22. Signal Subspaces and Sparsity

ECE 830, Spring 2014

Signal Subspaces and Sparsity

Recall the classical linear signal model:

$$X = H\theta + w, \quad w \sim N(0, \sigma_w^2 I)$$

where $S = H\theta$, is a linear-parametric model for the signal and w is noise. Here H is a known $n \times k$ matrix, whose columns span the signal subspace, and $\theta \in \mathbb{R}^k$ are the signal parameters. The MLE of θ is:

$$\widehat{\theta}_{MLE} = (H^T H)^{-1} H^T X$$

and the MLE of the signal is:

$$\widehat{S} = H\widehat{\theta} = \underbrace{H(H^T H)^{-1} H^T}_{P_H} X$$

where $P_H = H(H^H H)^{-1} H^T$ is the orthogonal projection operator on to the signal subspace.

The Bayes MMSE estimator based on a prior $\theta \sim \mathcal{N}(0, \sigma_\theta^2 I)$ is the Wiener filter (posterior mean and MAP estimator):

$$\widehat{\theta}_{\text{Wiener}} \;=\; H^T\Big(HH^T + \frac{\sigma_w^2}{\sigma_\theta^2} I\Big)^{-1} X \;=\; \Big(\frac{\sigma_\theta^2}{\sigma_\theta^2 + \sigma_w^2}\Big)\,\widehat{\theta}_{MLE}$$

This follows directly from the Gauss-Markov Theorem. And as the SNR grows

$$H^T\Big(HH^T + \frac{\sigma_w^2}{\sigma_\theta^2} I\Big)^{-1} \longrightarrow (H^T H)^{-1} H^T \qquad \text{as} \qquad \frac{\sigma_w^2}{\sigma_\theta^2} \longrightarrow \infty$$

So in the high SNR situation, the Wiener filter acts essentially the same as the MLE; it projects X onto the signal subspace. At low SNR the Wiener filter "shrinks" the MLE toward zero to balance the tradeoff between bias and variance.

Sparsity

In the classic set-up:

$$X = H\theta + w$$

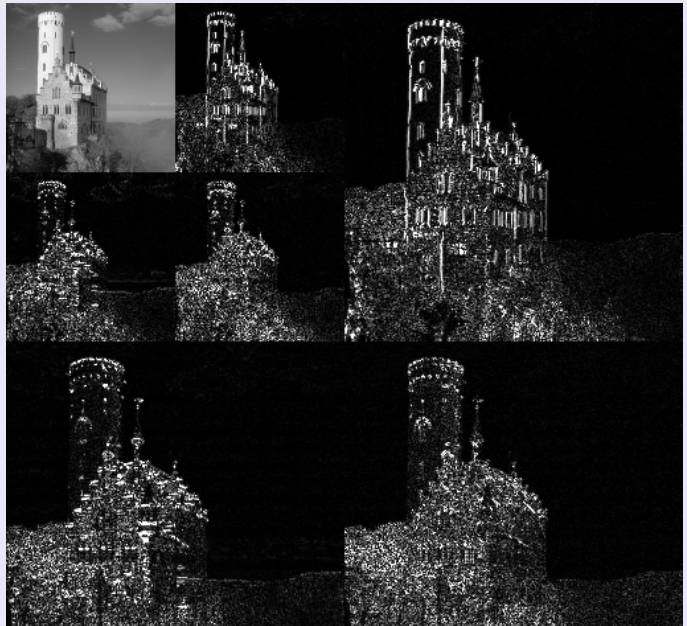
we assume that we know the low-dimensional signal subspace. In many problems we may not have this information, but we might know that the signal lies in one of many subspaces in a certain transform domain.

Example: Narrowband Communications

The communication signal lies in one of many narrow frequency bands, but we may not know which band it will be in (e.g. frequency hopping communication). If x is the signal and U is the DFT, then $\theta = U^T x$ is a sparse vector (i.e., there are just a few non-zero frequencies), but it is not known which frequencies will have non-zero coefficients.

Example: Wavelet-based Image Processing

The discrete wavelet transform (DWT) is very effective at compressing natural images. In fact it is the basis of the JPEG-2000 standard. The DWT of images tends to be "sparse" in the following sense. If x is an image and U denotes the DWT, then the DWT coefficients $\theta = U^T x$ tend to be mostly zero (or very nearly zero). The locations of the relatively few non-zero (or significant) coefficients in the vector θ depend on x in a complicated way. So, while images do approximately lie in a subspaces of the wavelet domain, the subspace is different for each different image.



Sparse Signal Models

Let U be an $n \times n$ matrix whose columns form an orthobasis for $\mathbb{R}^n$. For example, U could be the DFT or DWT. The signal of interest is represented in this domain as $s = U\theta$. Consider the observation model

$$x = U\theta + w, \; w \sim N(0, \sigma_w^2 I) \; .$$

An equivalent observation model is:

$$\begin{aligned} U^T x &= U^T U\theta + u^T w \\ &= \theta + w' \end{aligned}$$

where $w' \sim \mathcal{N}(0, \sigma_w^2 U^T U)$. Since the columns of U are orthonormal, $U^T U = I_{n\times n}$, and so $w' \sim \mathcal{N}(0, \sigma_w^2 I)$. Thus, after transforming the signal by U^T we have a direct observation of θ plus Gaussian white noise. The problem of estimating θ is called *denoising*.

If we make no assumption about the θ, then we could use the MLE:

$$\widehat{\theta}_{\text{MLE}} \;=\; U^T x \;.$$

The MLE of the signal is then $\widehat{s}_{\text{MLE}} = U\widehat{\theta}_{\text{MLE}} = UU^T x = x$, since $UU^T = I_{n\times n}$. If we suppose that the coefficients tend to have a certain energy, then we could use the prior: $\theta \sim \mathcal{N}(0, \sigma_\theta^2 I)$ and the Wiener filter:

$$\widehat{\theta}_{\text{Wiener}} \;=\; U^T\left(UU^T + \frac{\sigma_w^2}{\sigma_\theta^2} I\right)^{-1} x$$

Since U is an orthonormal transform $UU^T = I$ and the wiener filter simplifies to:

$$\begin{aligned}\widehat{\theta}_{\text{Wiener}} &= \; U^T\left(I + \frac{\sigma_w^2}{\sigma_\theta^2}\right)^{-1} x \\ &= \; \left(\frac{\sigma_\theta^2}{\sigma_\theta^2 + \sigma_w^2}\right) U^T x = \; \left(\frac{\sigma_\theta^2}{\sigma_\theta^2 + \sigma_w^2}\right) \widehat{\theta}_{\text{MLE}}\end{aligned}$$

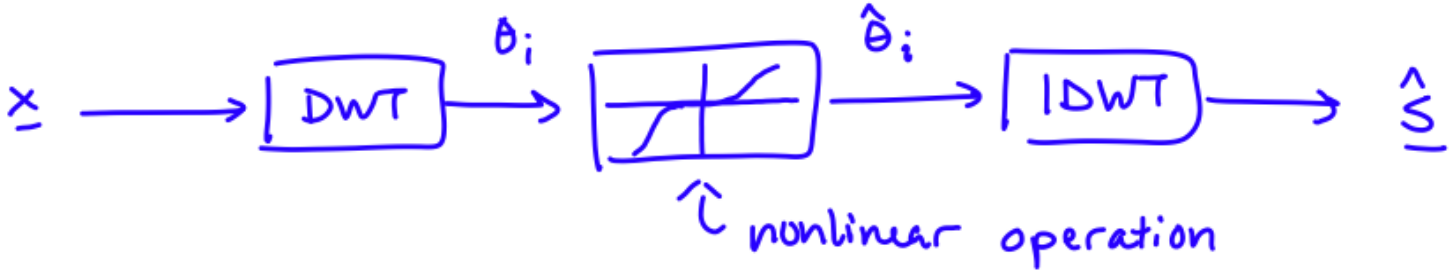
and we see that the Wiener filter is simply shrinking the MLE according to the SNR.

Now suppose our prior knowledge about θ is that it is sparse; i.e. many or most of the coefficients are zero (or near zero). This is not captured by the Gaussian prior, which models every coefficient as a Gaussian random variable with power σ_θ^2. If many coefficients are zero, then many should have approximately zero power! So we would like to design a prior probability density that reflects our belief that most of the coefficients are zero or near zero in magnitude.

The Big Picture

- We observe $x = s + v$, where $v \sim \mathcal{N}(\mathbf{0}, \sigma^2 I)$
- Compute $y = W^T x = \theta + z$ by taking wavelet transform of data
- View $y_i = \theta_i + z_i$, where $z_i \sim \mathcal{N}(0, \sigma^2)$, as independent estimation problems
- Leave "coarse" coefficients unprocessed: noise will be "averaged out" since these are local averages.
- Assume a sparse prior on the detail coefficients and estimate $\widehat{\theta}_i = \mathbb{E}[\theta_i | y_i]$
- Apply inverse wavelet transform to obtain

$$\widehat{s} = W\widehat{\theta}$$

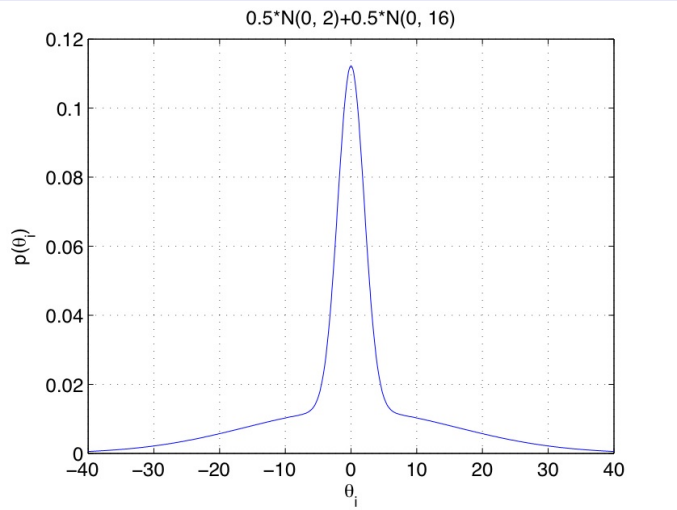


Example: Gaussian mixture

Let $\theta_1, \ldots, \theta_n$ denote the coefficients and model them as follows:

$$\theta_i \overset{\text{iid}}{\sim} \epsilon\,\mathcal{N}(0, \sigma_1^2) + (1-\epsilon)\,\mathcal{N}(0, \sigma_0^2)\ , \text{ for } i = 1, \ldots, n$$

with $\sigma_0^2 << \sigma_1^2$ and $\epsilon \approx 0$. In words this prior is saying that a large fraction $(1-\epsilon)$ of the coefficients tend to be very small in magnitude (i.e. $|\theta_i| \sim \sigma_0$) and ϵ tend to be large.



Example of Gaussian-mixture prior

Mixture Modeling

For instance, when U corresponds to the DWT, we view the detail coefficients $\theta_2, \theta_3, \ldots, \theta_N$ as realizations of a single random variable θ. We know:

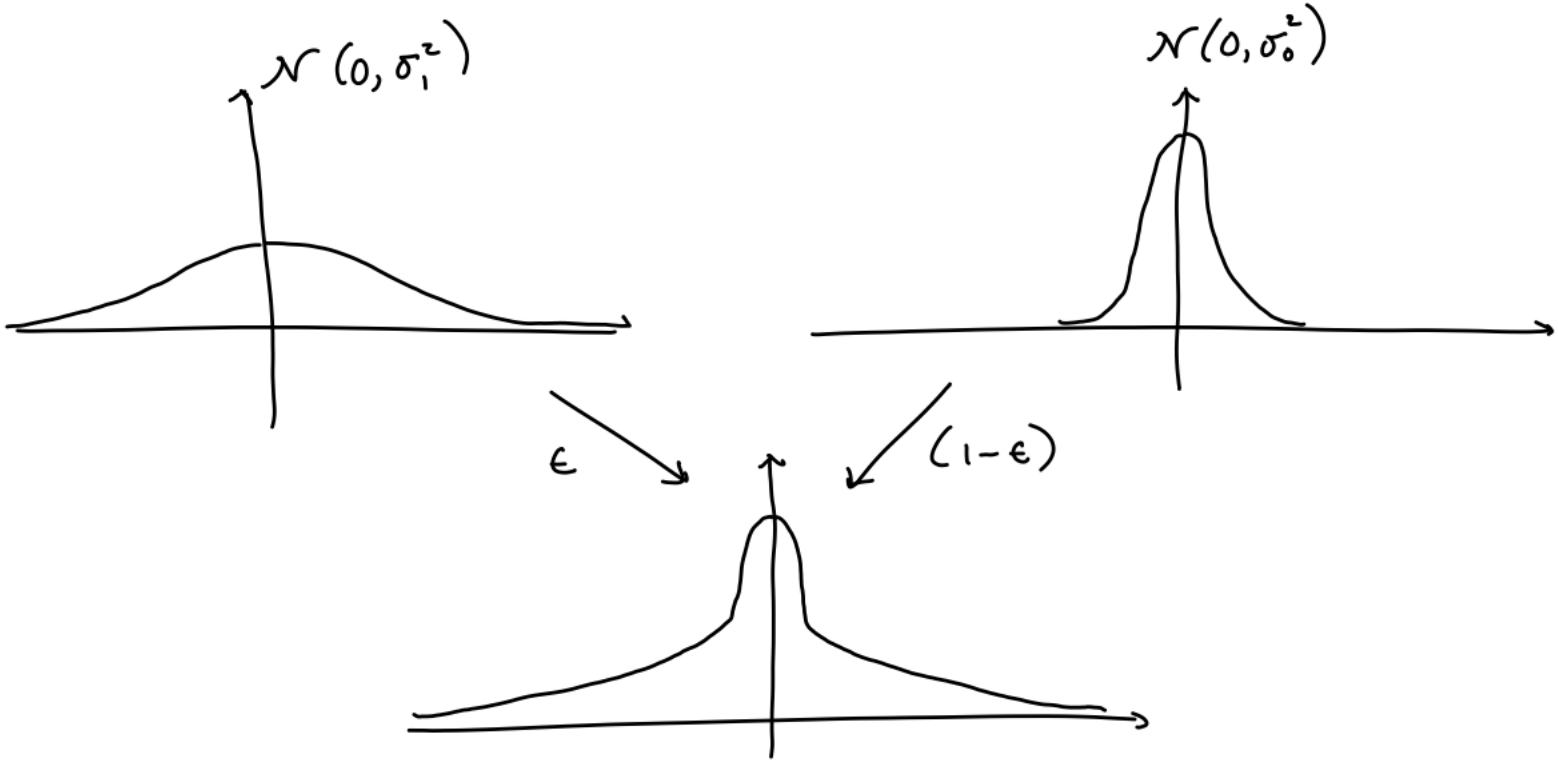
- Most θ_i are small (sparsity assumption)
- Some θ_i are large (W is orthogonal, so energy must be preserved)
- θ_i is zero-mean, since θ_i are local differences
- θ_i are "approximately" independent, since the θ_i are local differences

This suggests the following prior:

$$\theta_i \sim \epsilon \mathcal{N}(0, \sigma_1^2) + (1 - \epsilon)\mathcal{N}(0, \sigma_0^2)$$

where $\sigma_2 \ll \sigma_1$.

- $0 < \epsilon < 1$ is the proportion of "significant" coefficients
- σ_1^2 is the variance of "significant" coefficients
- σ_2^2 is the variance of "insignificant" coefficients



According to this prior, a detail coefficient θ_i is generated according to the following algorithm:

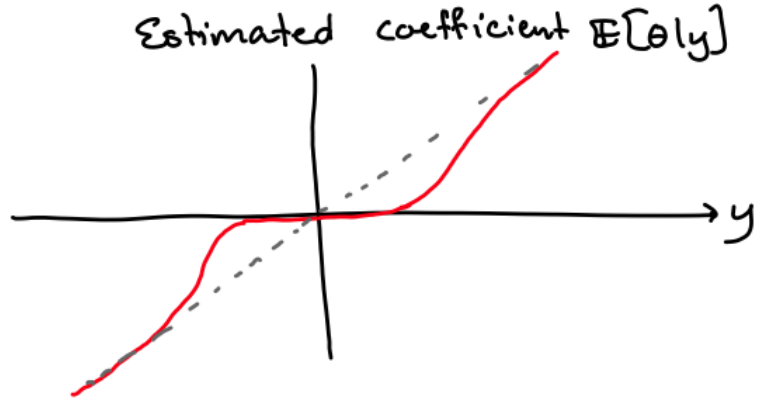
1. flip an ϵ-coin
2. if heads, $\theta_i \sim \mathcal{N}(0, \sigma_1^2)$, if talks, $\theta_i \sim \mathcal{N}(0, \sigma_2^2)$.

Gaussian Mixture: Shrinkage

You will show on the homework that

$$\widehat{\theta} = \mathbb{E}[\theta|y] = \tau(y) \cdot y$$

where $0 < \tau(y) < 1$ is called the shrinkage factor.



The effect of the shrinkage factor is that

- small coefficients are set nearly to zero
- large coefficients are virtually unaltered

This property is consistent with our understanding

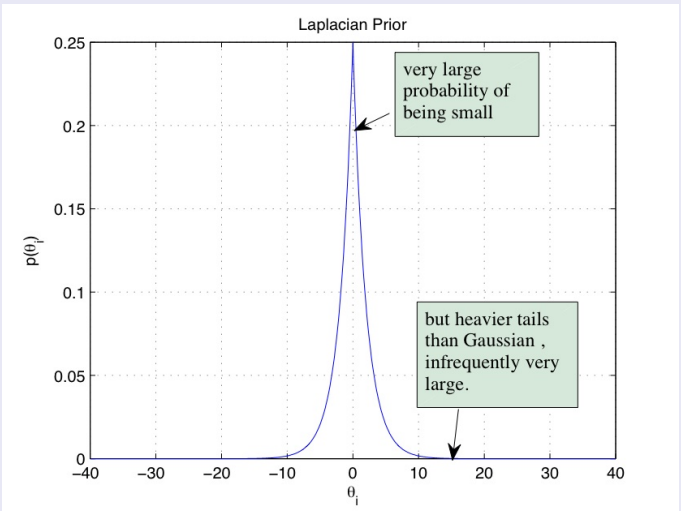
- small coefficients are mostly noise
- large coefficients contain actual signal

Example: Laplacian prior

Let $\theta_1, \ldots, \theta_n$ denote the coefficients and model them as follows:

$$\theta_i \sim \frac{\lambda}{2} e^{-\lambda|\theta_i|} , \qquad i = 1, \ldots, n$$

We will focus on the Laplacian prior because it leads to very simple and intuitive solutions to the denoising problem and it is log-concave, which makes it computationally tractable when used in inverse problems such as deconvolution.



Example of Laplacian prior

Laplacian priors for sparsity

Assume the prior

$$p(\theta) \;=\; \prod_{i=1}^{n} p(\theta_i) \;=\; \prod_{i=1}^{n} \frac{\lambda}{2}\, e^{-\lambda|\theta_i|}$$

and observation model

$$x = U\theta + w \quad , \quad w \sim \mathcal{N}((0, \sigma^2 I)$$

or equivalently

$$U^T x = \theta + U^T w$$

Recall that $U^T w \sim \mathcal{N}(0, \sigma^2 I)$. Defining $y = U^T x$, we have the model

$$y = \theta + w \quad , \quad w \sim \mathcal{N}(0, \sigma^2 I)$$

The likelihood of y given θ is

$$p(y|\theta) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y_i - \theta_i)^2}{2\sigma^2}}$$

The posterior distribution of θ is

$$\begin{aligned} p(\theta|y) &\propto p(y|\theta)p(\theta) \\ &= \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y_i-\theta_i)^2}{2\sigma^2}} \frac{\lambda}{2} e^{-\lambda|\theta_i|} \end{aligned}$$

Consider the MAP estimator

$$\begin{aligned} \widehat{\theta} &= \arg\max_{\theta} p(\theta|x) \\ &= \arg\max_{\theta} \log(p(\theta|x)) \\ &= \arg\max_{\theta} \sum_{i=1}^{n} \Big[-\frac{(y_i-\theta_i)^2}{2\sigma^2} - \lambda|\theta_i| \Big] + \text{constant} \\ &= \arg\min_{\theta} \sum_{i=1}^{n} \Big[\frac{(y_i-\theta_i)^2}{2\sigma^2} + \lambda|\theta_i| \Big] \end{aligned}$$

If $\theta_i \neq 0$, then we can differentiate to obtain

$$-\frac{(y_i - \theta_i)}{\sigma^2} + \lambda \mathsf{sign}(\theta_i) = 0$$
$$\Rightarrow \ \theta_i = y_i - \lambda\sigma^2 \mathsf{sign}(\theta_i)$$

and clearly the minimizer must have the same sign as y_i, and so

$$\widehat{\theta}_i = y_i - \lambda\sigma^2 \mathsf{sign}(y_i)$$

Plugging this into the argument of the minimization yields

$$\begin{aligned} \frac{(y_i - \widehat{\theta}_i)^2}{2\sigma^2} + \lambda|\widehat{\theta}_i| &= \frac{\lambda^2\sigma^4}{2\sigma^4} + \lambda|y_i - \lambda\sigma^2 \mathsf{sign}(y_i)| \\ &= \frac{\lambda^2\sigma^2}{2} + \lambda|y_i - \lambda\sigma^2 \mathsf{sign}(y_i)| \end{aligned} \tag{1}$$

On the other hand if $\widehat{\theta}_i = 0$, then the objective function's value is

$$\frac{(y_i - \widehat{\theta}_i)^2}{2\sigma^2} + \lambda|\widehat{\theta}_i| = \frac{y_i^2}{2\sigma^2} \tag{2}$$

Observe that

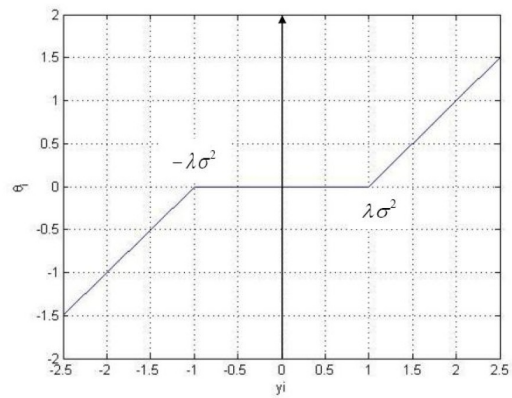
$$(1) < (2), \text{ when } |y_i| > \lambda\sigma^2$$
$$(1) > (2), \text{ when } |y_i| \leq \lambda\sigma^2$$

Therefore, the optimal solution is

$$\widehat{\theta}_i = \begin{cases} 0 & \text{if} \quad |y_i| \leq \lambda\sigma^2 \\ y_i - \lambda\sigma^2 \text{sign}(y_i) & \text{if} \quad |y_i| > \lambda\sigma^2 \end{cases}$$

This is called a "soft-threshold" function. It can be written compactly as

$$\widehat{\theta}_i = \text{sign}(y_i)\ \max(|y_i| - \lambda\sigma^2, 0)$$



The "soft-threshold" estimator is:

$$\widehat{\theta} = \begin{bmatrix} \widehat{\theta}_1 \\ \vdots \\ \widehat{\theta}_n \end{bmatrix} \quad , \quad \widehat{s} = U\widehat{\theta} = \sum_{i:|\widehat{\theta}_i| \neq 0} \widehat{\theta}_i u_i$$

where u_i is the ith column (basis vector) of U. Note that the soft-threshold estimator automatically selects a signal subspace based on the magnitude/energy of the observed data in each 1-dimensional subspace.

Summary

We studied the signal plus noise model

$$y \;=\; U^T x + w, \qquad w \sim \mathcal{N}(0, \sigma^2 I)$$

The MLE is

$$\begin{aligned}
\widehat{\theta}_{\text{MLE}} &= \arg\min_{\theta} \frac{||x - u\theta||_2^2}{2\sigma^2} \\
&= \arg\min_{\theta} \frac{||x - u\theta||^2}{2\sigma^2} \\
&= U^T x \\
&= y \;.
\end{aligned}$$

The Wiener filter (based on a Gaussian prior) is given by

$$\widehat{\theta}_{\text{Wiener}} = \left(\frac{\sigma_\theta^2}{\sigma_\theta^2 + \sigma^2}\right) y \quad , \qquad \theta \sim \mathcal{N}(0, \sigma_\theta^2 I)$$

or $$\widehat{\theta}_{\text{Wiener},i} = \left(\frac{\sigma_{\theta i}^2}{\sigma_{\theta i}^2 + \sigma^2}\right) y_i \quad , \qquad \theta_i \sim \mathcal{N}(0, \sigma_{\theta i}^2 I)$$

The soft-thresholding estimator based on the Laplacian prior $\theta_i \overset{\text{iid}}{\sim} \frac{\lambda}{2} e^{-\lambda|\theta_i|}$ has the form

$$\widehat{\theta} = \arg\min_{\theta} \frac{||y-\theta||_2^2}{2\sigma^2} + \lambda||\theta||_1$$

$$\widehat{\theta}_i = \text{sign}(y_i)\ \max(|y_i| - \lambda\sigma^2, 0)$$

$\longrightarrow$ Data-adaptive shrinkage to trade off bias and variance.

Example:

Consider the following observation

$$\sigma^2 = 1, \qquad y = \begin{bmatrix} 10 \\ 1 \end{bmatrix} \swarrow \text{ probably just noise}$$

MLE:

$$\widehat{\theta}_{\text{MLE}} = y = \begin{bmatrix} 10 \\ 1 \end{bmatrix} \qquad \text{full dimension}$$

Example: (cont.)

Wiener filter:

$$\widehat{\theta}_{\text{Wiener}} = \left(\frac{\sigma_\theta^2}{\sigma_\theta^2 + 1}\right) \begin{bmatrix} 10 \\ 1 \end{bmatrix} \quad \propto \quad \begin{bmatrix} 10 \\ 1 \end{bmatrix} \qquad \text{full dimension}$$

Soft-Threshold:

$$\widehat{\theta}_{\text{ST}} = \begin{bmatrix} \max(10 - \lambda, 0) \\ \max(1 - \lambda, 0) \end{bmatrix} \quad \overset{\lambda=1}{=} \quad \begin{bmatrix} 9 \\ 0 \end{bmatrix} \qquad \text{shrink to 1-dimension}$$

Inverse problems

Suppose we observe a distorted signal s in noise:

$$\begin{aligned} x &= As + w \\ &= AU\theta + w \quad , \quad w \sim \mathcal{N}(0, \sigma^2 I) \end{aligned} \tag{1}$$

A is a known matrix, suppose s is sparse in basis U, and write $s = U\theta$.

Wiener Filter (with Gaussian Prior): $\theta \sim \mathcal{N}(0, \lambda I)$

$$\widehat{\theta}_{\text{Wiener}} = \arg\min_{\theta} \left(\frac{||x - AU\theta||_2^2}{2\sigma^2} + \lambda ||\theta||_2^2 \right)$$

$\Rightarrow$ linear, non-adaptive.

Sparse Solution (Laplacian Prior): "Lasso"

$$\widehat{\theta}_L = \arg\min_{\theta} \underbrace{\left(\frac{||x - AU\theta||_2^2}{2\sigma^2} + \lambda ||\theta||_1 \right)}_{\text{"Lasso"}}$$

$\Rightarrow$ non-linear, adaptive.

Both are convex optimizations. The Wiener filter, which is linear, has a linear-algebraic solution. The Lasso (Least absolute shrinkage and selection operator) is nonlinear, and does not have a simple closed-form solution (except when $A = I$). Convex optimization methods like GPSR or SPARSA (`http://www.lx.it.pt/~mtf/GPSR/`) or SPARSA (`http://www.lx.it.pt/~mtf/SpaRSA/`) can be used to solve the Lasso optimization.