Lecture 15

Image Reconstruction as Optimization: ℓ_2 Formulations

15.1 Lecture Objectives

- Review quadratic formulations (based on ℓ_2 norm squared terms) typically used for image reconstruction
- Connect this material to previous lectures where we studied how to solve such formulations

15.2 Overview

In this lecture we will review optimization-based image reconstruction formulations and algorithms of the form:

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} f(\mathbf{x};\mathbf{d}) + \lambda R(\mathbf{x}) \tag{15.1}$$

where the first term in the cost function $f(\mathbf{x};\mathbf{d})$ is the data fidelity term, and penalizes mismatch between our reconstructed image and the measured data $\mathbf{d}$. The second term in the cost function, $R(\mathbf{x})$, promotes certain properties (eg: smoothness, sparsity, compressibility) of the image that are known a priori. In today's lecture, we will focus on a priori terms of quadratic form, ie:

$$R(\mathbf{x}) = \|\mathbf{C}\mathbf{x}\|_2^2$$

Finally, the "regularization" parameter λ balances the importance of the data fidelity term and the prior term. Note that, as we covered in homework 1, sometimes this type of formulation can be interpreted as a Maximum a Posteriori (MAP) estimation including a log-likelihood term ($f(\mathbf{x};\mathbf{d})$), and a prior term ($\lambda R(\mathbf{x})$).

15.3 Data term

15.3.1 Formulation

In this lecture, we will consider quadratic data terms of the form:

$$f(\mathbf{x}; \mathbf{d}) = \|\mathbf{A}\mathbf{x} - \mathbf{d}\|_2^2 \tag{15.2}$$

where the system matrix $\mathbf{A}$ represents the physics of our imaging system, as well as some acquisition parameters. For instance, $\mathbf{A}$ may perform a discrete Fourier transform (DFT), or a radon transform (ie: calculate a set of projections). The use of a quadratic data term is generally justified when our data can be considered to contain Gaussian i.i.d. noise, since the least-squares solution provided by minimizing equation 15.2 above corresponds to the maximum-likelihood solution (as derived in our first homework set). A quadratic data term is often also reasonable when our noise is not exactly Gaussian but our data have relatively high signal-to-noise ratio (SNR) and the noise can be considered approximately Gaussian.

Let us now consider a specific choice of system matrix $\mathbf{A}$. In this case, our data consists of undersampled DFT measurements. In other words,

$$\mathbf{A} = \mathbf{A_S}\mathbf{F} \tag{15.3}$$

where $\mathbf{F}$ performs a 2D DFT, and $\mathbf{A_S}$ performs an undersampling operation (ie: $\mathbf{A_S}\,\mathbf{x}$ picks some subset of the entries of $\mathbf{x}$ and ignores the rest, as we discussed in the previous lecture.

These illustrative elements lead to our first image reconstruction formulation:

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \|\mathbf{A_S}\mathbf{F}\mathbf{x} - \mathbf{d}\|_2^2 \tag{15.4}$$

15.3.2 Computational solution

As we know from our previous lectures (based on the normal equations), the solution to the formulation in equation 15.4 above satisfies the following equality:

$$(\mathbf{A_S}\mathbf{F})^H \mathbf{A_S}\mathbf{F}\mathbf{x} = (\mathbf{A_S}\mathbf{F})^H \mathbf{d} \tag{15.5}$$

Which is simply a linear system of equations $\mathbf{Q}\mathbf{x} = \mathbf{b}$, with $\mathbf{Q} = (\mathbf{A_S}\mathbf{F})^H \mathbf{A_S}\mathbf{F}$ and $\mathbf{b} = (\mathbf{A_S}\mathbf{F})^H \mathbf{d}$. Importantly, solving this system of equations is equivalent to minimizing the following quadratic problem:

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \frac{1}{2}\mathbf{x}^H \mathbf{Q}\mathbf{x} - \mathbf{x}^H \mathbf{b} \tag{15.6}$$

which we can solve relatively easily using a steepest descent or conjugate gradients algorithm, as long as we know how to calculate $\mathbf{Q}\mathbf{x}$ for an arbitrary $\mathbf{x}$. See the previous lecture notes for details on how to do this.

15.3.3 Further extensions

Sometimes we want our reconstructed image to fit our data in a *weighted* ℓ_2 norm sense, ie: we trust some data points more than others, so we would like to find the image $\mathbf{x}$ that minimizes the following cost function:

$$f(\mathbf{x};\mathbf{d}) = \|\mathbf{W_d}\left(\mathbf{Ax}-\mathbf{d}\right)\|_2^2 \tag{15.7}$$

where $\mathbf{W_d}$ is a diagonal matrix with non-negative diagonal elements, where the nth diagonal element specifies how much weight we place on that particular measurement (ie: how much we care about our reconstruction matching the nth measurement d_n).

Question: How can we transform this weighted ℓ_2 formulation in equation 15.7 into our standard problem $\mathbf{Qx} = \mathbf{b}$, as we did above with the unweighted ℓ_2 formulation?

15.4 Regularization (Smoothness) Term

15.4.1 Formulation

In this lecture, we will focus on quadratic regularization strategies:

$$R(\mathbf{x}) = \|\mathbf{Cx}\|_2^2 \tag{15.8}$$

where $\mathbf{C}$ is a matrix that will often include some form of high-pass filtering (eg: finite differences), although it could also be an identity matrix if we simply want to penalize high-energy solutions.

For example, $\mathbf{C}$ will typically perform finite differences ($\mathbf{C} = \mathbf{D}$ where $\mathbf{D}$ is something like the `D2` matrix shown in the previous lecture's notes). In addition, $\mathbf{C}$ may also include some kind of anatomically-derived weighting (eg: if we have prior information as to the likely locations of edges in this image), ie: $\mathbf{C} = \mathbf{W_e D}$ where $\mathbf{W_e}$ is a diagonal matrix with positive diagonal elements that are large for locations where we do not expect edges, and small (or zero) for locations where we do expect edges.

15.4.2 Computational solution

How would we include this regularization term (with regularization parameter λ) into our computational solution?

Based on previous lectures, we know we can combine our entire formulation:

$$\min \|\mathbf{Ax}-\mathbf{d}\|_2^2 + \lambda\|\mathbf{Cx}\|_2^2 \tag{15.9}$$

as follows:

$$\min \left\| \begin{bmatrix} \mathbf{A} \\ \sqrt{\lambda}\mathbf{C} \end{bmatrix} \mathbf{x} - \begin{bmatrix} \mathbf{d} \\ \mathbf{0} \end{bmatrix} \right\|_2^2 \tag{15.10}$$

which is our standard least-squares fitting problem and we know how to solve.

Now the only remaining question is, how can we efficiently perform matrix vector operations such as

$$\begin{bmatrix} \mathbf{A} \\ \sqrt{\lambda}\mathbf{C} \end{bmatrix} \mathbf{x}$$

or

$$\begin{bmatrix} \mathbf{A} \\ \sqrt{\lambda}\mathbf{C} \end{bmatrix}^H \mathbf{y}$$

For this, we will need to calculate the two different portions of the resulting vector, and stack/add them together as needed.