

Notes on reconstruction of optical spectra from 64-bit dFT spectrometer

Derek M. Kita¹ and Brando Miranda²

¹Department of Materials Science & Engineering, Massachusetts Institute of Technology
²Center for Brains, Minds, and Machines, Massachusetts Institute of Technology

February 1, 2018

Problem Statement & Objective

Given the measured interferogram y ($N \times 1$) and a calibration matrix A ($N \times D$), we seek to accurately reconstruct the input optical signal x that obeys:

$$y = Ax \quad (1)$$

where $D \gg N$, and in our case $D = 801$, $N = 64$. For our application, there are two types of signals of interest: (1) laser lines that produce sparse spectra, and (2) light sources (like an EDFA) with a broad spectrum (not-sparse).

L1 and L2 minimization

Since the problem we are solving is underdefined, there are an infinite number of solutions x that solve Eq. 1. However, with prior knowledge of the size of the correct solution's L1- and L2-norm, we can obtain better estimates of x by minimizing:

$$\min_x \left\{ \|y - Ax\|^2 + \alpha_1 \|x\|_1 + \alpha_2 \|x\|_2^2 \right\} \quad (2)$$

Solving the above corresponds to the “elastic net” regularized regression method (removing only the L1 norm corresponds to “ridge regression” and removing only the L2 norm corresponds to “LASSO”).

Radial Basis Function (RBF) Network

For some application, such as broad input sources, our prior information can consist of how “smooth” the spectrum is. In this case, we can construct a function h that represents our spectrum and we seek to minimize the following for appropriate α , and c_k .

$$\min_x \left\{ \|y - Ah(\lambda)\|^2 + \alpha \sum_k c_k \int \frac{\partial^k h(\lambda)}{\partial \lambda^k} d\lambda \right\} \quad (3)$$

The solution to the above minimization problem (that puts constraints on the solution's smoothness), is a RBF Network, which approximates h with radial basis functions:

$$h_c(\lambda) = \sum_{d=1}^D c_d e^{-\beta |\lambda - \lambda_d|^2} \quad (4)$$

We can then use an appropriate algorithm like stochastic gradient descent (SGD) to solve the following, simpler minimization problem:

$$\min_c \left\{ \|y - Ah_c(\lambda)\|^2 \right\} \quad (5)$$

non-negative Elastic-D smoothing Regularization

For some application, such as broad input sources, our prior information can consist of how “smooth” the spectrum is. To induce such smoothness with respect to the first derivative one can use the forward finite difference matrix D to define the following regularizer $\|Dx\|_2^2$. The matrix D is defined as follows:

$$D = \begin{bmatrix} 1 & 0 & 0 & \cdots & \cdots & \cdots & 0 \\ 0 & -1 & 1 & \cdots & \cdots & \cdots & 0 \\ 0 & 0 & -1 & \cdots & \cdots & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \cdots & \cdots & \vdots \\ 0 & 0 & 0 & \cdots & -1 & 1 & 0 \\ 0 & 0 & 0 & \cdots & 0 & -1 & 1 \\ 0 & 0 & 0 & \cdots & 0 & 0 & 1 \end{bmatrix}$$

note that there are other matrices that also approximate the first derivative, for example the central difference matrix with derivatives approximated by $[-1, 2, 2]$. We did not try these or other first derivative approximations because we found forward difference to be sufficiently accurate for our application. Also note that is easy to generalize the difference if points are not equidistant by simply taking the real distance between between points.

We cast our reconstruction problem with L1, L2 and first derivative smoothness prior as follow:

$$\min_{x, x \geq 0} \left\{ \|y - Ax\|^2 + \alpha_1 \|x\|_1 + \alpha_2 \|x\|_2^2 + \alpha_3 \|Dx\|_2^2 \right\} \quad (6)$$

by using the standard trick that $\|Mx\|_2^2 = x^T Mx$ we can write the above as a non-negative quadratic program by using the fact that $\|Dx\|_2^2 = x^T Dx$ and $\|x\|_2^2 = x^T Ix$ where I is the identity. We also use the fact that $x \geq 0$ thus $\|x\|_1 = \sum_{d=1}^D x_i = \mathbf{1}^T x$ to write the above into the following equivalent non-negative quadratic program:

$$\min_{x, x \geq 0} \left\{ x^T \left(A^T A + \alpha_2 I + \alpha_3 D^T D \right) x + (\alpha_1 \mathbf{1} - A^T y)^T x \right\} \quad (7)$$

this is easy to show using basic linear algebra and matrix rules. Then we simply plug this in to a standard quadratic program solver available in python to find the reconstruction of the signal x .