BIOMATH 208 Week 7

Aidan Jan

February 26, 2025

Review

[FILL]

In general we cannot find analytic solutions. We used the gradient descent algorithm, an iterative algorithm that has... [FILL]

Natural Gradient Descent

1. Start with an initial guess for our parameter, p.
2. Compute $f(p)$, $\mathrm{d}f(p)$, $g(p)$
3. Replace the i-th component of p: $p^i \mapsto p^i - \varepsilon(g(p)^{-1ij})\mathrm{d}f_j(p)$. (Note that g^{ij} means (g_{ij}^{-1}))
4. Same as gradient descent
5. Same as gradient descent

- This is especially helpful when compoennts of p have different units
- Also gives a better search direction allowing a bigger step size ε.

How to choose a metric?

- In a d dimensional manifold, $d(d-1)/2$ functions, and they have to always be positive definite.
- Pull back metrics
  - Find a map $\phi : \mathcal{M} \to$ a nice space where choosing a metric is easy

$$g_p^{\mathcal{M}}(x, y) = g_{\phi(p)}^{\text{nice space}}(\phi_{\phi(p)}X, \phi_{\phi(p)}Y)$$

- We showed how to use this metric for affine transforms acting on points, because it's easy to define a metric on the points themselves.
- Invariant metric, for Lie groups:
  - Define a metric at one point $p = I$, then for any other point p, we just use a pull back metric, pull back with $\phi = p^{-1}$.

Image Registration

(for images as functions, NOT landmark points.)

Consider a piar of grayscale images $I, J : \mathbb{R}^d \to \mathbb{R}$, and a transformation acting via $[T \cdot I](x) = I(T^{-1}x)$. We will minimize the integral square error objective function

$$f(T) = \int (I(*T^{-1}x) - J(x))^2 \mathrm{d}x$$

using gradient based methods.

Note that even though this is a square error cost, it is **not quadratic** in T, (because I is not a linear function of its argument) and therefore cannot be solved analytically

Translation

We will start with a very simple transformation group: the translation.

Consider T as a translation in $\mathbb{R}^d$ with $[T \cdot I](x) = I(x - T)$. (The $\cdot$ is a group action.) The gradient is

$$\mathrm{d}f(T) = -2 \int [IU(x - T) - J] \mathrm{d}I(x - T) \mathrm{d}x$$

- f is integral square error
- $I(x - T) - J$ is the error term.
- $\mathrm{d}I$ is the gradient of the image with respect to space
- $I(x - T)$ is the transformed images, not the original

Proof

Consider a curve $\gamma(t) = T + t\delta T$ for δT an arbitrary translation, and consider the velocity

$$\frac{\mathrm{d}}{\mathrm{d}t} f(T + t\delta t)\bigg|_{t=0} = v_{\gamma,T}(f)$$

Plugging in our definition of f gives:

$$\frac{\mathrm{d}}{\mathrm{d}t} \int (I(x - T - t\delta T) - J(x))^2 \mathrm{d}x \bigg|_{t=0}$$

By the chain rule,

$$= \int 2(I(x - T) - J(x)) \mathrm{d}I(x - T)(-\delta T)$$
$$= [-2 \int (I(x - T) - J(x)) \mathrm{d}I(x - T) \mathrm{d}x] \delta T$$

- T is a vector
- everything before the $\mathrm{d}x$ is the direction, or $\mathrm{d}f(T)$.

Affine Group Transformations

The gradient covector, as a $(d+1) \times (d+1)$ matrix, with bottom row all zeros, is given by

$$\mathrm{d}f(T) = -2 \int [I(T^{-1}x) - J(x)] \mathrm{d}[I(T^{-1}x)](T^{-1}x)^T \mathrm{d}x$$

This can be written in a matrix as:

$$\begin{pmatrix} L & T \\ 0 & 1 \end{pmatrix}$$

where

- L is linear
- T is translation
- bottom row is fixed (homogeneous coordinates)
- $[I(T^{-1}x) - J(x)]$ is the error term, transformed I minus J.
- $\mathrm{d}[I(T^{-1}x)]$ is the gradient of the transformed I.

To prove this equation, we first need to define some other things:

Part 1: Derivative of inverse matrix

$$\frac{\mathrm{d}}{\mathrm{d}t}(T + t\delta T)^{-1} = -T^{-1}\delta T T^{-1}$$

where

- δT is direction
- t is time
- This is very well defined if t is small, and δT has zeros on its bottom row.

The derivative of a matrix with respect to another matrix is multiplying the inverse matrix on both sides of the direction. In this case, it is the $T^{-1}\delta T T^{-1}$ section. We can show this because

$$I = (T + t\delta T)(T + t\delta T)^{-1}$$

Now, we take the derivative of both sides:

$$0 = \frac{\mathrm{d}}{\mathrm{d}t} I = \frac{\mathrm{d}}{\mathrm{d}t}(T + t\delta T)(T + t\delta T)^{-1}$$

By product rule,

$$= \delta T(T + t\delta T)^{-1} + (T + t\delta T)\frac{\mathrm{d}}{\mathrm{d}t}(T + t\delta T)^{-1}\Big|_{t=0}$$

When we evaluate at $t = 0$, a lot of things disappear

$$\begin{aligned} &= \delta T T + T\frac{\mathrm{d}}{\mathrm{d}t}(T + t\ deltaT)^{-1}\Big|_{t=0} \\ -\delta T T^{-1} &= T\frac{\mathrm{d}}{\mathrm{d}t}(T + t\ deltaT)^{-1}\Big|_{t=0} \\ -T^{-1}\delta T T^{-1} &= \frac{\mathrm{d}}{\mathrm{d}t}(T + t\delta T)^{-1}\Big|_{t=0} \end{aligned}$$

If big T were a scalar, this is equivalent to the "quotient rule" for taking derivatives

Part 2: Derivative of image with affine

$$\mathrm{d}I(T^{-1}x)T^{-1} = \mathrm{d}[I(T^{-1}x)]$$

- Left side: gradient of image, transformed
- Right side: gradient of the transformed image

Proof:

Start with the right side and apply the chain rule.

$$\begin{aligned}\mathrm{d}[I(T^{-1}x)] &= \mathrm{d}I\Big|_{T^{-1}x} \cdot \frac{\mathrm{d}}{\mathrm{d}x}T^{-1}x \\ &= \mathrm{d}I\Big|_{T^{-1}x} \cdot T^{-1} \\ &= \mathrm{d}I[T^{-1}x]T^{-1}\end{aligned}$$

Note: $\mathrm{d}I \doteq [\mathrm{d}I] \neq [\mathrm{d}][I]$

Part 3: The gradient

Find the directional derivative

$$\frac{\mathrm{d}}{\mathrm{d}t}\int [I((T+t\delta T)x) - J(x)]^2 \mathrm{d}x\Big|_{t=0}$$

$$\begin{aligned}&= \int 2[I((T+t\delta T)_x) - J(x)] \cdot \mathrm{d}I((T+t\delta T)^{-1}x) \cdot (-T^{-1}\delta T T^{-1})x\mathrm{d}x \\ &= \int 2[I(T^{-1}x) - J(x)]\mathrm{d}I(T^{-1}x)T^{-1}\delta T T^{-1}x\mathrm{d}x \\ &= \int -2[I(T^{-1}x) - J(x)]\mathrm{d}[I(T^{-1}x)]\delta T T^{-1}x\mathrm{d}x\end{aligned}$$

At this point, we would like to isolate δT, but it doesn't factor nicely because these are matrices. Therefore, we will work with the trace of the matrix.

Since the whole quantity is scalar, we can take the trace:

$$= -2\int (I(T^{-1}x) - J(x))\mathrm{tr}\left(\mathrm{d}[I(T^{-1}x)]\delta T T^{-1}x\right)\mathrm{d}x$$

With the trace, we have the cyclic permutation property, so we can rearrange the terms.

$$\begin{aligned}&= \cdots \mathrm{tr}\left(\delta T T^{-1}x\mathrm{d}[I(T^{-1}x)]\right) \\ &= \cdots \mathrm{tr}\left(\mathrm{d}(I(T^{-1}x))^T(T^{-1}x)^T\delta T^T\right) \\ &= \det(\mathrm{d}(I(T^{-1}x))(T^{-1}x)^T, \delta T) \\ &[FILL] \\ &= \mathrm{tr}\left(-2\int (I(T^{-1}x) - J(x))\mathrm{d}(I(T^{-1}x))^T x^T (T^{-1})^T \mathrm{d}x \cdot \delta T^T\right)\end{aligned}$$

Notice that at this point, we have factored the equation into a vector times a covector, and therefore the group action on the derivative. This completes the proof.

Metrics for Image Registration

One simple metric is to choose every voxel as a point, and use our pull back metric for point sets. (Use what we already do for point sets. [FILL])

Automatic Differentiation

Automatic differentiation works by defining a computation graph with data as edges, and functions as nodes.

For every function ($f(x)$), we define push forward ($f_*(x)[X]$), and a pull back ($f^*(x)[\chi]$). The former tells us how a perturbation of inputs affects outputs. The latter tells us how gradients should be pulled back (chain rule) for optimization.

- The push forward represents the forward pass (calculating the gradients)
- The pull back is how the nodes should be modified through backpropagation on the gradient to adjust parameters.
- In some applications, the push forward isn't needed sometime.

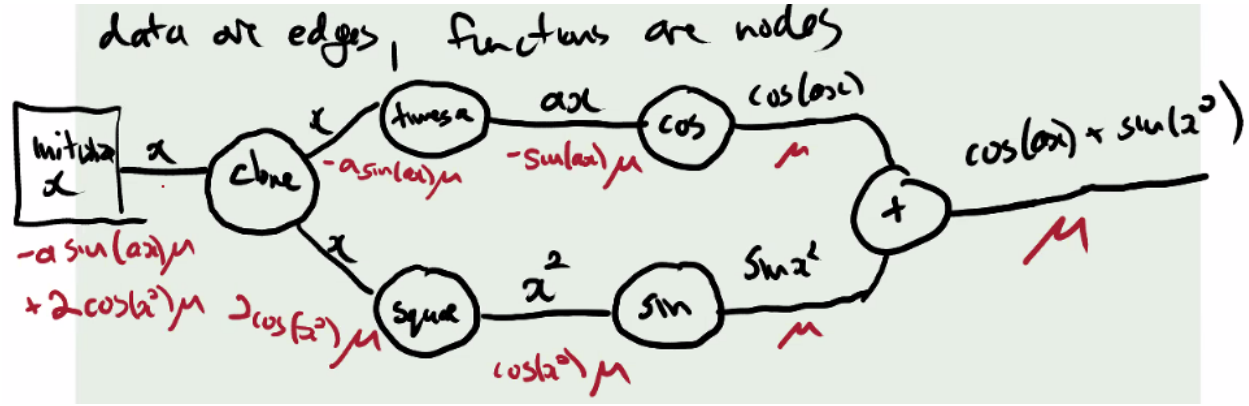
Example: (Computation graph)

Build a computation graph for the function:

$$f(x) = \cos(ax) + \sin(x^2)$$

Forward propagation in black, backpropagation in red.

- When you go backwards, you apply the derivative of each function.



- pull back, multiply by transpose of Jacobian, often can be done without computing or storing Jacobian matrix.

Riemannian Manifolds and Geodesics

- A Riemannian manifold is one with a metric defined at every point
- Geodesics is the shortest path between two points

Motivation

- Here we consider the implications of putting an inner product on a manifold.
- This will allow us to measure lengths and angles, and define straight lines.
- These operations will allow us to compute a distance between any pair of points, which can serve as an input to many machine learning algorithms.
- These operations will extend the definitions of a lot of familiar data processing techniques: filtering, averaging, regression

Riemannian Manifolds

- A Riemannian manifold is a smooth manifold with a (0, 2) tensor field that describes an inner product at every point
- This is usually denoted by the symbol g_p for a metric tensor at the point p. It is a nonlinear map $T_p\mathcal{M} \times T_p\mathcal{M} \to \mathbb{R}$.

FILL

The length of a curve

Given a curve $\gamma : [0, 1] \to \mathcal{M}$, its length is given by

$$L(\gamma) = \int_0^1 \sqrt{g_{\gamma(t)}(v_{\gamma,\gamma(t)}, v_{\gamma,\gamma(t)})}\mathrm{d}t$$

This corresponds to the familiar definition "distance equals speed times time", but note the difference between speed and velocity.

Velocity is more fundamental, whereas speed requires us to add additional structural to our manifolds.

Distances and Geodesics

We define the distance between two points on a manifold as the length of the shortest curve that connects them.

$$d(p, q) = \min_{\gamma : [0,1]\to\mathcal{M}\gamma(0)=p,\gamma(1)=q} L(\gamma)$$

These length minimizing curves are called geodesics.

"Geodesic" has two definitions.

- the shortest path between two points
- a stationary point in the above optimization problem

Most of the time, they coincide, but consider the following:

The red path meets the second definition, but not the first.

Action Integrals

Because there are an infinite number of parameterizations of the same curve γ, we choose to work with a constant speed geodesics. These are minimizers of the action integral

$$A(\gamma) = \int_0^1 g_{\gamma(t)}(v_{\gamma,\gamma(t)}, v_{\gamma,\gamma(t)})\mathrm{d}t$$

In a coordinate chart x, with $x(\gamma(t)) \doteq q(t)$ and $\dot{\gamma}_{(x)}(t) \doteq \dot{q}(t)$, and $g_{x^{-1}(q(t))}\left(\frac{\partial}{\partial x^i}, \frac{\partial}{\partial x^j}\right) \doteq g_{ij}(q(t))$, we have:

$$A(q) = \int_0^1 g_{ij}(q(t))\dot{q}^i(t)\dot{q}^j(t)\mathrm{d}t$$

Constant Speed Geodesics

With fixed endpoints minimizers of A are constant speed, and are also minimizers of L.

Proof:

Let $f(t) = \sqrt{g_{\gamma(t)}} \dots$ [FILL] [FILL]

Now, consider another function $h(t)$ (arbitrary), and consider the L_2 inner product $\int_0^1 f(t)h(t)\mathrm{d}t$. By Cauchy-Schwartz, we have

$$\left(\int_0^1 f(t)h(t)\mathrm{d}t\right)^2 \leq \int |f|^2(t)\mathrm{d}t \int |h|^2(t)\mathrm{d}t$$

Choosing $h(t) = 1$ gives

$$\left(\int_0^1 f(t)\mathrm{d}t\right)^2 \leq \int_0^1 |f|^2(t)\mathrm{d}t \cdot 1$$

Notice that this is the action integral. This implies that $L^2(\gamma) \leq A(\gamma)$. If γ has a constant speed c, then the left and right side are equal to c^2, and the inequality becomes an equality (f, h are colinear).

So A achieves the lower bound of L over reparameterizations.

- L does not change when we reparameterize γ.
- A obtains its smallest value when it is constant speed and when A is using a [FILL]
- Suppose for the purpose of contradiction that γ is a minimizer of A but not a minimizer of L. Then there is another curve α with constant speed reparameterization $\tilde{\alpha}$ such that

$$L(\alpha) < L(\gamma)$$

- But the left side is $\sqrt{A(\tilde{\alpha})}$ and the right side is $\sqrt{A(\gamma)}$, meaning γ is not a minimizer of A. We therefore reject our assumption.

The constant speed geodesic equation

In a given coordinate chart, with components of a metric tensor field g written as g_{ij}, and its inverse written g^{ij}, constant speed geodesics are determined by

$$\ddot{q^i} + \frac{1}{2}g^{ij}(-\partial_j g_{kl})[FILL]$$