Denoising Lena

http://freemind.pluskid.org/machine-learning/denoising-lena

posted on Free Mind on March 19, 2015
generated with pandoc on December 3, 2015
category: Machine Learning

tags: Sparsity, Signal Processing, Algorithm

在之前讲 Projected Gradient Method 的文章中，我们举了一个对加了白噪声的 Lena 的照片进行 denoising 的例子，文中提到了两种方法，一种是直接对 DWT 的系数进行 hard thresholding，将数值较小的值设为零，再用逆向离散小波变换得到 denoising 之后的图片。另一种方法是解一个 ℓ_1 正则化的线性回归，我们选了后者因为刚好那个优化问题经过变化之后可以用 Projected Gradient Method 来解，这也是当时选这个问题作为那篇文章的原因。但是当时并没有解释为什么这些算法可以实现降噪，而这就是今天的话题。

当然，直观来讲，是不难理解的，因为 natural image 在小波基下呈现稀疏性，而白噪声，也就是 Gaussian Noise，则没有稀疏性，另外假设 noise 的 scale 和原始信号相对来说比较小的话，那么通过 hard thresholding，去掉那些较小的系数之后得到的稀疏系数会达到一定的降噪效果。我们在这里试图将问题 formally 定义出来。首先，我们假设 Lena 的图片是这样生成的

$$y = Xw^* + \epsilon \tag{1}$$

其中 $X \in \mathbb{R}^{d\times d}$ 是正交小波基，$w^* \in \mathbb{R}^d$ 是真实的 Lena 的照片在小波基下的系数，而 $\epsilon \sim \mathcal{N}(0, \sigma^2 I_d)$ 是零均值的高斯分布噪音。实际上我们并不需要高斯分布，在后面的分析中我们只需要 ϵ 的分布的 tail decay 得足够快就可以。比如任何 bounded 随机变量，或者 sub-Gaussian 随机变量都是可以的。当然我们所能达到的降噪的效果会取决于噪音的 variance σ^2 的大小。

因此，我们所观测到的 y，会是原始图像加上未知噪音 ϵ 的结果，我们知道作为小波基的 X，但是却不知道原始的小波系数 w^*，而我们的目的正是要构造一个 estimator $\hat{w}$，而我们这里将使用 mean square error (MSE) 来衡量一个 estimator 的好坏：

$$\mathsf{MSE}(X\hat{w}) = |Xw^* - X\hat{w}|_2^2 \tag{2}$$

注意 MSE 是一个随机变量，随机性来自 $\hat{w}$，因为 $\hat{w}$ 是根据 y 构造出来的，而 y 则依赖于随机变量 ϵ。这里看起来有点像 machine learning 的设定：X 是数据，w^* 是模型参数，而 y 是带噪声的 label，但是实际上设定并不太一样，首先这里的 X 是固定的 (fixed design)，并不是像 machine

learning 里是从一个数据分布中随机采样得到的 (random design)，其次这里的目的是估计 w^*，并且衡量标准用重构出来的图片 $X\hat{w}$ 和真实的图片 Xw^* 进行比较，而在 machine learning 中衡量标准则是要对于未知的，新的数据下的预测结果和真实结果进行比较。

在构造 estimator 之前，我们先对问题进行一些简单的变换，首先注意到小波基是一个 orthonormal basis，也就是说 $X^\top X = I$，因此，我们在模型 (1) 两边同时乘以 $X^\top$ 就可以得到：

$$X^\top y = w^* + X^\top \epsilon$$

记 $X^\top y = Y$，$X^\top \epsilon$ 为 ξ，我们可以得到如下的新模型

$$Y = w^* + \xi \tag{3}$$

其中 Y 是观测到的量，ξ 是 (变换过的噪音)，由于 $X^\top$ 是 orthonormal 的，所以 $\xi \sim \mathcal{N}(0, \sigma^2 I_d)$，和原来的噪音是同分布的。类似地，MSE 可以变换为

$$\begin{aligned}\mathsf{MSE}(X\hat{w}) &= (w^* - \hat{w})^\top X^\top X (w^* - \hat{w}) \\ &= |w^* - \hat{w}|_2^2\end{aligned}$$

现在问题变得简介了许多：我们观测到一个未知的向量 w^* 加成了高斯噪音的结果，现在想要估计 w^* 使得估计值和真实值的 ℓ_2 距离平方最小。光这样似乎并不是特别明显我们可以做什么，如果有多个观测值的话我们似乎还可以求求平均之类的，现在只有 Y 这一个观测值。

接下来我们不妨来分析一下 Lena 和噪声各自的性质。首先 Lena 作为一张 natural image，在小波基下的系数 w^* 是应当呈现稀疏性的。另一方面注意到高斯噪声有一个很好的性质就是它的分布的 tail decay 得很快。例如，根据 Wikipedia，一个高斯分布的随机变量取值在均值加减 3σ 的范围内的概率是 99.7%，这里 σ^2 是这个随机变量的方差。

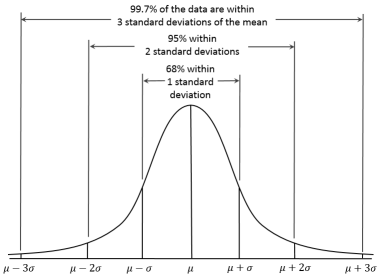

Figure 1: 正态分布的 tail decay。图片来自 Wikipedia。

根据这个，我们可以以很大的概率确定如下情况：$|\xi_i|$ 都是很小的；因此如果观测值 $|Y_i|$ 是一个很大的值，那么说明在 i index 下的真实值 $|w_i^*| \neq 0$，因此我们观测到的是真实值加噪音；另一方面，如果 $|Y_i|$ 是一个很小的值，那么有两种情况，一是由于稀疏性，原始信号在这个位置的系数就是零，或者是原始信号虽然非零但是绝对值很小。这些分析下下面的所谓 hard threshold estimator 就变得 make sense 了：

$$\hat{w}^{\mathrm{HRD}}(Y)_i = \begin{cases} Y_i & |Y_i| > 2\tau \\ 0 & |Y_i| \leq 2\tau \end{cases} \tag{4}$$

这里的 threshold 2τ 是什么我们接下来会讨论，简单来说就是，如果 $|Y_i|$ 很大，那么观测到的是信号加噪音，由于我们不知道噪音具体是多少反正噪音相对于信号来说比较小，就索性留着；但是反过来 $|Y_i|$ 很小的情况，如果信号原来在这里是稀疏的，那么正好我们设为零估计正确，但是即使原始信号在这里不稀疏，其绝对值也是很小的，因此我们设为零之后造成的估计误差也不会太大。

接下来就让我们把这个 idea 具体地用数学语言描述出来。首先，让我们来具体刻画一下高斯分布的 tail decay。具体来说，假设 Z 是均值为零，方差为 σ^2 的高斯分布，对于任意的 $t > 0$，我们希望计算 $Z > t$ 的概率：

$$\begin{aligned} P(Z > t) &= \int_t^\infty p_Z(z)\,dz \\ &= \int_t^\infty \frac{1}{\sqrt{2\pi\sigma^2}} e^{-z^2/2\sigma^2}\,dz \end{aligned}$$

排除 Q-function 这种耍赖的存在的话，这个积分并没有一个表达式可以直接写出来，如果我们通过数值方法，可以算出类似刚才的 $e\sigma$ 那样的数值来，不过我们这里希望有一个表达式，由于我们只是希望噪音 decay 的很快，也就是说在 t 变大的时候 $P(Z > t)$ 变小得非常快，因此我们并不需要得到 exact 的等式，只要得到一个足够好的上界不等式就好了。这里有一个简单的方法，注意到当 $z \geq t$ 时，$z/t \geq 1$，因此

$$\begin{aligned} \int_t^\infty \frac{1}{\sqrt{2\pi\sigma^2}} e^{-z^2/2\sigma^2}\,dz &\leq \int_t^\infty \frac{z}{t} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-z^2/2\sigma^2}\,dz \\ &= \frac{\sigma}{\sqrt{2\pi}t} e^{-t^2/2\sigma^2} \end{aligned}$$

从这里可以看出，高斯分布的 tail 是以 $e^{t^2/2\sigma^2}$ 来 decay 的，这将随着 t 的增大以非常快的速度趋向于 0，σ^2 越小速度越快。这里我们暂时忽略前面的 $\sigma/\sqrt{2\pi}t$ 的系数，注意到当 t 变得很小的时候这个系数变得非常大，此时这个上界就失去意义了，因为当 $t > 0$ 时我们显然有 $P(Z > t) < 1/2$。实际上，通过 Chernoff Bound 可以直接得到这样的方便处理的上界：

$$P(Z > t) \leq e^{-t^2/2\sigma^2}$$

虽然推导并不复杂，但是篇幅有限，为了避免扯得太远，这里就直接使用这个结论了。现在我们回到 ξ，刚才的分析中已经知道它是一个 d 维，零均值，协方差矩阵为 $\sigma^2 I_d$ 的高斯分布变量，由于各个维度互相独立，通过简单的 union bound 和对称性，我们可以得到：

$$P(|\xi_i| > t) \leq P(\xi_i > t) + P(\xi_i < -t) \leq 2e^{-t^2/2\sigma^2} \tag{5}$$

如果我们想要同时控制所有的 ξ_i 的话，同样利用 union bound 可以得到

$$\begin{aligned} P\left(\max_{1\le i\le d}|\xi_i|>t\right) &= p\left(\bigcup_{i=1}^{d}\{|\xi_i|>t\}\right) \\ &\le \sum_{i=1}^{d} P(|\xi_i|>t) \\ &\le 2de^{-t^2/2\sigma^2} \end{aligned} \tag{6}$$

令右边的式子等于 δ，反解出 t，我们可以得到，对于任意 $\delta>0$，我们可以得到，以至少 $1-\delta$ 的概率，如下式子成立：

$$\max_{1\le i\le d}|\xi_i| \le \sigma\sqrt{2\log(2d/\delta)} \tag{7}$$

也就是说，以很高的概率，我们可以将所有 $|\xi_i|$ 控制在上面的 bound 以内，大约就是 σ 的 scale，当然会随着 $1/\delta$ 和 d 增大，但是都是根号下的 log-scale，增长比较缓慢，基本上可以接受。既然知道了噪音的大致 scale，那么我们不妨将 hard threshold estimator 里的 τ 取成这里的噪音上界。具体来说，我们将得到如下结论。

定理 1 假设模型满足 (3)，那么令

$$\tau = \sigma\sqrt{2\log(2d/\delta)}$$

则对于任意 $\delta>0$，如 (4) 所定义的 *Hard Threshold Estimator* 可以以至少 $1-\delta$ 的概率实现

$$|\hat{w}^{HRD}-w^*|_2^2 \lesssim k\sigma^2\log(2d/\delta)$$

其中 $k=|w^*|_0$ 是真实系数的稀疏性。此外，如果最小的非零 $|w_i^*|$ 数值都大于 3τ 的话，以同样的概率可以实现

$$supp(\hat{w}^{HRD}) = supp(w^*)$$

也就是说估计出来的系数有正确的稀疏性。

这里 $\lesssim$ 是用于省略掉其中的一些常量的写法。在证明之前我们先来看一下结论。首先，MSE 随着 σ^2 的增大而增大，噪音越大，估计就越差，这是理所当然的事情。通常，我们都会要求噪音是足够小的，例如，如果 σ 是在 $1/d$ 的 scale 上的话，上面的式子就会变得非常好看。另外前面的系数 k 相当于是必须要 pay 的 price，因为我们这里需要估计的参数有 k 个（w^* 的稀疏性）。实际上，假设 w^* 不具有稀疏性，此时我们直接用 least square estimator，也就是

$$\hat{w} = \arg\min_{w} |Y - w|_2^2$$

很显然最优解就是 $\hat{w} = Y$，此时的 MSE 为 $|\hat{w} - w^*|_2^2 = |\xi|_2^2$，根据我们刚才得到的高斯分布的 tail bound，再加上 union bound，可以得到

$$\begin{aligned} P(|\xi|_2^2 > t) &\leq \sum_{i=1}^{d} P\left(|\xi_i|^2 > \frac{t}{d}\right) \\ &\leq \sum_{i=1}^{d} P\left(|\xi_i| > \sqrt{\frac{t}{d}}\right) \\ &\leq 2d \exp\left(-\frac{t}{2\sigma^2 d}\right) \end{aligned}$$

令右边等于 δ，我们可以得到，以至少 $1-\delta$ 的概率，

$$|\hat{w} - w^*|_2^2 \leq 2d\sigma^2 \log(2d/\delta)$$

可以看到我们得到的 bound 和 $\hat{w}^{\text{HRD}}$ 是类似的，只是现在需要估计的参数是 d 个（由于没有稀疏性）。现在假设我们知道 w^* 的稀疏性是 k，并且知道 w^* 在哪 k 个位置上是非零的，此时我们可以只对这 k 个位置通过 least square 进行估计，将会得到和 $\hat{w}^{\text{HRD}}$ 一样的 bound。也就是说，hard threshold estimator 在对 k 的值以及是哪 k 个位置上非零这些情报毫不知情的情况下，达到了和知道这些情报的情况下所能得到的差不多的估计误差，因此 hard threshold estimator 又被称为 sparsity adaptive thresholding estimator。接下来我们来证明该定理。

为方便起见，以下我们就记 $\hat{w}^{\text{HRD}}$ 为 $\hat{w}$，首先注意到

$$\begin{aligned} |\hat{w}_i - w_i^*| &= |\hat{w}_i - w_i^*| \mathbf{1}_{Y_i > 2\tau} + |\hat{w}_i - w_i^*| \mathbf{1}_{Y_i \leq 2\tau} \\ &= |\xi_i| \mathbf{1}_{Y_i > 2\tau} + |w_i^*| \mathbf{1}_{Y_i \leq 2\tau} \\ &\leq \tau \mathbf{1}_{Y_i > 2\tau} + |w_i^*| \mathbf{1}_{Y_i \leq 2\tau} \end{aligned}$$

其中最后一个不等式根据刚才对高斯分布的 tail 的分析 (7)，是以至少 $1-\delta$ 的概率成立，其中 τ 就是取定理中所指定的值。此外，注意到，根据三角不等式

$$\begin{aligned} |Y_i| > 2\tau &\Rightarrow |w_i^*| = |Y_i - \xi_i| \geq |Y_i| - |\xi_i| > \tau \\ |Y_i| \leq 2\tau &\Rightarrow |w_i^*| = |Y_i - \xi_i| \leq |Y_i| + |\xi_i| \leq 3\tau \end{aligned}$$

因此，接着上面的不等式

$$|\hat{w}_i - w_i^*| \leq \tau \mathbf{1}_{|w_i^*| > \tau} + |w_i^*| \mathbf{1}_{|w_i^*| \leq 3\tau}$$

我们将右边的式子分情况展开可以得到

$$\begin{aligned}
\tau \mathbf{1}_{|w_i^*|>\tau} + |w_i^*|\mathbf{1}_{|w_i^*|\leq 3\tau} &= \begin{cases} \tau + |w_i^*| & \tau < |w_i^*| \leq 3\tau \\ |w_i^*| & |w_i^*| \leq \tau \\ \tau & |w_i^*| > 3\tau \end{cases} \\
&\leq \begin{cases} 4\tau & \tau < |w_i^*| \leq 3\tau \\ |w_i^*| & |w_i^*| \leq \tau \\ \tau & |w_i^*| > 3\tau \end{cases} \\
&\leq \begin{cases} 4\tau & \tau < |w_i^*| \\ |w_i^*| & |w_i^*| \leq \tau \end{cases} \\
&\leq \begin{cases} 4\tau & \tau < |w_i^*| \\ 4|w_i^*| & |w_i^*| \leq \tau \end{cases}
\end{aligned}$$

也就是说

$$|\hat{w}_i - w_i^*| \leq 4\min(\tau, |w_i^*|)$$

从而，我们可以直接得到

$$\begin{aligned}
|\hat{w} - w^*|_2^2 &= \sum_{i=1}^{d} |\hat{w}_i - w_i^*|^2 \\
&\leq 16 \sum_{i=1}^{d} \left(\min(\tau, |w_i^*|)\right)^2 \\
&\leq 16|w_i^*|_0 \tau^2
\end{aligned}$$

带入定理中 τ 的式子即证第一个结论。第二个结论很好证明，只要利用三角不等式即可。首先，如果 $w_i^* \neq 0$，此时根据假设我们有 $|w_i^*| > 3\tau$，于是

$$|Y_i| = |w_i^* + \xi_i| \geq |w_i^*| - |\xi_i| > 2\tau$$

于是 $\text{supp}(w^*) \subset \text{supp}(\hat{w})$。反过来，如果 $|Y_i| > 2\tau$，则

$$|w_i^*| = |Y_i - \xi_i| \geq |Y_i| - |\xi_i| > \tau > 0$$

于是 $\text{supp}(\hat{w}) \subset \text{supp}(w^*)$。定理即证。

结束之前，有几点注意事项。除了上面的 hard thresholding 之外，还可以定义 soft thresholding，也就是先对所有的系数的绝对值减去 2τ，然

后将不够减的这些系数设为零。通过类似的分析可以得到差不多的结论。此外这里的噪音并不限于高斯噪音，从上面的分析中看到我们只要求噪音的 tail decay 足够快即可，因此对于其他有类似于高斯噪音这样的 tail decay 速度的噪音是同样适用的。另外就是，当 X 并非正交的时候，我们也能得到一些相似的定理，但是此时必须对 X 加一些额外的条件，否则，比如一个极端的情况，如果 w^* 在 X 的 null space 里，此时 $Xw^* = 0$，那么测量到的将会完全是 noise，此时没有任何可能恢复原来系数的希望。这里需要对 X 所加的限制基本上是要求 X "近似于" 正交，具体地 formulate 出来将会得到 compressive sensing 里那些常用的诸如 incoherence、null space property 之类的性质。此外，当 X 并非正交之后，它的行数并不要求等于它的列数，compressive sensing 里的许多结论就是在说，当 w^* 本身满足一定的稀疏性之后，我实际上并不需要 d 行那么多的测量数，而是只要远远小于这个数目的行数（依赖于稀疏性 k）就能恢复原来的系数。这个时候我们得到的类似于定理中的 bound 里，上界里将会出现行数 n，并且 error bound 会随着 n 的增加而减小。

最后，理论分析里得到的 τ 的取值通常并不适用于直接在实际中带入。因为在求上界的过程中经过了许多不等式的放松，而且即便是 "tight bound"，通常都是指不考虑常数系数，以及在最坏情况下和下界达到一致之类的情况，因此主要还是 bound 里的各个参数的阶对于 bound 变化的影响是比较有用的。实际操作中通常会通过 cross validation 之类的另外的方法来选取 estimator 的参数。

由于本文没有什么图片，看起来比较枯燥，下面随便给了一段用来通过 hard thresholding 和 soft thresholding 对 Lena 的图片进行去噪的 julia 代码。

```
# Requires the following packages:
#
#   Pkg.add("Wavelets")
#   Pkg.add("Images")

using Images
using Wavelets

sigma = 0.1
dim = 512*512
delta = 0.1

img = reinterpret(Float64, float64(imread("lena512gray.bmp")))
imwrite(img, "threshold-lena-orig.jpg")

# add noise to the image
img.data += 0.2*randn(size(img.data))
```

```
imwrite(img, "threshold-lena-noisy.jpg")

the_wavelet = wavelet(WT.sym4)
img_coef = dwt(img.data, the_wavelet)

for config in ["bound", "choose"]
  if config == "bound"
    tau = sigma * sqrt(2*log(2*dim/delta))
  else
    tau = 0.25
  end

  # hard thresholding
  img_coef_hrd = copy(img_coef)
  img_coef_hrd[abs(img_coef_hrd) .<= 2tau] = 0

  rcv_img = idwt(img_coef_hrd, the_wavelet)
  imwrite(rcv_img', "threshold-lena-hrd-$config-rcv.jpg")

  # soft thresholding
  if config == "choose"
    tau = 0.1
  end
  img_coef_sft = copy(img_coef)
  coef = 1 - 2tau ./ abs(img_coef_sft)
  coef = coef .* (coef .> 0)
  img_coef_sft = img_coef_sft .* coef

  rcv_img = idwt(img_coef_sft, the_wavelet)
  imwrite(rcv_img', "threshold-lena-sft-$config-rcv.jpg")
end
```

我们对比了 hard thresholding 和 soft thresholding 使用定理中给定的 τ 和随便尝试了一下选取的 τ 值（注意“随便尝试一下”并不是一个很科学的 parameter selection 的方法）。结果图所示。

可以看到，定理中给出的 τ 值过大，通过那个值进行 thresholding 之后原本图像的细节都被去得所剩无几了。还有就是 hard thresholding 比较暴力，因此造成的图像上的 artifact 也比较严重一点，相对应的 soft thresholding 的结果就相对更加 smooth 一点。当然，总体来说，效果都只能算一般，真正要做降噪的应用的话，应该会利用更多的 prior knowledge 和更加复杂的模型的。

(a) 原图。

(b) 加白噪声图。

(c) **Hard thresholding**，用定理中的 **tau**。

(d) **Hard thresholding**，取 **tau=0.25**。

(e) **Soft thresholding**，用定理中的 **tau**。

(f) **Soft thresholding**，取 **tau=0.1**。

Figure 2: . . . . . .