11

Overview of Optimization

"[N]othing at all takes place in the universe in which some rule of maximum or minimum does not appear."
— Leonhard Euler

Mathematical optimization, traditionally also known as mathematical programming, is the theory of optimal decision making. Other than in machine learning, optimization problems arise in a large variety of contexts, including scheduling and logistics problems, finance, optimal control and signal processing. The underlying mathematical problem always amounts to finding parameters that minimize (cost) or maximize (utility) an objective function in the presence or absence of a set of constraints. In the context of machine learning, the objective function is usually related to the empirical risk, but we first take a step back and consider optimization problems in greater generality.

What is an optimization problem?

A general mathematical optimization problem is a problem of the form

$$\begin{aligned} &\text{minimize} && f(\boldsymbol{w}) \\ &\text{subject to} && \boldsymbol{w} \in \Omega \end{aligned} \tag{11.1}$$

where $f\colon \mathbb{R}^d \to \mathbb{R}$ is a real-valued **objective function** and $\Omega \subseteq \mathbb{R}^d$ is a set defining the **constraints**. In the context of machine learning, the variables $\boldsymbol{w}$ are usually the parameters defining a classifier or predictor, such as the weights of a support vector machine or neural network. If $\Omega = \mathbb{R}^d$, then the problem is an **unconstrained optimization problem**. The constraint set Ω often consists of $\boldsymbol{w} \in \mathbb{R}^d$ that satisfy certain equations and inequalities,

$$f_1(\boldsymbol{w}) \leq 0, \ldots, f_m(\boldsymbol{w}) \leq 0, g_1(\boldsymbol{w}) = 0, \ldots, g_p(\boldsymbol{w}) = 0.$$

A vector $\boldsymbol{w}^*$ satisfying the constraints is called an **optimum**, a **solution**, or a **minimizer** of the problem (11.1), if $f(\boldsymbol{w}^*) \leq f(\boldsymbol{w})$ for all other $\boldsymbol{w}$ that satisfy the constraints. Note that replacing f by $-f$, we could equivalently state the problem as a maximization problem.

Optimality conditions

In what follows we will study the unconstrained problem

$$\underset{\boldsymbol{w}\in\mathbb{R}^d}{\text{minimize}}\, f(\boldsymbol{w}). \tag{11.2}$$

Optimality conditions aim to identify properties that potential minimizers need to satisfy in relation to $f(\boldsymbol{w})$. We will review the well known local optimality conditions for differentiable functions from calculus. We first identify different types of minimizers.

Definition 11.1. A point $\boldsymbol{w}^* \in \mathbb{R}^d$ is a

- *global minimizer* of (11.2) if for all $\boldsymbol{w} \in \mathbb{R}^d$, $f(\boldsymbol{w}^*) \leq f(\boldsymbol{w})$;
- a *local minimizer*, if there is an open neighbourhood U of $\boldsymbol{w}^*$ such that $f(\boldsymbol{w}^*) \leq f(\boldsymbol{w})$ for all $\boldsymbol{w} \in U$;
- a *strict local minimizer*, if there is an open neighbourhood U of $\boldsymbol{w}^*$ such that $f(\boldsymbol{w}^*) < f(\boldsymbol{w})$ for all $\boldsymbol{w} \in U$;
- an *isolated minimizer* if there is an open neighbourhood U of $\boldsymbol{w}^*$ such that $\boldsymbol{w}^*$ is the only local minimizer in U.

Without any further assumptions on f, finding a minimizer is a hopeless task: we cannot simply examine the function at *all* points in $\mathbb{R}^d$. The situation becomes more tractable if we assume some *smoothness* conditions. Recall that $C^k(U)$ denotes the set of functions that are k times continuously differentiable on some set U. The following *first-order* necessary condition for optimality is well known. We write $\nabla f(\boldsymbol{w})$ for the gradient of f at $\boldsymbol{w}$, i.e., the vector

$$\nabla f(\boldsymbol{w}) = \left(\frac{\partial f}{\partial w_1}(\boldsymbol{w}), \ldots, \frac{\partial f}{\partial w_d}(\boldsymbol{w})\right)^\top$$

Theorem 11.2. *Let $\boldsymbol{w}^*$ be a local minimizer of f and assume that $f \in C^1(U)$ for a neighbourhood of U of $\boldsymbol{w}^*$. Then $\nabla f(\boldsymbol{w}^*) = \mathbf{0}$.*

There are simple examples that show that this is not a sufficient condition: maxima and saddle points will also have a vanishing gradient. If we have access to *second-order information*, in form of the second derivative, or Hessian, of f, then we can

say more. Recall that the Hessian of f at $\boldsymbol{w}$, $\nabla^2 f(\boldsymbol{w})$, is the $d \times d$ symmetric matrix given by the second derivatives,

$$\nabla^2 f(\boldsymbol{w}) = \left(\frac{\partial^2 f}{\partial w_i \partial w_j} \right)_{1 \leq i,j \leq d}.$$

In the one-variable case we know that if w^* is a local minimizer of $f \in C^2([a,b])$, then $f'(w^*) = 0$ *and* $f''(w^*) \geq 0$. Moreover, the conditions $f'(w^*) = 0$ *and* $f''(w^*) > 0$ guarantee that we have a local minimizer. These conditions generalise to higher dimension, but first we need to know what $f''(w) > 0$ when we have more than one variable.

Recall also that a matrix $\boldsymbol{A}$ is **positive semidefinite**, written $\boldsymbol{A} \succeq \boldsymbol{0}$, if for every $\boldsymbol{w} \in \mathbb{R}^d$, $\boldsymbol{w}^\top \boldsymbol{A} \boldsymbol{w} \geq 0$, and positive definite, written $\boldsymbol{A} \succ \boldsymbol{0}$, if $\boldsymbol{w}^\top \boldsymbol{A} \boldsymbol{w} > 0$. The property that the Hessian matrix is positive semidefinite is a multivariate generalization of the property that the second derivative is nonnegative. The known conditions for a minimizer involving the second derivative generalize accordingly.

Theorem 11.3. *Let $f \in C^2(U)$ for some open set U and $\boldsymbol{w}^* \in U$. If $\boldsymbol{w}^*$ is a local minimizer, then $\nabla f(\boldsymbol{w}^*) = 0$ and $\nabla^2 f(\boldsymbol{w}^*)$ is positive semidefinite. Conversely, if $\nabla f(\boldsymbol{w}^*) = \boldsymbol{0}$ and $\nabla^2 f(\boldsymbol{w}^*)$ is positive definite, then $\boldsymbol{w}^*$ is a strict local minimizer.*

Unfortunately, the above criteria are not able to identify global minimizers, as differentiability is a local property. For **convex** functions, however, local optimality implies global optimality.

Examples

We present two examples of optimization problems that can be interpretred as machine learning problems, but have mainly been studied outside of the context of machine learning. The examples below come with associated Python code and it is not expected that you understand them in detail; they are merely intended to illustrate some of the problems that optimization deals with, and how they can be solved.

Example 11.4. Suppose we want to understand the relationship of a quantity y (for example, sales data) to a series of *predictors* $x_1, \ldots, x_p$ (for example, advertising budget in different media). We can often assume the relationship to be *approximately linear*, with distribution

$$Y = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p + \varepsilon,$$

with noise ϵ that satisfies $\mathbb{E}[\epsilon] = 0$. In the setting of statistical learning, the input space is $\mathcal{X} = \mathbb{R}^p$, the output space is $\mathcal{Y} = \mathbb{R}$, and the set $\mathcal{H}$ consists of functions of the form

$$h(\boldsymbol{x}) = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p. \tag{11.3}$$

The goal is to determine the *model parameters* $\boldsymbol{w} = (\beta_0, \ldots, \beta_p)^\top$ from observed data. To determine these, we can collect $n \geq p$ sample realizations (from observations or experiments),

$$\{(x_{i1}, \ldots, x_{ip}, y_i)\}_{i=1}^n.$$

and minimize the empirical risk with respect to the (normalized) ℓ_2 loss function:

$$\hat{R}(h) = \frac{1}{2n} \sum_{i=1}^{n} (y_i - (\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip}))^2.$$

Collecting the data in matrices and vectors,

$$\boldsymbol{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}, \quad \boldsymbol{X} = \begin{pmatrix} 1 & x_{11} & \cdots & x_{1p} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & \cdots & x_{np} \end{pmatrix}, \quad \boldsymbol{w} = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{pmatrix},$$

we can write the empirical risk concisely as

$$\hat{R}(h) = \frac{1}{2n} \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}\|^2.$$

Minimizing over $h \in \mathcal{H}$ means minimizing over vectors $\boldsymbol{w} \in \mathbb{R}^{p+1}$, and the best $\boldsymbol{w}$ is then the vector that solves the unconstrained optimization problem

$$\underset{\boldsymbol{w} \in \mathbb{R}^{p+1}}{\text{minimize}} \; \frac{1}{2n} \|\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y}\|_2^2.$$

This is an example of an optimization problem with variables $\boldsymbol{w}$, no constraints (*all* $\boldsymbol{w}$ are valid candidates and the constraint set is $\Omega = \mathbb{R}^{p+1}$), and a *quadratic* objective

$$\begin{aligned} f(\boldsymbol{w}) = \frac{1}{2n} \|\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y}\|_2^2 &= \frac{1}{2n} (\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y})^\top (\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y}) \\ &= \frac{1}{2n} \boldsymbol{w}^\top \boldsymbol{X}^\top \boldsymbol{X} \boldsymbol{w} - 2\boldsymbol{y}^\top \boldsymbol{X} \boldsymbol{w} + \boldsymbol{y}^\top \boldsymbol{y}, \end{aligned} \tag{11.4}$$

where $\boldsymbol{X}^\top$ is the matrix transpose. If the columns of $\boldsymbol{X}$ are linearly independent (which, by the way, requires there to be more data than the number of parameters p), this simple optimization problem has a unique closed form solution,

$$\boldsymbol{w}^* = (\boldsymbol{X}^\top \boldsymbol{X})^{-1} \boldsymbol{X}^\top \boldsymbol{y}. \tag{11.5}$$

In practice one would not compute $\boldsymbol{w}^*$ by evaluating (11.5). There are more efficient methods available, such as gradient descent, the conjugate gradient method, and variations of these. It is important to note that even in this simple example, solving the optimization problem can be difficult if the number of samples is large.

To illustrate the least squares setting using a concrete example, assume that we have data relating the basal metabolic rate (energy expenditure per time unit)

in mammals to their mass.[1] Using data for 573 mammals from the PanTHERIA database[2], we see that the relationship is approximately linear (see Figure 11.1), and can be modelled as

$$Y = \beta_0 + \beta_1 X + \epsilon.$$

From the data we can assemble the vector $\boldsymbol{y}$ and the matrix $\boldsymbol{X} \in \mathbb{R}^{n\times(p+1)}$ in order to compute the $\boldsymbol{w} = (\beta_0, \beta_1)^\top$ (here, $p = 1$ and $n = 573$). We can find β_0 and β_1 by solving an optimization problem as described above (or solving the problem in closed form). Solving the problem, we get the values $\beta_0 = 1.362$ and $\beta_1 = 0.702$, and we can plot the line and see how it fits the data.

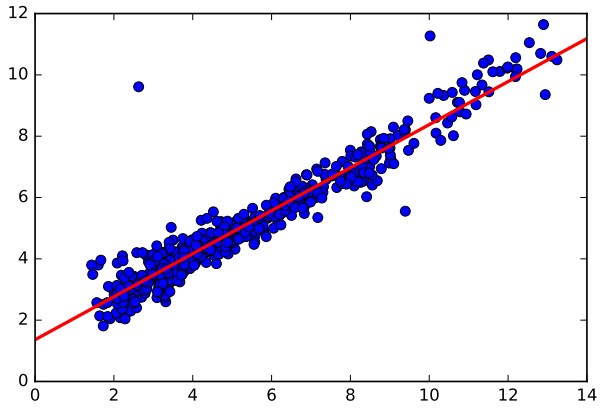

Figure 11.1: Fitting a regression line

Example 11.5. (Image inpainting) Even problems in image processing that do not appear to be machine learning problems can be cast as such. An image can be viewed as an $m \times n$ matrix $\boldsymbol{U}$, with each entry u_{ij} corresponding to a light intensity (for greyscale images), or a colour vector, represented by a triple of red, green and blue intensities (usually with values between 0 and 255 each). For simplicity the following

[1] This example is from the episode "Size Matters" of the BBC series Wonders of Life.
[2] http://esapubs.org/archive/ecol/E090/184/#data

discussion assumes a greyscale image. For computational purposes, the matrix of an image is often viewed as an mn-dimensional vector $\boldsymbol{u}$, with the columns of the matrix stacked on top of each other.

In the *image inpainting* problem, one aims to *learn* the true value of missing or corrupted entries of an image. There are different approaches to this problem. A conceptually simple approach is to replace the image with the *closest* image among a set of images satisfying typical properties. But what are typical properties of a typical image? Some properties that come to mind are:

- Images tend to have large homogeneous areas in which the colour doesn't change much;
- Images have approximately low rank, when interpreted as matrices.

Total variation image analysis takes advantage of the first property. The **total variation** or TV-norm is the sum of the norm of the horizontal and vertical differences,

$$\|\boldsymbol{U}\|_{\mathrm{TV}} = \sum_{i=1}^{m}\sum_{j=1}^{n}\sqrt{(u_{i+1,j}-u_{i,j})^2+(u_{i,j+1}-u_{i,j})^2},$$

where we set entries with out-of-bounds indices to 0. The TV-norm naturally increases with increased variation or sharp edges in an image. Consider for example the two following matrices (imagine that they represent a 3×3 pixel block taken from an image).

$$\boldsymbol{U}_1 = \begin{pmatrix} 0 & 17 & 3 \\ 7 & 32 & 0 \\ 2 & 9 & 27 \end{pmatrix}, \quad \boldsymbol{U}_2 \begin{pmatrix} 1 & 1 & 3 \\ 1 & 0 & 0 \\ 0 & 0 & 2 \end{pmatrix}$$

The left matrix has TV-norm $\|\boldsymbol{U}_1\|_{\mathrm{TV}} = 200.637$, while the right one has TV-norm $\|\boldsymbol{U}_2\|_{\mathrm{TV}} = 14.721$ (verify this!) Intuitively, we would expect a natural image with artifacts added to it to have a higher TV norm.

Now let $\boldsymbol{U}$ be an image with entries u_{ij}, and let $\Omega \subset [m]\times[n] = \{(i,j) \mid 1 \le i \le m, 1 \le j \le n\}$ be the set of indices where the original image and the corrupted image coincide (all the other entries are missing). One could attempt to find the image with the *smallest* TV-norm that coincides with the known pixels u_{ij} for $(i,j)\in\Omega$. This is an optimization problem of the form

$$\text{minimize } \|\boldsymbol{W}\|_{\mathrm{TV}} \quad \text{subject to} \quad w_{ij} = u_{ij} \text{ for } (i,j)\in\Omega. \tag{11.6}$$

The TV-norm is an example of a convex function and the constraints are linear conditions which define a convex set. This is an example of a **convex optimization problem** and can be solved efficiently by a range of algorithms.

In our first example, we consider an image that has been corrupted. We first determine which entries of the corrupted image are known (the complement of the corrupted pixels), and these will specify the constraints in (11.6). As the problem is

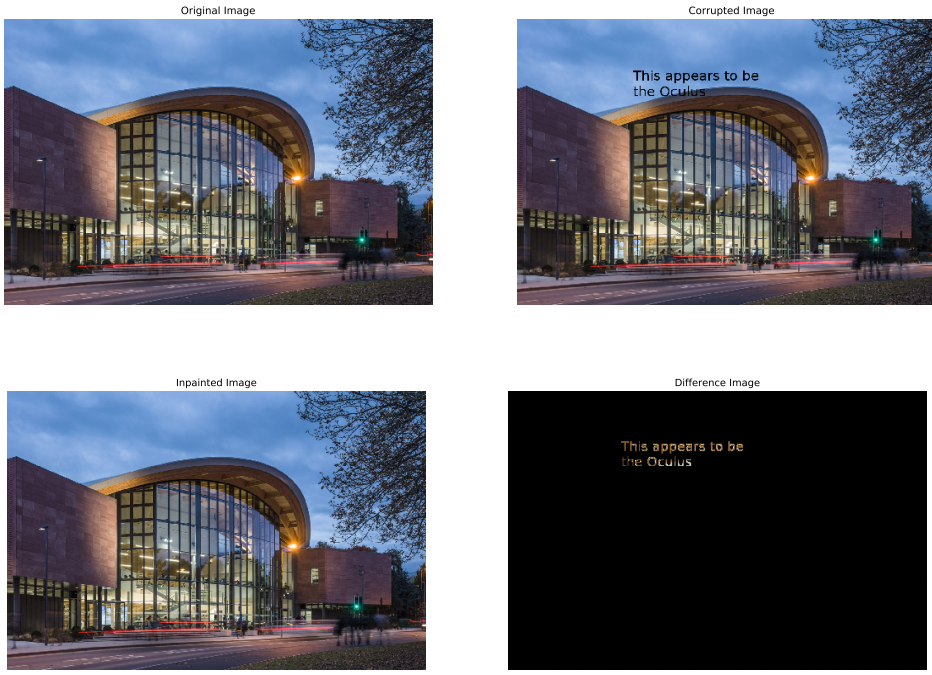

Figure 11.2: Removing writing from the Oculus

rather large (more than a million variables), it is important to choose a good solver that will solve the problem to sufficient accuracy in an acceptable amount of time.

Another typical structure of images is that the **singular values** of the image, considered as matrix, decay quickly. The **singular value decomposition** (SVD) of a matrix $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ is the matrix product

$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T,$$

where $\boldsymbol{U} \in \mathbb{R}^{m \times m}$ and $\boldsymbol{V} \in \mathbb{R}^{n \times n}$ are orthogonal matrices, and $\boldsymbol{\Sigma} \in \mathbb{R}^{m \times n}$ is a diagonal matrix with entries $\sigma_1, \ldots, \sigma_{\min\{m,n\}}$ on the diagonal. Instead of minimizing the TV-norm of an image $\boldsymbol{X}$, one may instead try to minimize the **Schatten 1-norm**, defined as the sum of the singular values, $\|\boldsymbol{U}\|_{S_1} = \sigma_1 + \cdots + \sigma_{\min\{m,n\}}$. The problem is then

$$\text{minimize } \|\boldsymbol{W}\|_{S_1} \quad \text{subject to} \quad w_{ij} = u_{ij} \text{ for } (i,j) \in \Omega.$$

This is an instance of a type of convex optimization problem known as **semidefinite programming**. Alternatively, one may also use the 1-norm of the image applied to a discrete cosine transform (DCT) or a discrete wavelet transform (DWT). As this examples (and many more to come) shows: there is no unique choice of loss function, and hence of the objective function, for a particular problem. These choices depend on model assumptions and require some knowledge of the problem one is trying to solve.

We conclude by using total variation inpainting to liberate a parrot.

Caged parrot

Free parrot

Notes

In the past, optimization theory has largely been confined to applications from physics, engineering, and in operations research, economics and finance. In these settings, the objective function has a clear physical interpretation: either as some form of energy or cost to be minimized, or as a utility to be maximized. Optimization problems have featured implicitly in statistics, either as a measure of data misfit (as in linear regression) or in parameter estimation (as in maximum likelihood estimation). Despite this, the fields of statistics and of optimization theory have only recently started to come together through machine learning. A good overview of classical optimization theory is [2]. A reference for applications in imaging, such as those mentioned in this chapter, is [1].

[1] Stephane Mallat. *A wavelet tour of signal processing: the sparse way*. Academic press, 2008.

[2] J. Nocedal and S. Wright. *Numerical Optimization*. Springer Series in Operations Research and Financial Engineering. Springer New York, 2006.