A Framework for Studying Animal Behavior Using Deep Learning

Marcel Gietzmann-Sanders, Michael Courtney, Andrew Seitz, Curry Cunningham

University of Alaska Fairbanks

This study explores using deep learning to build Individual-Based Models (IBMs). Using Chinook salmon (*Oncorhynchus tshawytscha*) PSAT data as a case study, we introduce a methodology that leverages deep learning to transform the construction of IBMs into a systematic tool for investigating behaviors and their covariates. By employing a "log-odds model" to reduce dimensionality, this approach positions modeling as a central driver of discovery, enabling efficient hypothesis testing and iterative refinement.

Introduction

Modeling organism behavior can be thought of as equivalent to modeling choices and selections. For example if modeling animal movement, we can imagine each timestep as a decision d_i where the animal is selecting from a series of grid cells in its vicinity. Each such grid cell would represent a choice $c_{i,j}$. For each of these cells we may have a series of features $\vec{v}_{i,j}$ such as environmental covariates, descriptors such as distance or direction from the animal, and/or features of the animal itself or its state. Then we could pose our model as the function $F(\vec{v}_{i,j})$ which gives the probability of selecting $c_{i,j}$ given the information at hand. That is, we model the animal's behavior as a discrete Markov process.

The fact that this process produces probabilities per choice as opposed to explicit predictions of behavior is advantageous for two reasons:

1. With a deterministic model that predicts a specific trajectory you require a high level of $\vec{v}_{i,j}$ to use the model beyond a small number of steps as errors can accumulate. Here because we are predicting probabilities across all trajectories we can see what $\vec{v}_{i,j}$ elucidates and what it doesn't.

2. Because a specific likelihood is provided for each choice and trajectory we can identify specific decisions, individual organisms, places, or periods of time in which the model gives low likelihoods relative to the rest of the data making the model a tool for exploratory data analysis as well.

The fact that we have posed it in terms of distinct choices means that this is precisely a probabilistic machine learning problem (Oliver Durr, 2020) meaning that we can take advantage of machine learning to learn $F(\vec{v}_{i,j})$. This carries with it a few advantages as well.

1. Machine learning models, especially deep learning models, allow for very flexible pattern matching.

2. Rather than having to rebuild a model for each addition or subtraction to $\vec{v}_{i,j}$ we can reuse the same tool set to build and validate models regardless of $\vec{v}_{i,j}$.

This combination presents a kind of virtuous cycle in that the modeling process can be used to provide the very exploratory data analysis that can point to improvements in the model itself, and then whatever new features are derived can be incorporated into the model using the same tooling as before, thus allowing model development to supercharge itself.

There is significant precedent here as Hidden Markov Models (HMMs) and machine learning have both seen extensive use in movement modeling (Dhanushi A. Wijeyakulasuriya, 2020). However while each of these in turn bring some of the advantages outlined above they do not typically combine them.

Hidden Markov Models, by definition, do provide a conditioned probability distribution but the examples from the literature require explicit assumptions about how the data and behavior interact and therefore aren't able to take advantage of the flexibility and automation provided by machine learning (Dhanushi A. Wijeyakulasuriya, 2020). Therefore they provide they can technically provide the first set of advantages but not the second.

The applications of machine learning in the space do the opposite. They allow for the flexible and automated training of models, but those models typically make explicit predictions on movement, such as regressions on step length and angle thereby providing the latter set of advantages but not the former (Dhanushi A. Wijeyakulasuriya, 2020)(Daniel Clark and Clark, 2021)(Daniel Einarson and Sennersten, 2024).

Only with the two together can you...

Moreover with this more abstract and less explicit framing the applications are far broader...

With this in mind this paper has two goals:

1. To provide a guide to using probabilistic deep learning to model animal behavior that demonstrates how to overcome some of the practical issues that arise in its application

2. To demonstrate, through an illustrative example, how the artifacts of these models can be used as tools for EDA and thereby drive a virtuous model development cycle.

In order to further reduce the barrier to taking advantage of these techniques, the authors have provided a codebase at https://github.com/networkearth/mimic with all of the tooling required to setup probabilistic modeling in the cloud.

Applying Probabilistic Machine Learning to Behavior

Theory

Standard probabilistic deep learning networks are typically framed as a classification problem, using categorical cross-entropy as the loss function (Oliver Durr, 2020). Each output neuron represents a potential choice, with the model predicting the probability of each choice being correct based on this loss formulation. For these choices, we provide the network with features encapsulating the relevant information. Training is then comprised of providing a series such decisions.

However, this formulation introduces a critical challenge: if there are N features per choice and M potential choices, the overall dimensionality of the input space becomes $N \cdot M$. Adding even a single feature increases the dimensionality by M not just 1.

This growth poses a significant challenge due to the "curse of dimensionality", where the amount of data required to effectively train models can grow exponentially with the dimensionality of the input space (Verleysen and François, 2005).

Log-Odds Modeling

To address this issue we could take advantage of the order invariance of choices in the traditional probabilistic problem framing. Specifically, the order in which choices are presented to the model should not matter. For instance, whether a particular choice appears in the first or the thirteenth position should have no impact on the model's operation. This property allows for data augmentation by reordering choices.

In essence, for each training example, $M!$ (factorial of the number of choices) augmented examples can be created.

The issue with this approach is that as your augmented data size grows to match the needs of the greater dimensionality, so too does the time required for training. So while the problem remains theoretically possible, the potential exponential uptick in time complexity poses a significant practical issue.

Instead we propose an adjustment to the standard framing of probabilistic machine learning. Instead of predicting the probabilities directly, we predict the log-odds ϕ_m for each choice and calculate the probability p_m using the softmax function:

$$p_m = \frac{e^{\phi_m}}{\sum_{m=1}^{M} e^{\phi_m}}$$

This approach reduces the feature space dimensionality to N and effectively increases the number of training examples by a factor of M.

We can implement this log-odds model using standard probabilistic deep learning techniques by replicating the "log-odds model" weights across all M choices. The outputs are fed into a softmax layer with M units, where the layer's weights are set to the identity matrix and biases are set to zero. Using categorical cross-entropy as the loss function ensures compatibility with standard probabilistic deep learning while enabling us to train the log-odds weights and significantly reduce the problem's dimensionality.

Contrast Sampling

A practical issue with our log-odds framing is that as M grows large most instances of the internal log-odds model would ideally report very low log-odds, resulting in low probabilities. Ideally, only one choice should produce $p_m = 1$. This is analogous to a class imbalance problem, where the model becomes prone to predicting the most common class.

To address this, we balance the training data. Instead of presenting the model with full decisions containing all M choices, we create training pairs, or contrasts, where each pair consists of one selected choice and one unselected choice. This approach is valid because the log-odds model focuses on the relative likelihood of choices, making the number of choices considered at any one time irrelevant.

The primary risk in using contrasts is introducing bias by disproportionately sampling certain combinations of choices. To mitigate this, we randomly sample pairs from each decision, ensure an equal number of contrasts per decision, and an equal number of decisions per individual. This preserves the balance across the training data and avoids skewing the model's predictions.

Taking Advantage of the Cloud

In deep learning the specific layer sizes, depth, optimizer, learning rate, and other parameters best suited to a particular problem are not usually known from the start. Therefore it is important to do hyperparameter tuning in order to discover the best parameters for a particular problem. Practically this means training large numbers of models - a process that is usually very compute intensive. Even with just 3 parameters each of which can take on five results in 125 distinct models. Furthermore as with any search over a non-convex space (as backpropogation is) the initial seed can have impacts on the final solution. Therefore, training deep learning models requires training many models per problem.

This issue can be overcome by virtue of the fact that these models can all be trained independently and therefore in parallel. This is a perfect use case for cloud compute where one can spin up, for a short time and for a reasonable cost, a large number of machines and therefore do all of the hyperparameter tuning in parallel.

Summary

In summary the application of probabilistic modeling proposed here follows three steps:

1. Discretize the behaviors into distinct decisions d_i and choices c_i, j.

2. Resample the choices using contrast sampling

3. Apply log-odds modeling to the contrasts, taking advantage of cloud compute to efficiently search the hyperparameter space

4. Use the trained model to make inferences on the original decisions

The following example application demonstrates these steps in practice.

Application to Chinook Salmon Movement Data

Outline

This example will involve building three models of Chinook salmon (*Oncorhynchus tshawytscha*) movement that build upon one another. These specific models were chosen to illustrate the capabilities of this technique as an exploratory data analysis (EDA) tool. The features are described in the following table:

Model	Features
Distance Model	distance to choice
Heading Model	+ heading to choice
Food Model	+ primary productivity, mixed layer thickness

and their purposes (in terms of EDA) are described here:

Model	Purpose
Distance Model	A null model for comparison
Heading Model	Look at heading tendencies in the data
Food Model	Explore deviations from averages due to productivity

We will begin by describing the data and associated features, then move onto a description of the building of the models, and finally, in the results and discussion, illustrate how the models can be used to explore the data.

Data and Features

The data used is a series of tracks from 111 Chinook salmon (*Oncorhynchus tshawytscha*) caught and monitored between 2013 and 2022 (Michael B Courtney, 2021) (Michael B Courtney, 2019). These tracks were obtained from pop-up satellite archival tags which collect temperature, light level, and depth information at specified (sub day) intervals. This data is then passed through a proprietary algorithm from Wildlife Computers to determine likely longitude and latitude during each day of of monitoring (Wildlife-Computers, 2024).

Environmental data was derived from the Global Ocean Biogeochemistry Hindcast dataset (10.48670/moi-00019) and the Global Ocean Physics Reanalysis (10.48670/moi-00021) from the E.U. Copernicus Marine Service Information. Net primary production (mg/m3/day) and mixed layer thickness (m) were aggregated per Uber h3 resolution 4 cell in the Northern

Pacific.

Movement heading in radians and distance to the centroid of the choice cell were computed and then mixed layer thickness and net primary production were joined to the choices on cell and day.

Distance was normalized to a range of 0-1 by division by 100, while mixed layer thickness and net primary production were both log-scaled and then centered at zero.

Building the Models

Train and Validation Sets

The first step in any other machine learning problem is splitting training and validation sets. Given the low number of individuals in the sample and the intention to demonstrate the technique's value as an EDA tool it was decided to not maintain a hold out test set.

71 individuals were randomly selected for training and 40 for validation.

Formulation

The next step in building a log-odds model is to decide on the formulation of our choices. In our case we decided to grid space by Uber h3 cells at resolution 4. Specifically, the cell containing each salmon location from our data was identified and then, assuming a maximum travel distance of 100km (centeroid to centeroid) all adjacent cells within the 100km were identified as choices (including the currently occupied cell). In general this represented $\sim$ 19 choices per decision with the intention being to predict the probability of moving to any particular cell. Training data was derived by identifying the actual cell moved to.

Contrast Sampling

The next step in building a log-odds model is determining the specifics of the contrast sampling. For this example, after inspecting the distribution of number of choices per salmon and number of choices per decision, we decided on random sampling (with replacement) 200 decisions per individual and 19 choices per decision.

Over a validation/training split of 40, 71 this resulted in 421,800 contrasts of which 269,800 were used in training and the rest in validation.

Note that only 14,200 training examples would've been available to a traditional probabilistic approach representing a large increase in the number of available training examples.

Training and Model Selection

For each of the three models trained architectures/hyperparameters for the internal log-odds component of the model were parametrized in the following ways:

Component	Options
Layers	3, 4
Units per Layer	24, 32
Batch Size	10000
Learning Rate	0.0005

5 separate seeds were used for each combination. Models were trained in Keras using an Adam optimizer for 100 epochs. Training was done on AWS Batch using Fargate instances of 2 vcpu's and 4 GB of memory. By taking advantage of AWS Batch, models could be all trained in parallel allowing for short (1 hour) turn around times.

Lowest loss (categorical cross entropy) at the end of the 100 epochs over the validation dataset was used to select the best set of parameters for each of the three models trained.

Results

The loss, categorical cross entropy, is equivalent to the average (normalized) log probability per contrast. For an equivalent metric over the actual decisions made by the individuals we computed the average log probability per decision for each individual and then computed an average over those across individuals (in order to not favor individuals with many decisions). This is the D-NLP reported in the table below.

Model	Train C-NLP	Val C-NLP	Train D-NLP	Val D-NLP
No Model	-0.693	-0.693	-2.944	-2.944
Distance	-0.172	-0.154	-1.336	-1.223
Heading	-0.156	-0.150	-1.281	-1.200
Food	-0.147	-0.146	-1.248	-1.180

”No Model” assumes all decisions are equally likely, ”Distance” is the distance only model, ”Heading” adds the movement heading, and ”Food” adds the net productivity and mixed layer thickness features.

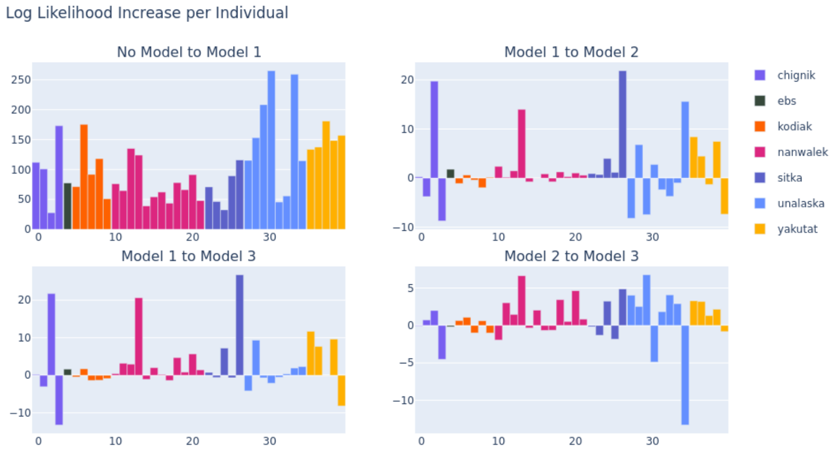

Figure 1: Log likelihood increase (unnormalized) per individual of each model change.

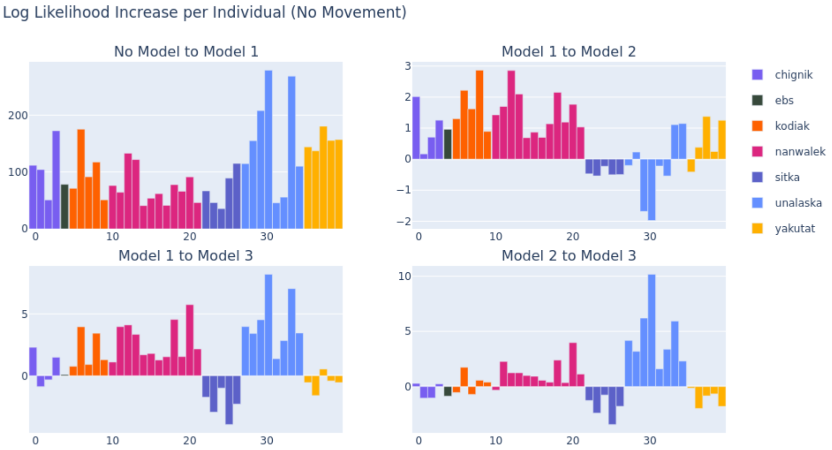

Figure 2: Log likelihood increase (unnormalized) per individual of each model change in cases where the selection involved no movement.

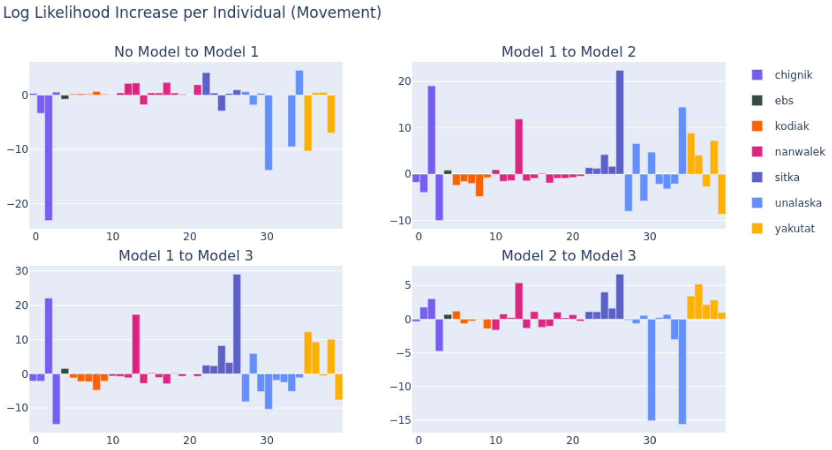

Figure 3: Log likelihood increase (unnormalized) per individual of each model change in cases where the selection involved movement.

Figure 1 shows the unnormalized changes in log likelihood for each individual colored by the region in which they were initially tagged as we go from one model to the next. Figures 2 and 3 break this down for decisions that resulted in no movement and movement respectively. Only individuals in the validation set were considered.

The appendix contains a series of mapped examples (figures 4-9) from each of these regions (besides EBS) that shows how the log likelihood per decision (when moving) changed in moving from model 1 to model 2 and then from model 2 to model 3. As in the above, only individuals in the validation set were considered.

Finally as much of the behavior seems to be modulated by movement distances (or lack of movement) a summary table of empirical likelihood of movement of a specific distance is given (taken over all individuals).

Distance Bin (km)	Likelihood of Selection	# Training Decisions
No Movement	65.6%	3163
Up to 50km	32.5%	1567
50km to 100km	1.9%	92

Discussion

The Distance Model

In any probabilistic classification problem model performance has to be evaluated against some kind of "null" baseline. For example in our case, given there are normally 19 choices available in each decision any model must be able to produce likelihoods of selecting the correct choice greater than $1/19 \approx 0.05$ on average. Anything below this and a purely random guess is better.

In our example however a random guesser can be smarter than this without having any information on the environment or organisms in question because there are simple descriptive features that can be derived from the formulation alone. I.e., someone with just those statistics could become an "informed random guesser". In this analysis we look at two of them - distance and movement heading.

The first, distance, provides a significant drop in our loss from -2.944 in the no model case to -1.223 in the distance model. We can also plot the change in the log likelihood per individual track (figure X) and see that adding distance to the model represents an improvement across every single individual considered. Our random guesser is very usefully informed by the distance associated with each choice and so our ”null model” should include this feature.

The Heading Model

Next we turn to the model that includes the movement heading per decision as well. Here we also see a shift in the loss but of far lesser magnitude than in going from the no model case to the distance model. Specifically we go from a validation loss of -1.223 to -1.200. While this is useful, far more informative is a comparison plot that shows us the shift in log likelihood per individual track when going from the distance model to the movement heading model (figure X). Unlike the latter figure this one restricts itself to choices where movement occurred. This is our first example of the usefulness of these models in doing EDA as this plot is effectively showing us, per individual, the alignment with average tendencies of movement in the data.

The individuals with marked positive increases represent individuals well aligned to the movement heading tendencies in the data whereas those quite negatively aligned show a contrary pattern. Using this we can then plot the movements themselves for these ”exemplar” individuals colored by their changes in likelihood compared to the distance model (blue is positive, red is negative) to see what those general tendencies in the data are (figure X). From this it is clear that the model is learning a south westerly pattern as being the best direction to randomly guess.

This is all incredibly useful to the model builder as we are now in possession of exemplars who need little explanation beyond these basic descriptive features as well as specific exemplars that act contrary to the ”norm”. These individuals can now be studied more extensively to look for discriminating patterns we can use to further improve the model.

The Food Model

The examples in yakutat presented an especially interesting example in that the exhibit ”low likelihood” behavior at the beginning of their tracks and then switch to higher likelihood behaviors (according to the heading model) later on as they then proceed to move down the coast. This gave the authors a very specific shift to watch and observe for patterns and one pattern seemed to be that when the mixed layer thickness began to shrink the fish began to move south. To see more clearly how these features might interact with the ones already given mixed layer thickness and primary productivity were added and comparisons as before were done (figure X).

First, the addition of these features did make consistent improvements across individuals' movement choices in the yakutat and sitka groups and much of this improvement was on the tracks south. However these same groups all saw a decline in the likelihood of decisions that resulted in no movement - something that would certainly be worth additional investigation. For unalaska the opposite pattern is seen - decisions that resulted in no movement saw a strong positive increase whereas there were very strong drops in likelihood for movement decisions. The other groups had far more mixed responses. Looking at the specific likelihood changes per movement decision (figure X) in unalaska we see that the likelihoods of decisions in an easterly direction actually dropped as compared to the movement heading model whereas for yakutat the directions that were already boosted by the movement heading model were further boosted by the addition of these features. Clearly understanding these marked differences between the unlaska and sitka and yakutat groups is key to understanding some important aspect of these individuals' movement behavior.

Summary of EDA

The purpose of these three example models was to show how by using this modeling process and looking at the likelihood changes per individual, decision, or group of choices, the modeling can be used to identify classes of behaviors that warrant additional investigation as well as general tendencies in the data. These kinds of outcomes are precisely the purpose of exploratory data analysis and are essential to building, and understanding, robust models.

Beyond EDA

While the primary focus of this example has been directed at the use of probabilistic models as tools for exploratory data analysis, their predictive capabilities should not be overlooked. One of the key strengths of probabilistic modeling lies in its ability to perform well even when the available features are only weakly predictive. By outputting probabilities rather than deterministic predictions, the model can effectively capture uncertainty and express it in a way that is both interpretable and actionable.

In scenarios where features have low predictive power, deterministic models of behavior over time can struggle because of an accumulation of errors. Probabilistic models, on the other hand, distribute probability mass across choices, reflecting the underlying uncertainty in the data. This enables better decision-making under uncertainty, as the model not only identifies the most likely choices but also quantifies the likelihood of alternatives, given the data available.

Moreover, the iterative refinement of these models makes them particularly valuable in predictive contexts. As new information comes to light or existing information is improved there is no need to rebuild or rethink models from scratch, instead the same framework can be used to build the old as well as new models. This allows for faster evaluation, more flexibility and freedom in considering new information, and therefore a better chance of picking features in a purely data driven manner.

Wide Applicability

Finally while this example focused on one species - Chinook salmon - and one aspect of behavior - movement - this methodology applies to any problem that can be framed in terms of discreet decisions d_i and choices c_i, j. This could include movements of other species such as birds or mammals and other behaviors such as food selection, mating choices, or timing of spawning. By adapting the feature set and decision structure to the system of interest, researchers can reuse the same tools to study a wide variety of different behaviors.

Conclusion

This study highlights the dual power of probabilistic machine learning in providing both predictive insights and a robust framework for exploratory data analysis. The log-odds modeling technique exemplifies how dimensionality challenges can be overcome, and along with parallelism in the cloud enables rapid iteration and refinement of models to uncover patterns that would otherwise remain hidden. Ultimately, this framework is more than a predictive tool; it is an investigative lens, empowering researchers to probe the intricacies of behavior and the factors driving it.

Code

The tooling used to build these models as well as the means to deploy them using Amazon Web Services has been packaged at:
https://github.com/networkearth/mimic

References

Daniel Clark, David Shaw, A. V. S. W. J. S. and Clark, J. D. (2021). Using machine learning methods to predict the movement trajectories of the using machine learning methods to predict the movement trajectories of the louisiana black bear louisiana black bear. *SMUDataScienceReview*, 5.

Daniel Einarson, Fredrik Frisk, K. K. and Sennersten, C. (2024). A machine learning approach to simulation of mallard movements. *Applied Sciences*, 14.

Dhanushi A. Wijeyakulasuriya, Elizabeth W. Eisenhauer, B. A. S. E. M. H. (2020). Machine learning for modeling animal movement. *PLOS One*.

Michael B Courtney, Mark D Evans, J. F. S. A. H. R. A. C. S. (2019). Behavior and thermal environment of chinook salmon oncorhynchus tshawytscha in the north pacific ocean, elucidated from pop-up satellite archival tags. *Environmental Biology of Fishes*, 102:1039–1055.

Michael B Courtney, Mark Evans, K. R. S. A. C. S. (2021). Understanding the behavior and ecology of chinook salmon (oncorhynchus tshawytscha) on an important feeding ground in the gulf of alaska. *Environmental Biology of Fishes*, 104:357–373.

Oliver Durr, Beate Sick, E. M. (2020). *Probabilistic Deep Learning*. Manning Publications.

Verleysen, M. and François, D. (2005). The curse of dimensionality in data mining and time series prediction. *Computational Intelligence and Bioinspired Systems*, 3512:758–770.

Volker Grimm, S. R. (2005). *Individual-based Modeling and Ecology*. Princeton University Press.

WildlifeComputers (2024). Minipat.

Appendix: Mapped Examples

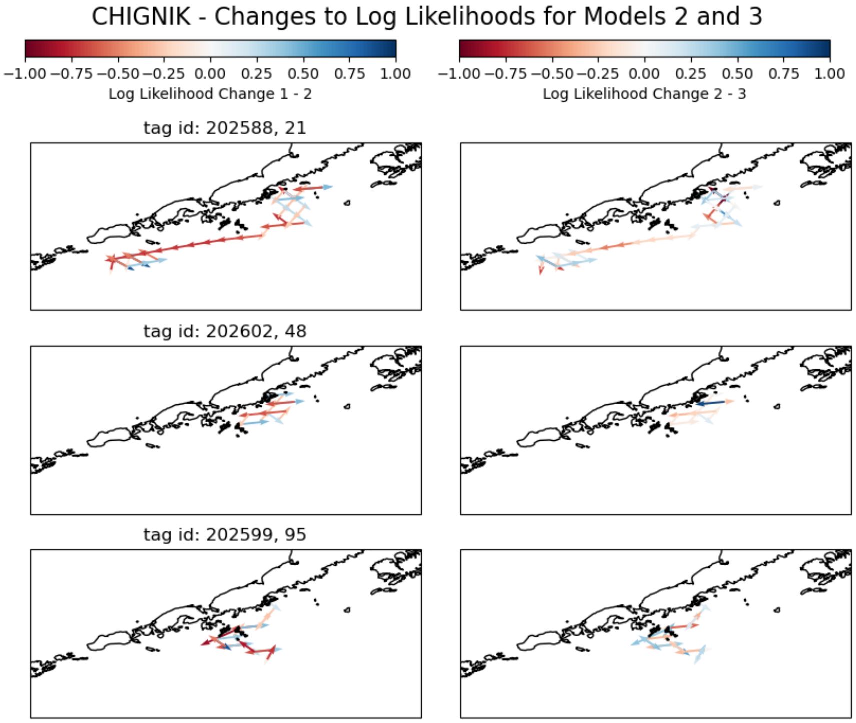

Figure 4: Log likehood changes per decision - Chignik

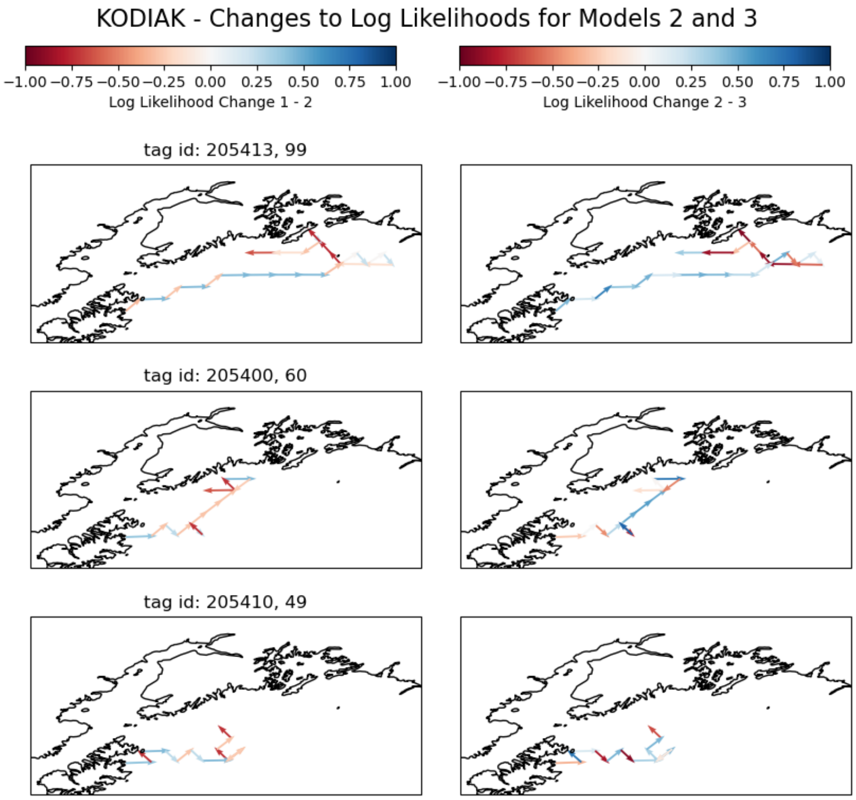

Figure 5: Log likehood changes per decision - Kodiak

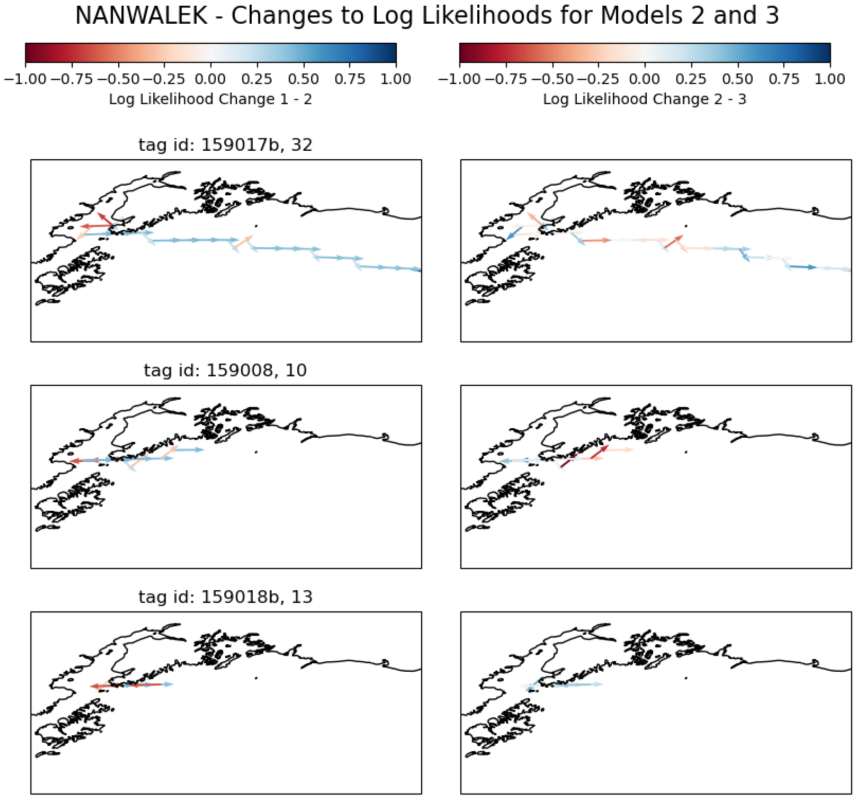

Figure 6: Log likehood changes per decision - Nanwalek

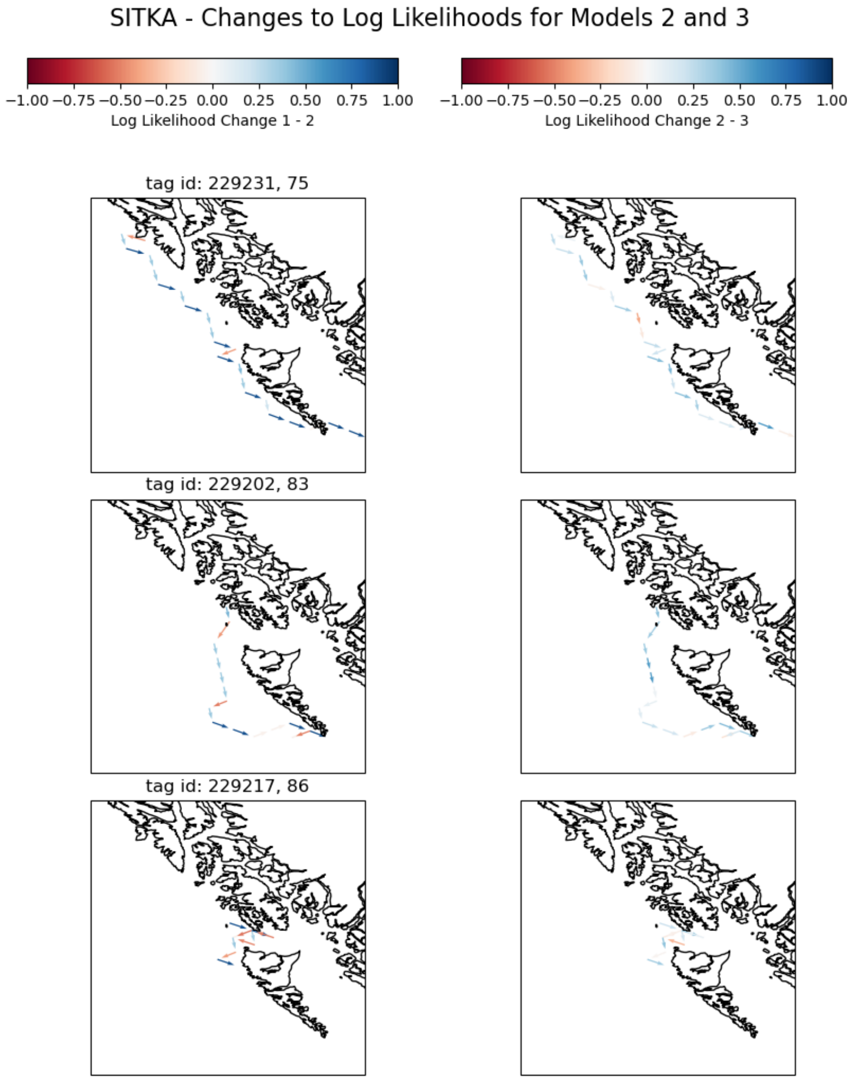

Figure 7: Log likehood changes per decision - Sitka

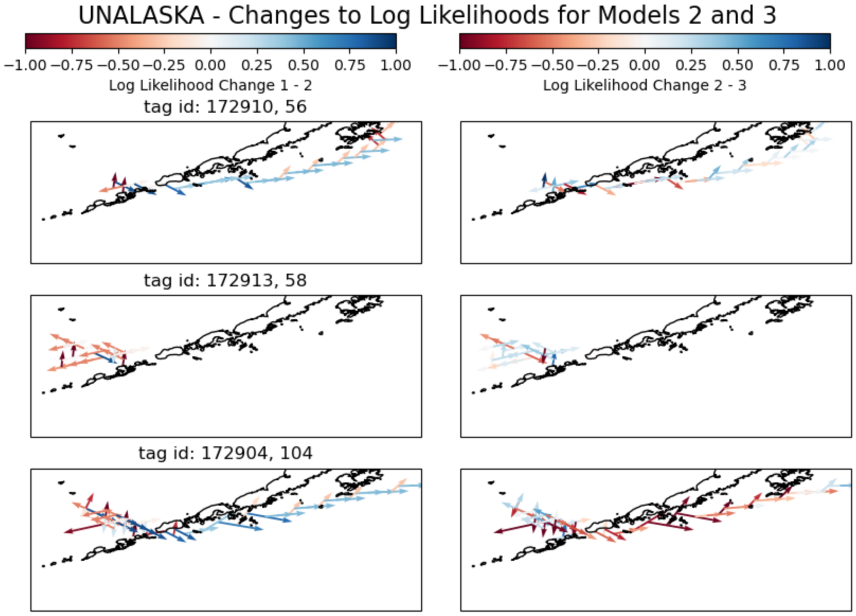

Figure 8: Log likehood changes per decision - Unalaska

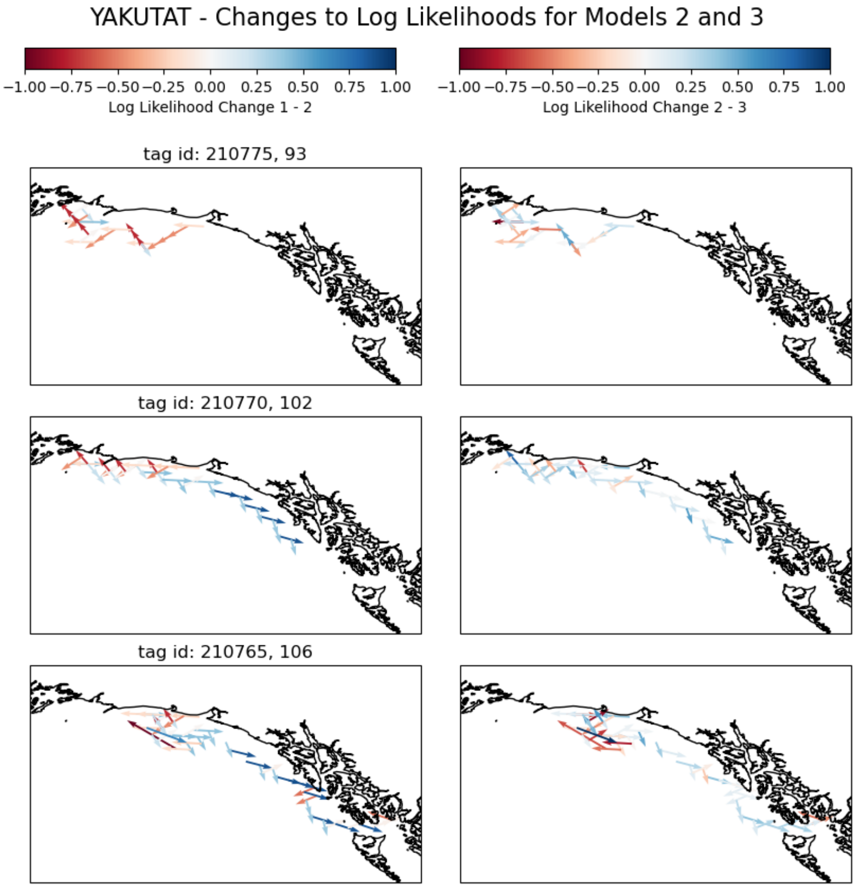

Figure 9: Log likehood changes per decision - Yakutat