Foundations of Variational Image Analysis
Lecture Notes, HCI, 4.10.2011

Chapter 2

Gradient Descent and Implementation

Solving the Euler-Lagrange Equations in Practice

Bastian Goldlücke
Computer Vision Group
Technical University of Munich

Overview

Overview

Gradient descent

Suppose you have an energy $E : \mathcal{C}^1(\Omega) \to \mathbb{R}$ as required for the theorem (3.15), and your goal is to find the minimum

$$\operatorname*{argmin}_{u \in \mathcal{C}^1(\Omega)} E(u).$$

A simple approach you can always try is the gradient descent or steepest descent algorithm, which works as follows:

1. Start at an initial point $u \in \mathcal{V}$.
2. Find the direction of steepest descent $\hat{h} \in \mathcal{V}$ which minimizes $\delta E(u; h)$.
3. Determine t such that $E(u + t\hat{u})$ is as small as possible. Alternatively, choose just any t such that $E(u + t\hat{h}) < E(u)$. Then update u with $u + t\hat{h}$.
4. Repeat from (2) until convergence (i.e. E does not get smaller anymore).

Key question: what is the steepest descent direction $\hat{h}$ given u?

Steepest descent direction

Key question: what is the steepest descent direction h given u?
Fortunately, we already know the answer.

Theorem (steepest descent direction)

We normalize directions with respect to the $\mathcal{L}^2$ norm, i.e. only consider vectors h in the unit sphere of $\mathcal{L}^2$. The direction of steepest descent is then given by

$$\operatorname*{argmin}_{h \in \mathcal{C}^1_c(\Omega), \|h\|_2 = 1} \delta E(u; h) = -\frac{\phi_u}{\|\phi_u\|_2},$$

where ϕ_u is the left-hand side of the Euler-Lagrange equation viewed as a function on Ω,

$$\phi_u : x \mapsto \partial_a L(u, \nabla u, x) - \operatorname{div}_x \left[\nabla_b L(u, \nabla u, x) \right].$$

The proof of this theorem is not too difficult and quite enlightening, especially if you compare to the situation on $\mathbb{R}^n$ and think about it thoroughly (exercise).

Gradient descent properties

- Gradient descent can easily be implemented for any functional for which you can compute the Euler-Lagrange equation.
- It will nearly always “work”, if you manage to initialize it with a good initial value u which is close to the minimum.
- However, you can usually only hope for a local minimum.
- Gradient descent is very slow.

Overview

1. Gradient descent
   - Basic descent algorithm
   - Descent direction

2. Discretization and Implementation
   - Finite difference operators
   - Gradient descent for linear inverse problems

3. Summary

Discretization

- When one wants to solve the Euler-Lagrange PDE, one has to discretize it and work in the discrete setting.
- We will discuss discretization only for the case of a 2D image, and $\Omega \subset \mathbb{R}^2$ is a rectangle, it is easy to generalize to higher dimensions which follow the same pattern.
- Thus, in the following, we will assume that we have given an image on a regular grid of size $M \times N$. This means that an image u is represented as a matrix $u \in \mathbb{R}^{M \times N}$. We write the gray value at a pixel (i, j) as $u_{i,j}$.
- Analogously, a vector field ξ on Ω is described by two component matrices $\xi^1, \xi^2 \in \mathbb{R}^{M \times N}$.

Discretization of the gradient

The main question is how to discretize the derivative operators on functions u and vector fields ξ.

Discretization of the gradient

The gradient acts on functions u. It can be most easily discretized with forward differences:

$$(\nabla u)_{i,j} = \begin{bmatrix} u_{i+1,j} - u_{i,j} \\ u_{i,j+1} - u_{i,j} \end{bmatrix}.$$

Neumann boundary conditions $\frac{\partial u}{\partial \mathbf{n}} = 0$ are used for functions, i.e. if the grid has size $M \times N$, then

$$(\nabla u)^1_{M,j} = 0 \text{ and } (\nabla u)^2_{i,N} = 0.$$

Note that for example the regularizer total variation can then be written in discrete form as

$$J(u) = \sum_{i=1}^{M} \sum_{j=1}^{N} \left| (\nabla u)_{i,j} \right|_2.$$

Discretization of the method

Discretization of the divergence

The divergence operator acts on vector fields ξ. For a consistent discretization, it must be discretized so that the key property $\operatorname{div} = -\nabla^*$, the Gauss theorem, is satisfied.

$$(\operatorname{div}(\xi))_{i,j} = \begin{cases} \xi^1_{i,j} - \xi^1_{i-1,j} & \text{if } 1 < i < M \\ \xi^1_{i,j} & \text{if } i = 1 \\ -\xi^1_{i-1,j} & \text{if } i = M \end{cases} + \begin{cases} \xi^2_{i,j} - \xi^2_{i,j-1} & \text{if } 1 < j < N \\ \xi^2_{i,j} & \text{if } j = 1 \\ -\xi^2_{i,j-1} & \text{if } j = N \end{cases}$$

In short, div is discretized with backward differences using Dirichlet boundary conditions - this makes sense, since test functions ξ vanish on the boundary.

You should check for yourself that the desired property is satisfied when div is defined this way (exercise).

Gradient descent for the linear inverse problem

Having decided on a discretization, implementation of gradient descent is extremely simple. We show it here for the case of smoothened total variation on linear inverse problems.

- Input: An observed image f related to an unknown original image u by $Au = f$ in an ideal world, but which is degraded by Gaussian noise of standard deviation σ.
- Goal: Recover an estimate for u by computing a solution to

$$E(u) = \int_\Omega |\nabla u|_\epsilon + \frac{1}{2\sigma^2}(Au - f)^2 \, dx.$$

- Reminder: The Euler-Lagrange equation for this functional is

$$-\mathrm{div}\left(\frac{\nabla u}{|\nabla u|_\epsilon}\right) + \frac{1}{\sigma^2} A^*(Au - f) = 0.$$

The model is convex, so any solution is a global minimizer of the energy.

Gradient descent for the linear inverse problem

Algorithm

Choose a step size $\tau > 0$. Initialize $u_0 = f$. Then iterate the following steps.

- Compute the left-hand size of the Euler-Lagrange equation, using the discrete difference schemes on the previous slides.

$$\Delta_n = -\text{div}\left(\frac{\nabla u_n}{|\nabla u_n|_\epsilon}\right) + \frac{1}{\sigma^2} A^*(Au_n - f).$$

- Update the solution

$$u_{n+1} = u_n + \tau \Delta_n.$$

The step size τ depends on a number of factors and is best determined experimentally. Later, we will show much better algorithms where we know the maximum step size we can take. It is also possible to do a “line search” to determine the maximum step size automatically in each step.

Summary

- A local minimum of a differentiable energy can always be found via gradient descent.
- Non-differentiable problems can often be approximated by a smoothened version, which however usually does not obtain the exact solution to the original problem.
- If the problem is convex, gradient descent converges to the global minimizer.

Open questions

- The regularizer of the ROF functional is

$$\int_\Omega |\nabla u|_2 \, dx,$$

which requires u to be differentiable. Yet, we are looking for minimizers in $\mathcal{L}^2(\Omega)$. It is necessary to generalize the definition of the regularizer (chapter 4).

- The total variation is not a differentiable functional, so the variational principle is not applicable. We need a theory for convex, but not differentiable functionals (chapter 2).