Part I

Fundamentals of Optimization

Lecture 2

Introduction to Optimization in Imaging

2.1 Lecture Objectives

- Set a common mathematical framework for describing optimization problems (regardless of the specific application).
- Understand the difference between an optimization problem (formulation) and an algorithm used to solve such problem.
- Develop intuition for the kinds of optimization problems we face in imaging applications.
- Preview some of the types of formulations to be covered in this course.
- Preview some of the types of algorithms to be covered in this course.

2.2 Standard Optimization Problem

A standard (continuous-valued) optimization problem can be written as follows

$$\begin{aligned} &\text{minimize } f(\mathbf{x}) \\ &\text{such that } g_k(\mathbf{x}) \leq b_k, \text{ for } k = 1, \ldots, K \\ &\text{such that } h_l(\mathbf{x}) = c_l, \text{ for } l = 1, \ldots, L \end{aligned} \tag{2.1}$$

where $\mathbf{x} = [x_1, \cdots, x_N] \in \mathbb{R}^N$ is the vector we are trying to optimize (please see below for specific examples of $\mathbf{x}$ in imaging applications), the function $f : \mathbb{R}^N \to \mathbb{R}$ is the objective (or 'cost') function that drives our optimization, the functions $g_k : \mathbb{R}^N \to \mathbb{R}$ together with the scalar values b_k determine the inequality constraints on $\mathbf{x}$, and the functions $h_l : \mathbb{R}^N \to \mathbb{R}$ together with the scalar values c_l determine the equality constraints on $\mathbf{x}$.

Together, the inequality and equality constraints determine the allowable values of $\mathbf{x}$ (ie: the feasible region).

Note that the formulation given in Equation 2.1 is very general, and includes a wide array of optimization problems encountered in various applications. Very different optimization problems will arise depending on the nature of the objective function f_0 and the constraint functions f_k. Some of these problems are easy to solve, and some are very hard. In this course, we are interested in high-dimensional optimization problems (ie: where N is large, typically on the order of thousands or even millions), so finding appropriate algorithms that work well in practice is critical. The remainder of this portion of the course will dissect the general formulation in Equation 2.1 and various algorithms used to solve it. Importantly, this leads us to the first fundamental concept of this course: the distinction between a formulation and an algorithm.

- *Formulation:* in the context of this course, a formulation defines an optimization problem, ie: "find a vector $\mathbf{x}$ that minimizes some objective function, subject to a set of constraints". Note that the formulation makes no claims about how one might go about finding such optimal vector; that is where the algorithm comes in.
- *Algorithm:* a step-by-step procedure for solving a problem such as the optimization problem (Equation 2.1) described above.

Why is it important to distinguish these two concepts? Because, in the instructor's experience, this is likely to lead to better research (eg: technical development), improved ability to communicate one's research (eg: writing papers), and improved ability to understand others' research (eg: reading papers). For instance, an optimization problem (formulation) may be solved using a specific algorithm today, but a different algorithm tomorrow (assuming the new algorithm is better, eg: faster) without altering the fundamental nature of the solution.

Exceptions: There are cases where the algorithm itself is used to drive the specific solution that is reached, which is therefore not determined solely by the formulation. Notable examples of this approach can be found in the training of neural networks for machine learning. For instance, consider the "dropout" technique[1] often used during the training of neural networks in order to avoid convergence to certain solutions in favor of others. One way to view this is that the choice of algorithm parameters introduces implicit modifications in the formulation. In these cases, the tidy separation between formulation and algorithm becomes blurred.

2.3 Optimization Applications

Next, we will preview a few of the applications of optimization to be covered in this course, including the possible meanings of the objective function and constraint functions

[1]For details on this technique, see for instance Srivastava et al, 2014, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting", https://www.cs.toronto.edu/ hinton/absps/JMLRdropout.pdf

in image reconstruction, segmentation, and registration.

2.3.1 Image Reconstruction

In image reconstruction, regardless of imaging modality (MRI, CT, PET, SPECT, etc) we typically seek to estimate an image from a set of measurements (denoted c_l). This desired image is indeed our vector $\mathbf{x}$ (even if the image is to be displayed as a 2D or 3D array after reconstruction, conceptually we can consider it a vector for mathematical manipulation). In a simple image reconstruction formulation, the objective function calculates the energy of the image, and the equality constraints force the estimated image to match the acquired data:

$$\begin{aligned} &\text{minimize } \|\mathbf{x}\| = \sum_{n=1}^{N} |x_n|^2 \\ &\text{such that } h_l(\mathbf{x}) = c_l, \text{ for } l = 1, \ldots, L \end{aligned} \tag{2.2}$$

where the functions h_l capture the imaging physics, eg: calculate different Fourier Transform coefficients in the case of MRI, and perform projections along different angles in the case of CT. This simple formulation would seek the image with lowest energy among all the possible images that match the acquired data.

Now, why would we formulate the image reconstruction this way (by minimizing the image energy)? This type of formulation (similar to Tikhonov regularization) helps prevent noise and artifacts from blowing up in image reconstruction. We will revisit this topic in the image reconstruction portion of the course.

2.3.2 Image Segmentation

A traditional formulation for segmentation of an input image $\mathbf{y}$ is the Potts model, which can be written as:

$$\text{minimize } \gamma\|\nabla\mathbf{x}\|_0 + \|\mathbf{x} - \mathbf{y}\|_2^2 \tag{2.3}$$

where ∇ performs a gradient operation, and the $\|\cdot\|_0$ notation is the ℓ_0 metric, that simply calculates the number of non-zero entries in a vector (in this case, the number of points with non-zero gradient). In other words, this formulation seeks a piecewise-constant approximation $\mathbf{x}$ to the original image $\mathbf{y}$. Further, the 'regularization' parameter γ balances the importance of the two terms in the objective function (ie: 'flatness' vs closeness to the input image). Note that this specific optimization problem is 'unconstrained', ie: it has an objective function but no constraints.

Question: Can you devise a constrained formulation that accomplishes similar goals to the unconstrained segmentation formulation shown in Equation 2.3 above?

Also, note that this segmentation formulation is defined in terms of a continuous-valued vector. However, image segmentation is generally a bit different from the formulation

given in Equation 2.1 in that the desired vector $\mathbf{x}$ is often not real-valued, but rather a discrete-valued vector that may take on one of several possible values (2 in the case of binary segmentation) at each location. We will briefly review discrete optimization later in the course.

2.3.3 Image Registration

Registration of input images $\mathbf{y_1}$ and $\mathbf{y_2}$ can be described as a pixel-wise geometric operation on $\mathbf{y_2}$ that makes it closely aligned (co-registered) with $\mathbf{y_1}$.

$$\text{minimize } E(\mathbf{y_1}, r(\mathbf{y_2}, \mathbf{x})) \tag{2.4}$$

where the function r applies a geometric distortion to $\mathbf{y_2}$, as parameterized by $\mathbf{x}$, and the function E calculates a measure of dissimilarity between $\mathbf{y_1}$ and $r(\mathbf{y_2}, \mathbf{x})$. Typical choices for E are based on measures of mutual information using entropy, and seek to penalize images that contain features (eg: edges) at different locations. We will review more advanced image registration formulations and algorithms later in the course.

Question: Can you devise a more advanced version of the registration formulation shown in Equation 2.4 above, including additional terms or constraints on the specific geometric distortion? Note that, for most registration applications, we consider certain geometric distortions (moderate, spatially smooth, physically feasible, etc) as more likely than others (very severe, spatially rough, physically infeasible, etc).

2.4 Formulations

A few essential (not necessarily mutually exclusive) types of formulations include:

- Linear vs non-linear
- Quadratic vs non-quadratic
- Convex vs non-convex
- Constrained vs unconstrained

In particular, the concept of convexity is of enormous theoretical and practical importance, as it leads to the ability to guarantee global optimality for iterative algorithms. These concepts will be studied over the next few lectures.

2.5 Algorithms

Most optimization problems are solved iteratively using an algorithm that provides a progressively better and better solution (ie: one that yields a smaller and smaller value

of the objective function). An enormous variety of algorithms for optimization exist, including:

- Coordinate descent
- Gradient (steepest) descent
- Conjugate gradients
- Newton's method and its variants
- Stochastic gradient descent

as well as many others. Note that the choice of iterative algorithm often presents a trade-off between the complexity required to complete each iteration, and the number of iterations required to reach the solution. We will explore several foundational algorithms and their trade-offs towards the end of the optimization portion of this course.