Lecture 7: Trajectory Optimisation Part III

Scribes: Daniel Filan and Davis Foote

7.1 Summary

This lecture will have two sections: first, an explanation of why and how to change the inner product implicitly used to define functional gradient descent, and secondly, a discussion of how student paper presentations will work, and why they will work that way.

7.2 Changing the inner product

For ease of understanding, we will think about our trajectories as discretised vectors, whose i^{th} component is the position in configuration space of the trajectory at the ith point in (discrete) time. This vector will look something like

$$\xi = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ \vdots \\ q_N \end{bmatrix} \tag{7.1}$$

Since we won't consider changing the start or end points of the trajectory, we leave out the initial configuration q_0 and the final configuration q_{N+1}. We should also remember that each element of this vector is itself a vector, rather than simply a scalar.

So far, we have been using the Euclidean inner product, defined as

$$\langle \xi_1, \xi_2 \rangle_E = \xi_1^\top \xi_2 = q_{1,1}^\top q_{2,1} + \cdots + q_{1,N}^\top q_{2,N} \tag{7.2}$$

Why would we want to change this? Well, an inner product defines a norm by $\|\xi\| = \sqrt{\langle \xi, \xi \rangle}$, and a norm defines a distance metric defined by $\text{distance}(\xi_1, \xi_2) = \|\xi_1 - \xi_2\|$. Therefore, we might ask ourselves whether the Euclidean metric induced by the Euclidean inner product is any good.

7.2.1 Why the Euclidean metric isn't good

Consider three trajectories of a robot with one degree of freedom:

DIAGRAM HERE

$$a = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \ b = \begin{bmatrix} 0 \\ 0 \\ 10 \\ 0 \\ 0 \end{bmatrix}, \ \text{and } c = \begin{bmatrix} 0 \\ 5 \\ 10 \\ 5 \\ 0 \end{bmatrix} \tag{7.3}$$

Which is closer to a: b or c? Well,

$$\|a - b\|_E^2 = \langle b, b \rangle_E = 10 \times 10 = 100 \tag{7.4}$$

and

$$\|a - c\|_E^2 = \langle c, c \rangle_E = 5 \times 5 + 10 \times 10 + 5 \times 5 = 150 \tag{7.5}$$

Therefore, according to the Euclidean metric, b is closer to a than c is. However, intuitively, there's something wrong with this. a involves very smooth motion, b is super jerky, and c involves a bit of a jerk, but less so than b. The Euclidean metric can't notice this because it doesn't take the orderings of elements of the vector into account, and therefore can't talk about speeds, but it would be nice if we could take time into account somehow.

Can we give a more rigorous argument for why we should change metric?

7.2.1.1   Detour 1

The formula for gradient descent, as we know, is $\xi_{i+1} = \xi_i - (1/\alpha)\nabla_{\xi_i}\mathcal{U}$. But where does this come from?

Suppose instead that we approximate $\mathcal{U}$ by its first order Taylor expansion, and try to minimise that. Then, we would use the rule

$$\xi_{i+1} = \arg\min_{\xi} \left\{ \mathcal{U}[\xi_i] + \nabla_{\xi_i}\mathcal{U}^\top(\xi - \xi_i) \right\} \tag{7.6}$$

However, we know that this Taylor expansion will be off if we get far away in norm from ξ_i, so we add a regularisation term to penalise this:

$$\xi_{i+1} = \arg\min_{\xi} \left\{ \mathcal{U}[\xi_i] + \nabla_{\xi_i}\mathcal{U}^\top(\xi - \xi_i) + \frac{1}{2}\alpha\|\xi - \xi_i\|_E^2 \right\} \tag{7.7}$$

This is a quadratic problem in ξ, so we can take the gradient of the term inside the curly brackets, set the gradient to zero, and thereby solve the problem. Doing this, we get

$$0 + \nabla_{\xi_i}\mathcal{U} + \alpha(\xi - \xi_i) \tag{7.8}$$

Setting that expression to 0, we obtain

$$\xi_{i+1} = \xi_i - \frac{1}{\alpha}\nabla_{\xi_i}\mathcal{U} \tag{7.9}$$

which is exactly the formula for gradient descent.

7.2.1.2   Back to the main track

Remember the example of vectors a, b, and c? Well, gradient descent tends to go to things that are closer in norm, all else being equal. So, if b is closer to a than c is, even if c is better than b, gradient descent might well reach b first, because of how much further away c is.

What's the solution? We will change the distance metric (by changing the inner product) so that the sphere of elements of constant distance away from a becomes an ellipsoid, thereby making c closer than b.

DIAGRAM HERE

7.2.2 How to make a new inner product

All inner products are of the form $\xi_1^\top A \xi_2$, where A is a positive semi-definite matrix. Therefore, we just have to choose a matrix A. How?

Since we want our metric to like smooth deformations, let's think of other things that like that. Remember $\mathcal{U}_{\text{smooth}}$? That was defined as

$$\mathcal{U}_{\text{smooth}}[\xi] = \frac{1}{2} \int_0^T \|\xi'(t)\|^2 dt \tag{7.10}$$

When we discretise, the analogue of this is

$$\mathcal{U}_{\text{smooth}}[\xi] = \frac{1}{2} \sum_{i=0}^N \|q_{i+1} - q_i\|^2 \tag{7.11}$$

Suppose we compute the partial derivative of $\mathcal{U}_{\text{smooth}}$ with respect to q_i. This is

$$\frac{\partial}{\partial q_i} \mathcal{U}_{\text{smooth}} = (\nabla_\xi \mathcal{U}_{\text{smooth}})_i = -(q_{i+1} - q_i) + (q_i - q_{i-1}) = 2q_i - q_{i-1} - q_{i+1} \tag{7.12}$$

We're going to form a matrix A such that

$$\nabla_\xi \mathcal{U}_{\text{smooth}} = A\xi + c \tag{7.13}$$

where the constant c will appear because $\nabla_\xi \mathcal{U}_{\text{smooth}}$ depends on the endpoints q_0 and q_{N+1}, which don't appear in ξ. The matrix that does this is

$$A = \begin{bmatrix} 2 & -1 & & & & \\ -1 & 2 & -1 & & & \\ & -1 & 2 & -1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & -1 & 2 & -1 \\ & & & & -1 & 2 \end{bmatrix} \tag{7.14}$$

This A (the Hessian of $\mathcal{U}_{\text{smooth}}$) will be what we use for our inner product:

$$\langle \xi_1, \xi_2 \rangle_A = \xi_1^\top A \xi_2 \text{ and therefore } \|\xi_1 - \xi_2\|_A^2 = (\xi_1 - \xi_2)^\top A(\xi_1 - \xi_2) \tag{7.15}$$

Going back to our motivational example, we now have that

$$\|b - a\|_A^2 = \|b\|_A^2$$

$$= \begin{bmatrix} 0 & 0 & 10 & 0 & 0 \end{bmatrix} \begin{bmatrix} 2 & -1 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & -1 & 2 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 10 \\ 0 \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 0 & 10 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ -10 \\ 20 \\ -10 \\ 0 \end{bmatrix}$$

$$= 200$$

$$\|c - a\|_A^2 = \begin{bmatrix} 0 & 5 & 10 & 5 & 0 \end{bmatrix} \begin{bmatrix} 2 & -1 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & -1 & 2 \end{bmatrix} \begin{bmatrix} 0 \\ 5 \\ 10 \\ 5 \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 5 & 10 & 5 & 0 \end{bmatrix} \begin{bmatrix} -5 \\ 0 \\ 10 \\ 0 \\ -5 \end{bmatrix}$$

$$= 100$$

So it worked!

To better understand the structure of the A inner product, we can consider doing gradient descent with respect to $\mathcal{U}_{\text{smooth}}$ on a straight line trajectory with three points, one of which is closer to the end than the start. The gradient of $\mathcal{U}_{\text{smooth}}$ will be the sum of the vector pointing from the start point to the middle, and the vector from the end point to the middle. This will push the middle point closer to the middle of the trajectory, until it's exactly in the middle, meaning that the C-space velocity of the trajectory is constant.

DIAGRAM HERE

7.2.3 How to do gradient descent now

Now that we have our new inner product, how should we do gradient descent? Well, if we modify Equation (7.7) to use the A-norm instead of the Euclidean norm, we get

$$\xi_{i+1} = \arg \min_{\xi} \left\{ \mathcal{U}[\xi_i] + \nabla_{\xi_i} \mathcal{U}^\top (\xi - \xi_i) + \frac{1}{2} \alpha \|\xi - \xi_i\|_A^2 \right\} \tag{7.16}$$

Taking the gradient of the terms inside the curly braces, we get

$$0 + \nabla_{\xi_i} \mathcal{U} + \alpha A (\xi - \xi_i) \tag{7.17}$$

Setting this gradient to zero, we obtain

$$\xi_{i+1} = \xi_i - \frac{1}{\alpha} A^{-1} \nabla_{\xi_i} \mathcal{U} \tag{7.18}$$

So here's our equation for gradient descent! But is it correct? Since we're using the new inner product, the term $\nabla_{\xi_i} \mathcal{U}^\top (\xi - \xi_i)$ should be replaced by $\langle \nabla_{\xi_i} \mathcal{U}, \xi - \xi_i \rangle_A$, right? Well, that's true, but it's only half of the story: we also have to change from the Euclidean gradient ∇_{ξ_i} to the A-gradient $\nabla_{\xi_i}^A$. We can actually calculate this gradient directly by considering the first-order Taylor expansion of $\mathcal{U}$ using both inner products:

$$\mathcal{U}[\xi] \approx \mathcal{U}[\xi_i] + \nabla_{\xi_i}^A \mathcal{U}^\top A (\xi - \xi_i) \tag{7.19}$$

$$\approx \mathcal{U}[\xi_i] + \nabla_{\xi_i} \mathcal{U}^\top (\xi - \xi_i) \tag{7.20}$$

These expansions have to be the same, since we're talking about the same function, and the inner products are topologically equivalent, so they have the same notion of limits of small distances. Therefore, the terms on the right must be equal, meaning that

$$\nabla_{\xi_i}^A \mathcal{U} = A^{-1} \nabla_{\xi_i} \mathcal{U} \tag{7.21}$$

This agrees with our other attempt at an equation for gradient descent! It also makes clear that the new gradient takes the old gradient and 'smears it out' along the trajectory. To get a better sense of this, try calculating it for some problems and see what happens!

In summary, it's like we've turned trajectories into plastic bands: it's hard to move one point in the trajectory without moving the other ones along as well.

7.3 Student paper presentations

Each paper has pro and con presenters. The pro presenter will get approximately 15 minutes to talk, and the con presenter will get approximately 10 minutes to talk.

7.3.1 Advice for the pro presenter

Essentially, you should pretend that you wrote the paper, and are presenting it at a conference. Tell the best story you can about the paper.

- Don't just procedurally go through the paper explaining definitions and theorems in order.

- Identify the key insight! There will be one main insight in the paper, like "if the final point of the trajectory can be on a manifold, it's easier to find solutions". Explain the insight and how it helps.

- Daniel Dennet: "You should attempt to re-express your [paper's] position so clearly, vividly, and fairly that [the author would say], 'Thanks, I wish Id thought of putting it that way'."

- Use an hourglass structure, where the big ideas come in at the start and the end. First should be the motivation, then the problem statement and why it's hard, then the key insight of the paper, then the nitty-gritty details and results, and at the end restating the key insight and how it helped solve the problem.

- DIAGRAM HERE

- It would also be good to include demos, videos, etc.

- Relate back to the lecture material as much as possible. This involves not only translating the notation, but explaining how it fits in with the material we covered, such as showing that something we said was hard was actually easy, or how an algorithm we talked about was developed, or alternatives to an algorithm, etc.

7.3.2 Advice for the con presenter

Essentially, you should talk about how you would improve the paper.

- Life is easier for you.

- Be constructive, instead of just saying what's wrong with the paper.

- One question you could ask is "What paper would you write?"

- You can criticise the technical material in the paper, the experiments, or even the key idea, if you can propose improvements.

- Relate back to the lecture material as much as possible. This involves not only translating the notation, but explaining how it fits in with the material we covered, such as showing that something we said was hard was actually easy, or how an algorithm we talked about was developed, or alternatives to an algorithm, etc.

7.3.3 Advise about presenting

- It's probably a good idea for many to choose slides instead of a board talk, since board talks are harder to architect, more messy, etc.

- Board talks are OK too, especially for more theoretical papers. If you do one, you should probably plan what you're going to write ahead of time, be linear (instead of going back and editing things you wrote before), and don't try to invent things on the spot.