Spatio-temporal density modeling of Malaria outbreaks in Africa.

By

Armand BANDIANG MASSOUA (barmand@aims.ac.rw)
African Institute for Mathematical Sciences (AIMS), Rwanda
Supervised by: Dr. Marc Deisenroth
Imperial College London, United Kingdom

January 2019

AN ESSAY PRESENTED TO AIMS RWANDA IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE AWARD OF MASTER OF SCIENCE IN MATHEMATICAL SCIENCES

DECLARATION

This work was carried out at AIMS Rwanda in partial fulfilment of the requirements for a Master of Science Degree.

I hereby declare that except where due acknowledgement is made, this work has never been presented wholly or in part for the award of a degree at AIMS Rwanda or any other University.

Student: Armand Bandiang Massoua

Supervisor: Dr. Marc Deisenroth 

ACKNOWLEDGEMENTS

First of all, I would like to thank the Almighty God for giving me the strength and courage to finish the project.

I would like also to be very grateful to Dr. Marc Deisenroth who accepted my request to be my supervisor. I have learned a lot from this brilliant and wise man, through his advice, comments, and guidance hugely helped me to achieve this work. Thank also to my tutor Alex Rogers, with great assistance contributed to the achievement of this work.

I would like also to thank especially Prof. Blaise TCHAPNDA the academic director at AIMS Rwanda, for the great job he has done for us students to have good education. For us to be taught by top lecturers from all over the world.

Lastly, I would like to thank my fellow colleagues and everyone who contributed to my success.

DEDICATION

I would like to dedicate this work to my family, always there supporting me.

Abstract

Malaria is a vector-borne disease, which is transmitted to humans after having been bitten by an infected Anopheles mosquito. A Risk of malaria outbreaks in a given geographical regions can be modeled using density estimation model. This procedure is demonstrated using a geo-coded inventory of malaria vectors in Sub-Saharan Africa from 1924 to 2016. In order to estimate density, many methods are used including histograms, kernel density estimation, etc. In our case we used a parametric density estimator which is Gaussian mixture model.

We observe that high density of malaria vectors distribution are inuenced by the amount of rainfall, the presence and extent of local water bodies, and the coasts.

Keywords: Malaria, Anopheles mosquito, Africa, Spatio-temporal model, Density estimation, Gaussian mixture model, Blending of Gaussian mixture model.

Contents

Declaration i

Acknowledgements ii

Dedication iii

Abstract iv

1 Introduction 1

1.1 Background and Motivation 1

1.2 Statement of the Problem 2

1.3 Objective 3

1.4 Structure of the Essay 3

2 Method 4

2.1 Machine Learning 4

2.2 Gaussian Mixture Model 5

2.3 How many components? 9

3 Results 11

3.1 Materials 11

3.2 Model Selection 11

3.3 Model Explanation 12

3.4 Blending of Models 15

3.5 Difficulties 18

3.6 Results and Discussion 18

4 Conclusion 19

References 21

1. Introduction

This chapter describes the problem and the approaches that we are going to do through some experiments in this essay, the objective that we set for the project, and how the essay is structured.

1.1 Background and Motivation

Malaria is a vector-borne disease and is one causes most of morbidity (Kakmeni et al., 2018). This endemic disease is caused by the parasites belonging to the Plasmodium group, transmitted to humans after having been bitten by an Anopheles mosquito which is infected. The vectors of malaria are located in the areas in tropical and subtropical and the sub-Saharan part of the African continent, this disease is one of the most difficult to control by the public health (Kakmeni et al., 2018). According to WHO Report 2016 estimation in 2015 over 212 million cases of malaria occurred in the world and $429,000$ people died, where the majority are children from the African continent. The WHO Report 2016 gave in 2016 an estimation 216 million of malaria cases in 91 countries, which shows a slight increase about five million cases over 2015.

Spatio-temporal models arise when data are collected across time as well as geographical location and has at least one spatial and one temporal property. An event in a spatio-temporal dataset describes a spatial and temporal phenomenon that exists at a certain time t and location x. So, the model has as input space and time. An example would be that of the patterns of female breast cancer mortality in the United States between 1990-2010, where the spatial property is the location and geometry of the object United States, states with breast cancer mortality rate information, and the temporal property is the time interval for which the spatial object is valid $1990-2010$ breast cancer mortality years (Columbia University Mailman School of Public Health).

What we are going to do here is to use machine learning approach to build an unsupervised model which will estimate malaria vectors density in Africa. For this problem we use a geo-coded inventory of Anophelines in the Sub-Saharan Africa dataset.

The dataset use for modeling the problem is a geo-coded inventory of malaria vectors in Afro-tropical region south of the Sahara from 1924 to 2016. Its format is csv (comma-separated values) file. The dimension of our dataset is $(12710, 32)$ which is a high dimension dataset, with 12710 observations and 32 variables. The variables are described as the following:

- Country name: describe the country name in which the mosquito type is present;
- City: the city of the country in which this mosquito is present;
- Latitude: give us the latitude coordinate for a given location;
- Longitude: give us the longitude coordinate for a given location;
- the other remaining columns describe the presence or not of different mosquito types.

1.2 Statement of the Problem

The purpose of density modeling is to approximate the true probability density function of a random variable from an observed dataset. Density modeling falls under two principal families of density estimator methods: parametric and non-parametric. In the parametric case we restrict the approximating family to specific (parametrized class of distributions), whilst in the nonparametric case, this is not the case, but we still need to make assumptions.

Many methods have been used for density modeling, including histograms, kernel density estimation (KDE), etc. Histograms and Kernel density estimation are non-parametric methods, they differ from parametric methods in that the model structure is not specified in advance but is instead determined from data. Histograms are of widely use density estimator.

A histogram splits the data into discrete bins, counts the number of points that belong to each bin, then visualized it (Silverman, 2018).

The kernel density estimator is a commonly used for density estimation. It is given by the following function:

$$\hat{p}(x) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{X_i - x}{h}\right), \tag{1.2.1}$$

and $K(x)$ is the *kernel function*, and $h > 0$ is the smoothing bandwidth, h is used to control the amount of smoothing. When h becomes large there is an increasing in the smoothing. The choice of h one can use the cross-validation technique to get an optimal h. The KDE smooths each data point X_i into a small density bumps and then do the summation of all these small bumps together to obtain the final density estimate (Silverman, 2018). Figure 1.1 is its illustration.

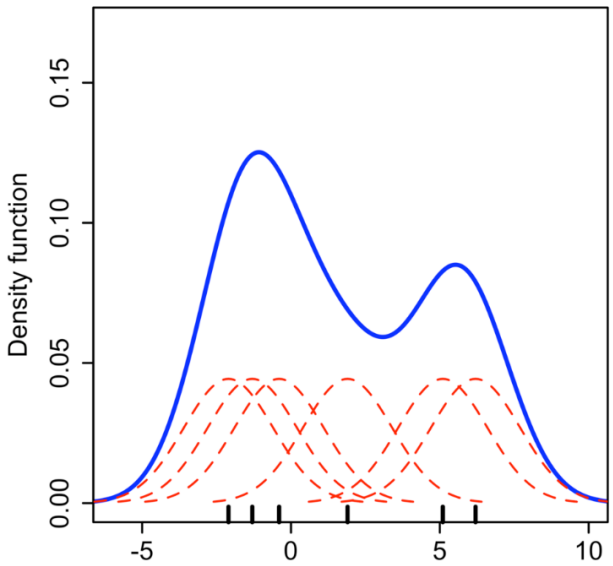


Figure 1.1: KDE for six data points. The blue curve is the kernel density estimate. Source Wikipedia.

There are several types of kernel functions $K(x)$, the three most common kernel functions are

the following:

1. Gaussian define as:
$$K(x) = \frac{1}{\sqrt{2\pi}} \exp\{-\frac{x^2}{2}\}, \tag{1.2.2}$$
2. Uniform define as:
$$K(x) = \frac{1}{2} I(-1 \leq x \leq 1), \tag{1.2.3}$$
3. Epanechnikov define as:
$$K(x) = \frac{3}{4} \cdot \max\{1 - x^2, 0\}. \tag{1.2.4}$$

The kernel density estimator is likely the most used, however it suffers from a slight shortcoming when it is used on long-tailed distribution data. Because the bandwidth is the same for the whole sample, there is a tendency for a spurious noise to appear in the tails of the estimates (Silverman, 2018).

In this project we are going use Gaussian mixture models (GMMs) for density modeling of Malaria vectors in Africa.

1.3 Objective

The objective of this project is to find a spatio-temporal density model of Malaria vectors in Africa over the last 93 years.

1.4 Structure of the Essay

This essay report is divided into four chapters. The chapter one states the introduction. Chapter two talks about the method used to tackle the problem, and defining some useful concepts and tools used in machine learning. Chapter three describes the materials, results that we got, and the interpretation. Chapter four is where we concluded the report and state the possible of future work.

2. Method

Toward our goal of density modeling using unsupervised machine learning techniques, in this section we are going to describe that technique and resources.

2.1 Machine Learning

Machine learning is a science that allows computers to learn from given data to progressively improve their performance without being explicitly programmed.

2.1.1 Definition of concepts. The definitions below have been taken from (Mohri et al., 2012). A dataset is a collection of information that is used for training or testing a model.

Features are the set of variables or columns, represented usually as a vector within a dataset.

Labels are the values or categories or classes assigned to observations. The observations are the lines in dataset.

A training set is a dataset used to train a model.

A validation set is a dataset used to adjust the parameters of a model.

A test set is a dataset used to evaluate the performance of a model. It is separated from the data use for training and validation.

A cost (loss) function is a function used to measure the difference, or loss between a prediction value and a true value.

2.1.2 Types of Machine Learning. Machine learning algorithms fall under different learning classes. Supervised learning, conjointly refered to as learning algorithms, learn from a data set in which the input variables and target variable are both provided and, the target variable is the response that the algorithm should produce. Supervised learning deals with regression and classification problems (Marsland, 2015). Regression predictive modeling is a way of approximating a mapping function (f) from the input variables (X) to a continuous output variable (y). Classification predictive modeling is a way of approximating a mapping function (f) from the input variables (X) to the discrete output variables (y).

Unsupervised learning builds a model that tries to find clusters of similar inputs in the data without being explicitly told to which class the data point belongs. The statistical approach to unsupervised learning is known as density estimation (Marsland, 2015). The modeling technique that we are going to use in this project falls under this type of machine learning.

Reinforcement learning closes the hole between supervised learning, where we train model on true labels in a given response variable, and unsupervised learning, where the model tries to look for correspondence in an unlabeled data to create clusters. The middle ground is where information is provided about whether or not the answer is correct, but not the way of improving it. So, the reinforcement algorithm must try out several strategies and choose the best one (Marsland,

2015).

2.2 Gaussian Mixture Model

A Gaussian mixture model (GMM) falls under the family of parametric model which is a probability density function which is represented as a sum of weighted Gaussian component densities. The parameters of Gaussian mixture model are estimated from a given training dataset by using an Expectation-Maximization (EM) algorithm or Maximum A Posteriori (MAP) estimation.

2.2.1 Gaussian distribution. The Gaussian also known as the normal distribution, for the case of a single variable x can be written as

$$\mathcal{N}(x|\mu,\sigma^2) = \frac{1}{(2\pi\sigma^2)^{\frac{1}{2}}} \exp\left\{-\frac{1}{2\sigma^2}(x-\mu)^2\right\} \tag{2.2.1}$$

where μ is the mean and σ^2 is the variance. For a $D-$dimensional vector x, the multivariate normal distribution is of the form

$$\mathcal{N}(x|\mu,\Sigma) = \frac{1}{(2\pi)^{\frac{D}{2}}|\Sigma|^{\frac{1}{2}}} \exp\left\{-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right\} \tag{2.2.2}$$

where μ is a $D-$dimensional mean vector and Σ is a $D \times D$ covariance matrix (Bishop, 2006).

2.2.2 Gaussian mixture model. A mixture model is a density model which is a convex combination of a finite number of M Gaussian distributions of the form:

$$p(x|\theta) = \sum_{k=1}^{M} \pi_k \mathcal{N}(x|\mu_k,\Sigma_k) \tag{2.2.3}$$

$$0 \leq \pi_k \leq 1, \quad \sum_{k=1}^{M} \pi_k = 1 \tag{2.2.4}$$

where θ is defined by $\theta := \{\mu_k, \Sigma_k, \pi_k : k = 1, ..., M\}$ and π_k are the mixture weights (Deisenroth et al., 2018).

2.2.3 Learning Gaussian Mixture Model Parameters. Here we are going to see how to learn the parameters of the GMM. There are several methods for parameter learning but we are going to focus only on the maximum likelihood (ML). The following are taken from (Deisenroth et al., 2018).

2.2.4 Maximum Likelihood. Let's say we have a given dataset $\mathcal{X} = \{x_1, ..., x_N\}$ in which x_n, $n = \{1, ..., N\}$ are drawn independently and identically distributed from a distribution $p(x)$ not known. The aim here is to find the best parameters through the Gaussian mixture model. Let's summarize the parameters in $\theta := \{\mu_k, \Sigma_k, \pi_k : k = 1, ..., M\}$. By taking advantage of our assumption for independent and identically distributed, the likelihood is the form

$$p(\mathcal{X}|\theta) = \prod_{n=1}^{N} p(x_n|\theta) \tag{2.2.5}$$

with

$$p\big(x_n|\theta\big) = \sum_{k=1}^{M} \pi_k \mathcal{N}\big(x|\mu_k, \Sigma_k\big). \tag{2.2.6}$$

By taking logarithm of the likelihood we get log-likelihood as follow

$$\log p(\mathcal{X}|\theta) = \sum_{n=1}^{N} \log p(x_n|\theta) = \underbrace{\sum_{n=1}^{N} \log \sum_{k=1}^{M} \pi_k \mathcal{N}\big(x|\mu_k, \Sigma_k\big)}_{=:\mathcal{L}}. \tag{2.2.7}$$

Our purpose is to find the best parameters that maximize the log-likelihood $\mathcal{L}$. For that we are going to compute the derivative of the log-likelihood regarding to the parameters θ, after setting to 0 and solving for θ. For each parameter we get:

$$\frac{\partial \mathcal{L}}{\partial \mu_k} = 0 \iff \sum_{n=1}^{N} \frac{\partial \log p(x_n|\theta)}{\partial \mu_k} = 0 \tag{2.2.8}$$

$$\frac{\partial \mathcal{L}}{\partial \Sigma_k} = 0 \iff \sum_{n=1}^{N} \frac{\partial \log p(x_n|\theta)}{\partial \Sigma_k} = 0 \tag{2.2.9}$$

$$\frac{\partial \mathcal{L}}{\partial \pi_k} = 0 \iff \sum_{n=1}^{N} \frac{\partial \log p(x_n|\theta)}{\partial \pi_k} = 0 \tag{2.2.10}$$

For the case of μ_k we will get by apply the chain rule

$$\sum_{n=1}^{N} \frac{\partial \log p(x_n|\theta)}{\partial \mu_k} = \sum_{n=1}^{N} \frac{1}{p(x_n|\theta)} \frac{\partial p(x_n|\theta)}{\partial \mu_k} \tag{2.2.11}$$

with

$$\begin{aligned} \frac{\partial p(x_n|\theta)}{\partial \mu_k} &= \sum_{j=1}^{N} \pi_j \frac{\partial \mathcal{N}\big(x_n|\mu_j, \Sigma_j\big)}{\partial \mu_k} \\ &= \pi_k \frac{\partial \mathcal{N}\big(x_n|\mu_k, \Sigma_k\big)}{\partial \mu_k} \\ &= \pi_k \big(x_n - \mu_k\big)^T \Sigma_k^{-1} \mathcal{N}\big(x_n|\mu_k, \Sigma_k\big). \end{aligned} \tag{2.2.12}$$

Putting everything together:

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial \mu_k} &= \sum_{n=1}^{N} \frac{\partial \log p(x_n|\theta)}{\partial \mu_k} \\
&= \sum_{n=1}^{N} \frac{1}{p(x_n|\theta)} \frac{\partial p(x_n|\theta)}{\partial \mu_k} \\
&= \sum_{n=1}^{N} (x_n - \mu_k)^T \Sigma_k^{-1} \underbrace{\frac{\pi_k \mathcal{N}\big(x_n|\mu_k, \Sigma_k\big)}{\sum_j \pi_j \mathcal{N}\big(x_n|\mu_j, \Sigma_j\big)}}_{:=r_{nk}} \\
&= \sum_{n=1}^{N} r_{nk} \big(x_n - \mu_k\big)^T \Sigma_k^{-1} = 0 \\
&\iff \sum_{n=1}^{N} r_{nk} x_n = \sum_{n=1}^{N} r_{nk} \mu_k \\
&\iff \mu_k^* = \frac{\sum_{n=1}^{N} r_{nk} x_i}{\underbrace{\sum_{n=1}^{N} r_{nk}}_{N_k}} = \frac{1}{N_k} \sum_{n=1}^{N} r_{nk} x_n
\end{aligned} \tag{2.2.13}$$

For the case of Σ we will typically operate as previously.

$$\frac{\partial \mathcal{L}}{\partial \Sigma_k} = \sum_{n=1}^{N} \frac{\partial \log p(x_n|\theta)}{\partial \Sigma_k} = \sum_{n=1}^{N} \frac{1}{p(x_n|\theta)} \frac{\partial p(x_n|\theta)}{\partial \Sigma_k} \tag{2.2.14}$$

with

$$\frac{\partial p(x_n|\theta)}{\partial \Sigma_k} = \pi_k \mathcal{N}\big(x_n|\mu_k, \Sigma_k\big) \cdot \big[-\frac{1}{2} \big(\Sigma_k^{-1} - \Sigma_k^{-1} \big(x_n - \mu_k\big)\big(x_n - \mu_k\big)^T\big) \Sigma_k^{-1} \big]. \tag{2.2.15}$$

Now let us put everything together, taking the partial derivative of the log-likelihood will give us

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial \Sigma_k} &= \sum_{n=1}^{N} \frac{\partial \log p(x_n|\theta)}{\partial \Sigma_k} \\
&= \sum_{n=1}^{N} \frac{1}{p(x_n|\theta)} \frac{\partial p(x_n|\theta)}{\partial \Sigma_k} \\
&= \sum_{n=1}^{N} \frac{\pi_k \mathcal{N}\big(x_n|\mu_k, \Sigma_k\big)}{\sum_{j=1}^{N} \pi_j \mathcal{N}\big(x_n|\mu_j, \Sigma_j\big)} \cdot \big[-\frac{1}{2} \big(\Sigma_k^{-1} - \Sigma_k^{-1} \big(x_n - \mu_k\big)\big(x_n - \mu_k\big)^T \Sigma_k^{-1} \big] \\
&= -\frac{1}{2} \sum_{n=1}^{N} r_{nk} \big(\Sigma_k^{-1} - \Sigma_k^{-1} \big(x_n - \mu_k\big)\big(x_n - \mu_k\big)^T \Sigma_k^{-1}\big).
\end{aligned} \tag{2.2.16}$$

By setting the derivative to 0 and solving for Σ_k, we get

$$\Sigma_k^* = \frac{1}{N_k}\sum_{n=1}^{N} r_{nk}\big(x_n - \mu_k\big)\big(x_n - \mu_k\big)^T \tag{2.2.17}$$

Let's now look at the last case for π_k. let's look for the partial derivative of log-likelihood respect to π_k. We are going consider the constraint $\sum_{k=1}^{K}\pi_k = 1$ by using the Lagrange multiplier as follow

$$\begin{aligned} \mathrm{L} &= \mathcal{L} + \lambda\big(\sum_{k=1}^{M}\pi_k - 1\big) \\ &= \sum_{n=1}^{N}\log\sum_{k=1}^{M}\pi_k\mathcal{N}\big(x_n|\mu_k,\Sigma_k\big) + \lambda\big(\sum_{k=1}^{M}\pi_k - 1\big). \end{aligned} \tag{2.2.18}$$

$$\begin{aligned} \frac{\partial \mathrm{L}}{\partial \pi_k} &= \sum_{n=1}^{N}\frac{\partial \log\sum_{k=1}^{M}\pi_k\mathcal{N}\big(x_n|\mu_k,\Sigma_k\big)}{\partial\pi_k} + \frac{\partial\lambda\big(\sum_{k=1}^{M}\pi_k - 1\big)}{\partial\pi_k} \\ &= \sum_{n=1}^{N}\frac{\mathcal{N}\big(x_n|\mu_k,\Sigma_k\big)}{\sum_{k=1}^{M}\pi_k\mathcal{N}\big(x_n|\mu_k,\Sigma_k\big)} + \lambda \\ &= \frac{1}{\pi_k}\underbrace{\sum_{n=1}^{N}\frac{\pi_k\mathcal{N}\big(x_n|\mu_k,\Sigma_k\big)}{\sum_{k=1}^{M}\pi_k\mathcal{N}\big(x_n|\mu_k,\Sigma_k\big)}}_{:=N_k} + \lambda \\ &= \frac{N_k}{\pi_k} + \lambda, \end{aligned} \tag{2.2.19}$$

and also by taking the partial derivative with respect to the Lagrange multiplier λ as

$$\frac{\partial \mathrm{L}}{\partial\lambda} = \sum_{k=1}^{M}\pi_k - 1. \tag{2.2.20}$$

Equating both partial derivatives to 0 yields to the system of equations as follow

$$\sum_{k=1}^{M}\pi_k = 1, \tag{2.2.21}$$

$$\pi_k = -\frac{N_k}{\lambda}. \tag{2.2.22}$$

Using 2.2.22 in 2.2.21 and by solving for π_k, we get

$$\sum_{k=1}^{M}\pi_k = 1 \iff -\sum_{k=1}^{M}\frac{N_k}{\lambda} = 1 \iff -\frac{N}{\lambda} = 1 \iff \lambda = -N. \tag{2.2.23}$$

By substituting $-N$ for λ in 2.2.22 we get

$$\pi_k^* = -\frac{N_k}{-N} = \frac{N_k}{N}. \tag{2.2.24}$$

2.2.5 Expectation-Maximization Algorithm. The Expectation-Maximization (EM) algorithm was proposed by Dempster et al. (1977) and it is a very nice and powerful way to find solutions for maximum likelihood for models with latent variables. Expectation Maximization algorithm is a general iterative algorithm for learning parameters. The steps of the EM algorithm that estimate the Gaussian mixture model parameters are as following :

1. Initialize μ_k, Σ_k, π_k
2. E-step: Evaluate responsibilities r_{nk} for each data point x_n using current parameters μ_k, Σ_k, π_k:
$$r_{nk} = \frac{\pi_k \mathcal{N}\big(x_n|\mu_k, \Sigma_k\big)}{\sum_{j=1}^{N} \pi_j \mathcal{N}\big(x_i|\mu_j, \Sigma_j\big)}. \tag{2.2.25}$$
3. M-step: Re-estimate parameters μ_k, Σ_k, π_k using the current responsibilities r_{nk} (from E-step):
$$\begin{aligned} \mu_k^* &= \frac{1}{N_k} \sum_{i=1}^{N} r_{ik} x_i, \\ \Sigma_k^* &= \frac{1}{N_k} \sum_{n=1}^{N} r_{nk} \big(x_n - \mu_k\big) \big(x_n - \mu_k\big)^T, \\ \pi_k^* &= \frac{N_k}{N}. \end{aligned} \tag{2.2.26}$$

2.3 How many components?

In this section we are going to describe the techniques used for getting the optimal number of components for the GMM. They are several techniques like Cross-validation, Akaike information criterion (AIC), Bayesian information criterion (BIC), etc.

2.3.1 Cross-validation. Cross-validation is a technique for re-sampling used for evaluating machine learning models on a sample dataset. They are several type of cross-validation technique:

- Leave-one-out cross-validation (LOOCV) is a cross-validation technique that uses one data point as validation data, and the $m-1$ data points remaining are used for training data. The process is repeated until each datapoint in the dataset is used as validation data.

  The LOOCV algorithm is the following:

  (a) Split the data set of size m into:
  - i. Training data set of size $m-1$
  - ii. Testing data set of size 1

  (b) Fit the model using the training data

(c) Validate the model using the testing data, and compute the probability of belonging in any category of the testing data or compute the related mean square error(MSE) where the formula is given as follow:

$$CV_m = \frac{1}{m}\sum_{i=1}^{m}\left(y_i - \hat{y}_i\right)^2, \tag{2.3.1}$$

where $\hat{y}_i$ is y_i predicted based on the model trained with ith case left out.

(d) Repeat the algorithm m times.

- K-fold cross-validation is the same as the LOOCV but here the dataset is divided in K equal parts, the $K-1$ form the training dataset and the last part as validation dataset. That process is repeated until K different parts are the validation datasets.

  The K-fold cross-validation algorithm is the following:

  (a) Split the data set into equal parts of size K

  (b) For each part

  i. Form a training data set of size $K-1$
  ii. Form a testing data set of size 1
  iii. Fit the model using the training data
  iv. Evaluate(compute the error) the model using the testing data by computing the probability of belonging in any category of the testing data or compute the related mean square error(MSE) where the formula is given as follows:

$$CV_m = \frac{1}{m}\sum_{i=1}^{m}\left(y_i - \hat{y}_i\right)^2, \tag{2.3.2}$$

  (c) Repeat the algorithm K times.

  (d) Take the average of the errors

In techniques described above, the cross-validation error is camputed: For each validation step an error is calculated and at the end of all possible validation steps we take the mean of all validation errors, and that is the cross-validation error.

2.3.2 Akaike Information Criterion (AIC). AIC is a technique for selecting one model among others. The best model is the one with a minimum AIC among all the other models. The AIC for a model is estimated as follow (Akaike, 1974):

$$AIC = 2k - 2\log L \tag{2.3.3}$$

where k is the number of estimated parameters and L the value of the likelihood.

2.3.3 Bayesian Information Criterion (BIC). BIC is a tool for selecting a model among others. The model selected is one which minimizes the BIC. The model BIC is estimated as follows (Schwarz et al., 1978):

$$BIC = \log(n)k - 2\log L \tag{2.3.4}$$

where k is the number of estimated parameters, n is the number of data points and L the value of the likelihood.

3. Results

This chapter describes all process have done during experiments from beginning until the end and shows what we have gotten from that process following by interpretation. This chapter describes all the processes performed during experimentation and presents our interpretation of the results.

3.1 Materials

In this section we are going to describe the resources used in this project.

3.1.1 Dataset. The dataset used in this project for the experiment was taken from Snow (2017). This dataset is a geo-coded inventory of anopheles vectors of malaria in sub-Saharan in Africa. The dimension of the dataset is $(12899, 32)$. A data point represents anopheles species in a given location, the location is characterized by the geolocation coordinates latitude and longitude.

3.1.2 Tools Used for Experiments. The implementation of the model used is carried out on a computer with the following characteristics:

- Operating system: Windows 10 Pro 64-bit
- Processor: Intel(R) Core(TM) $i5$-$7200U$ CPU $2.50GHz$ $(4CPUs)$, $2.7GHz$
- Memory: $8GB$ RAM

An online platform like Google Colab was also used to contribute to the implementation process. It is a free cloud service provided by Google.

3.2 Model Selection

This section will cover the methods used, and the results obtained through the experiments.

3.2.1 Modeling Technique. As we mentioned earlier in this project, the modeling technique used here is Gaussian mixture model. The model was implemented using python programming language. Since a Gaussian mixture model is a combination of set of finite number of M components which are the Gaussian distributions. In order to select the best model with the optimal number of components M, we did a cross-validation. The cross-validation technique was a $10-$folds cross-validation. It means we divided our dataset into 10 equal parts, and we trained the model on the 9 parts and the one kept out was used for validation. The process was repeated until the 10 parts being validation set. In each validation step we computed the error which is based on computing the negative $\log$likelihood, and at the end we computed the mean of errors. We select the one with lowest mean error.

3.3 Model Explanation

We treat the years independently, which means we take the temporal component out. After filtering our data by year, we ended up with 93 different datasets. So, we decided to now build a model for each different year. In this section we are going to used two models for two different years to explain our model. For illustration let's take the models for the years 2005 and 2006.

3.3.1 Data Exploration. In this we tried to use visual exploration to understand what is in the data and the characteristics of the data. The characteristics that we checked include length of the data, completeness of the data, correctness of the data, possible relationships among data variables.

3.3.2 Data Cleaning. After the data exploration process we performed data cleaning. This allowed us to correct inconsistent data format for the latitude and longitude columns of our dataset, as well as handle missing values in latitude and longitude columns. This process allowed us to have consistent data for our model.

3.3.3 Model Fitting. After cleaning our data and making it ready for further analysis. We fitted our model with that clean data.

For the figure 3.1, it give us a visualization the different cross-validation errors. So, we choose the model with lowest cross-validation error. In this case the best model with the optimal number component is the model with 4 components.

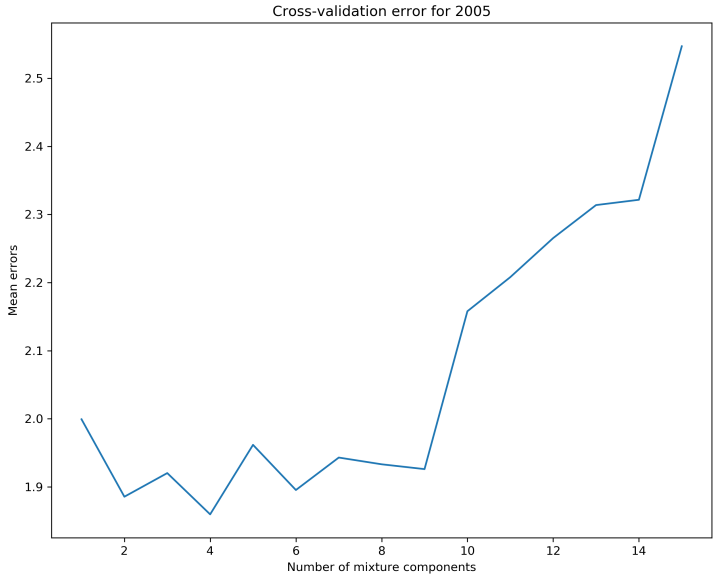


Figure 3.1: Cross-validation error for year 2005

The figure 3.2 give us the visualization of our model for year 2005. One can see that we plotted the model in map of Africa. The ellipses represent the different components, which are four for

this model. The cross in the center of each ellipse represents the mean parameter of the GMM, and the co-variance matrix parameter describe the width of an ellipse. The model shows us how malaria vectors distribution across Africa for this given year. The density covers some countries in West Africa and central Africa near the Gulf of Guinea, and the model captures a high density of vectors in Burkina Faso. A second pattern is located in slightly in some of East African countries, with a high concentration around Lake Victoria.

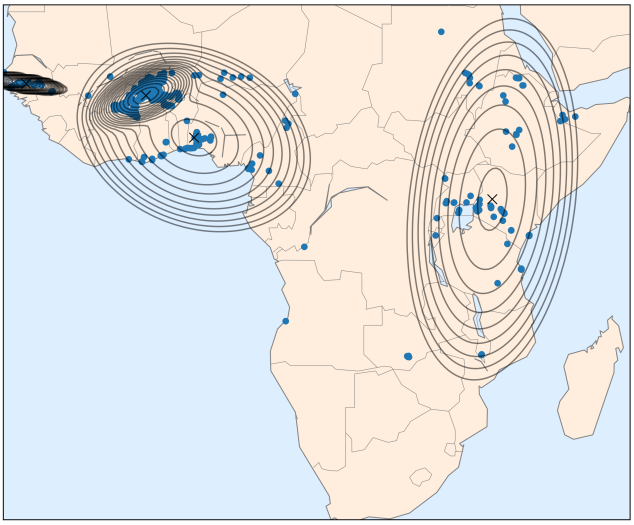


Figure 3.2: GMM for year 2005

For the year 2006 the figure 3.3 give us a visualization of the different cross-validation errors. So, we choose the model with lowest cross-validation error. In this case the best model with the optimal number component is the model with 3 components.

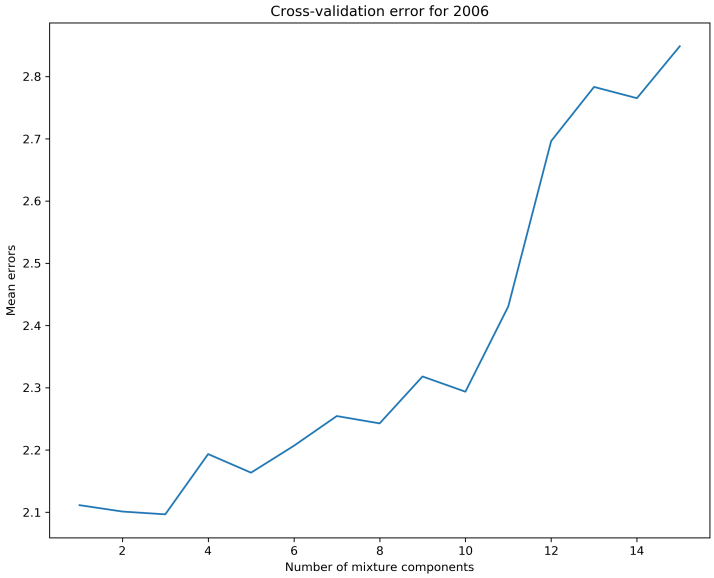


Figure 3.3: Cross-validation error for year 2006

The figure 3.4 give us the visualization of our model for year 2006. The three ellipses represent the mixing components of the model. It shows how malaria vector is distributed along with the density that it covers. It goes from Madagascar to East Africa up to countries in West Africa near the Gulf of Guinea by passing through central Africa.

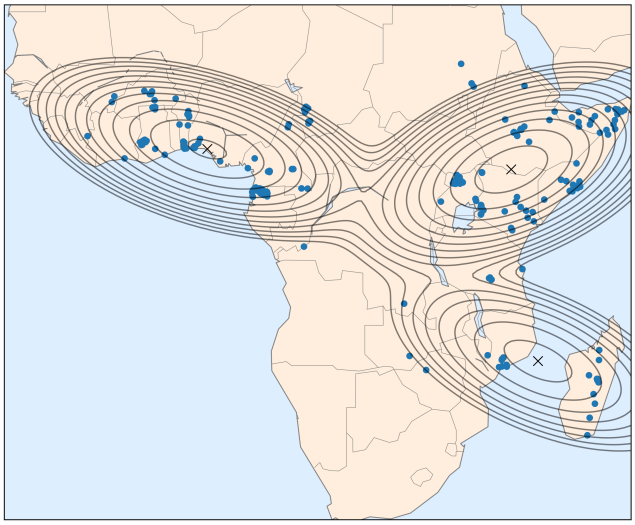


Figure 3.4: GMM for year 2006

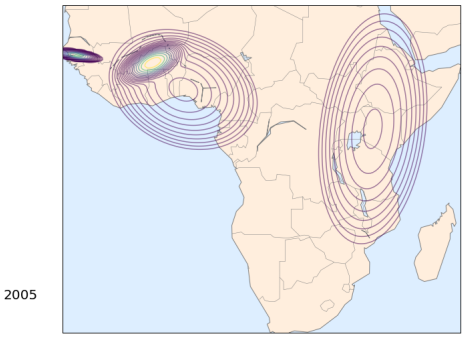
3.4 Blending of Models

The temporal aspect of our dataset led us to do a blending of models. The blending that we have done formally is defined as follows. Let's consider two Gaussian mixture models for different years GMM1 and GMM2 and let GMM be the linear combination of GMM1 and GMM2.

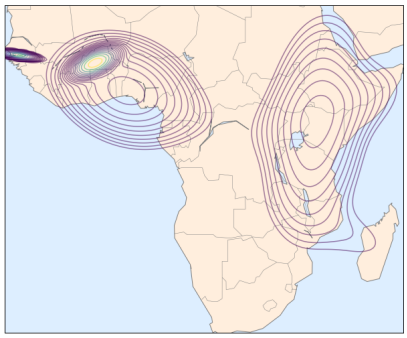
$$\begin{aligned} GMM &= (1-\lambda)GMM1 + \lambda GMM2 \\ &= (1-\lambda)\sum_{k=1}^{M}\pi_k\mathcal{N}\big(x|\mu_k,\Sigma_k\big) + \lambda\sum_{k=1}^{M}\pi_k\mathcal{N}\big(x|\mu_k,\Sigma_k\big) \\ &= \sum_{k=1}^{K}(1-\lambda)\pi_k\mathcal{N}\big(x|\mu_k,\Sigma_k\big) + \sum_{k=1}^{M}\lambda\pi_k\mathcal{N}\big(x|\mu_k,\Sigma_k\big), \end{aligned} \tag{3.4.1}$$

and $\lambda \in \{0, \frac{1}{12}, \frac{2}{12}, \frac{3}{12}, \frac{4}{12}, \frac{5}{12}, \frac{6}{12}, \frac{7}{12}, \frac{8}{12}, \frac{9}{12}, \frac{10}{12}, \frac{11}{12}, 1\}$.

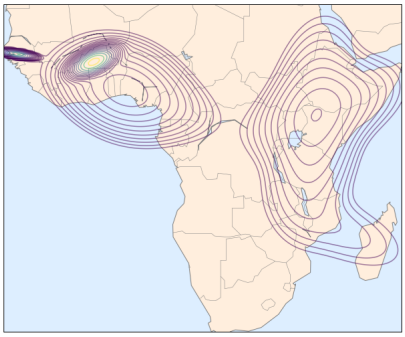
In order to smoothly passe from one year to another we have taken the months of the year and divided each month by 12. By proceeding like that if one wants to visualize the GMM when $\lambda = 0$ only the GMM1 will show up, when $\lambda \neq 0$ the GMM2 starts showing up also until if $\lambda = 1$ when the GMM1 disappears completely and only the GMM2 is visible. This technique allow us an smooth transition between one year ton another. The following figures illustrate the blending of the model from year 2005 to 2006.



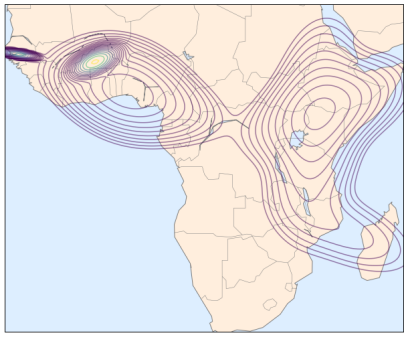
(a) Lambda : 0



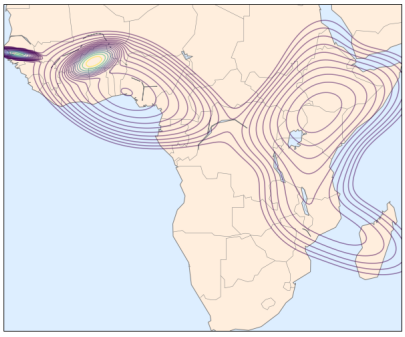
(b) Lambda : 0.08



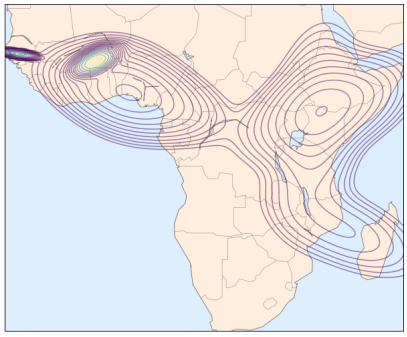
(c) Lambda : 0.16



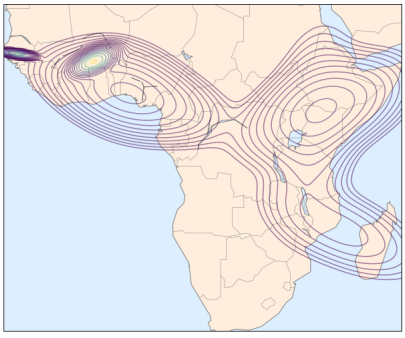
(d) Lambda : 0.25



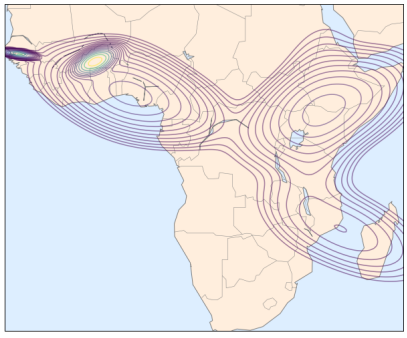
(e) Lambda : 0.33



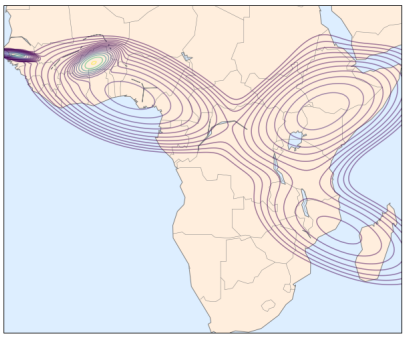
(f) Lambda : 0.42



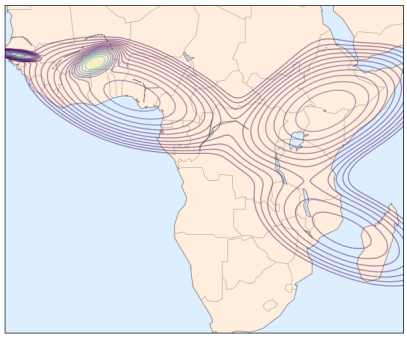
(g) Lambda : 0.5



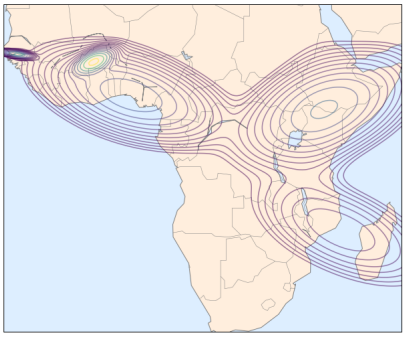
(h) Lambda : 0.58



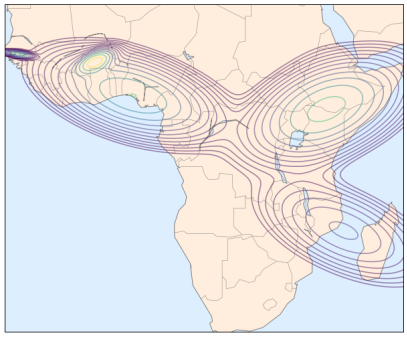
(a) Lambda : 0.66



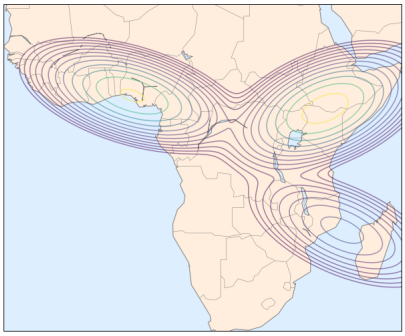
(b) Lambda : 0.75



(c) Lambda : 0.83



(d) Lambda : 0.92



(e) Lambda : 1

Those figures show us how to pass smoothly from one year to another using our blending technique.

3.5 Difficulties

The difficulties that we had, first is about our dataset. The dataset we use in this project is about malaria vectors in Africa, this dataset was really complex du the sparsity of the data points. This sparsity is due to some islands like Cape Verde, Sao Tome and Principe, Mayotte, and Comoros. So when we build , it used to give us abnormal behaviour. The issue persist even after using model selection technique. So we decided to remove those data points corresponding to the islands that we mentioned above. So, we focus on the continent plus Madagascar. The second difficulty is about English, coming from French background it was a bit challenging for me to do the writing up of my report in English. By making daily effort and with the great support of my supervisor and tutor, thank God I made it.

3.6 Results and Discussion

The overall result that our model produces is that we observed some patterns. These patterns will help governments, health care organizations, and NGOs in their actions for fighting against malaria for its eradication in the continent. The patterns will help those different entities mostly to understand how malaria vectors are distributed across Africa, to see where there is a possibility of outbreaks, and helping them in their plan and actions to undertake.

The model gave us a great insight that the high density of malaria vectors distribution are inuenced by the amount of rainfall at a given area, the presence and extent of local water bodies, and the coasts.

The figure 3.7 below give us a visual understanding of what we mentioned just above.

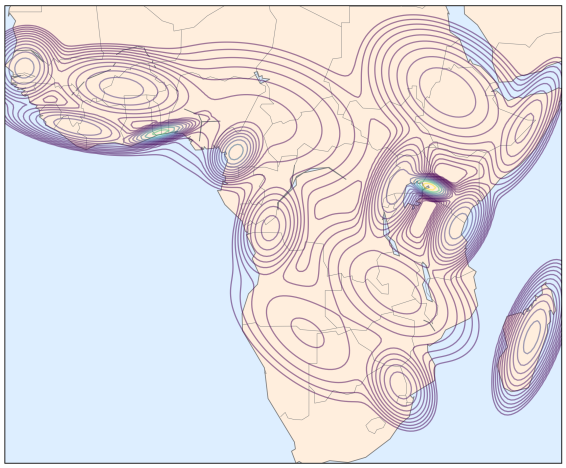


Figure 3.7: Gaussian mixture model for the overall malaria vectors in Africa.

4. Conclusion

In this project we have used unsupervised machine learning techniques to build a model that estimates density based on geo-coded inventory of malaria vectors in African data. Data exploration, data cleaning, and model selection technique are important steps in order to build our model. We have used a parametric probability density to build our model.

After implementing our model, we managed to get, through several experiments patterns that can be very useful for health care professionals, researchers, and organizations in their fight against malaria.

In this work we managed to have a spatial version our model. The spatio-temporal aspect can be seen in the blending technique, however we did not reach the point where we explicitly modeled time (temporal).

In the future work, we will take into account the time aspect as an input in order to build upon our existing model and also explore the possibility of predicting mosquitoes population growth trends.

References

Hirotugu Akaike. A new look at the statistical model identification. *IEEE transactions on automatic control*, 19(6):716–723, 1974.

Victor A Alegana, Simon P Kigozi, Joaniter Nankabirwa, Emmanuel Arinaitwe, Ruth Kigozi, Henry Mawejje, Maxwell Kilama, Nick W Ruktanonchai, Corrine W Ruktanonchai, Chris Drakeley, et al. Spatio-temporal analysis of malaria vector density from baseline through intervention in a high transmission setting. *Parasites & vectors*, 9(1):637, 2016.

C. M. Bishop. *Pattern recognition and machine learning*. Springer, 2006.

Columbia University Mailman School of Public Health. Spatio-temporal example. https://www.mailman.columbia.edu/research/population-health-methods/spatiotemporal-analysis, Accessed December 2018.

M. P. Deisenroth, A. A. Faisal, and C. S. Ong. *Mathematics for Machine Learning*. Cambridge University Press, 2018.

Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38, 1977.

Emmanuel Hakizimana, Corine Karema, Dunia Munyakanage, John Githure, Jean Baptiste Mazarati, Jon Eric Tongren, Willem Takken, Agnes Binagwaho, and Constantianus JM Koenraadt. Spatio-temporal distribution of mosquitoes and risk of malaria infection in rwanda. *Acta tropica*, 182:149–157, 2018.

M. Ibanez, B.and Ugarte and A Militino. Spatio-temporal modeling in disease mapping. *Technical report*, 2008.

Francois M Moukam Kakmeni, Ritter YA Guimapi, Frank T Ndjomatchoua, Sansoa A Pedro, James Mutunga, and Henri EZ Tonnang. Spatial panorama of malaria prevalence in africa under climate change and interventions scenarios. *International journal of health geographics*, 17(1):2, 2018.

Stephen Marsland. *Machine learning: an algorithmic perspective*. Chapman and Hall/CRC, 2015.

Mehryar Mohri, Ameet Talwalkar, and Afshin Rostamizadeh. *Foundations of machine learning (adaptive computation and machine learning series)*. Mit Press Cambridge, MA, 2012.

Gideon Schwarz et al. Estimating the dimension of a model. *The annals of statistics*, 6(2): 461–464, 1978.

Bernard W Silverman. *Density estimation for statistics and data analysis*. Routledge, 2018.

Robert W. Snow. A geo-coded inventory of anophelines in the afrotropical region south of the sahara: 1898-2016. Webots, https://doi.org/10.7910/DVN/NQ6CUN, 2017.

WHO Report 2016. Who (world health organization) global malaria programme: World malaria report 2016. Webots, http://www.who.intmalaria/media/world-malaria-report-2016/en/, Accessed December 2018.

Wikipedia. Kernel density estimation. https://en.wikipedia.org/wiki/Kernel_density_estimation, Accessed January 2019.