Probabilistic Modelling for Evolution and Biodiversity Research

Moritz Schauer
Chalmers University of Technology | University of Gothenburg

DDLS Research area symposium: Data-Driven Evolution and Biodiversity Research 22

with Frank v.d.Meulen (VU Amsterdam), Frank Schaefer (MIT/Julialab), Stefan Sommer (CPH)

Machine Learning + Science + Statistics

- Machine learning revolution
  - $\rightarrow$ Backpropagation of derivatives
- Scientific modelling
  - $\rightarrow$ Branching (stochastic) differential equation phylogenetic models.
- Probabilistic programming
  - $\rightarrow$ Automated Bayesian inference/MCMC
  - $\rightarrow$ Backpropagation of likelihood
  - $\rightarrow$ Felsenstein's algorithm (back-propagation of likelihood on a tree)

Models

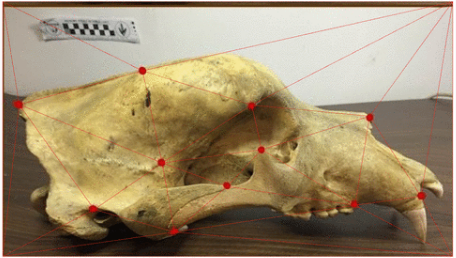
Functional (stochastic) model

$$\text{output} = F(\text{input}, \text{noise})$$

We want F to be anything: A ODE model, a neural network layer, a branching process.

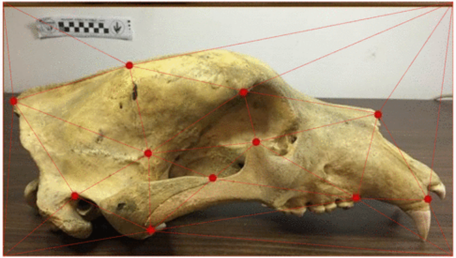
Anything?

Stochastic differential equation landmark dynamics for shape evolution



Anything?

Stochastic differential equation landmark dynamics for shape evolution



Derivatives

Derivatives are all about change.

Highly parametrized model: F predicting y given x and trained weights w

$$\hat{y} = F(x, w)$$

with "pullback" $\partial x, \partial w = \mathcal{J}(F, x, w, \partial \hat{y})$.

Backpropagation of derivatives

Nested model: L loss function, x input, y target

$$\begin{aligned} \hat{y} &= F(x, w) \\ \text{score} &= L(\hat{y}, y) \end{aligned}$$

Forward pass: Plug $\hat{y} = F(x, w)$ into $L(\cdot, y)$.

Backpropagation of derivatives

Nested model: L loss function, x input, y target

$$\hat{y} = F(x, w)$$
$$\text{score} = L(\hat{y}, y)$$

Forward pass: Plug $\hat{y} = F(x, w)$ into $L(\cdot, y)$.

Backward pass: Plug $\mathcal{J}(L, \hat{y}, \partial\,\text{score})$ into $\mathcal{J}(F, x, w, \partial y)$. Tells me: How to change w to reduce prediction loss.

Backpropagation of derivatives

Nested model: L loss function, x input, y target

$$\hat{y} = F(x, w)$$
$$\text{score} = L(\hat{y}, y)$$

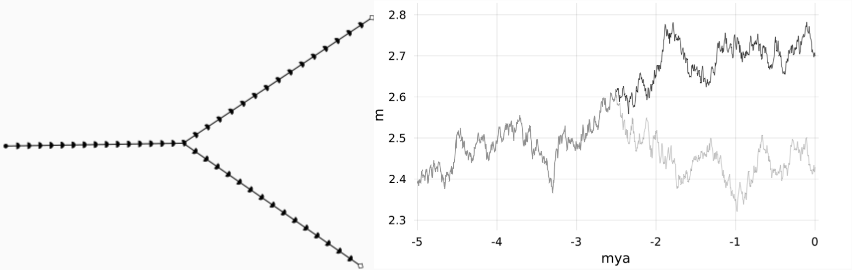
Forward pass: Plug $\hat{y} = F(x, w)$ into $L(\cdot, y)$.

Backward pass: Plug $\mathcal{J}(L, \hat{y}, \partial\,\text{score})$ into $\mathcal{J}(F, x, w, \partial y)$. Tells me: How to change w to reduce prediction loss.

Missing piece when using parameter twice: total derivative/adding up derivatives.

Tensorflow/Flux/JAX does this automatically.

Trait evolution



Possible latent evolution of trait $x(t)$ according to a random walk model. Species A in grey, species B in black separate at time τ.

Backpropagation of likelihood

Inference is all about conditional distributions.

Stochastic model/emulator for evolution of trait x on time interval $[t_0, T]$ with speciation event

$$
\begin{aligned}
\tau, x_\tau &= \mathrm{evolve}(t_0, x_0, \theta, \mathrm{noise}, \mathrm{until} = \mathrm{speciation}) \\
x^1, x^2 &= \mathrm{branch}(x_\tau) \\
x^1_T &= \mathrm{evolve}(\tau, x^1, \theta, \mathrm{noise}, \mathrm{until} = T) \\
x^2_T &= \mathrm{evolve}(\tau, x^2, \theta, \mathrm{noise}, \mathrm{until} = T) \\
&\mathrm{observe}(x^1_T, x^2_T)
\end{aligned}
$$

Backpropagation of likelihood

Inference is all about conditional distributions.

Stochastic model/emulator for evolution of trait x on time interval $[t_0, T]$ with speciation event

$$
\begin{aligned}
\tau, x_\tau &= \mathrm{evolve}(t_0, x_0, \theta, \mathrm{noise}, \mathrm{until} = \mathrm{speciation}) \\
x^1, x^2 &= \mathrm{branch}(x_\tau) \\
x^1_T &= \mathrm{evolve}(\tau, x^1, \theta, \mathrm{noise}, \mathrm{until} = T) \\
x^2_T &= \mathrm{evolve}(\tau, x^2, \theta, \mathrm{noise}, \mathrm{until} = T) \\
&\mathrm{observe}(x^1_T, x^2_T)
\end{aligned}
$$

Backpropagate the log-likelihood with e,g
$\mathrm{likelihood} = \mathcal{L}(\mathrm{emulator}, \mathrm{inputs}, \mathrm{observation/likelihood})$

Missing piece when branching: fusion/adding up log-likelihoods.

Next steps

Master plan: ML infrastructure for inferential phylogenetics.

Contact: `smoritz@chalmers.se`

Link: `http://mschauer.github.io/ddls/`