbf{f}}{\mathrm{d}x}\right\|^2 \tag{16}$$

In the discrete setting, the derivative can be implemented as a **finite difference matrix** giving

$$\Psi(\mathbf{f}) = \frac{1}{2}||\mathrm{D}\mathbf{f}||^2 = \frac{1}{2}\mathbf{f}^{\mathrm{T}}\mathrm{D}^{\mathrm{T}}\mathrm{D}\mathbf{f} = \frac{1}{2}\mathbf{f}^{\mathrm{T}}\Gamma\mathbf{f} \tag{17}$$

where $\Gamma = \mathrm{D}^{\mathrm{T}}\mathrm{D}$ is the discrete second order derivative.
In 2D this would be the **Laplacian operator** $\Gamma_{2\mathrm{D}} \equiv -\nabla^2$. (change of sign due to integration by parts!)

Another option is to penalise *derivatives* of the image. Staying in 1D we would have

$$\Psi(\mathbf{f}) = \frac{1}{2}\left\|\frac{\mathrm{d}\mathbf{f}}{\mathrm{d}x}\right\|^2 \tag{16}$$

In the discrete setting, the derivative can be implemented as a **finite difference matrix** giving

$$\Psi(\mathbf{f}) = \frac{1}{2}||\mathrm{D}\mathbf{f}||^2 = \frac{1}{2}\mathbf{f}^{\mathrm{T}}\mathrm{D}^{\mathrm{T}}\mathrm{D}\mathbf{f} = \frac{1}{2}\mathbf{f}^{\mathrm{T}}\Gamma\mathbf{f} \tag{17}$$

where $\Gamma = \mathrm{D}^{\mathrm{T}}\mathrm{D}$ is the discrete second order derivative.
In 2D this would be the **Laplacian operator** $\Gamma_{2\mathrm{D}} \equiv -\nabla^2$. (change of sign due to integration by parts!)
This is known as *First Order Tikhonov regularisation*, because it is penalising derivatives of order 1. Naturally, higher order derivatives could be penalised too.

Regularisation

1st order Tikhonov Regularisation

What is the Bayesian interpretation of the derivative penalty ?

Regularisation

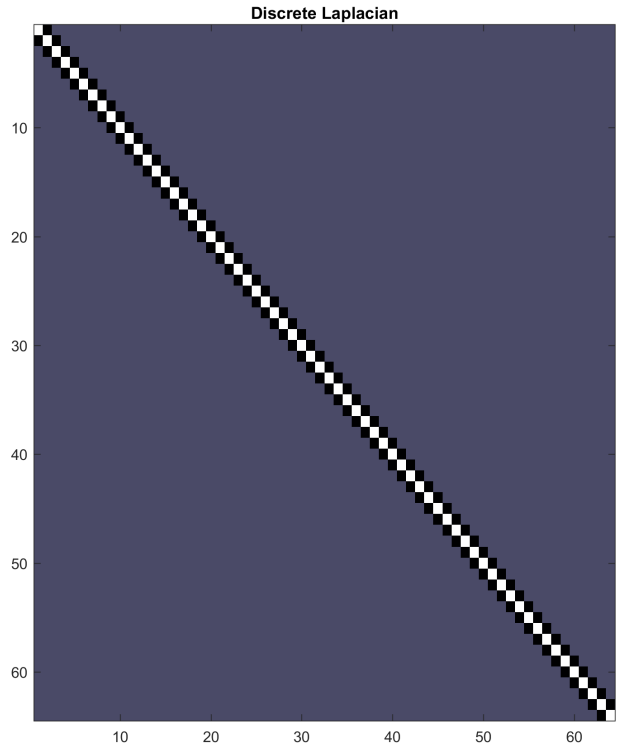
1st order Tikhonov Regularisation

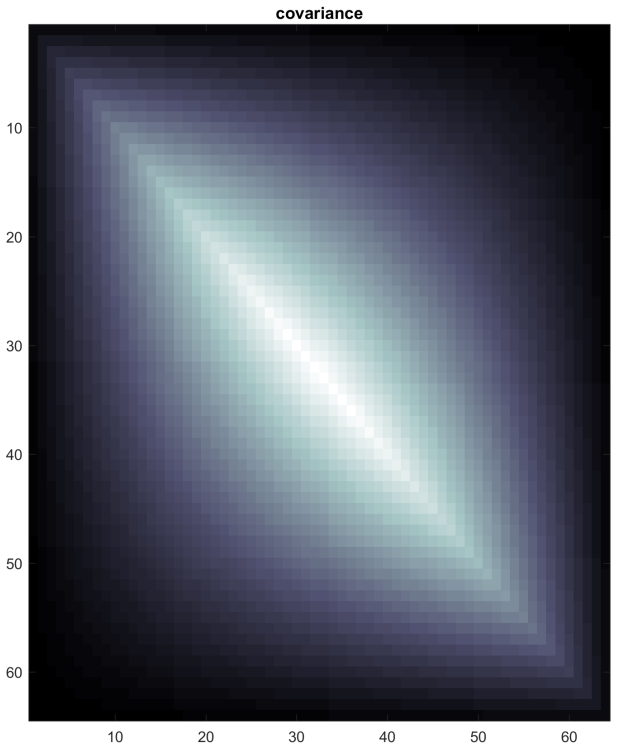
What is the Bayesian interpretation of the derivative penalty ?
Technically, the Laplacian $-\nabla^2$ doesn't have an inverse because it has a Null-Space. But by Specifying boundary conditions we can get an inversion of the discrete operator

Regularisation

1st order Tikhonov Regularisation

What is the Bayesian interpretation of the derivative penalty ?
Technically, the Laplacian $-\nabla^2$ doesn't have an inverse because it has a Null-Space. But by Specifying boundary conditions we can get an inversion of the discrete operator





The effect is to strongly correlate between neighbouring pixels.

Comment : Another interesting feature of using derivatives in the penalty is that the singular functions of a convolution operator are still diagonalised in this prior.

Comment : Another interesting feature of using derivatives in the penalty is that the singular functions of a convolution operator are still diagonalised in this prior.
Recall that $\mathrm{e}^{-\mathrm{i}\boldsymbol{k}\cdot x}$ is a singular function of any convolution operator $g = f * h$.

Comment : Another interesting feature of using derivatives in the penalty is that the singular functions of a convolution operator are still diagonalised in this prior.
Recall that $\mathrm{e}^{-\mathrm{i}\boldsymbol{k}\cdot x}$ is a singular function of any convolution operator $g = f * h$.
But $-\nabla^2 \mathrm{e}^{-\mathrm{i}\boldsymbol{k}\cdot x} = k^2 \mathrm{e}^{-\mathrm{i}\boldsymbol{k}\cdot x}$.

Comment : Another interesting feature of using derivatives in the penalty is that the singular functions of a convolution operator are still diagonalised in this prior.
Recall that $\mathrm{e}^{-\mathrm{i}\boldsymbol{k}\cdot x}$ is a singular function of any convolution operator $g = f * h$.
But $-\nabla^2 \mathrm{e}^{-\mathrm{i}\boldsymbol{k}\cdot x} = k^2 \mathrm{e}^{-\mathrm{i}\boldsymbol{k}\cdot x}$.
This means that the optimisation problem

$$\arg\min_f \left[\frac{1}{2}||g - h * f||^2 + \alpha\frac{1}{2}|\nabla f|^2\right]$$

is solved by Fourier Domain filtering as

$$\hat{F}_\alpha(k) = \hat{G}(k)\left(\frac{\hat{H}(k)}{\hat{H}(k)^2 + \alpha k^2}\right)$$

Comment : Another interesting feature of using derivatives in the penalty is that the singular functions of a convolution operator are still diagonalised in this prior.
Recall that $\mathrm{e}^{-\mathrm{i}\boldsymbol{k}\cdot x}$ is a singular function of any convolution operator $g = f * h$.
But $-\nabla^2 \mathrm{e}^{-\mathrm{i}\boldsymbol{k}\cdot x} = k^2 \mathrm{e}^{-\mathrm{i}\boldsymbol{k}\cdot x}$.
This means that the optimisation problem

$$\arg\min_f \left[\frac{1}{2}||g - h * f||^2 + \alpha\frac{1}{2}|\nabla f|^2\right]$$

is solved by Fourier Domain filtering as

$$\hat{F}_\alpha(k) = \hat{G}(k)\left(\frac{\hat{H}(k)}{\hat{H}(k)^2 + \alpha k^2}\right)$$

In signal processing it is known as the *Generalised Wiener Filter*.

Regularisation

Bias Variance

- As we have seen, introduction of regularisation prevents instability in the solution. But it also means that the model no longer fully fits the data.

Regularisation

Bias Variance

- As we have seen, introduction of regularisation prevents instability in the solution. But it also means that the model no longer fully fits the data.
- The statistical distribution of the solution in the presence of noise has a reduced variance, but it has introduced *bias*.

Regularisation

Bias Variance

- As we have seen, introduction of regularisation prevents instability in the solution. But it also means that the model no longer fully fits the data.
- The statistical distribution of the solution in the presence of noise has a reduced variance, but it has introduced *bias*.
- We test these formally using a bias-variance curve. This plots mean of distribution of solution errors vs the trace of their covariance.

Regularisation

Bias Variance

- As we have seen, introduction of regularisation prevents instability in the solution. But it also means that the model no longer fully fits the data.
- The statistical distribution of the solution in the presence of noise has a reduced variance, but it has introduced *bias*.
- We test these formally using a bias-variance curve. This plots mean of distribution of solution errors vs the trace of their covariance.

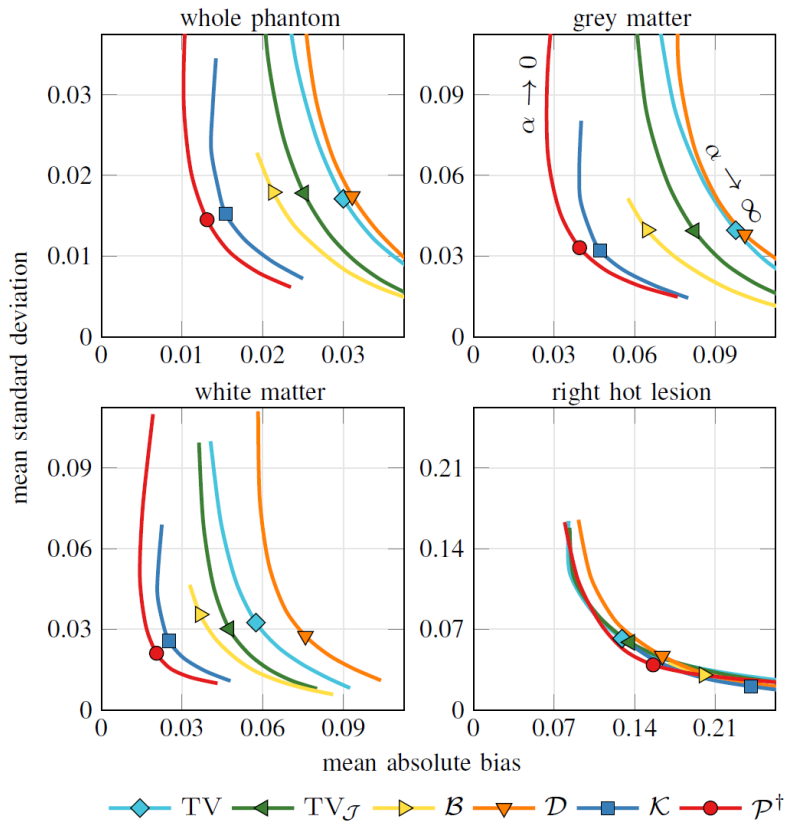
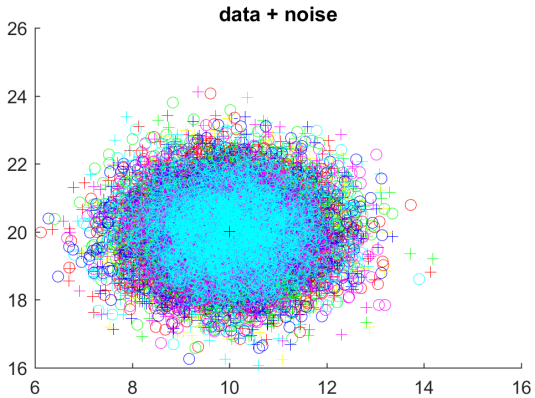
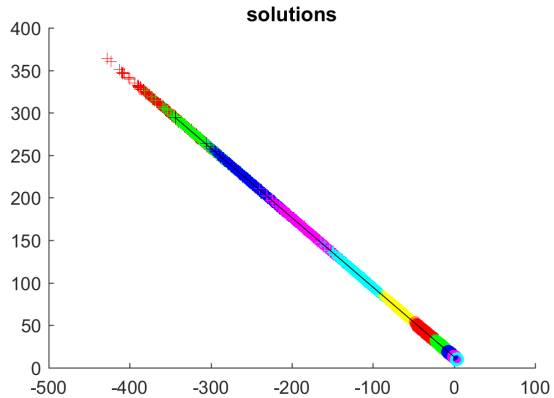


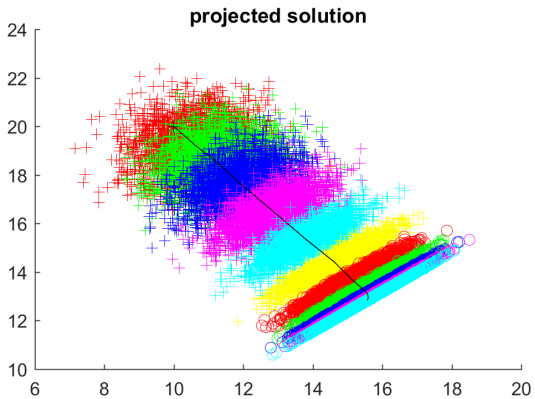
Fig. 6. Mean absolute bias-versus-mean standard deviation trade-off of in different regions of interest. $\mathcal{K}$ and $\mathcal{P}$ have the best trade-off for the whole phantom, grey matter and white matter as their curves lie "underneath" the other curves but all methods have similar curves for the right hot lesion (bottom right). Moreover, the solution that has the smallest expected mean squared error for the whole phantom (distance from the origin in the top left plot) is marked in all four graphs. It can be seen that these solutions have all roughly the same standard deviation for the whole phantom, grey matter and white matter but $\mathcal{P}$ has always the smallest bias. In addition, the "optimal" solution for $\mathcal{K}$ has a larger bias for the hot lesion.

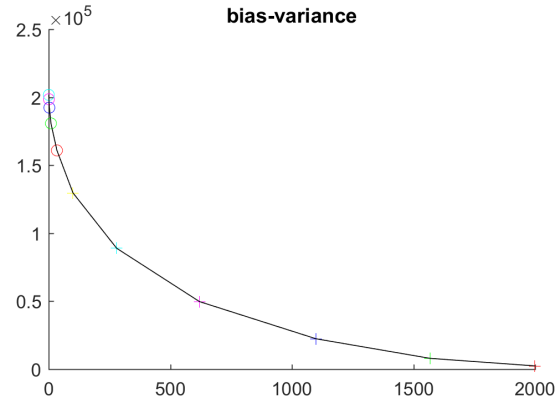
Regularisation

Bias Variance Results 1









Results of `RunIPtoyReg`. Top left : true data with added noise. Top right : solution distributions for different regularisation parameters. Bottom left : forward projection of solutions obtained. Bottom right : bias variance curve.

Outline

A generalisation of Tikhonov Regularisation is to use a regulariser of the form

$$\Psi(f) = \sum_{j=1}^{J} \psi(\langle f, \phi_j \rangle) = \sum_{j}^{J} \psi(c_j) \tag{18}$$

where the coefficients c_j are the projection of the function f onto a set of representative functions $\{\phi_j; j = 1 \dots J\}$

- The mapping $\mathcal{T} : X \mapsto C$ is called the *analysis* operator.
- The adjoint mapping $\mathcal{T}^* : C \mapsto X$ is called the *synthesis* operator.
- If $\{\phi_j\}$ are orthonormal then they form a *basis* and $\mathcal{T}^* \equiv \mathcal{T}^{-1}$.
- More generally $\{\phi_j\}$ may be *over complete* and are said to form a *frame*

Sparsity

Introduction

The idea behind *sparsity regularisation* is that most images, despite having potentially millions of pixels, do not contain an equivalent amount of information.
This means they can be represented quite compactly provided the appropriate representation $\{\phi_j\}$ is used.

Sparsity

Introduction

The idea behind *sparsity regularisation* is that most images, despite having potentially millions of pixels, do not contain an equivalent amount of information.
This means they can be represented quite compactly provided the appropriate representation $\{\phi_j\}$ is used.
There are three commonly used approaches

1. Total Variation : the function representation is in terms of the derivatives of the function *f*.
2. Wavelet basis : the function representation is in terms of wavelets or their variations (curvelets, shearlets etc.)
3. Dictionary basis : the function is represented in terms of problem specific components derived from learning.

Sparsity

Total Variation

The motivation for the Total Variation (TV) regulariser is that natural scenes tend to consist of fairly homogeneous regions with sharp edges between them.
The basic idea is the same as the first order Tikhonov regulariser with the 2-norm replaced by a 1-norm. [1]

$$\Psi_{\mathrm{TV}}(f) = \int |\nabla f| \mathrm{d}x$$

This can be seen as a special form of eq. 18 with the basis function $\{\phi_j\} = \delta'(x - x_j)$ chosen as the derivatives of a delta-function.

[1] In statistics this method goes by the name **Lasso** [Tibishirani 1996]

Sparsity

Total Variation

The motivation for the Total Variation (TV) regulariser is that natural scenes tend to consist of fairly homogeneous regions with sharp edges between them.
The basic idea is the same as the first order Tikhonov regulariser with the 2-norm replaced by a 1-norm. [1]

$$\Psi_{\mathrm{TV}}(f) = \int |\nabla f| \mathrm{d}x$$

This can be seen as a special form of eq. 18 with the basis function $\{\phi_j\} = \delta'(x - x_j)$ chosen as the derivatives of a delta-function.
This function has a value given by the length of the perimeter of a region, multiplied by its contrast (the height of the jump between the region interior and its background).

[1] In statistics this method goes by the name **Lasso** [Tibishirani 1996]

Sparsity

Total Variation

The motivation for the Total Variation (TV) regulariser is that natural scenes tend to consist of fairly homogeneous regions with sharp edges between them.
The basic idea is the same as the first order Tikhonov regulariser with the 2-norm replaced by a 1-norm. [1]

$$\Psi_{\mathrm{TV}}(f) = \int |\nabla f| \mathrm{d}x$$

This can be seen as a special form of eq. 18 with the basis function $\{\phi_j\} = \delta'(x - x_j)$ chosen as the derivatives of a delta-function.
This function has a value given by the length of the perimeter of a region, multiplied by its contrast (the height of the jump between the region interior and its background).
We could also have used the 0-norm

$$\Psi_{\partial_0}(f) = \int |\nabla f|^{(0)} \mathrm{d}x$$

This would simply record the length of all edges in an image. But minimisation of this function is a *non-convex* problem and very difficult to compute.

[1] In statistics this method goes by the name **Lasso** [Tibishirani 1996]

Definition of Total Variation

Let f be a real-valued function on the interval $[a, b]$. The Total Variation of f, denoted $\mathrm{TV}(f)$ is given by

$$\mathrm{TV}(f) := \sup \sum_{j=1}^{J} |f(x_j) - f(x_{j-1})|$$

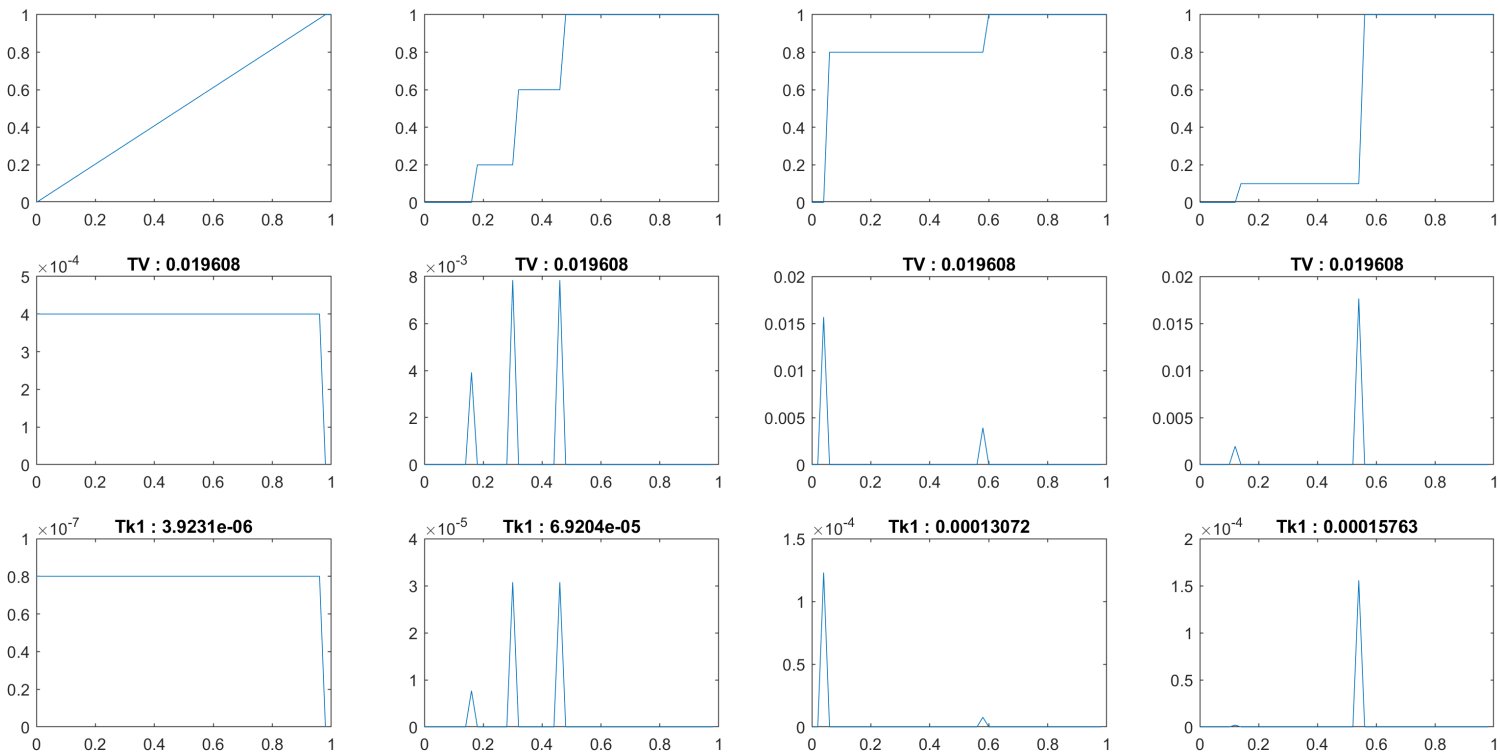
where the supremum is over all partitions $\{a = x_0 < x_{-1} < \ldots x_{J-1} < x_J = b\}$ of the interval $[a, b]$.

Note that if f is piecewise constant with a finite number of jumps, then the TV value is the sum of the magnitude of the jumps.

This means that any function increasing monotonically in the interval $[a, b]$ with the same start and end values will have the same TV value.

Sparsity

Total Variation Illustration



These functions all monotonically increase from 0 to 1. They have the same value of TV and different values of TK1.

Code : `Sparsity/lin1d_TV_qaudprog.m`

If f is differentiable, letting $\Delta x = x_j - x_{j-1}$

$$\mathrm{TV}(f) := \sup \sum_{j=1}^{J} \frac{|f(x_j) - f(x_{j-1}|}{\Delta x} \Delta x$$

and letting $\Delta x \to 0$ results in

$$\mathrm{TV}(f) := \int_a^b |f'(x)| \mathrm{d}x$$

In higher dimensions this generalises to

$$\mathrm{TV}(f) := \int_\Omega |\nabla f(x)| \mathrm{d}x$$

as already defined above. An alternative expression that will prove useful is

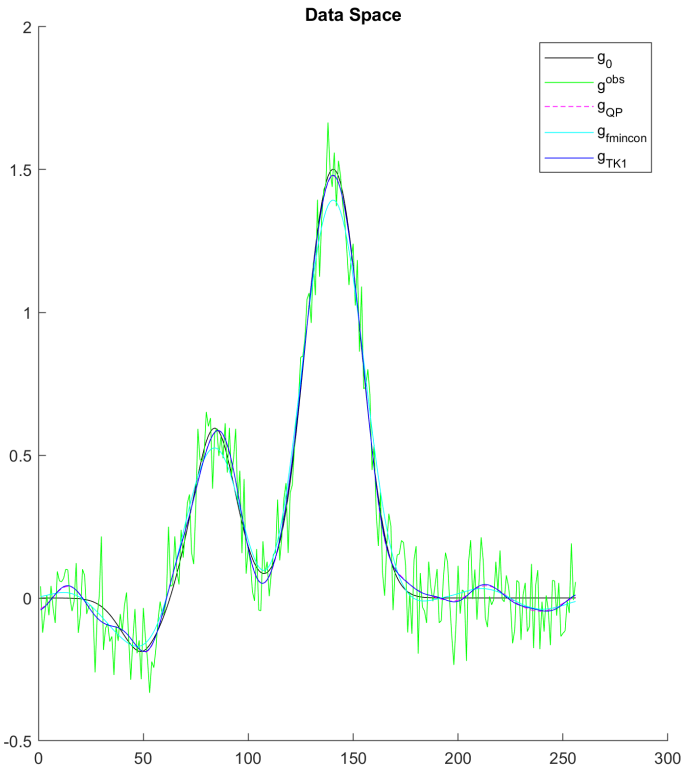
$$\mathrm{TV}(f) := \sup_{\mathbf{v} \in \mathcal{V}} \int_\Omega f(x,y) \nabla \cdot \mathbf{v} \, \mathrm{d}x \mathrm{d}y$$

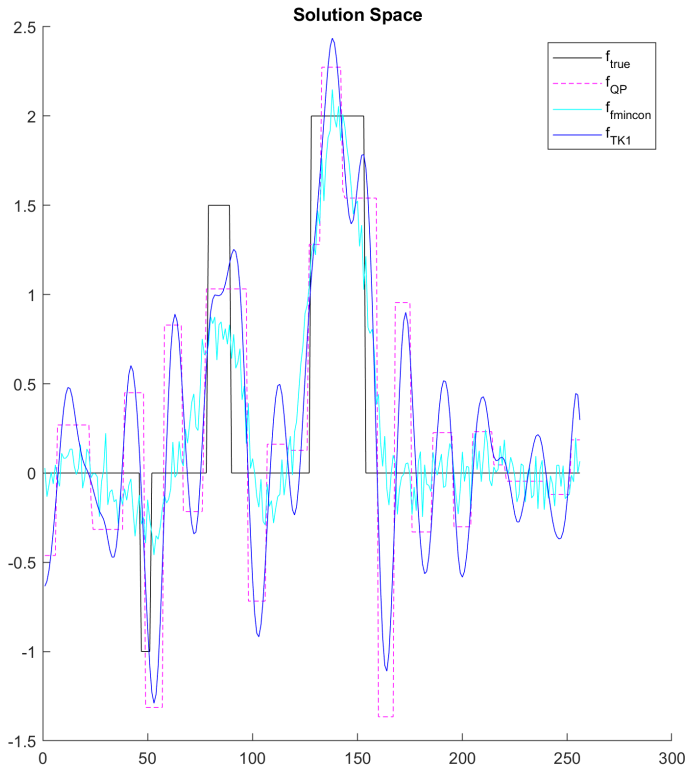
where $\mathcal{V}$ is the space of vector valued functions with norm $|\mathbf{v}| \leq 1$ with continously differentiable components $(v_1(x,y), v_2(x,y)$ that vanish on the boundary of the unit square.

- There is a technical problem with using Ψ_{TV} as a regulariser. It is because the function $\psi(s) = |s|$ is not differentiable at $s = 0$. We say that it belongs to the class of functions that are **convex but non-smooth**.
- This fact motivated the use of the various "TV-like" functions such as Perona-Malik, Smoothed-TV, and Huber that we saw earlier.
- For small size problems we can use a form of **constrained optimisation** that will help to demonstrate the effectiveness of TV as a regulariser.

Sparsity

Total Variation Results





Comparison of results for the 1D deblurring problem, using TK1 and Total Variation.

Code : `Sparsity/lin1d_TV_qaudprog.m`

2D example : `Sparsity/TVgrad.m`

Sparsity

Wavelets

- Wavelets are a powerful way to provide **multiresolution analysis** of signals and images.
- They are the underlying concept behind **compression** and **compressed sensing**. For example “JPEG2000” image compression standard.
- The basic idea is to take a function $\phi^{\text{Mother}}(x)$ that is compact in the spatial and frequency domains simultaneously, and to produce a representative set by *scaling* and *dilation*.
- In contrast to the Fourier Transform, the Wavelet Transform produces a set of “smoothed” (i.e. low-resolution) data and their remainder “details” at a mutiple of scales.
- Truncation of the wavelet transformed image gives a way to remove extraneous detail at mutliple scales, while retaining local detail.

Wavelets are defined by a pair of filters $\{\psi, \phi\}$. Typically ψ (the *Mother Wavelet*) has one or more vanishing moments and behaves as a generalised differential operator, whereas ϕ (the *Father Wavelet*) is a smoothing filter. In signal processing this pair is thought of as "high-pass" and "low-pass" respectively. The wavelet transform at a single scale σ is obtained by convolution with each filter scaled by σ

$$\begin{pmatrix} \xi_S(x) \\ \xi_D(x) \end{pmatrix} = \begin{pmatrix} \mathcal{H}_{S,\sigma} \\ \mathcal{H}_{D,\sigma} \end{pmatrix} f = \frac{1}{\sqrt{\sigma}} \begin{pmatrix} \phi\left(\frac{x}{\sigma}\right) * f \\ \psi\left(\frac{x}{\sigma}\right) * f \end{pmatrix} \tag{19}$$

(In most cases) we will have orthogonality within and across scales

$$\mathcal{H}^*_{S,\sigma_i} \mathcal{H}_{S,\sigma_j} = \delta_{i,j} \mathbf{I} \quad \mathcal{H}^*_{D,\sigma_i} \mathcal{H}_{D,\sigma_j} = \delta_{i,j} \mathbf{I} \quad \mathcal{H}^*_{S,\sigma_i} \mathcal{H}_{D,\sigma_j} = 0 \tag{20}$$

In the discrete setting the above transform at the lowest scale is a matrix $\mathrm{H}_1 \in \mathbb{R}^{N\times N}$ split into the "smoothing" and "detail" parts

$$\begin{pmatrix}\xi_{S,1}\\ \xi_{D,1}\end{pmatrix} = \begin{pmatrix}\mathrm{H}_{S,1}\\ \mathrm{H}_{D,1}\end{pmatrix} \boldsymbol{f} \qquad \mathrm{H}_{S,1}, \mathrm{H}_{D,1} \in \mathbb{R}^{N/2\times N} \tag{21}$$

with orthogonality $\mathrm{H}_{S,1}^{\mathrm{T}}\mathrm{H}_{S,1} = \mathrm{I} \quad \mathrm{H}_{D,1}^{\mathrm{T}}\mathrm{H}_{D,1} = \mathrm{I} \quad \mathrm{H}_{S,1}^{\mathrm{T}}\mathrm{H}_{D,1} = 0$ (22)

Note that, in contrast to the normal definition of discrete convolution, the effect of H_1 is convolve the corresponding sampled filter with the signal and downsample by a factor of 2. In the convolution neural network syntax this is a convolution with *stride* of 2.
A *multiscale* decomposition is obtained by applying eq. (21) recursively

$$\begin{pmatrix}\xi_{S,j}\\ \xi_{D,j}\end{pmatrix} = \begin{pmatrix}\mathrm{H}_{S,j}\\ \mathrm{H}_{D,j}\end{pmatrix} \xi_{S,j-1} \qquad \mathrm{H}_{S,j}, \mathrm{H}_{D,j} \in \mathbb{R}^{N/2^j\times N/2^{j-1}} \tag{23}$$

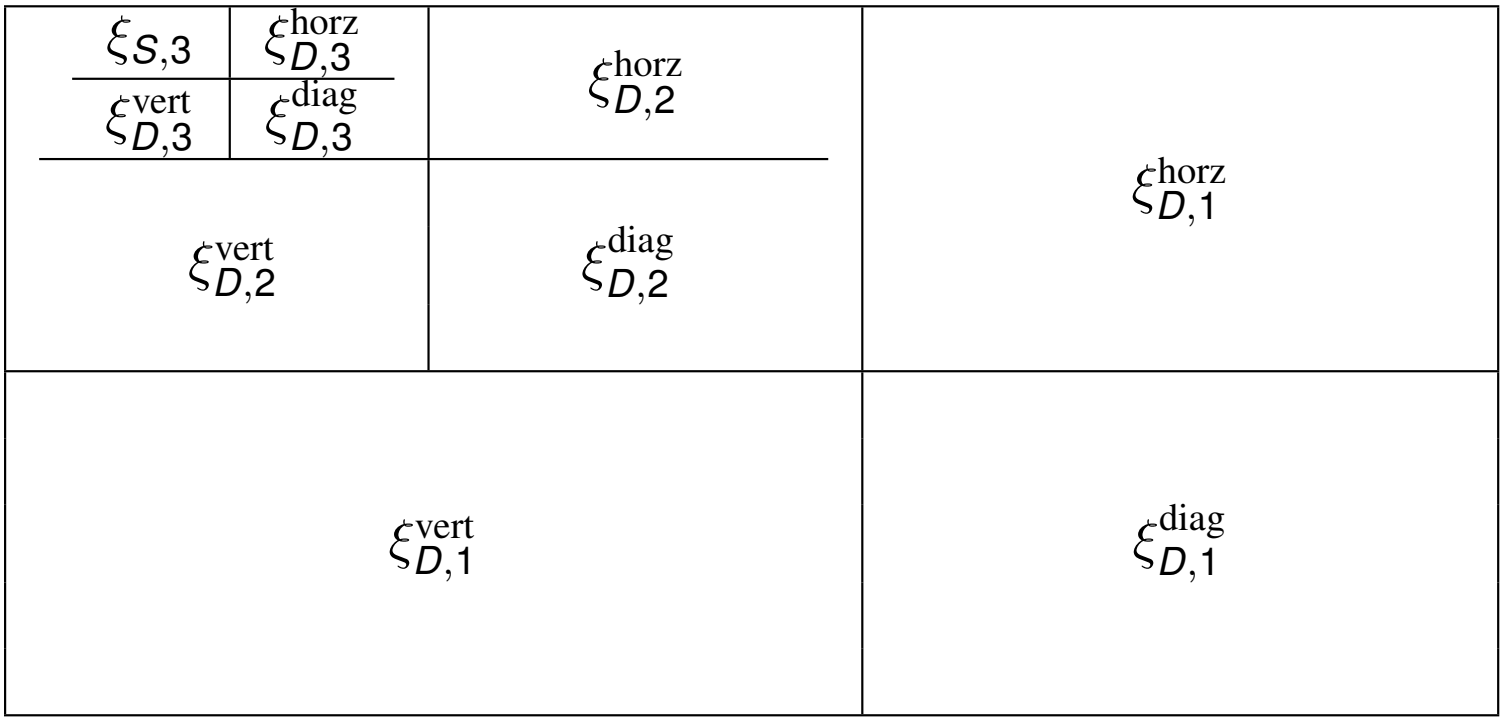
with the final encoding

$$\xi = \mathrm{H}f = \begin{pmatrix}\xi_{S,J}\\ \xi_{D,J}\\ \xi_{D,J-1}\\ \vdots\\ \xi_{D,1}\end{pmatrix}$$

Sparsity

Wavelets

In multiple dimensions a multiscale wavelet decomposition provides includes smoothing and detail at each scale and across dimensions :

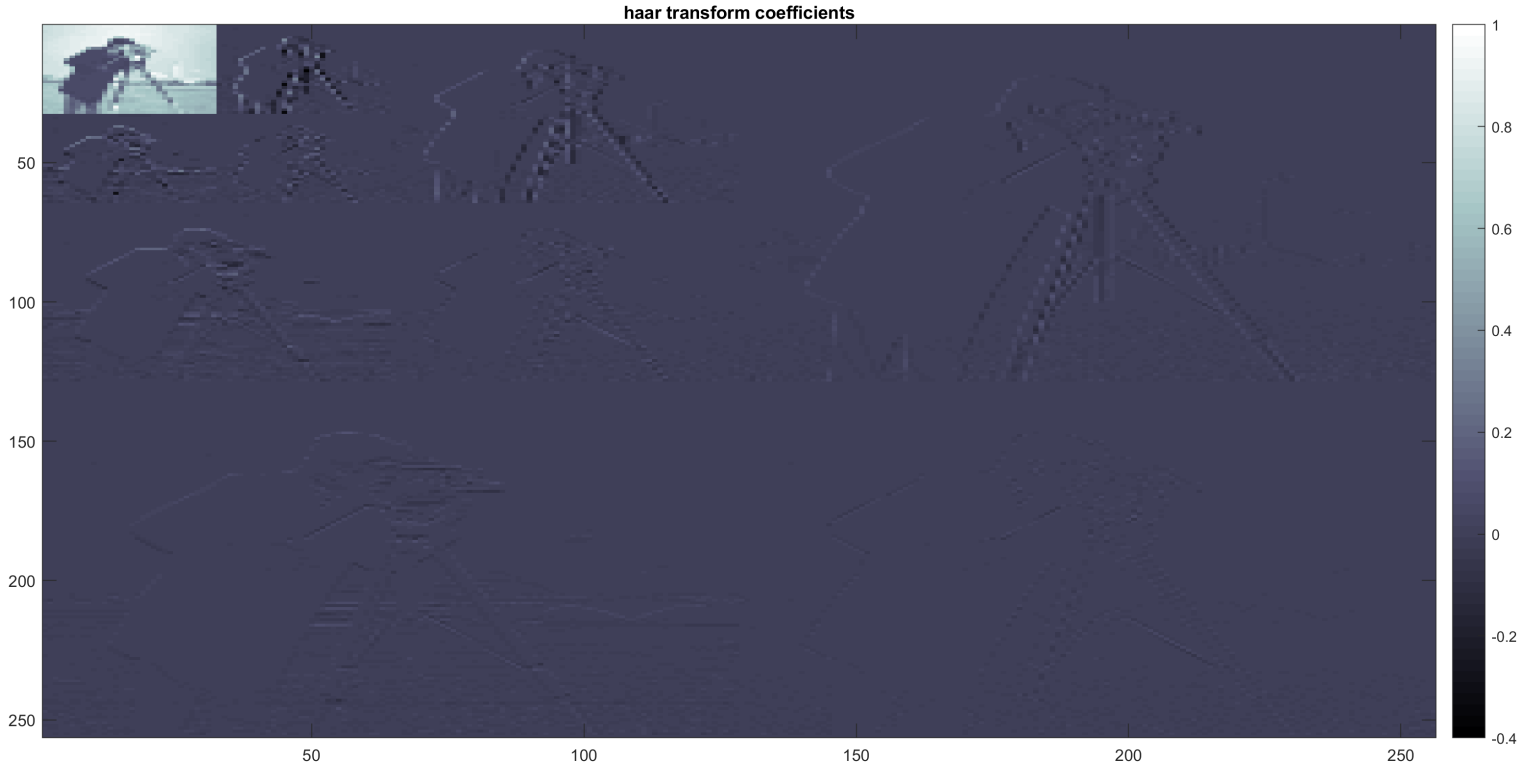


(24)

Many wavelet operators consist of atoms that are orthogonal and invertible through their adjoint i.e. $\mathrm{H}^{\mathrm{T}}\,\mathrm{H} = \mathrm{Id}$

Sparsity

Wavelets



Code : `WaveletExample/WaveletTest2D.m`

Sparsity

Wavelets

setting small wavelet coefficients to zero allows compression

Code : `WaveletExample/WaveletTest2D.m`

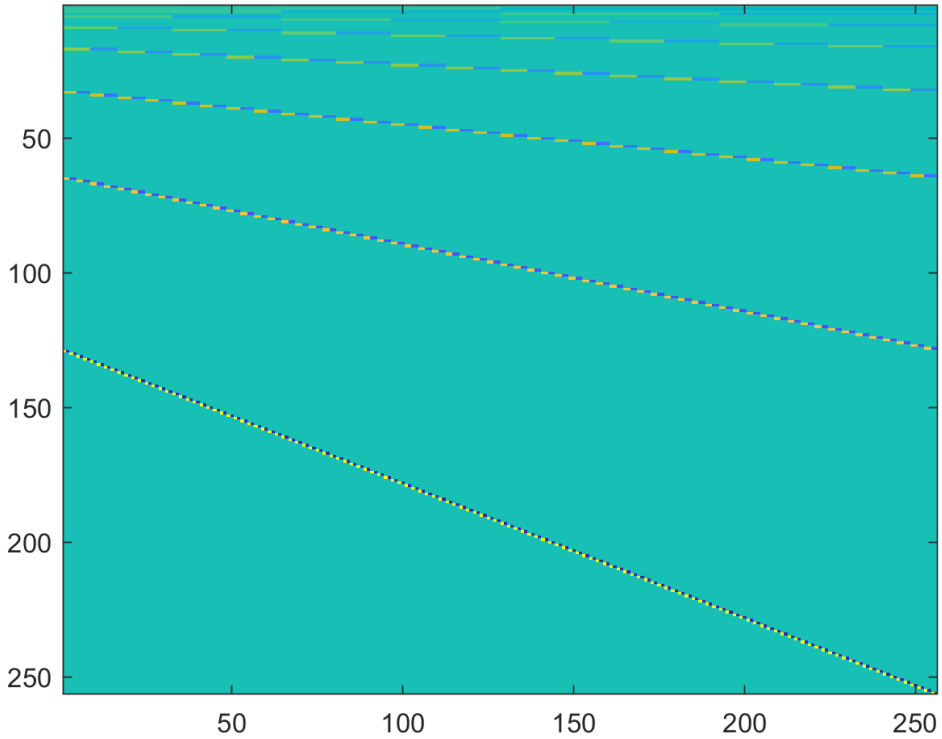
Sparsity

Wavelets

There are a great many variations on wavelets; see Matlab's **Wavelet Toolbox**. The simplest one, and the one most commonly used in Inverse Problems is the *Haar Wavelet*
We define

$$\phi^{\text{Mother}}(x) = \begin{cases} 1 & 0 \leq x < 1/2 \\ -1 & 1/2 \leq x < 1 \\ 0 & \text{otherwise} \end{cases}$$

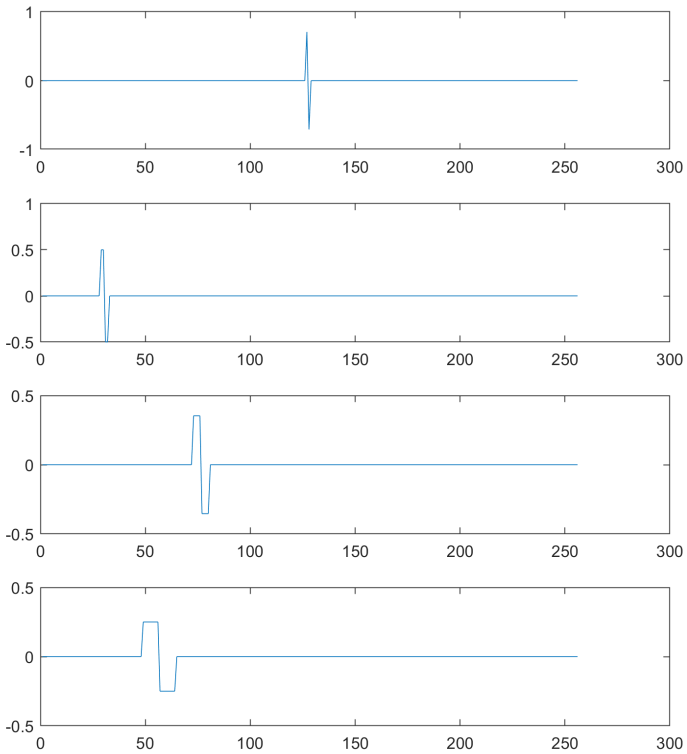
and $\phi_{m,n}(x) = 2^{-m/2}\phi^{\text{Mother}}(2^{-m}x - n)$

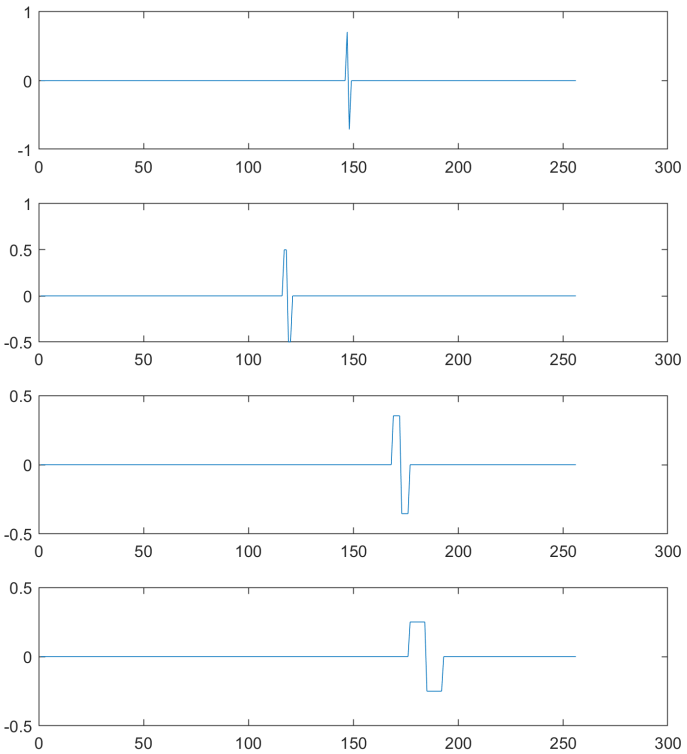


Matrix representation of the discrete Haar Wavelet Transform. Note the increasing resolution at the larger indices. Code : `Sparsity/lin1d_WaveletL1_qaudprog.m`

Sparsity

Wavelets



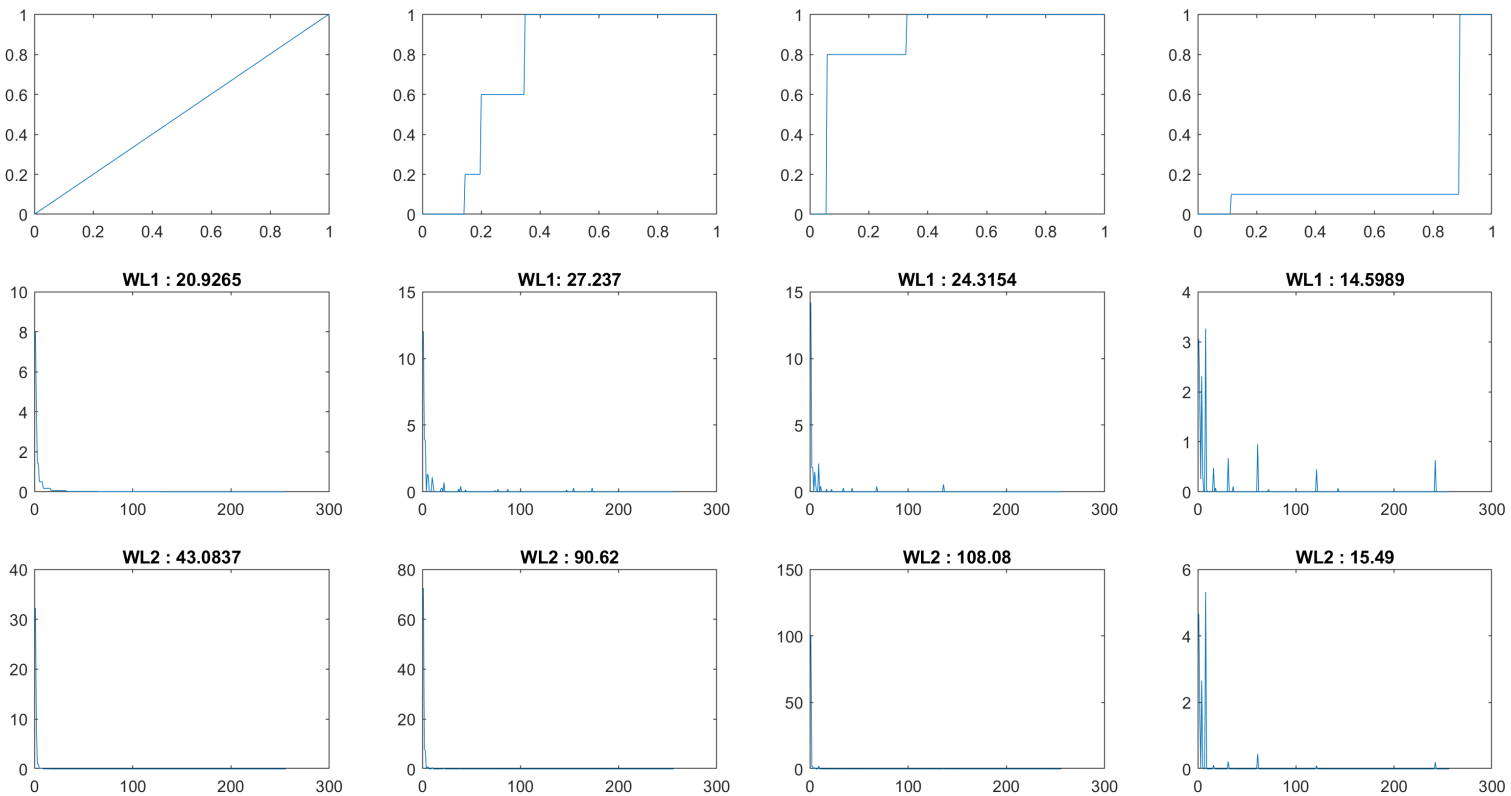


Rows of the Haar Wavelet Matrix at increasing scales.

Code : `Sparsity/lin1d_WaveletL1_qaudprog.m`

Sparsity

Wavelet Compression Illustration



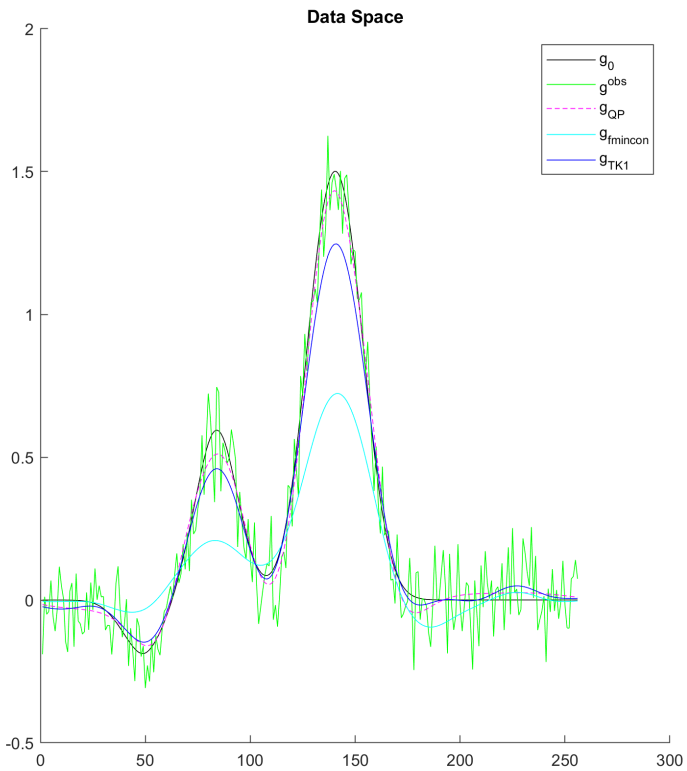
These functions all monotonically increase from 0 to 1. 2nd row are the absolute wavelet coefficients. 3rd row are square of these values.

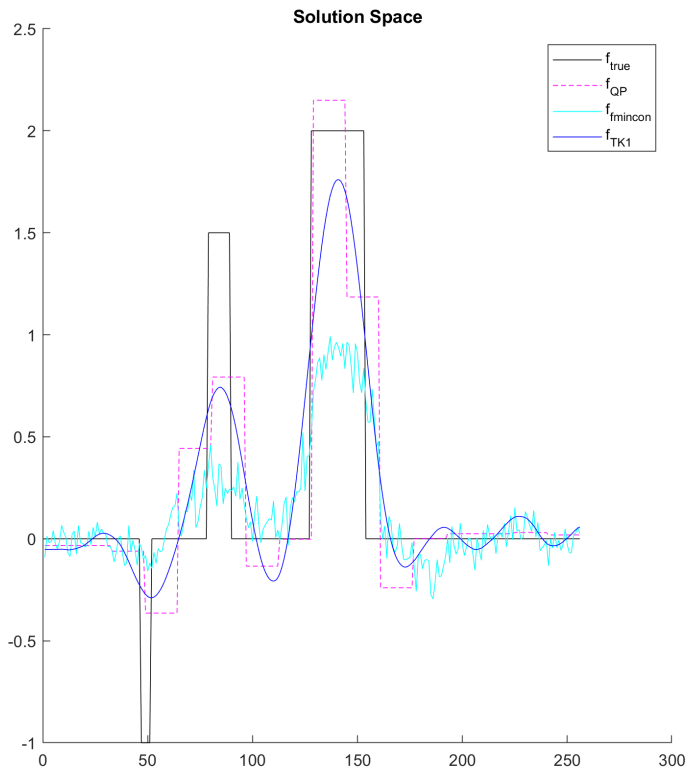
Code : `Sparsity/lin1d_WaveletL1_qaudprog.m`

Sparsity

Wavelet L1 Results

Wavelet transform can be used in place of finite-differencing as a regulariser.





Comparison of results for the 1D deblurring problem, using TK1 and Wavelet-L1. Code : `Sparsity/lin1d_WaveletL1_qaudprog.m`

For general kernels, the sparse regularisation functional

$$\Psi(f) = \langle 1, \varrho(Kf)\rangle \equiv \|\varrho(Kf))\|_1 \qquad \text{(if } \varrho \text{ positive)}$$

leads to variation

$$\Psi'(x) = K^{\mathrm{T}}\varrho'(Kf)$$

e.g. for the Perona-Malik function

$$\varrho(s) = \frac{1}{2}\log(1+s^2)$$

we arrive at

$$\Psi'(x) = K^{\mathrm{T}}\frac{1}{1+(Kf)^2}(Kf)$$

where the equivalent *diffusivity* $\kappa = \frac{1}{1+(Kf)^2}$ is spatially varying.

For a bank of N kernels $\mathbf{K} = \begin{pmatrix} K_1 \\ K_2 \\ \vdots \\ K_N \end{pmatrix}$ we have to combine the filter into a diffusivity, e.g. to enforce rotational invariance [Alt et al 2021] For example, analagous to wavelets, (see eq. (24)) a natural idea would be

$$\varrho(f, \{T_0, \ldots, T_{NW}\}) = \frac{1}{2} \log \left(1 + \sum_{\ell=0}^{NW} \frac{\|\mathbf{K}_\ell f\|^2}{T_\ell} \right)$$

where $\|\mathbf{K}_\ell f\| = \sqrt{(K_\ell^{\text{horz}} f)^2 + (K_\ell^{\text{diag}} f)^2 + (K_\ell^{\text{vert}} f)^2}$, leading to

$$\Psi'(x) = \mathbf{K}^{\mathrm{T}} \text{diag} \left[\frac{1}{1 + (\|\mathbf{K}_\ell f\|)^2} \right] (\mathbf{K} f)$$

More generally, an RMS sum over kernels at level ℓ with the same number of vanishing moments;

Outline

Summary

- The basic concepts of Regularisation
- The concepts of filtering, penalty functions and log likelihood
- Sparsity and proximal operators
- Generalised regularisation