**Researching Mosquito Populations Using Particle Filter Modeling**

Professor: Nathanial Osgood, CMPT 394/858

Written by: Cong Peibo

Chin Wang

Date: December 28, 2014

**Researching Mosquito Populations Using Particle Filter Modeling**

**ABSTRACT:**

Modeling and researching the spread of the West Nile Virus relies heavily on knowledge of the mosquito population of a certain area or demographic; however it is getting increasingly difficult to accurately measure this population. Traditional methods epidemiologists use and experimental statistics are either non-reliable or extremely non-efficient. Here, we introduce particle filtering and we demonstrate how this approach can sample, update weights, and resample iteratively from a specified state space and other advantages to other methods. By exploring how these advantages can be applied to life sciences, we note important avenues for future work to make similar approaches.

**1. INTRODUCTION:**

Mosquitoes are the main transmitters of West Nile Virus disease. The degree of spread of the disease relates to the overall mosquito population, which has direct ties to the growth. The rate of growth can be further increased with factors such as amount of sunlight per day, relative humidity, temperature, etc.

Over the last decade, changes in the world’s climate have raised a red flag as being a potential catalyst for the development and increase of infectious diseases, especially the West Nile Virus; Therefore, the West Nile Virus has become of the most extensively circulated arboviruses in the world. In order to prevent and control the spread range of West Nile Virus, the primary objective is to accurately predict future populations of mosquitoes using historical data. Unfortunately, there are many limits and shortcomings in traditional methods. For example, epidemiology method requires gathering large data set, however collecting this data is severely restricted. As the time varies the population of mosquitoes varies from state to state, which then forms a state space. The change of the population of mosquitoes with time is non-linear and non-Gaussian; under this condition, we can think that one of the better methods to predict the population of mosquitoes in the future is the particle filtering model.

The particle filter is a sequential Monte Carlo algorithm, approximating a distribution using a large number of samples (particles) and as the number of particles N increases toward positive infinity, it will converge to the actual distribution in the state space.

The particle filter was built on the following principle of statistical theorems.

**(1). Bayesian inference**

In state space of our project, denotes the total number of mosquitoes in the week; it is a latent variable. denotes the number of trapped mosquitoes in the week, is an observation variable, and we assume it follows a binomial distribution. Obviously are forms of time series; Therefore, there are two equations:

***State equation***: is a Markov chain with initial probability density function (pdf) and transition density function, such that the prior distribution is

;

***Observation equation***: is such that the conditional joint density of given () has the likelihood function:

.

Our goal is to predict the state variable , given all of the observation values . One way is to find the posterior distribution by using Bayesian inference:

In the literature, ultimately we end up with two steps:

Update step:

Prediction step:

**(2) Importance sampling and resampling**

Sequential Monte Carlo is a method which is based on importance sampling and resampling. The basic idea of importance sampling is to assume that our goal is to compute . Where, is target distribution density, we wish to draw sample from it, but this is difficult, if we have a proposal distribution which is easy to sample from and keeps the main features of , we can sample and define the weight as , then the target distribution density can be estimated as , where is the delta-Dirac mass located at . Each sample is associated with a weight and state.

**(3) Sequential Monte Carlo Algorithm**

Step1. At time t=1, for i=1,2,...,N

1. Sample
2. Compute weights and

1. Resample to remove the particles with low weights and duplicate the particles with high weights.

Step2. For time implement a recursive update

1. Sample
2. Compete and Update the weights
3. Resample to remove the particles with low weights and duplicate the particles with high weights.

In our project, the goal of using the Particle Filter Model is to precisely predict(estimate) the total population of mosquitoes in a future time period.

**2. Deign Particle Filter Model**

**Step1: Going into AnyLogic**

The very first measure we had to ensure was to learn the interface and tools of AnyLogic. Basic models were created in order to experiment on many systems including table functions, functions, parameters, variables, feedback loops, stocks, flows and other useful tool. This step did not all added to the final project but gave us important insight and a base for the rest of the project.

**Step2: Trim the original model**

After vigorously investigating the original model, we trimmed the “Adding Predators to the System” section from the model. We completed this measure because of the following reasons:

* The purpose and goal of the build in the particle filter;
* The feasibility of the build in our model
* To simplify the process of the build in our model

After revised, we remain the main part of the original mosquito population mode that included Model weather data and Basic Exogenous Mosquito Model (Mosquitoes lifecycles: mosquito\_egg, mosquito\_larvae, mosquito\_pupae, and mosquito\_adult ).

**Step3: Building the particle filter**

**Background**

From the research concerning the purpose of our project, we realised that our goal is to estimate the total mosquito population (which would be a realistically sized latency variable) based on captured mosquito count in each period (an observation variable).The basic idea is that by taking samples of large quantities and weights we can update and resample to simulate the actual state.

**Structure**

After trimming the mosquito population model (system dynamic model) which acts as a foundation for our model, we continue structuring it as follows:

**Stage1**: We added the particle filter factor into all parts of the foundation in our model except the standing water. This was because we didn’t need the sample, update, weight or resample from it.

**Stage2**: We change the “Mean\_Time\_For\_Egg\_Hatch” dynamic variable from a fixed number (1.5) to a random number which mostly belonged to the interval. This change is according to the particle filter model where all variables are random variables.

**Stage3**: We added the parameter “ReportingProbability”. This probability denoted the chance of mosquitoes being trapped. The best value for this will get from calibration model.

**Stage4**: We inserted the “Diapause\_Adult\_Early” (stock and flow) between the “Mosquito\_Adult” and “Diapause\_Adult\_Late” which formed a first order delay to slow down the ratio of mosquitoes coming back from diapauses to being active in the winter. This can make our prediction smoother and closer to an actually situation.

**Stage5**: We changed “Adult\_Larvae\_Diapause” to become a particular filter so it doesn’t depend on day light because mosquitoes underground cannot sense day light. Lastly we added the following tables “posteriorHis2DSamplePosteriors” (after update), “PreHistSampledMosquito” (before update) and “chartHist2DSampledParticles”. The reason is to compare the results between the prior and posterior of update and also to compare the complete latency with empirical data.

**3. Calibration**

The purpose of creating the calibration model is that it applies an optimization algorithm to adjust unknown parameters so that it fits a large set of historical data. The calibration experiment will help us run the calibration model many times to find the” best values” of the unknown parameters until we find the best match for al historical data.

**Parameters:**

In our project, we attempted to search the following unknown parameters which we needed in our particle filter model:

Adult\_Mean\_Lifetime;

CountHistogramSamples;

Daylight\_Hours\_For\_diapause;

Developmental\_Ratio;

DurationForLowPassPrecip;

Egg\_Generation\_Period;

Initial\_Egg\_Density;

Initial\_Standing\_Water\_mm;

maxInitialMinTemperatureAdultOutOfDiapause;

MeanTimeEggHatchPerDayCMax;

MeanTimeEggHatchPerDayCMin;

minimumAcceptableEffectiveSampleSizeFraction;

minInitialMinTemperatureAdultOutOfDiapause;

minTempForIntoDiapusePreAdult ;

minTempForOutOfDiapausePreAdult;

OutOfHibernationRate;

ProbabliltyPreDay;

randomEggHatchStdDev;

ReportingProbablilty;

Starting \_Adult\_Mosquitoes;

Starting\_Aquatic\_density;

waterHeightForMaxEggLaying.

**“Best values” of parameters**

In the process of calibration, we imported the historical data set and after simulating and running the calibration model, we got the best values of unknown parameters, which is as follows:

***Table1: the Best Value by running the calibration model***

|  |  |
| --- | --- |
| Unknown parameters | Best values |
| Starting\_Aquatic\_Density | 0.346 |
| WaterHeightForMaxEggLaying | 11.499 |
| Initial\_Egg\_Density | 0.637 |
| Starting\_Adult\_Mosquitoes | 100 |
| minTempForOutOfDiapause | 17.834 |
| DuraitonForLowPassPrecip | 25 |
| minTempForIntoDiapause | 16.285 |
| reporting probability | 0.06 |

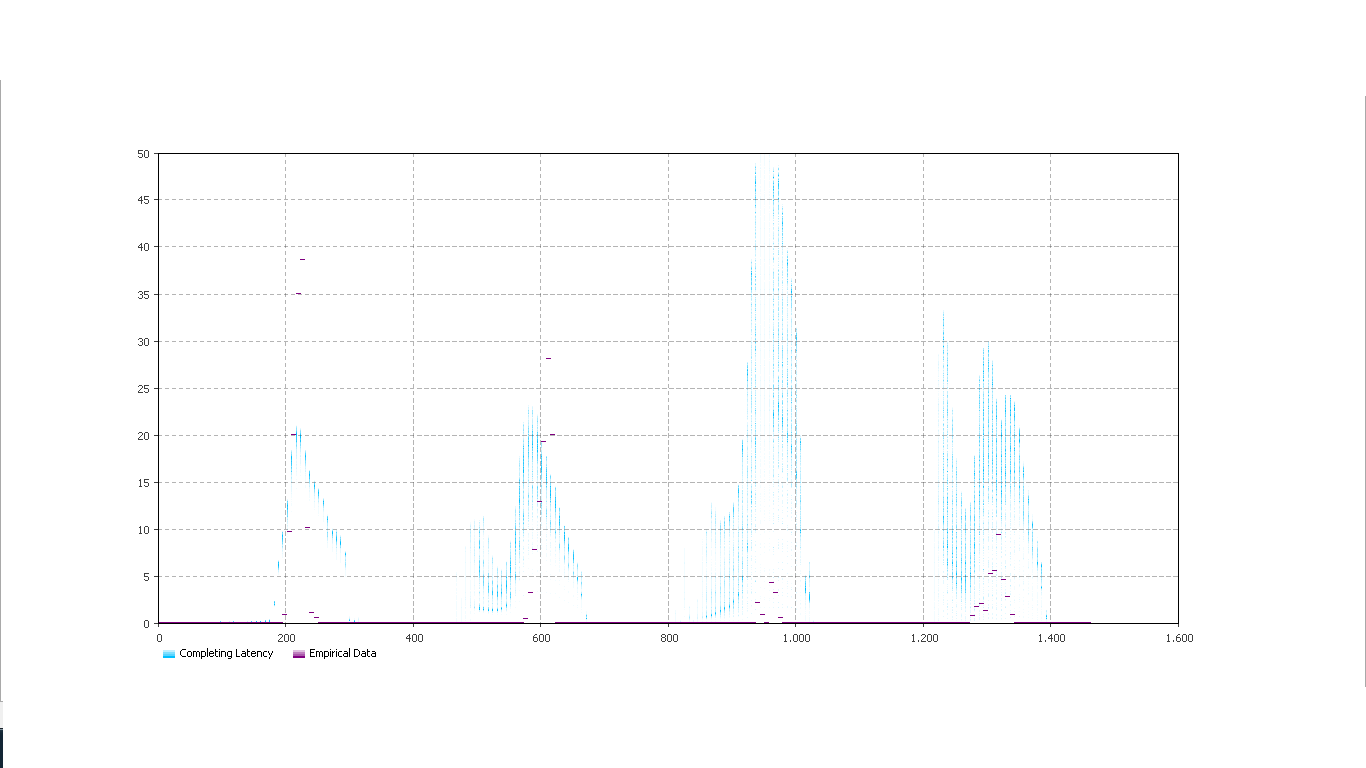
**Modify and adopted “best value”**

Unfortunately, we cannot directly use these parameters in our model. One reason is that calibrations can mislead us to falsify our model by the “best values” of. Another reason is the mosquito data set is reported in trapped counts and the population size is not accurately estimated; therefore, we have to modify (filter) some “best values” based on a realistic and reasonable situation. We adopted the following values:

|  |  |  |
| --- | --- | --- |
| Unknown parameters | Best values | Adjust and adopted Value |
| Starting\_Aquatic\_Density | 0.346 | 0.346 |
| WaterHeightForMaxEggLaying | 11.499 | 11.499 |
| Initial\_Egg\_Density | 0.637 | 0.637 |
| Starting\_Adult\_Mosquitoes | 100 | 50000 |
| minTempForOutOfDiapausePreAdult | 17.834 | 17.834 |
| DuraitonForLowPassPrecip | 25 | 25 |
| minTempForIntoDiapausePreAdult | 16.285 | 12 |
| ReportingProbability | 0.06 | 0.00001 |

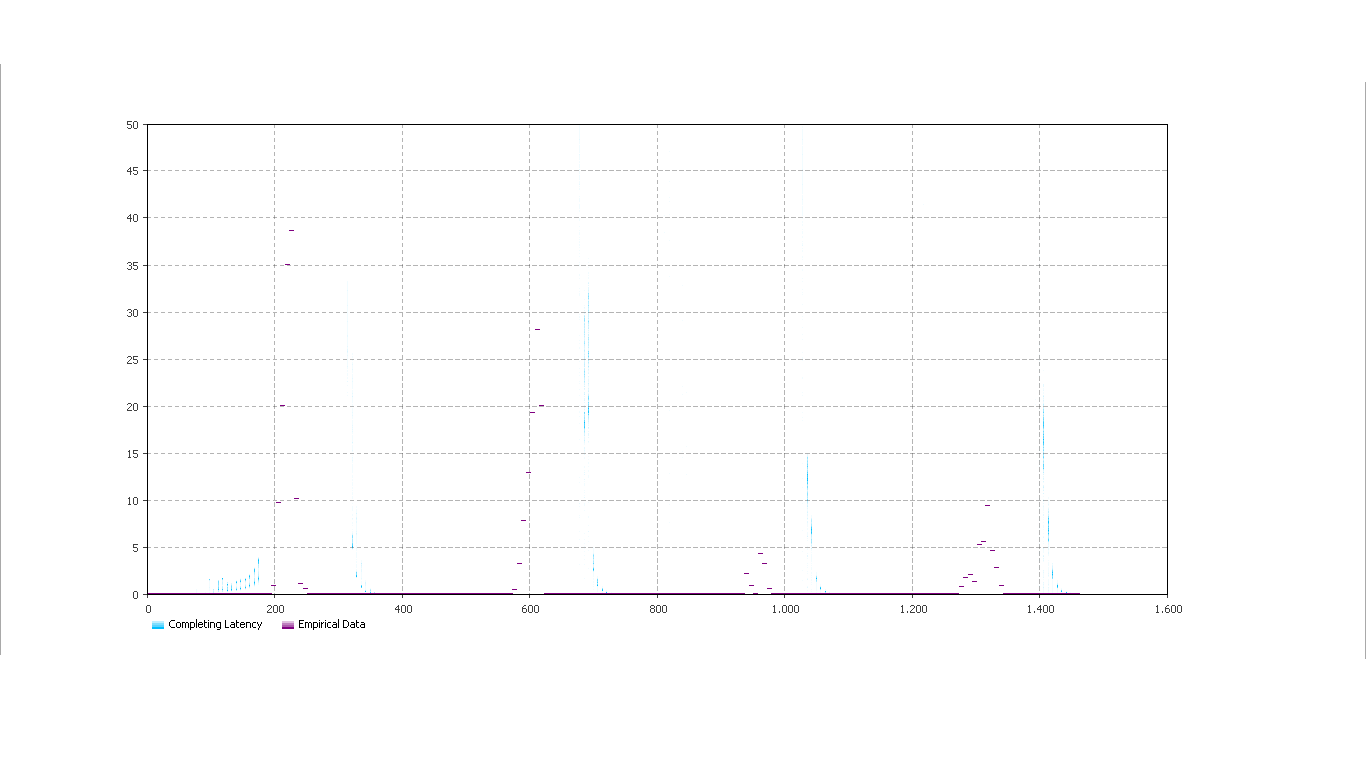
**4. Results**

While running our model under different situations, the outputs showed us the following figures 1-3; in each figure, the vertical axis represents the amount of the mosquito, the horizontal axis represents the time series (days). The order of time from 2010 to 2013 is in the graphs.

**Scenario1**: running the model under the No Particle Filter with No Best Value. 

From the above figure we can see the results of running model under no particle filter with the non-best values have similar data with actual numbers; however some years have a significant  differences (the second year has a smaller actual  mosquito number). There are large accumulated errors between the completing latency (amount of predicted mosquito) and the empirical data in 2012 and 2013. The key reason that led this result to appear is the model was run under a no particle filter situation; another reason is change of weather i.e. unexpected or irregular wind patterns or rainfall patterns. In 2011, the time of the appearance of mosquitoes matched the actual weather situation; from the weather data we can find that late April’s temperature in 2011 reached the temperature which the mosquitoes awaken from hibernation.

**Scenario 2**: running the model under the No Particle Filter with Best Value.



As figure-2 shows us, the results of the model under the no particle filters with the best value have significant differences between the amount of actual mosquitoes and amount of expected mosquitoes. The results of the prediction swerved from the empirical data, the reasons being as follows:

(1). The model was run under no a particle filter situation;

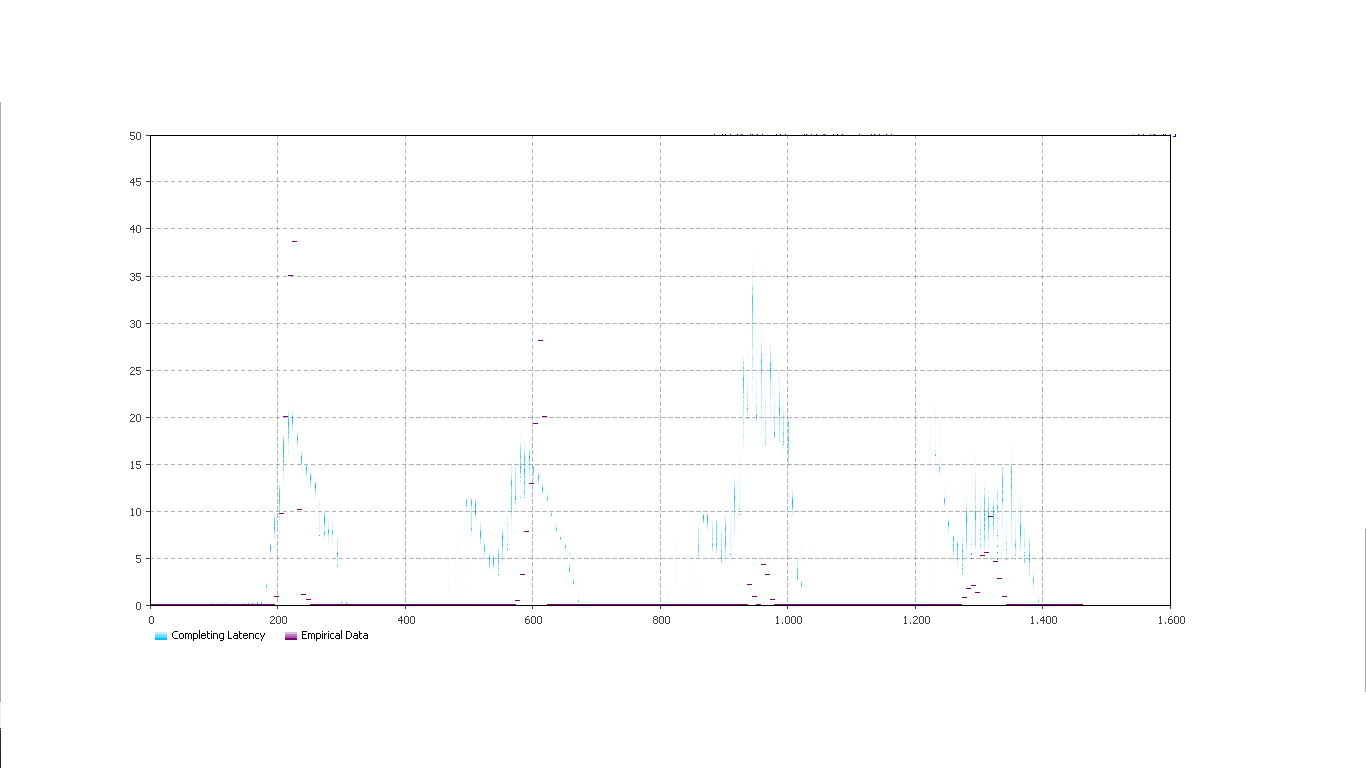
(2). The best values which were calculated by the calibration experiment were not the best

optimal values for our model;

(3). Empirical data may include some errors. For example, the errors were created in the process

of collecting data.

From the figure, we should also note that all the mosquitoes awakened from diapauses late because we added Diapause \_Adult\_early stock and changed the system dynamic model differently in other parts.

**Scenario 3**: running the model under the Particle Filter with Non Best Value.

Under the particle filter with the non-best value, the results of running the model (shown above) have real situations that happen in the actual world and are better at predicting estimated values from historical data. The reasonable explanation is that the particle filter deals with the process of sample data by using samples, update weights and resamples; this process eliminates noisy samples which have lower weight; another explanation is that non best values which were used by our model were modified and adjusted based on the best values and experimental values. After modification, these values were more fit to actual situations.

By comparing the no particle filter with non-best values against the particle filter with non-best value we can find that the accumulated error in the scenario 3 is less than the accumulated error in scenario 1. We can say that the variance between completing latency and empirical data by using particle filter method is less than the variance between completing latency and empirical data by using a non-particle filter method.

Note that in the scenario 4, running the model under the particle filter with best value we cannot obtain a figure; the reason being the actual value and expected value have too much of a difference; therefore, it is difficult to compare and analyze.

**5. Learning**

Through creating the structure of particle filter model, we learnt and realized the flexibility of AnyLogic’s interface. It allowed us to try different ideas on implementing data which affected the mosquito population size. We also noticed that we have not put enough factors into our model; we have only considered the environment in which mosquitoes live in and process of their life cycles in these environments; however, there are factors which have high correlations with change in mosquito population size were not included in our model. For example, predators such as frogs, dragonflies, birds, bats and other factors such as sanitation of environment, vegetation type and growth and mosquito adaptations to certain environments would be such examples. If we were to re-do the model, we would add these factors and at the very least add the predators’ factor into the particle filter model. In this way, our predicted result would have been closer to a realistic situation. But the one downside to this would be the difficulty of obtaining first hand sources for the predator factor. This would require extensive research into a very specific topic.

The calibration model made our project easier because it easily simulates and estimates the “best value” of the unknown parameter which was needed by the model which would be quite difficult for us to if we didn’t have the calibration model. Getting closer realistic results from our model was very hard, such as how to decrease the immediate surge of mosquitoes after a winter stasis in our model.

**6. Further Work**

The next step to develop the model is connecting the relation between the West Nile Virus and our mosquito population model to approach so that we can use an agency based model to simulate the situation of suspected, infected and recovered mosquitoes. We would also make similar models for predators such as species of birds and horses to get a closer look on how the disease develops in certain organisms. This would not only help us understand the disease better but it could also provide imperative insight on how we can reduce the spread and lethality of the virus.

**7. Conclusions**

(1). We have noticed that the benefits of the particle filter model is to allow iteration many times by samples, update weights, and resamples using a computational method. We’ve also noted that the results are only approximated which maybe lead to calculation errors.

(2). We have noticed that our model was not as complete as we would have liked it to be. We were missing a few factors such as predators and vegetation growth when creating the models.

(3). We noticed the some best values given by the calibration model were not realistic and did not make sense. We had to use modeller’s experience to modify the variation.

We hope that our models can be used as a basis for future endeavors regarding the West Nile Virus. Our models have given precisely estimate of current and future mosquito population sizes and it can provide useful insight for other West Nile Virus models.

**References**

Givens, Geof H., and Jennifer A. Hoeting. *Computational Statistics.* Hoboken, NJ: Wiley-

Interscience, 2005. Print.

Osgood, Nathaniel and Liu, Juxin. *Towards closed loop modeling: evaluating the prospects for creating recurrently aggregate simulation models using particle filtering.* Saskatoon, SK: University of Saskatchewan, 2014. Print.

Doucet, Arnaud and Johansen, M. Adam. *A Tutorial on Particle Filtering and Smoothing: Fifteen years later.* Coventry, UK: University of Warwick, 2008. Print.

Ricahrd, Mike and Theoret, Curtis. *Mosquito Population Model*. Saskatoon, SK: University of Saskatchewan, 2014. Print.