This is just a quick document for describing the differences between vanilla gradient descent, iterative thresholding and AMP.

1 Gradient Descent

Suppose we observe a vector $y \in \mathbb{R}^m$, through an operator $A \in \mathbb{R}^{m \times n}$, i.e. we have the model

$$y = Ax \tag{1.0.1}$$

$x \in \mathbb{R}^n$. One way to guess x, is to minimise the mean squared error:

$$J = \frac{1}{2} ||y - Ax||_2^2 = (y - Ax)^T (y - Ax) \tag{1.0.2}$$

We can do this by gradient descent, as

$$\frac{\partial J}{\partial x} = A^T (y - Ax) \tag{1.0.3}$$

So we have an iterative algorithm:

$$x^{t+1} = x^t + A^T (y - Ax^t)$$

Putting it in a more suggestive form:

$$\begin{aligned} x^{t+1} &= x^t + A^T z^t \\ z^t &= y - Ax^t \end{aligned}$$

2 Iterative Thresholding

The algorithm from the previous section will perform poorly, as we've not used any prior information about x. We assume that x is sparse, so we now seek solutions to:

$$J = \frac{1}{2} ||y - Ax||_2^2 + \lambda \, ||x||_1 \tag{2.0.4}$$

The presence of the ℓ_1 norm, is inconvenient: J in this case has no formal gradient. In this case we can pretend that J is differentiable by doing a step in the direction of the gradient, and correcting for any error using the soft thresholding operator:$S_\gamma \left(x\right)_i = \operatorname{sign}(x_i) \left(|x_i| - \gamma\right)^+$.

So our algorithm becomes:

$$\begin{aligned} x^{t+1} &= S_\lambda \left(x^t + A^T z^t\right) \\ z^t &= y - Ax^t \end{aligned}$$

The soft-thresholding operator can be derived by considering the MAP estimate of the following model:

$$y = x + w \tag{2.0.5}$$

where x is some (sparse) signal, and w is additive white Gaussian noise. We seek

$$\hat{x} = \arg\max_x \mathbb{P}_{x|y}(x|y) \tag{2.0.6}$$

This can be recast in the following form by using Bayes rule, noting that the denominator is independent of x and taking logarithms:

$$\hat{x} = \arg\max_x \left[\log \mathbb{P}_w(y-x) + \log \mathbb{P}(x)\right] \tag{2.0.7}$$

The term $\mathbb{P}_n(y-x)$ arises because we are considering $x+w$ with w zero mean Gaussian, with variance σ_n^2. So, the conditional distribution of y (given x) will be a Gaussian centred at x.

We will take $\mathbb{P}(x)$ to be a Laplacian distribution:

$$\mathbb{P}(x) = \frac{1}{\sqrt{2}\sigma} \exp -\frac{\sqrt{2}}{\sigma}|x| \tag{2.0.8}$$

Note that $f(x) = \log \mathbb{P}_x(x) \;\; -\frac{\sqrt{2}}{\sigma}|x|$, and so by differentiating $f'(x) = -\frac{\sqrt{2}}{\sigma}\text{sign}(x)$

Taking the maximum of 2.0.7 we obtain:

$$\frac{y-\hat{x}}{\sigma_n^2} - \frac{\sqrt{2}}{\sigma} sign(x) = 0 \tag{2.0.9}$$

Which leads the soft thresholding operation defined earlier, with $\gamma = \frac{\sqrt{2}\sigma_n^2}{\sigma}$ as (via rearrangement):

$$y = \hat{x} + \frac{\sqrt{2}\sigma_n^2}{\sigma}\text{sign}(x)$$

or

$$\hat{x}(y) = \text{sign}(y)\left(y - \frac{\sqrt{2}\sigma_n^2}{\sigma}\right)_+$$

i.e $S_\gamma(y)$.

3 AMP

Approximate message passing further corrects this idea of pretending our optimisation objective is differentiable and correcting, by making a quadratic approximation to the likelihood of the model.

The algorithm becomes

$$\begin{aligned} x^{t+1} &= S_\lambda\left(x^t + A^T z^t\right) \\ z^t &= y - Ax^t + b_t z^t \end{aligned}$$

which is similar to the iterative thresholding algorithm, but with an additional 'momentum' term added. A good choice of b is:

$$\frac{1}{m}\left|\left|x^t\right|\right|_0 \tag{3.0.10}$$

where $||t||_0$ is the number of non-zero elements of t.