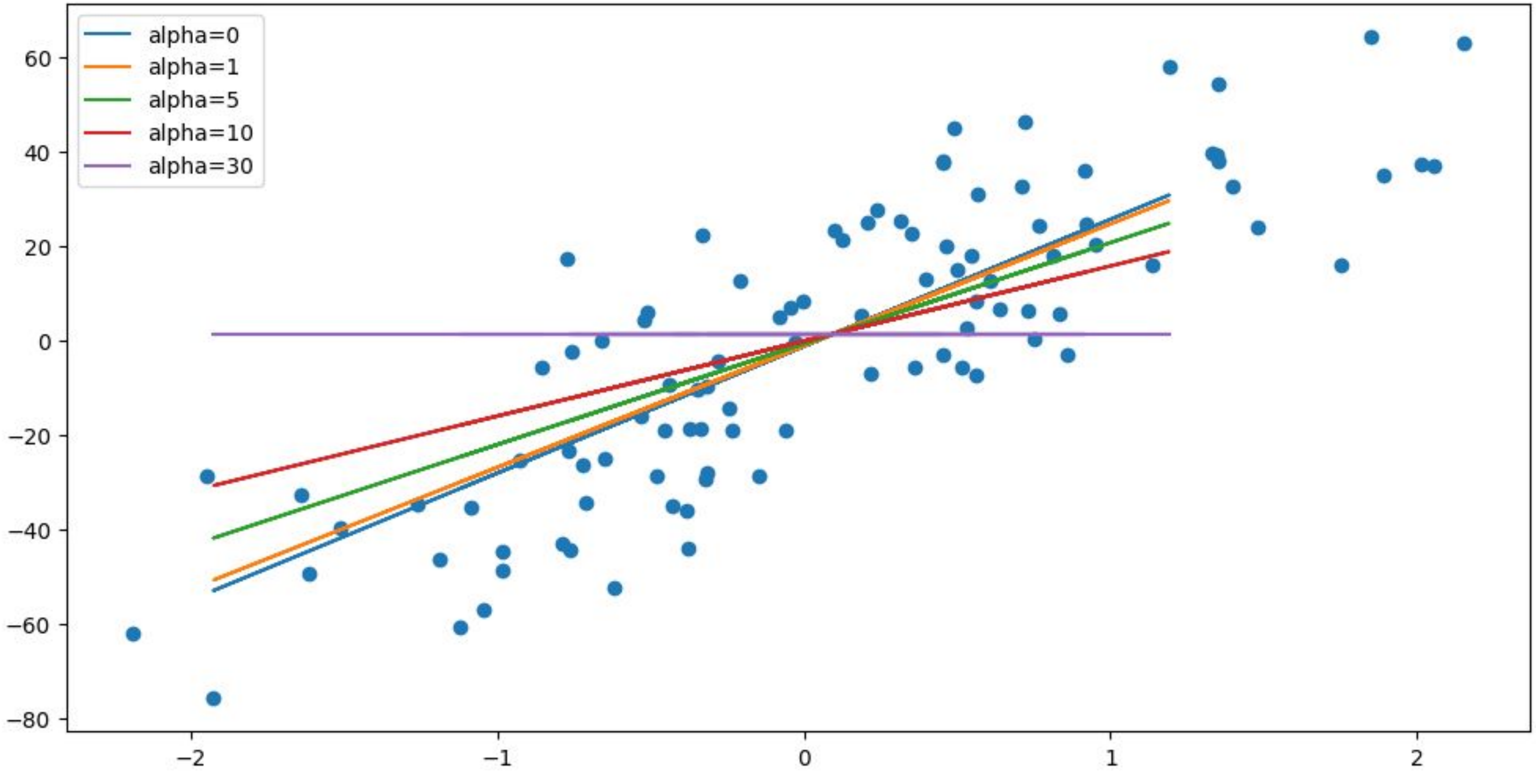
Implicit Differentiation: tutorial, and application to BirdFlow

Jacob Epstein

Lasso Regression



$$w^* \in \mathbf{argmin}_w \frac{1}{2n} \mid\mid X_{tr} w - y_{tr} \mid\mid_2^2 + \alpha \mid\mid w \mid\mid_1$$

$$g(w^*) = \frac{1}{2k} \mid\mid X_{val} w - y_{val} \mid\mid_2^2$$

How do we pick α to minimize $g(w^*)$?

Methods of hyperparameter selection

- *Manual*
  - *By-Hand*
  - *Grid Search*
- *Black-box*
  - *Bayesian optimization*
- *Differentiable*
  - *Unrolling*
  - Implicit differentiation

Solving, with implicit differentiation

$$w^* \in \text{argmin}_w \frac{1}{2n} \| X_{tr} w - y_{tr} \|_2^2 + \alpha \| w \|_1$$

$$g(w^*) = \frac{1}{2k} \| X_{val} w - y_{val} \|_2^2$$

- From implicit function theorem: $w^* = f(\alpha)$, for differentiable f
- Run gradient descent on the function $g(f(\alpha))$

Deriving Implicit Differentiation

$$w^* \in \mathbf{argmin}_w \mathcal{L}^{in}(w, \theta)$$

$$\min_\theta \mathcal{L}^{out}(w^*, \theta) = ?$$

$\mathcal{L}^{in}$ - inner loss, $\mathcal{L}^{out}$ - outer loss, $w^* \in \mathbb{R}^n$ - optimal parameters, $\theta \in \mathbb{R}^m$ - hyperparameters

$$\partial_w \mathcal{L}^{in}(w^*, \theta) = 0$$

By the implicit function theorem, there exists a function f defined near θ that satisfies:

- $f(\theta) = w^*$
- $\partial_w \mathcal{L}^{in}(f(\alpha), \alpha) = 0$ for all α

$\mathcal{L}^{in}$ - inner loss, $\mathcal{L}^{out}$ - outer loss, $w^* \in \mathbb{R}^n$ - optimal parameters, $\theta \in \mathbb{R}^m$ - hyperparameters

By the chain rule:

$$0 = \frac{d}{d\theta}\mathcal{L}^{in}(f(\theta),\theta) = \partial_w^2\mathcal{L}^{in}(f(\theta),\theta)\cdot f'(\theta) + \partial_{\theta w}\mathcal{L}^{in}(f(\theta),\theta)$$

$$A = \partial_w^2\mathcal{L}^{in}(f(\theta),\theta) \in \mathbb{R}^{n\times n}, B = \partial_{\theta w}\mathcal{L}^{in}(f(\theta),\theta) \in \mathbb{R}^{n\times m}$$

$$\boxed{\implies -Af'(\theta) = B}$$

In practice, this linear system is solved for $f'(\theta)$

$\mathcal{L}^{in}$ - inner loss, $\mathcal{L}^{out}$ - outer loss, $w^* \in \mathbb{R}^n$ - optimal parameters, $\theta \in \mathbb{R}^m$ - hyperparameters

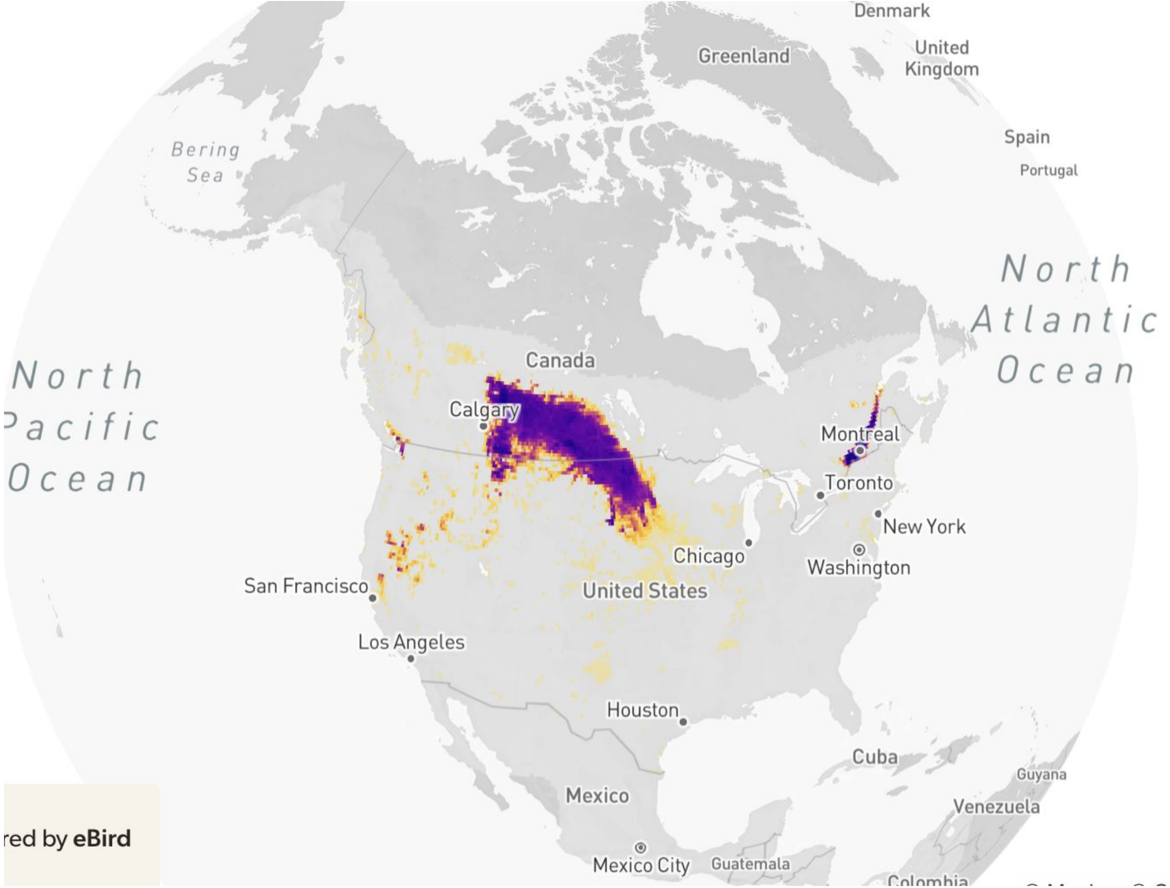
Finally, applying the chain rule again yields:

$$\nabla_\theta := \frac{d}{d\theta}\mathcal{L}^{out}(f(\theta),\theta) = \partial_w\mathcal{L}^{out}(f(\theta),\theta)f'(\theta) + \partial_\theta\mathcal{L}^{out}(f(\theta),\theta)$$

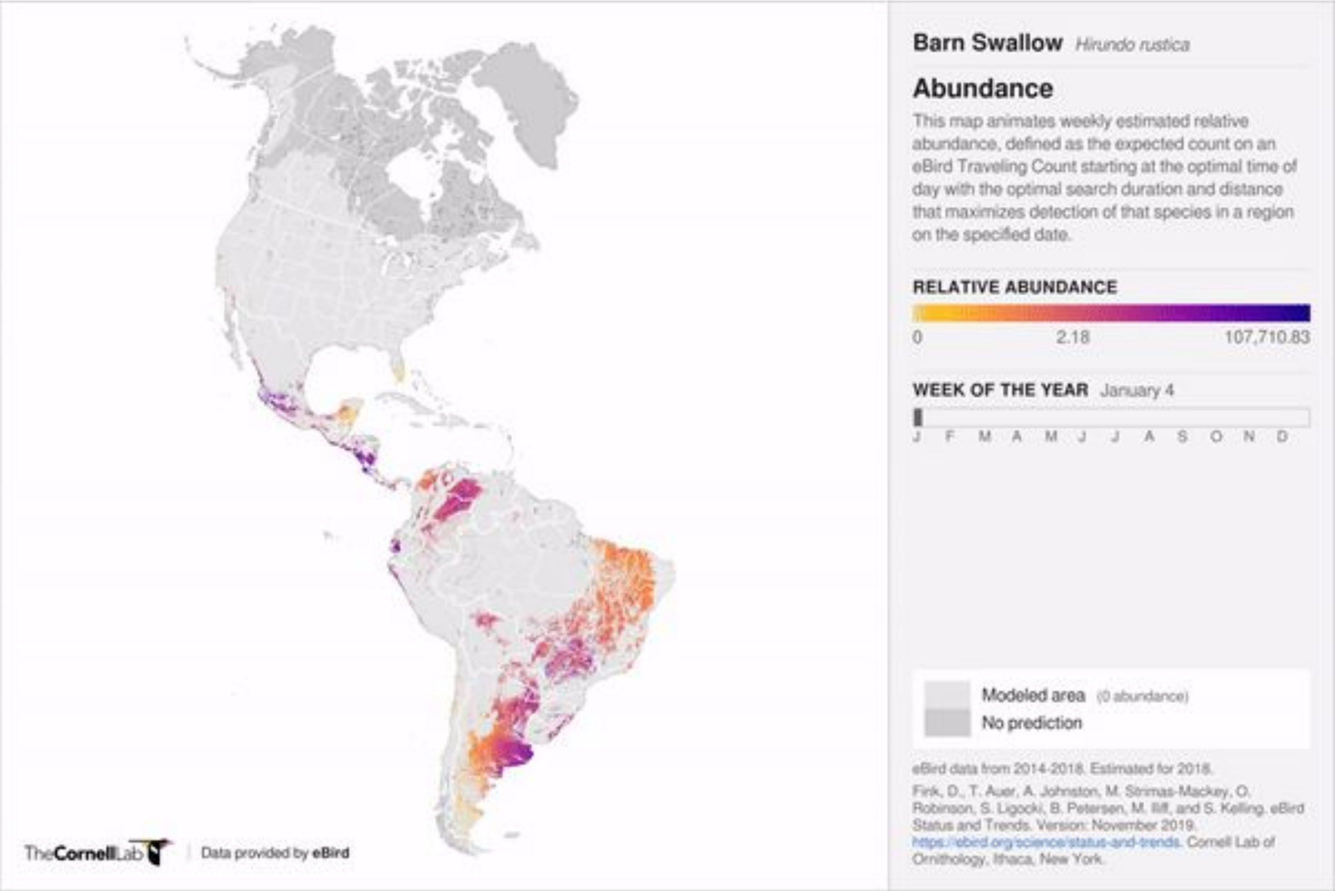
ADAM can be used with ∇_θ to find $\min_\theta \mathcal{L}^{out}(f(\theta),\theta)$

$\mathcal{L}^{in}$ - inner loss, $\mathcal{L}^{out}$ - outer loss, $w^* \in \mathbb{R}^n$ - optimal parameters, $\theta \in \mathbb{R}^m$ - hyperparameters

Live Demo - Implicit differentiation toy example



Snow goose population distribution, 4/12-4/19



Ebird weekly abundances for Barn Swallow

Can we learn a generative model for yearly flightpaths from weekly abundances?

BirdFlow

Model migration as a markov process, over a discrete sample space. Learn parameters θ, of a Markov chain.

From θ, we can compute weekly marginals μ_t (which should line up with the abundances), and pairwise marginals $\mu_{t,t+1}(i,j)$,

- $\mu_t(i)$ is the probability a bird is in grid cell i in week t
- $\mu_{t,t+1}(i,j)$ is the probability a bird is in grid cell i in week t, and in grid cell j in week $t+1$
- Assuming a discrete sample space of n grid cells, we can consider $\mu_t \in \mathbb{R}^n$, and $\mu_{t,t+1} \in \mathbb{R}^{n \times n}$

How do we define a loss function of θ?

Model loss is defined in terms of the marginals

The loss on the marginals of θ is a weighted sum of the **location loss**, plus a distance (depends on 2-way marginals) and entropy term

$$\mathcal{L}(\mu(\theta)) = \mathcal{L}_{loc}(\mu) + \mathcal{L}_{dist}(\mu) + \mathcal{L}_{ent}(\mu)$$

where $\mu = (\mu_1, \ldots, \mu_T, \mu_{1,2}, \ldots, \mu_{T-1,T})$, the vector of all 1 and 2-way marginals concatenated together. I'll be focusing on the location loss.

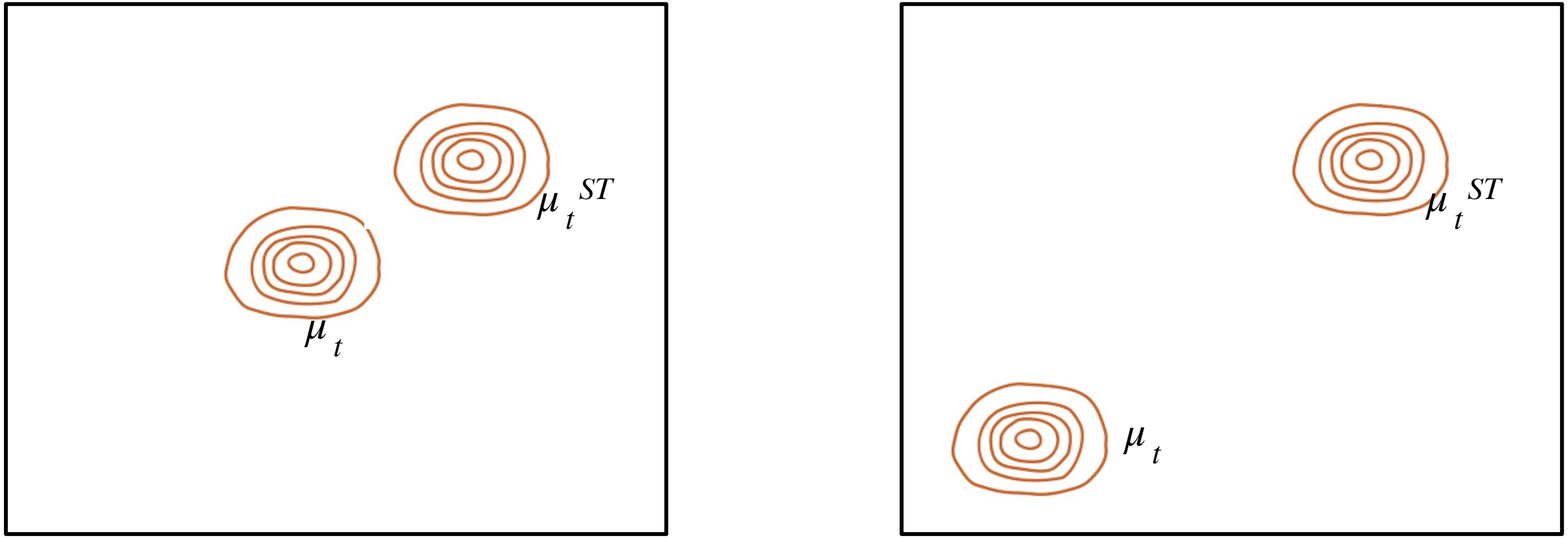
Location Loss

Makes rigorous the idea that model marginals should align with abundance estimates.

$$\mathcal{L}_{loc}(\mu) = \sum_{t=1}^{T} || \ \mu_t - \mu_t^{ST} \ ||_2^2$$

Use l^2 norm to compare model marginals μ_t with ebird status & trends marginals μ_t^{ST}.

Question - is it wise to use the l^2 norm for location loss comparison? We are comparing probability distributions. What if we used a transport-based metric instead?



l^2 norm is the same in both cases! Not so for W_2 …

2-Wasserstein Location Loss

2-Wasserstein distance is at a high level, the cost of transporting one probability distribution to another.

Rigorously, define a coupling of two distributions $\mu, \nu \in \mathbb{R}^n$ over n grid cells to be a matrix $\gamma \in \mathbb{R}^{n \times n}$ which satisfies:

- Elements of γ are positive, and sum to one
- $\sum_{j=1}^{n} \gamma_{ij} = \mu_i$
- $\sum_{i=1}^{n} \gamma_{ij} = \nu_j$

Let $\Gamma(\mu, \nu)$ be the set of all couplings between μ, ν. Then,

$$W_2^2(\mu, \nu) = \inf_{\gamma \in \Gamma(\mu,\nu)} \sum_{i=1}^{n} \sum_{j=1}^{n} || \, i - j \, ||_2^2 \, \gamma_{ij}$$

$W_2(\mu, \nu)$ is the minimum expected "transport distance" of any coupling of μ, ν.

Idea: modify $\mathcal{L}^{loc}$ to use W_2^2 instead of $||\cdot||_2^2$

$$\mathcal{L}^{loc}(\mu) = \sum_{t=1}^{T} W_2^2(\mu_t, \mu_t^{ST})$$

Note that computing $L^{loc}(\mu)$ now requires an inner (constrained) optimization process. We can do this efficiently with implicit differentiation...

Planned work for Spring '25

- Investigate to what extent a 2-Wasserstein location loss is helpful for birdflow
    - Train markov chains with original and 2-Wasserstein losses, evaluate validation metrics on each

- Interesting tidbits
    - Implicit differentiation and W_2 distance in a 'learn from marginals' setup
    - Applying implicit differentiation at scale, on a real-world problem

Questions?

asked ChatGPT to generate image of “implicit differentiation and bird migration” —---------->

Outline

- Part 1 - motivation
  - Automatic hyperparameter tuning
  - Suppose you are doing LASSO regression on some dataset, and wish to tune the regularization parameter, lambda
  - Options: tune by hand, grid search, bayesian optimization, unrolling, implicit differentiation
    - implicit differentiation exploits the fact that the computation of validation score from hyperparameters is end-to-end differentiable (assuming that the validation metric is suff. nice), allows gradient descent solvers to be leveraged to use gradient descent to find optimal hyperparameters.
- Part 2 - overview of implicit differentiation
  - Derivation
  - Live demo on toy example
- Part 3 - proposed application to BirdFlow
  - Background / motivation: replace euclidean-distance based distance loss with the W2 distance loss
  - W2 distance loss: requires an optimization problem to be solved as a step in computation. Now, we use implicit differentiation to get gradients with respect to parameters of birdflow
  - Plan for experiments: compare original and W2 models