**TMB study group meeting 3**

We will first go over some concepts.

1. State-space model vocabulary

2. Laplace approximation

3. Cross validation

4. Covariance matrix

5. Spatial model

*Problem 1*: Construct a state space model based on the diagram and process below. 1) Identify fixed and random effects. 2) Simulate some data with different number of Ds. 3) Estimate the MLE of parameters () in each scenario (varying #D).

).

A picture containing clock, photo

Description automatically generated

*Problem 2*: We will fit the catch data of witch flounder in 2J3KL in 2018 (witch\_flounder.csv). First, we need to deal with zeroes. We use a zero-inflated lognormal model:

We have a probability that data is zero, and if not a likelihood given a function. The goal is to estimate model parameters for linear spatial effects (latitude, longitude, probability of zero, observation error).

1) plot original data on a map.

2) Implement the likelihood in log space in TMB.

Hint1:

Hint 2: The lognormal likelihood is not built-in in TMB. You can add any type of functions via customized functions before your main function but after the headers. We will define the log-normal likelihood like this:

//log-normal likelihood

template<class Type>

Type dlognorm(Type x, Type meanlog, Type sdlog, int give\_log=0){

//return 1/(sqrt(2\*M\_PI)\*sd)\*exp(-.5\*pow((x-mean)/sd,2));

Type logres = dnorm( log(x), meanlog, sdlog, true) - log(x);

if(give\_log) return logres; else return exp(logres);

}

Make sure the model runs and calculate the AIC before proceeding.

3) Add a second likelihood function dinvgauss as an option. You can add a flag parameter to toggle the likelihood used. Compare AIC for both likelihoods to see which is better.

Hint 1:

Hint 2:

// Inverse Gaussian

template<class Type>

Type dinvgauss(Type x, Type mean, Type shape, int give\_log=0){

Type logres = 0.5\*log(shape) - 0.5\*log(2\*M\_PI\*pow(x,3)) - (shape \* pow(x-mean,2) / (2\*pow(mean,2)\*x));

if(give\_log) return logres; else return exp(logres);

}

4) Check the residuals.

Hint: Plot residuals vs lat/long [(pred-log(y))/sigma]

5) Assessing fit with cross validation.

The general principle of cross validation is first to divide data set onto K even partitions and calculate predictive probability for the one partition each time. To predict each piece in K, fit the model to all data except data in that partition. Calculate the predictive ability in partition that has not been used by the model. Repeat for all K partitions (called K-means cross validation). The model with highest predictive probability is the best.

Hint 1: use an Indicator vector of 0’s and 1’s to tell the program which part of the data to use.

Hint 2: Two nll’s are calculated for within sample and without sample.

*Problem 3*: Solve the residual problem via spatial modeling.

Here is the logic.

1) Each site (row of data) has a random effect (total n random effects).

2) Create a nxn pairwise distance matrix for all points.

3) Specify a function which relates distance to correlation, e.g. exponential decay.

4) Now we have a multivariate normal likelihood for all the data. (Hint: MVNORM\_t<Type> in TMB)

5) Fit the data to find parameters (latitude, longitude, probability of zero, observation error, decay rate and all random effects).