West Nile Virus Prediction

EE660 Final Project

Xuechao Wu, xuechaow@usc.edu, 12/07/2015

1. Table of Contents

2. Project Homepage

https://github.com/xuechaow/EE660_Final_Project

3. Abstract

West Nile Virus has been outbreak since 2006, according to Times.com. An prediction of next outbreak is demanded by the CRC. This paper uses data collected in Chicago Area to predict the next appearance of WNV in Chicago. This paper focus on 3 important features related to virus, which are Mosquito Species, Location and Time. Similar to the process of decision tree, establish three individual models for their conditions and the output of WNV presence. Each of these three models uses different features. With the regression result, surprisingly, the prediction turns out to be better matched with random tree method. Use Random Tree Approach to compare with mixed Model. With an Etest = 0.69, the result is partly useful to predict the presence. Use K-means clustering to cluster geographically close traps and calculate the probability of presence.

4. Problem Statement and Goals

Given weather, location, testing, and spraying data, this task requires to predict when and where different species of mosquitos will test positive for West Nile virus. Specifically, test input X contains Date and Location, try to solve to presence probability. The task has 2 stages, first use cross validation to build an accurate (low E_in) Model.

(1) Test Result is difficult to acquire.
Because this is a competition task, the final test result is unclear. There is only test result as probabilities. Pmtk only provides randomTree function with output without Probabilities.

(2) Significant amounts of preprocessing required.
The training data has only less than 13 features, but those features has high independency. Date and Weather has strong periodic patterns. Location Data has strong cluster features. And Mosquito need to be detected and translated into math features. The pre-processing of raw data is time consuming and easily corrupt into wrong data.

5. Literature Review

Several people on Leaderboard used Motion and Deep Learning Method. There are also Neural Network and Random Tree approaches with >0.73 accuracy.

6. Prior and Related Work

Prior and Related Work – None. Just a competition expired on Kaggle.

7. Project Formulation and Setup

The most tricky part and dangerous part is the features' patterns. Dimensions are complicated but there are several related features such as [Address trap # (Longitude, Latitude)], and [Date (or post-processed season #) Weather]. If treat those features independently, chances are model will over-fit with non-singular feature dimension.

To avoid such misleading approach, the basis of the entire model is constructed from three branch models, each weighing differently. Model feature selection is based on feature analysis. Each Model may fix one feature of another to simplify the compilation and reduce the feature reuse.

(1) Model 1: Date and Trap Model.

Date indicates seasons. Seasons controls raindrops. Humidity influence Mosquito Activities. So date may influence significantly on Virus. Using timeline analysis on specific place to investigate relations between date and WNVpresence. Here use reprocess date data into season and weather periods. Use K-means Clustering method to semi-supervise this model.

(2) Model 2: Area Propaganda Model with Weather Indicator
Virus outbreak need media, or mosquitoes. So there is need to research on presence of virus in a fixed time window. K-means Clustering and N-Nearest Neighbor Recognition is applied.

(3) Model 3: Species-Related Virus Break Model
Use a 2-D feature L2 regression to obtain within fixed area cluster.

The final model will be added by those three model with different weights which need to be tuned. Use add rather than multiplication in order to reduce influence the high-order noise in duplicated features.

8. Methodology

For Model 1, Logistic Regression would be best fit for a 2-class classifier. First of all analysis the appearance against time. The result are: (0 for no, 1 for Virus Presence)

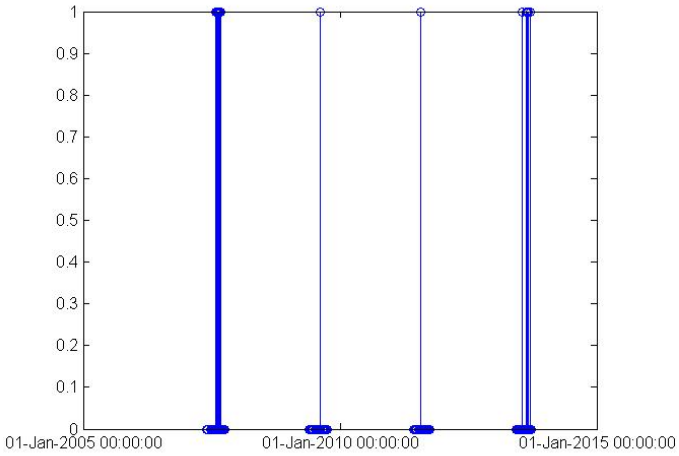

Figure 1 First Analysis of Date Data

It tells the appearance follows a periodic regulation. So further mode analysis each year.

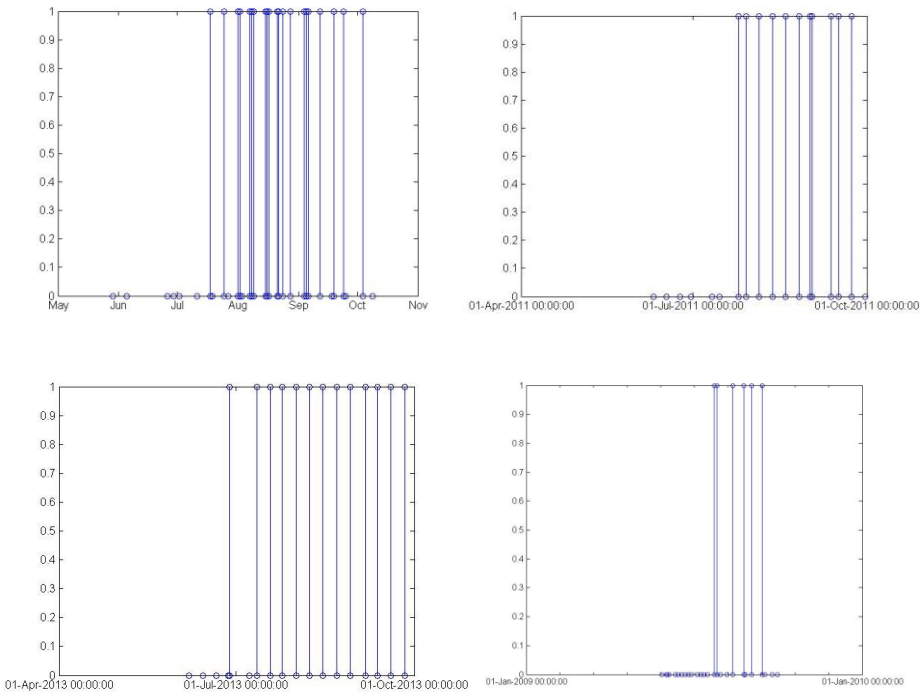

Figure 2 Clustered Date Analysis

Train the model with Train Data Set.

For model 2, cluster near trap results together to predict the probability.

Basically, K-means Clustering is introduced to this model.

The reason to choose clustering is because trap number cannot show geographical relationship among the traps. So the appearance data is separated into each single trap. Clustering the small trap data together to get a large area result. This result will be more accurate to predict than using each trap number.

In fact, the function of segmentation is equivalent to clustering. Cluster: collection of data samples (pixels). There are two ways of clustering: Divisive : Start with entire data set as a single cluster ,split cluster into components that yield largest inter-cluster distance ,and repeat until distance or clusters become small ;Agglomerative :Start with each point being a separate cluster ,merge clusters with smallest inter-cluster distance repeat until distance or clusters become large.

$$\sum_{i \in clusters} \left\{ \sum_{jth\ element\ of\ ith\ cluster} ||x_j - u_i||^2 \right\}$$

Implementation: Choose a fixed number of clusters: choose cluster centers and point-cluster allocations to minimize error: too many possible allocations for exhaustive search. But for fix cluster centers, allocate points to closest cluster, compute the cluster centers. Initial assignment may be random, and x could be any feature vector for which we can compute a distance.

For model 3, use another logistic regress to get the classifier.

9. Implementation

9.1. Feature Space

Name	Value	Usable
ID	Number	Yes
Address	String	No
Date	Date/Number	Yes
Speicies	String	No
Trap ID	Number	Yes
NumMosquitos	Number	Yes
Latitude Longitutde	Number	Yes
Block Street Address address accuracy	Number	No
WNVpresented	Boolean	yes
Weather data and spray location	Boolean	no

9.2. Pre-processing of Training Data

Name	Description	Usable
ID	The id of the record	Yes
Address	The Full Address of Trap	No
Date	Date the test is performed	Yes
Speicies	The species of mosquitos	No
Trap ID	The id of the trap	Yes
NumMosquitos	The number of the mosquitos	Yes
Latitude Longitutde	Geography data	Yes
Block Street Address address accuracy	Social location description	No
WNVpresented	Whether the virus presents	yes
Weather data and spray location	Additional data	no

While in Test set, for consideration from the Website, the mosquito number is not available, and thus became a secondary prediction target.

Additional Data Set is for enhancing accuracy, including spray data and weather data.

Model 1 Pre-process Code:

```
[row col] = size(Train_DataCopy);
val1 = 0;
black_cell = {};
block_cell_count = 1;
block_cell_vec_count = 1;
block_cell_vec = [];

datastring = '2013-Dec-31';
datastring_min = '2013-Jan-01';
formatin = 'yyyy-mmm-dd';
year_limit = datenum(datastring,formatin);
year_limit_min = datenum(datastring_min,formatin);

for i = 1:row
    if val1 ~= Train_DataCopy(i,3)
    black_cell{block_cell_count}= block_cell_vec;
    val1 = Train_DataCopy(i,3);
    block_cell_count = block_cell_count+1;
    block_cell_vec_count = 1;
    block_cell_vec = [];
    end
    if Train_DataCopy(i,1) < year_limit
        if Train_DataCopy(i,1) > year_limit_min
    block_cell_vec(block_cell_vec_count) = i;
    block_cell_vec_count = block_cell_vec_count+1;
        end
    end
end

cell_amount = size(black_cell);
data_cell = {};
for i = 1 : cell_amount(2)

    data_cell{i} = Train_DataCopy(black_cell{i},[1 9]);
    stem(data_cell{i}(:,1),data_cell{i}(:,2));
    hold on;
end

datetick('x',0);

%  [row col] = size(Train_DataCopy);
%
%  val1 = Train_DataCopy(1,3);
%  vec_mat = 1;
%  j = 1;
%  for i = 1 : col
```

```
%     if val1 == Train_DataCopy(i,3)
%         vec_mat(:,j) = [vec_mat(:,j) ; i];
%     else val1 = Train_DataCopy(i,3)
%         j = j+1;
%         vec_mat(:,j) = [vec_mat(:,j) ; i];
%     end
%  end
```

Feature Extraction

Locations features in this dataset are extremely duplicated. We only use Latitude, Longitude, block number and Trap Number for location features. Replace the mosquito species with number from 1 to 8. Change Date to Uniform Time Value

Type	Feature Value
CULEX PIPIENS/RESTUANS	1
CULEX RESTUANS	2
CULEX PIPIENS	3
CULEX SALINARIUS	4
CULEX TERRITANS	5
CULEX TARSALIS	6
UNSPECIFIED CULEX	7
CULEX ERRATICUS	8

Code:

```
for i = 1:size(MosType)
  find = strcmp(MosType{i}, 'CULEX PIPIENS/RESTUANS');
  if find
      Species_TF(:,i) = 1;
      continue;
  end
  find = strcmp(MosType{i},'CULEX RESTUANS');
  if find
      Species_TF(:,i) = 2;
      continue;
  end
  find = strcmp(MosType{i},'CULEX PIPIENS');
  if find
      Species_TF(:,i) = 3;
      continue;
  end
  find = strcmp(MosType{i},'CULEX TERRITANS');
  if find
      Species_TF(:,i) = 4;
```

```
        continue;
    end
    find = strcmp(MosType{i},'CULEX TARSALIS');
    if find
        Species_TF(:,i) = 5;
        continue;
    end
    find = strcmp(MosType{i},'CULEX ERRATICUS');
    if find
        Species_TF(:,i) = 6;
        continue;
    end
    find = strcmp(MosType{i},'CULEX SALINARIUS');
    if find
        Species_TF(:,i) = 7;
        continue;
    end
    find = strcmp(MosType{i},'UNSPECIFIED CULEX');
    if find
        Species_TF(:,i) = 8;
        continue;
    end
    Species_TF(:,i) = -1;

end

for i = 1:size(Species)
    find = strcmp(Species{i}, 'CULEX PIPIENS/RESTUANS');
    if find
        Species_TF(:,i) = 1;
        continue;
    end
    find = strcmp(Species{i},'CULEX RESTUANS');
    if find
        Species_TF(:,i) = 2;
        continue;
    end
    find = strcmp(Species{i},'CULEX PIPIENS');
    if find
        Species_TF(:,i) = 3;
        continue;
    end
    find = strcmp(Species{i},'CULEX TERRITANS');
    if find
        Species_TF(:,i) = 4;
        continue;
    end
    find = strcmp(Species{i},'CULEX TARSALIS');
    if find
        Species_TF(:,i) = 5;
        continue;
    end
    find = strcmp(Species{i},'CULEX ERRATICUS');
    if find
        Species_TF(:,i) = 6;
        continue;
```

```
  end
  find = strcmp(Species{i},'CULEX SALINARIUS');
  if find
      Species_TF(:,i) = 7;
      continue;
  end
  find = strcmp(Species{i},'UNSPECIFIED CULEX');
  if find
      Species_TF(:,i) = 8;
      continue;
  end
  Species_TF(:,i) = -1;

end

hist(MosType_TF,8);
title('Distribution of Species in Training Data');
xlabel('Specy');
ylabel('Numbers');

hist(Species_TF,8);
title('Distribution of Species in Testing Data');
xlabel('Specy');
ylabel('Numbers');
```

9.3. Training Process

For model 1, after the pre-process. Most of the data is usable. But date number is way too larger than others, and we need to normalize date number to avoid overfitting.

Code:

```
Xtrain  = Train_Data (:,[1 2 3 8]);
Ytrain  = Train_Data (:,9);
Xtest  = Test_Data (:,[1 2 3 8]);
Ytest  = Test_Data (:,9);

Xtest(:,1) = standardizeCols(Xtest(:,1));
Xtrain(:,1) = standardizeCols(Xtrain(:,1));

fitFn = @(X, y, param)logregFit(Xtrain, Ytrain, 'lambda', param,'L2');
predictFn = @logregPredict;
params = linspace(0, 0.1, 10)';
[model, bestParam] = fitCv(params,fitFn, predictFn,
@zeroOneLossFn,Xtrain, Ytrain,10);
model_1 = logregFit(Xtrain, Ytrain, 'lambda', bestParam,'L2');
yhat = logregPredict(model_1, Xtrain);
errorRate = mean(yhat ~= Ytrain)
yhat_test = logregPredict(model_1, Xtest);
errorRate_2 = mean(yhat_test ~= Ytest)
```

For model 2, training process is implemented by kmeans clustering. Cluster number is set to 20 at initial and iterate until a clear decision boundary is met. The most important part of model 2 training is to calculate the probability of virus presence within the cluster. Another way of training those clustered data is to tune the threshold of probability to represent presence/absence.

Code:

```
% K-means implementation of Final_Project.

indice_train = kmeans(Train_DataCopy(:,[5 6]),20);

P_presence = zeros(2,20);

[row col] = size(Train_DataCopy);

for i = 1 : row
    P_presence(1,indice_train(i)) = P_presence(1,indice_train(i)) +
Train_DataCopy(i,9);
     P_presence(2,indice_train(i)) = P_presence(2,indice_train(i)) + 1;
end

P_presence(3,:) = P_presence(1,:)./P_presence(2,:);

stem(P_presence(3,:));
xlabel('Cluster Index');
ylabel('Probability');
title('Distribution of Cluster Presence Probability');
```

For model 3, most of data is usable. Train with mosquito data and WNVpresence.

Model_3_implementation.m:

```
Xtrain  = Train_Data (:,[2 8]);
Ytrain  = Train_Data (:,9);
Xtest  = Test_Data (:,[2 8]);
Ytest  = Test_Data (:,9);

Xtest(:,1) = standardizeCols(Xtest(:,1));
Xtrain(:,1) = standardizeCols(Xtrain(:,1));

fitFn = @(X, y, param)logregFit(Xtrain, Ytrain, 'lambda', param,'L2');
predictFn = @logregPredict;
params = linspace(-10, 0.1, 10)';
```

```
[model, bestParam] = fitCv(params,fitFn, predictFn,
@zeroOneLossFn,Xtrain, Ytrain,10);
model_1 = logregFit(Xtrain, Ytrain, 'lambda', bestParam,'L2');
yhat = logregPredict(model_1, Xtrain);
errorRate = mean(yhat ~= Ytrain)
yhat_test = logregPredict(model_1, Xtest);
errorRate_2 = mean(yhat_test ~= Ytest)
```

9.4. Testing, Validation and Model Selection

Run the test and scatter the 2-D feature plot. The result plot in the first stage is:

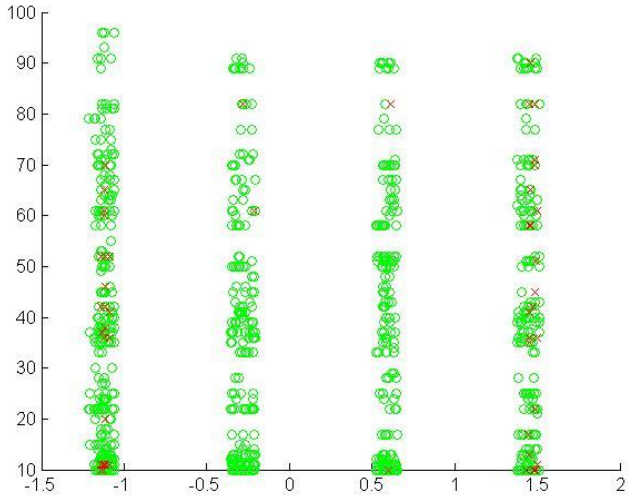

Figure 3 Scatter Plot of Figure 1

From this plot we can find the decision boundaries are split and difficult to draw. And the decision is very unstable but separable. So random tree is introduced to calibrate the model. The random tree model result will be presented after kmeans cluster result.

K-means Clustering Result:

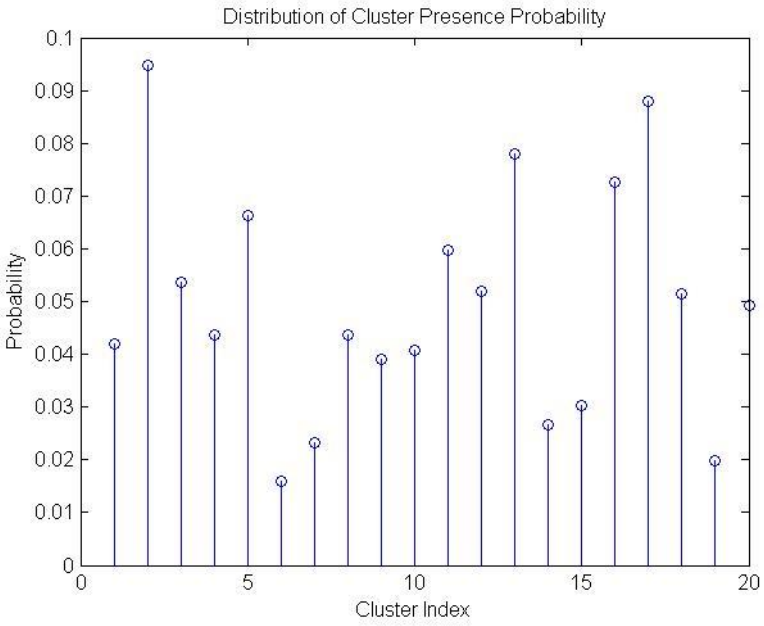

Figure 4 Scatter Plot of Model 2 Training

Notice the implementation of K-means takes only longitude and latitude data to cluster. Make the statistics of WNV presence to get the probability of presence.

The prediction of model 2 is implemented as follows: calculate the Euclidean distance of test sample data to the cluster centers, and label with the cluster's probability of presence. If the presence is over the threshold (initialize as 0.03 then tune), mark as presence.

For model 3, the result is:

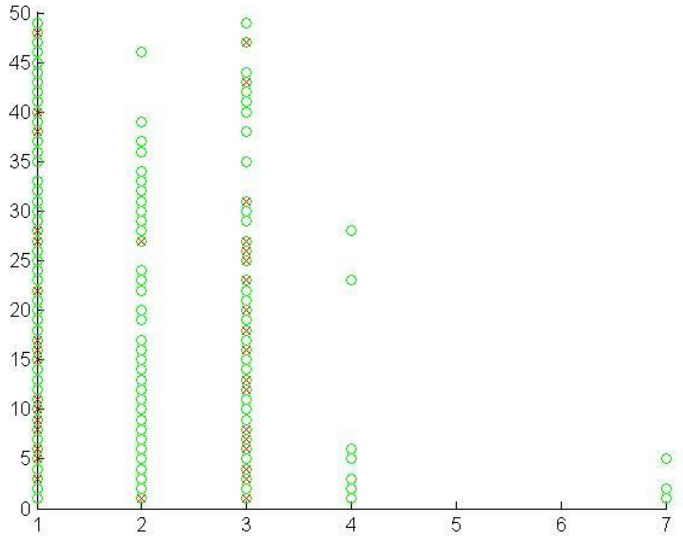

Figure 5 Model 3 Scatter Restlt

Category 1 stands for “Pippen/Rest”, and Category 3 stands for “Pippen”. From this result, Pippen Mosquito has high probability to be the media of the virus.

10. Final Results

For model 1,E_in = 5.21%, while best Prediction error rate is 5.6 %.

```
ans =L2

errorRate = 0.0521

errorRate_2 =0.0560
```

For Model_2, the prediction error rate is extremely high.

```
errorRate_3 = 0.9440
```

For Model_3, the prediction is a calibration weight for model 1 & 2.

11. Interpretation, Summary and conclusions

Date Model worked well, but Location clustering is not very successful, but there is still a lot to do to tune this model. Further work: Need to tune model 2, and implement Model 3. Need to add weather and spray data to calibrate the model.