2016

Hydro-Geological Flow Analysis Using Hidden Markov Models

Chandrahas Raj Venkat Gurram
University of South Carolina

HYDRO-GEOLOGICAL FLOW ANALYSIS
USING
HIDDEN MARKOV MODELS

by

Chandrahas Raj Gurram Venkat

Bachelor of Engineering
Osmania University, 2013

Submitted in Partial Fulfillment of the Requirements

For the Degree of Master of Science in

Computer Science and Engineering

College of Engineering and Computing

University of South Carolina

2016

Accepted by:

Csilla Farkas, Director of Thesis

John Rose, Reader

Gabriel Terejanu, Reader

Cheryl L. Addy, Vice Provost and Dean of the Graduate School

DEDICATION

To my Father… who is always an inspiration for me!

To my Mother… who has always guided me!

To my Sisters… who were always there for me!

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to Dr. Csilla Farkas and Dr. Gabriel A. Terejanu for taking time to be part of my committee.

I am very grateful to Dr. John Rose who gave me an excellent opportunity to work under his supervision and helped me at every step of my research.

Lastly, I want to thank Dr. Seth Willis Bassett for providing me the data used for the model.

ABSTRACT

Hidden Markov Models a class of statistical models used in various disciplines for understanding speech, finding different types of genes responsible for cancer and much more. In this thesis, Hidden Markov Models are used to obtain hidden states that can correlate the flow changes in the Wakulla Spring Cave. Sensors installed in the tunnels of Wakulla Spring Cave have recorded huge correlated changes in the water flows at numerous tunnels. Assuming the correlated flow changes are a consequence of system being in a set of discrete states, a Hidden Markov Model is calculated. This model comprising all the sensors installed in these conduits can help understand the correlations among the flows at each sensor and estimate the hidden states. In this thesis, using the Baum - Welch algorithm and observations from the sensors, hidden states are calculated for the model. The observations are converted from second order to first order observations using base 3 values. The generated model can help identify the set of discrete states for the quantized flow rates at each sensor. The hidden states can predict the correlated flow changes. This document further validates the assumption of the system being in a set of discrete states.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

The Woodville Karst Plain (WKP) is in the Northern Florida region with over 42 km of underwater cave passages. In this plain, one of the main caves is the Wakulla Springs cave. Numerous tunnels have been identified inside this cave. The main source of water in these caves is the seasonal rainfall. The flow of water in these caves varies from low or zero flow to flood proportions during the seasonal rainfall [3]. To understand such heavy flow variations, and various other parameters such as temperature, conductivity, salinity, depth and pressure, sensors have been installed in these tunnels. Dr. Bassett, who is also researching in generating a unique model at the Florida Geological Survey, provided the data from these sensors. This data contains fields such as temperature, conductivity, pressure, salinity, velocity along sensor x-axis, velocity along sensor y-axis, velocity along sensor z-axis, velocity along geographical north, velocity along geographical east, average north velocity, average east velocity, average speed, timestamp in the form of "Year-Month-Day Hour: Minute: Second" and data id. The figure below gives the view of the sensors used in creating the model. [1]

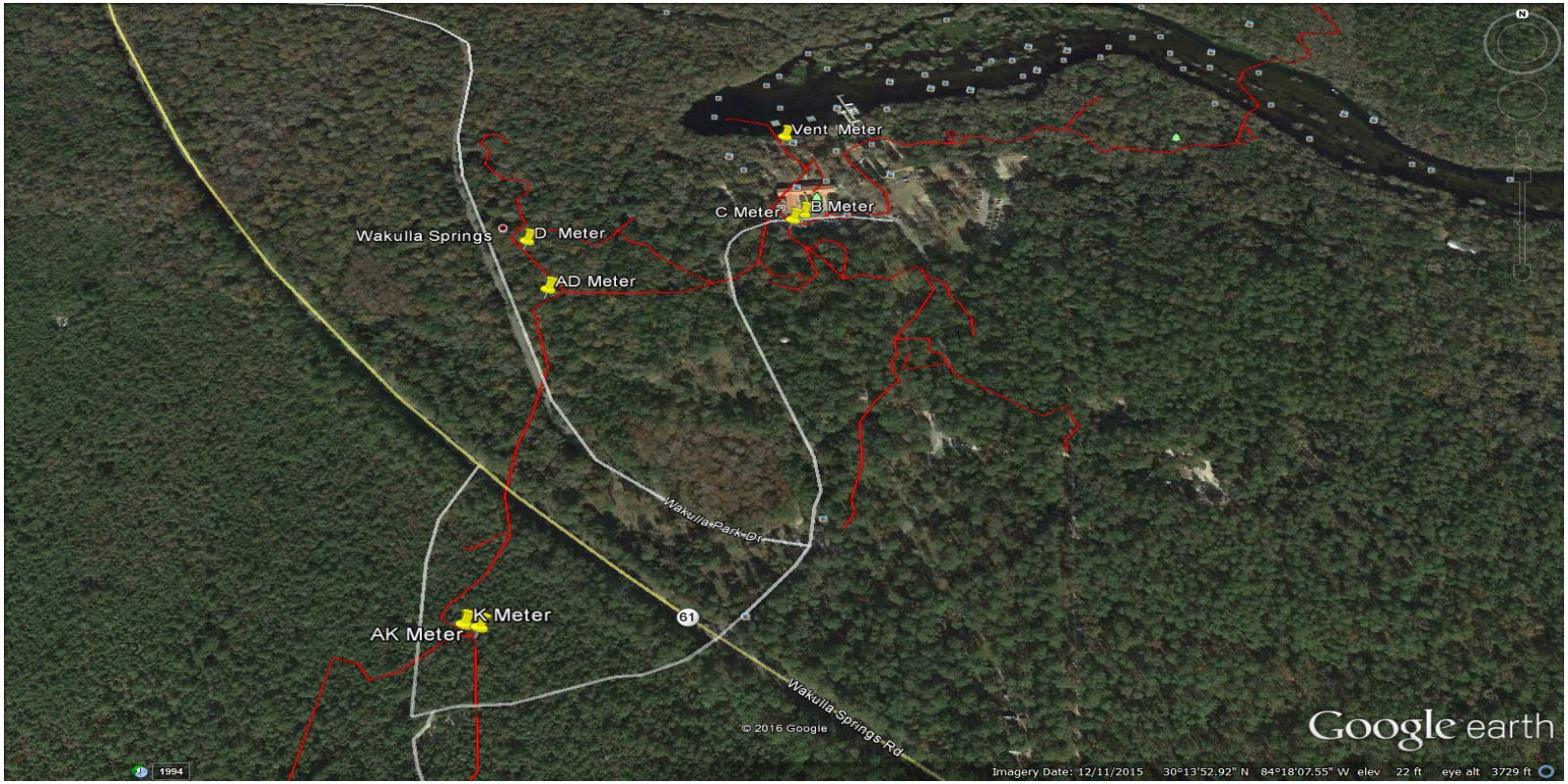


Figure 1.1 Insight of all sensors

1.1 Objective

Our objective was to analyze the data and generate a model, which identifies steady state flows in the historical data. To achieve this objective, we quantized flow rates and then looked for correlations between sensors in the system. The assumption is that the correlated flow rates are a consequence of the system being in a set of discrete states.

Among the sensors in the figure above, some sensors calculate temperature, conductivity, salinity, velocity and pressure at 15-minute intervals while some others calculate at 30-minute and 1-hour intervals. The data available ranges from 2003 to 2013. Since these states are unknown, but flow rates are known, we choose an approach to implement the design based on Hidden Markov Models. [4]

1.2 Background

The Hidden Markov Model is a tool for modelling time-series data. The Hidden Markov Model is a tool for representing probability distributions over sequences of observations [2]. L.E. Baum and his colleagues first proposed this paradigm in the late 1960's. The model was implemented for speech processing applications at the earlier stage; however, it has found use in other various fields such as molecular biology, artificial intelligence and pattern recognition.

The Hidden Markov Model gets its name from two defining properties. First, it assumes that an observation at time, t, is generated by some process whose state, S_t, is hidden from the observer. Second, it assumes that the state of this hidden process satisfies the Markov property. The Markov property states that, given the value of S_{t-1}, the current state, S_t is independent of all the states prior to t-1 [2]. In other words, the space at some

time encapsulates all we need to know about the space's history of the process in order to predict the future of the process.

1.2.1 Elements of HMM

1. "N, the number of states in a model. These states can be reached from any other state. These are normally represented as $S= \{S1, S_2 \ldots S_N\}$ and the state at time t is q_t.
2. M, the number of distinct observation symbols per state, i.e., the discrete alphabet size. These symbols are denoted as $V=\{ V_1, V_2, \ldots, V_M\}$
3. The state transition probability distribution $A = \{a_{ij}\}$ where $a_{ij} = P [q_{t+1} = S_j | q_t = S_i], 1 <= I, j <= N$. Any state can reach every other state $a_{ij} > 0$ for others $a_{ij} = 0$.
4. The observation symbol probability distribution in state j, $B = \{bj(k)\}$, where $bj(k) = P [V_k \text{ at } t | q_t = s_j], 1 <= j <= N, 1 <= k <= M$.
5. The initial state distribution $\pi = \{\pi_i\}$ where $\pi_i = P [q_1 = S_j], 1 <= I <= N$ "[2].

We represent a model as follows $\lambda = (A, B, \pi)$

HMM is used for three basic problems and various algorithms are used to solve these problems.

"Problem 1: Given the observation sequence $O = O_1, O_2 \ldots O_T$, and a model $\lambda = (A, B, \pi)$, how do we compute $P (O| \lambda)$, the probability of the observation sequence, given the model. This is called the evaluation problem. The Forward – Backward procedure is used to solve this problem.

Problem 2: Given the observation sequence $O = O_1, O_2 \ldots O_T$, and a model λ, how do we choose a corresponding state sequence $Q = q_1, q_2 \ldots q_T$ which is optimal. This is called the Decoding problem. Viterbi algorithm is used to solve this problem.

Problem 3: Given the observations, how do we adjust the model parameters $\lambda = (A, B, \pi)$, to maximize $P(O|\lambda)$. This is called the Learning problem. Baum – Welch algorithm is used to solve this problem" [2].

1.2.2 The Evaluation Problem

In this problem, we should compute the probability of an observation sequence given the model. For an HMM with P hidden states and Q observations there are P^Q possible hidden sequences. If P is very large, computing P^Q sequences is not possible. In order to reduce the number of sequences we can use the forward algorithm which uses the concept of dynamic programming and saves the intermediate values.

In the forward algorithm, we calculate the probability of an observation sequence ending at a state 'j' by summing up all the possible paths that lead to this state. Finally, we calculate the observation likelihood of the observation sequence at state 'j'. [5]

1.2.3 The Decoding Problem

Given a sequence of observations, calculating a sequence of hidden states for these observations using the HMM is called the Decoding Problem. The Viterbi algorithm is used to solve the decoding problem. The Viterbi algorithm also uses the concept of dynamic programming.

The observation sequences are processed from left to right, and a state for that observation is obtained. These states are saved into a table. Each element in the table represents the probability that the HMM is in state 'j' after seeing the first 't' observations and passing through the most probable state sequence, given the model. [5]

1.2.4 The Learning Problem

In the Hidden Markov Model learning problem, we should create a model using the observations by adjusting transitions and emission probabilities. The Baum-Welch algorithm takes the observed values, calculates the maximum log likelihood and updates the current model; this process is continued until an optimal model is obtained.

The Baum-Welch algorithm uses the well-known Estimation Maximization algorithm to calculate the maximum likelihood estimate of the parameters of a Hidden Markov Model given a set of observed vectors. The algorithm updates the parameters of the model until convergence using the forward-backward algorithm. [5]

1.3 Problem Statement

In our problem, we want to create a model that comprises all the network of sensors. This model can help us understand the flow changes at each sensor. We have assumed that the correlated flow changes are a consequence of the whole network of sensors been in some hidden states, so we want to consider 'n' number of states that can correspond to this changes. The problem we are addressing can be viewed as a learning problem in Hidden Markov Model. We can use the Baum – Welch algorithm to solve this problem.

In Baum – Welch algorithm, we need to adjust the model parameters and maximize the probability of the observations over the obtained model. To determine the final probabilities each state holds, we start by assigning some initial probabilities to these states. These initial probabilities are the probabilities each state hold between their transitions. They are randomly assigned and added to a transition matrix (A).

For example, consider each state as whether today will be a rainy day (state 1) or sunny day (state 1) wherein we walk or shop or clean.

So, our example can have transition state as follows A = [[0.7,0.3], [0.4,0.6]]. This represents that the probability of state rainy can move to state sunny is 0.3 and the probability that it will stay in its own state is 0.7. Similarly, the probability that state sunny can move to state rainy is 0.4 and the probability that it will stay in its own state is 0.6. We can also have random emission probabilities and random initial state probabilities. These are the probabilities, which we randomly assign. The figure below gives the view of this example.

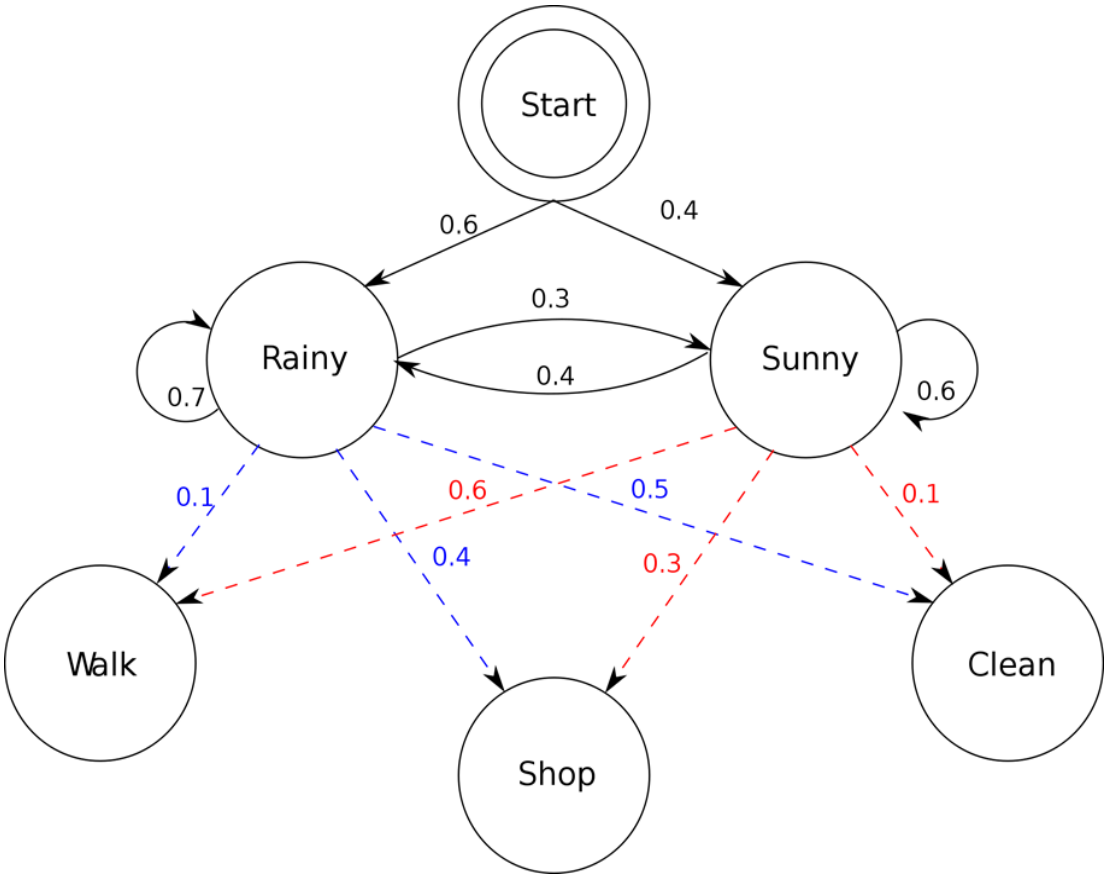


Figure 1.2: A sample Hidden Markov Model

The HMM can have many states at discrete times. These states transition from one to another based on the probabilities defined in the transition matrix. A good Hidden Markov Model is one where the final probabilities, after solving the Learning problem of HMM, would be like the expected probabilities.

In the problem, we are trying to solve, we want to create a model wherein each sensor has at least two levels. At least one level represents the probabilities of the flow going out of the cave and at least another level represents the probabilities of the flow going into the cave. There can be more emission levels if we consider different flow velocities.

The Baum – Welch algorithm is a special case of Expectation Maximization (EM) algorithm.

In this algorithm, we start with initial probability estimates, compute the expectations of how often each transition is used, re-estimate the probabilities based on those expectations and keep continuing until convergence. Using the data, we have a set of observation sequences are generated and applied to the Baum – Welch algorithm. The model gives us an estimate of the transitions between two hidden states.

This estimate helps us understand, the group of sensors as an entity whose flow can be noted using the model. We can also predict the flow over the next years using this model. Our main goal is to achieve a model λ and maximize $P(O|\lambda)$ using the observation sequences $O = O_1, O_2 \ldots O_T$.

CHAPTER 2

ANALYSIS OF DATA

To analyze and observe the direction of water flow at each sensor, graphs were created for each sensor with the flow direction on the x-axis and the frequency of the flow direction on the y-axis.

Figures 2.1, 2.3 and 2.5 show the flow direction at AD sensor, AK sensor, K sensor, B sensor, C sensor and D sensor respectively.

Figure 2.2 shows the outward flow at AD sensor following northeast direction and the inward flow following southwest direction while the outward flow at D sensor follows southeast direction and inward flow follows North West direction. Figure 2.3 shows the flow directions of AK and K sensors. AK sensor has outward flow towards north and inward flow towards south. K sensor has outward flow towards east and inward flow towards west. Figure 2.4 shows the flow directions of B and C sensors. Outward flow at B sensor is close to west and inward flow close to east. Similarly, outward flow at C sensor is close to north and inward flow close to south.

The figures show the directions where the flow is observed. A closer look at these graphs shows the above-mentioned directions in figures more accurately.

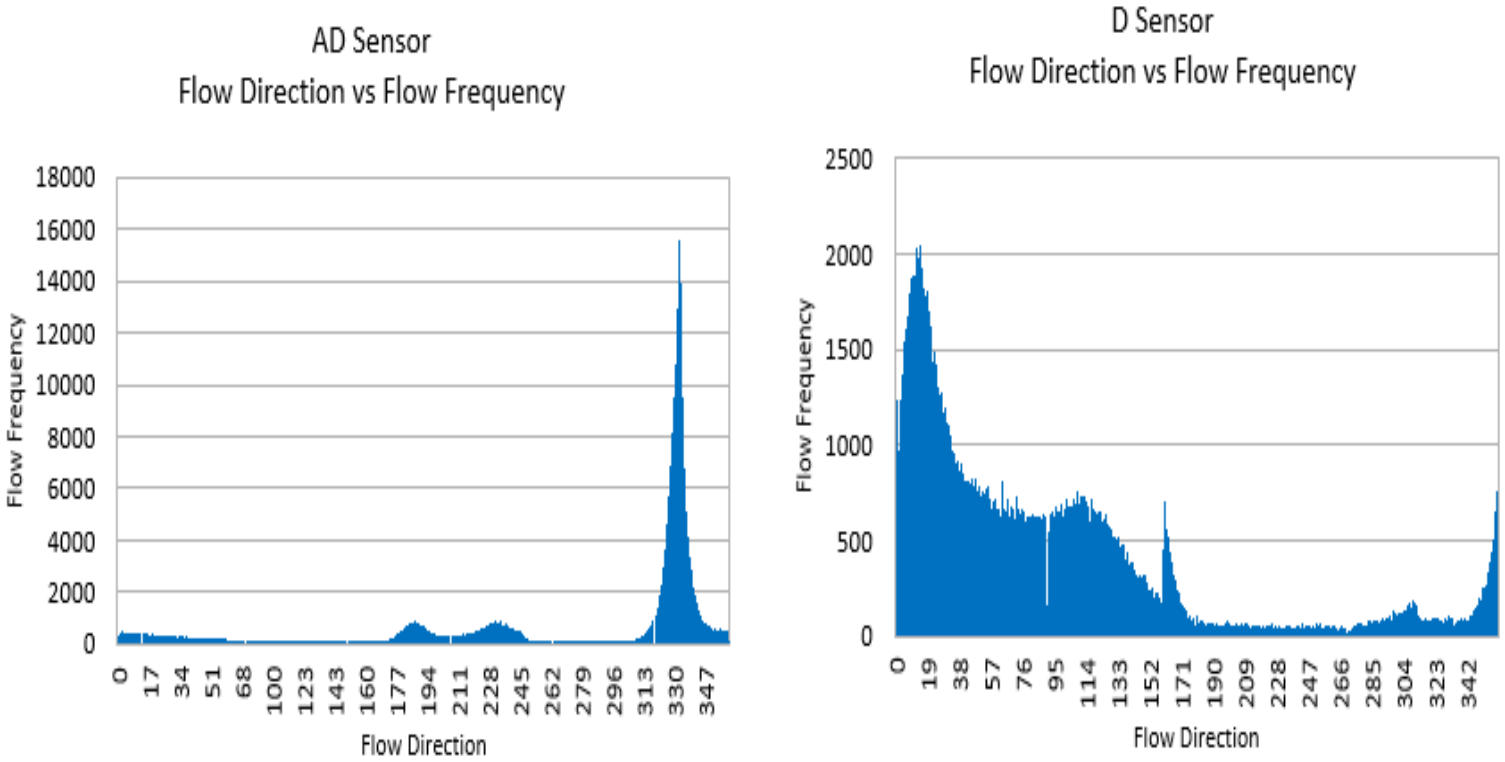


Figure 2.1 AD sensor, D sensor Flow Direction vs Flow Frequency

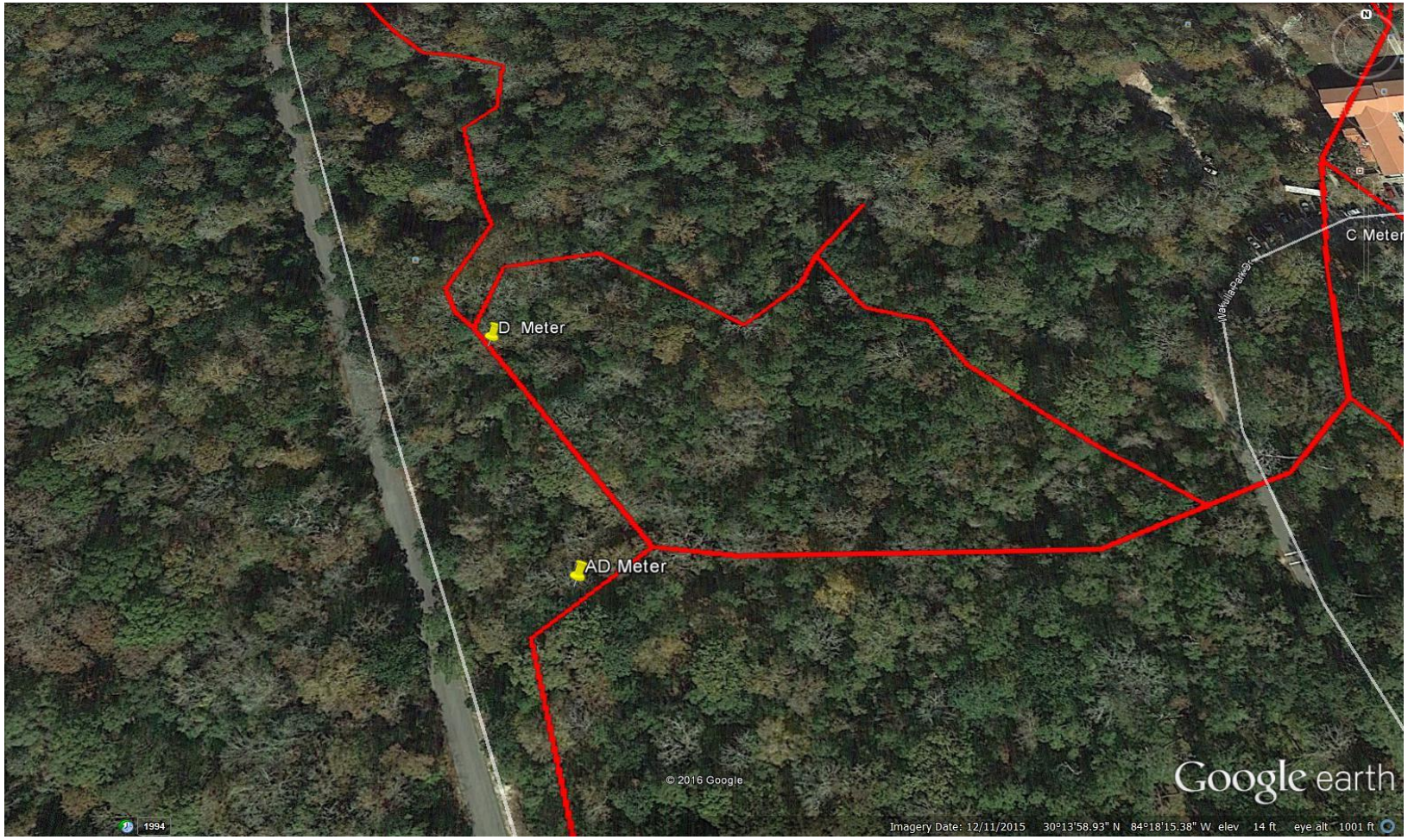
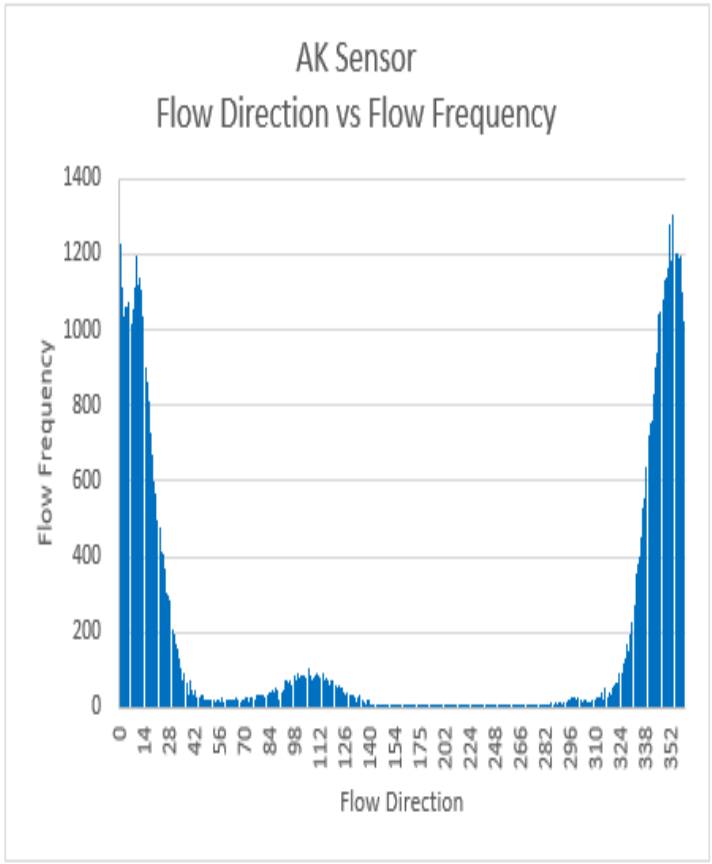


Figure 2.2 Insight of Sensors AD and D and the water flow between the sensors



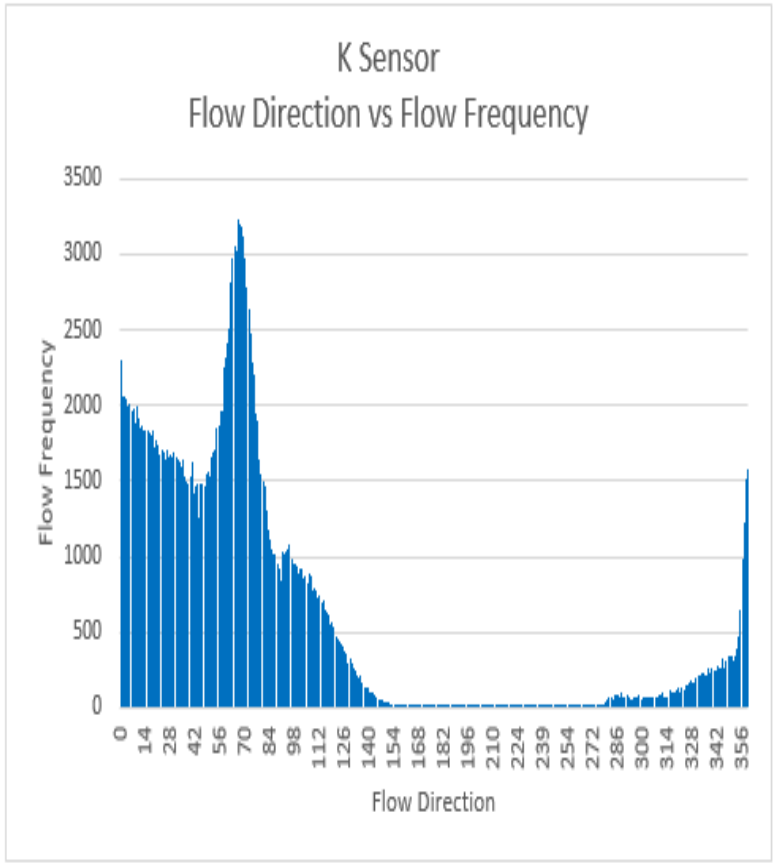
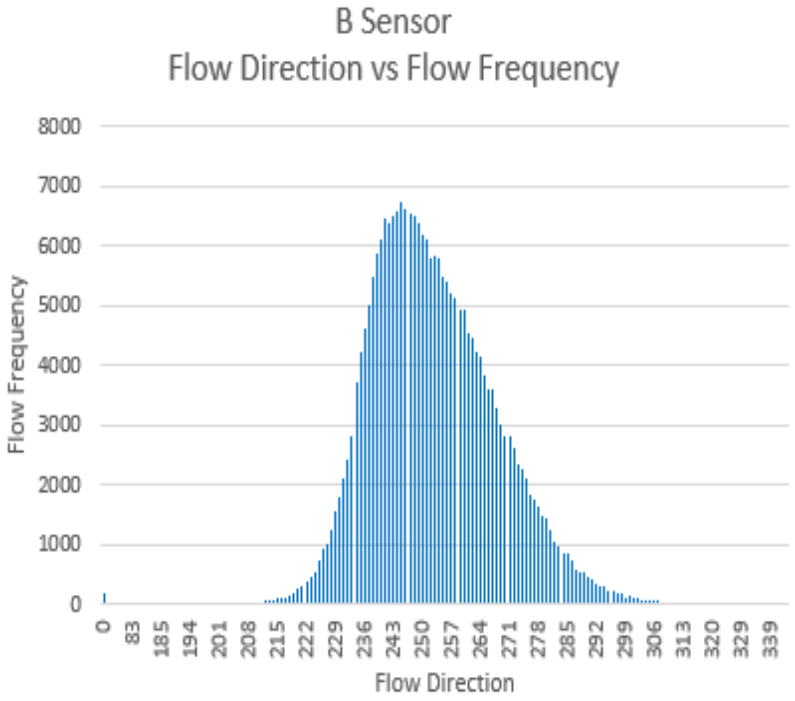


Figure 2.3 AK, K sensor Flow Direction Vs Flow Frequency

Figure 2.4 Insight of Sensors K and AK and the water flow between the sensors



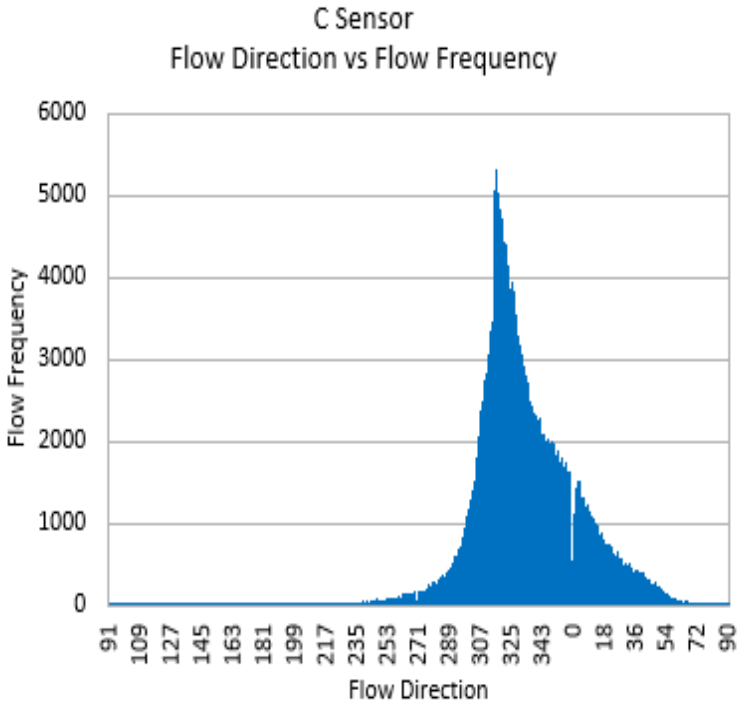


Figure 2.5 B, C sensor Flow Direction Vs Flow Frequency

Figure 2.6 Insight of Sensors B and C and the water flow between the sensors

CHAPTER 3

NORMALIZING DATA USING BAUM-WELCH ALGORITHM

In the data provided, we have 10 sensors, each designated with a unique name. Among the sensors available AK, AD, K, D, B and C are the sensors we have used to create our Hidden Markov Model. The figures 2.2, 2.4, and 2.6 depict the location of sensors AD and D, AK and K and B and C, respectively. To create an observation matrix, we need to have observations from all the sensors normalized into a unique format. As there are not many libraries that can work around second order emissions for a state we chose to create a first order emissions using this unique format.

3.1 Creating Hidden Markov Model

A flow vector is created for each data point at each sensor. We have used the parameters velocity north and velocity east from the sensor data and calculated the magnitude as sqrt $(V_n^2 + V_e^2)$ and direction as $\tan^{-1}(V_n/V_e)$.

Each data point at a sensor now holds the magnitude of the flow and the direction of the flow. Using the direction and magnitude, flow vector is created. This vector holds a negative magnitude if the direction of flow is into the cave else it holds positive magnitude. To understand the flow at a sensor throughout the year, scatter plots are created. Each scatter plot has days in a year on the x-axes and the calculated vector on the y-axes. These scatter plots also help validate the flow probabilities of each sensor.

Figures 3.1, 3.2, 3.3, 3.4, and 3.5 show subplots of each year for AD sensor, C sensor, K sensor, B sensor and AK sensor, respectively.

3.1.1 Creating Observations

Flow at each sensor spans different ranges. For example, in year 2006 and 2007 the flow at AD sensor ranged from a minimum of -4cm/s to a maximum of 12 cm/s. In year 2008, the range varied from a minimum of -6cm/s to a maximum of 35cm/s. The ranges of other sensors also varied similarly.

To obtain a unified model, which can correlate flow estimations at each sensor a set of emission range/ranges is required. To obtain these emission ranges individual Hidden Markov Models are created. These models gave probabilities of flow vector at each sensor. Using the probabilities of emissions at each sensor, ranges that can suffice all the sensors were chosen. For example, the probability of a flow vector below 1cm/s has good probability among all sensors except B sensor, while the probability of emission range above 9cm/s has good probability for AD and B sensor. To accommodate all the sensors, the following emission ranges are considered.

Flow vector less than -2cm/s is considered as emission-1, flow vector between -2cm/s and 2cm/s is considered as emission-2 and flow vector above 2cm/s is considered emission-3. The data from D sensor has lot of gaps in it, so we have not considered it for the model.

Our value array now contains 5 digits, one digit for each of the five sensors in the following order.

0	1	2	3	4
AD	AK	K	B	C

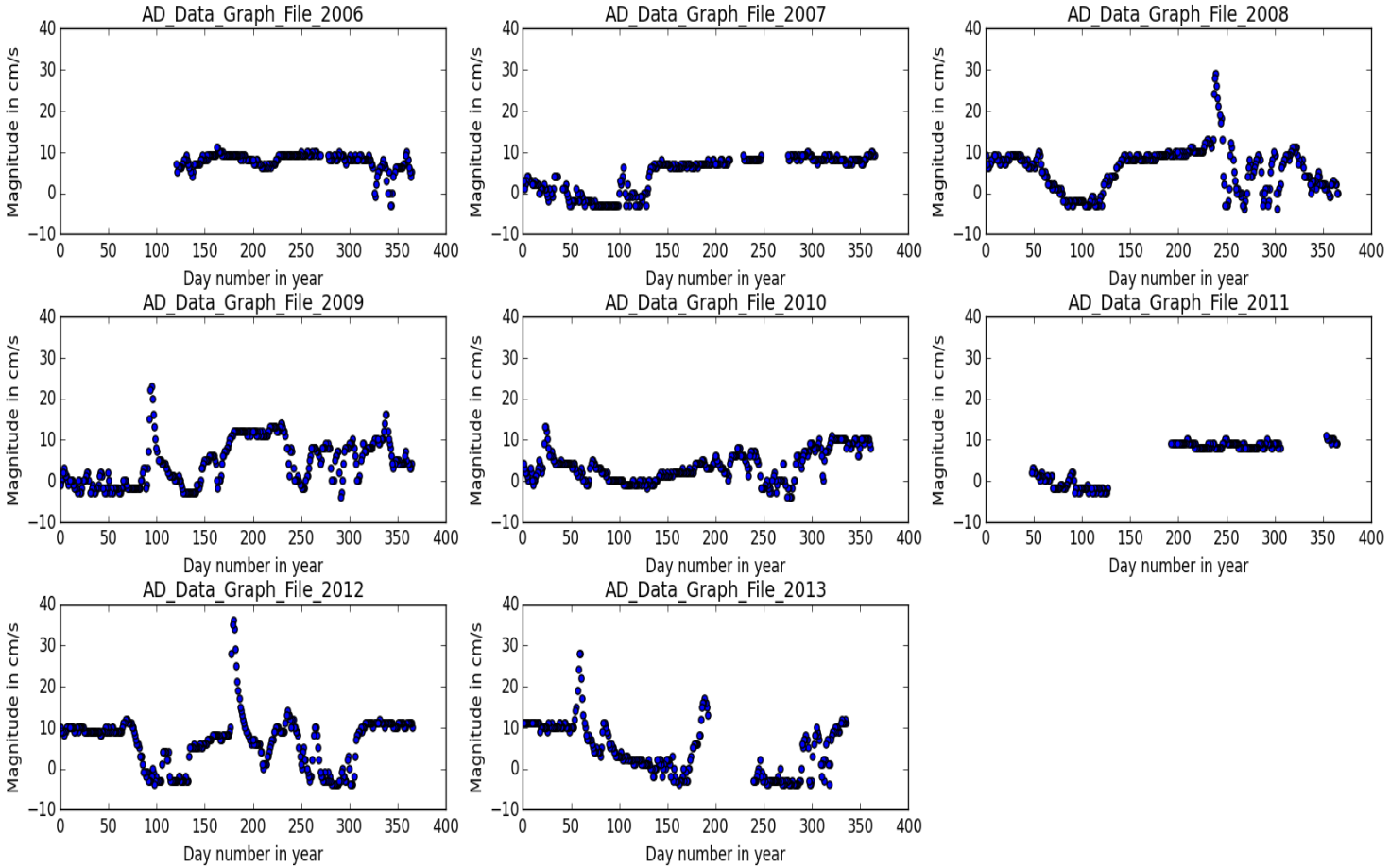


Figure 3.1 Days in the year vs Flow vector of AD sensor from 2006 to 2013

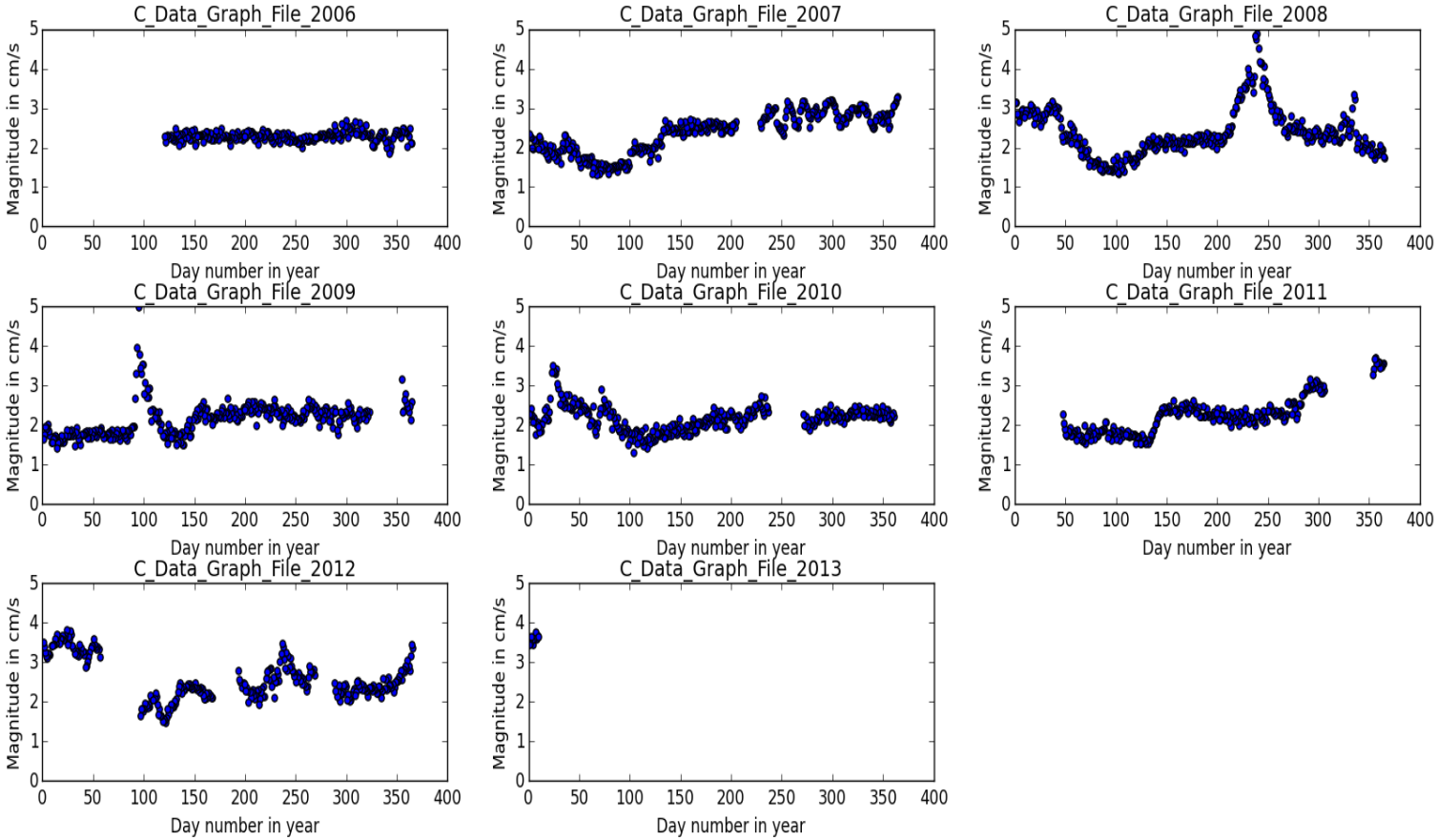


Figure 3.2 Days in the year vs Flow vector of C sensor from 2006 to 2013

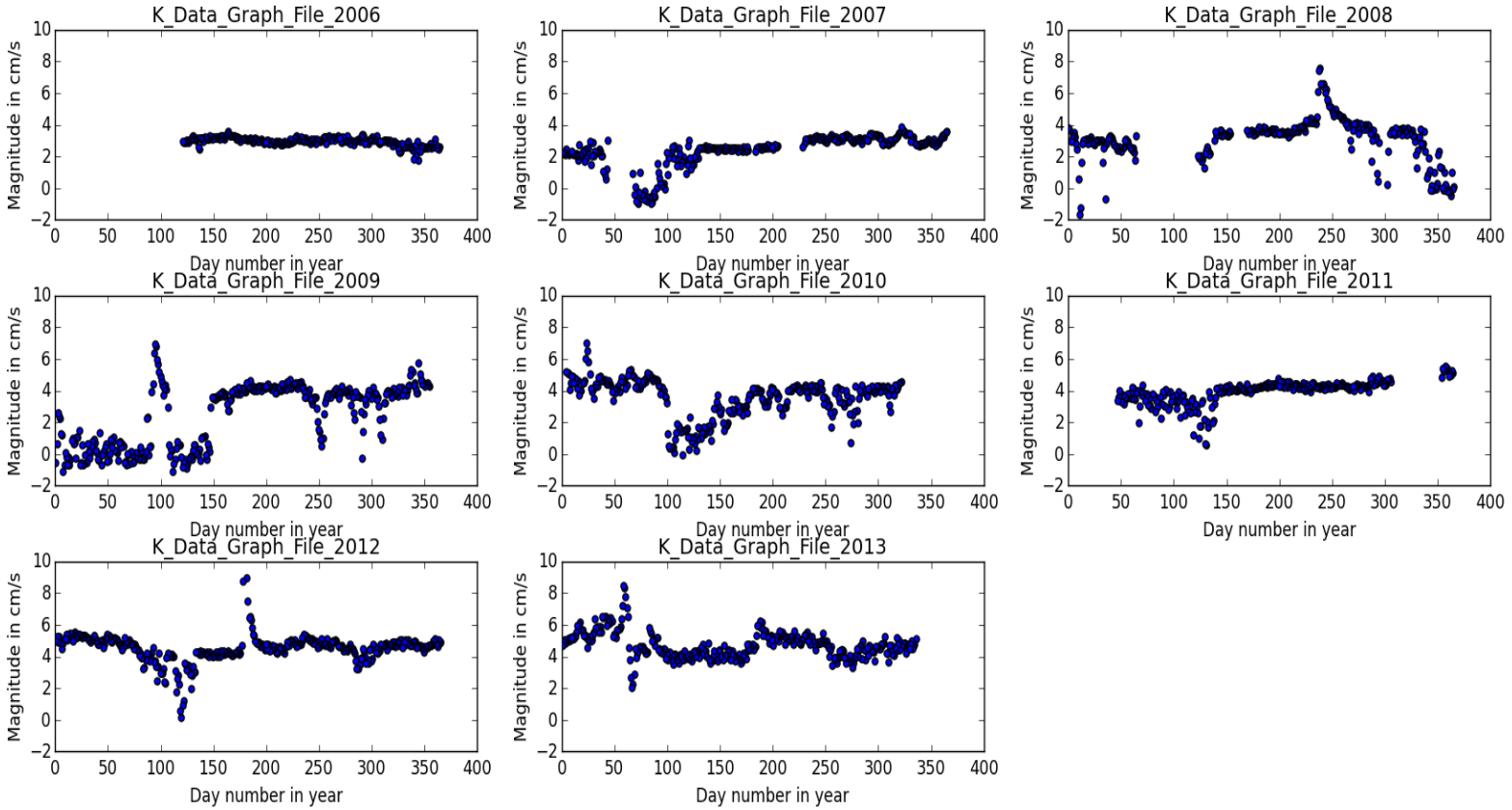


Figure 3.3 Days in the year vs Flow vector of K sensor from 2006 to 2013

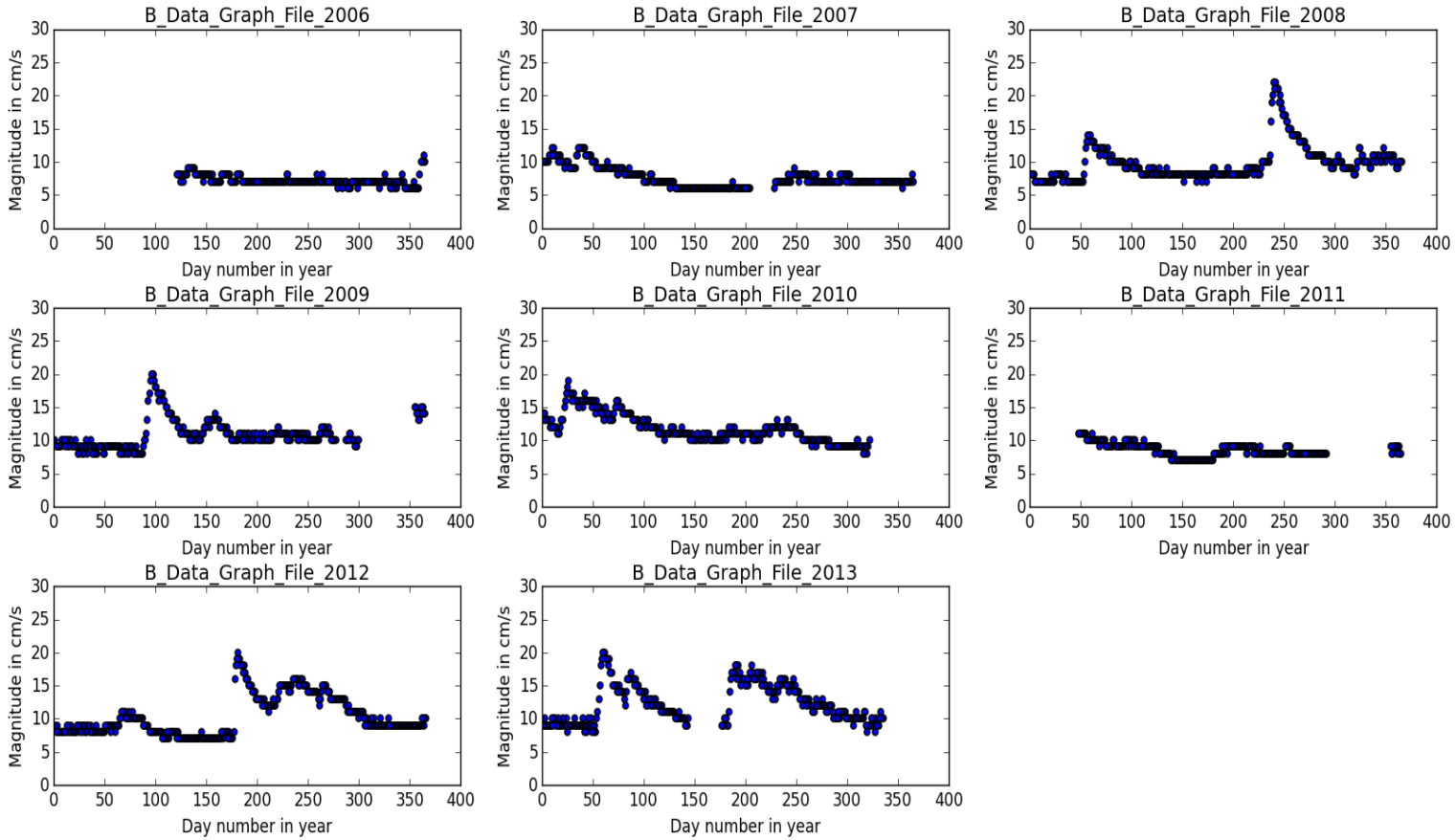


Figure 3.4 Days in the year vs Flow vector of B sensor from 2006 to 2013

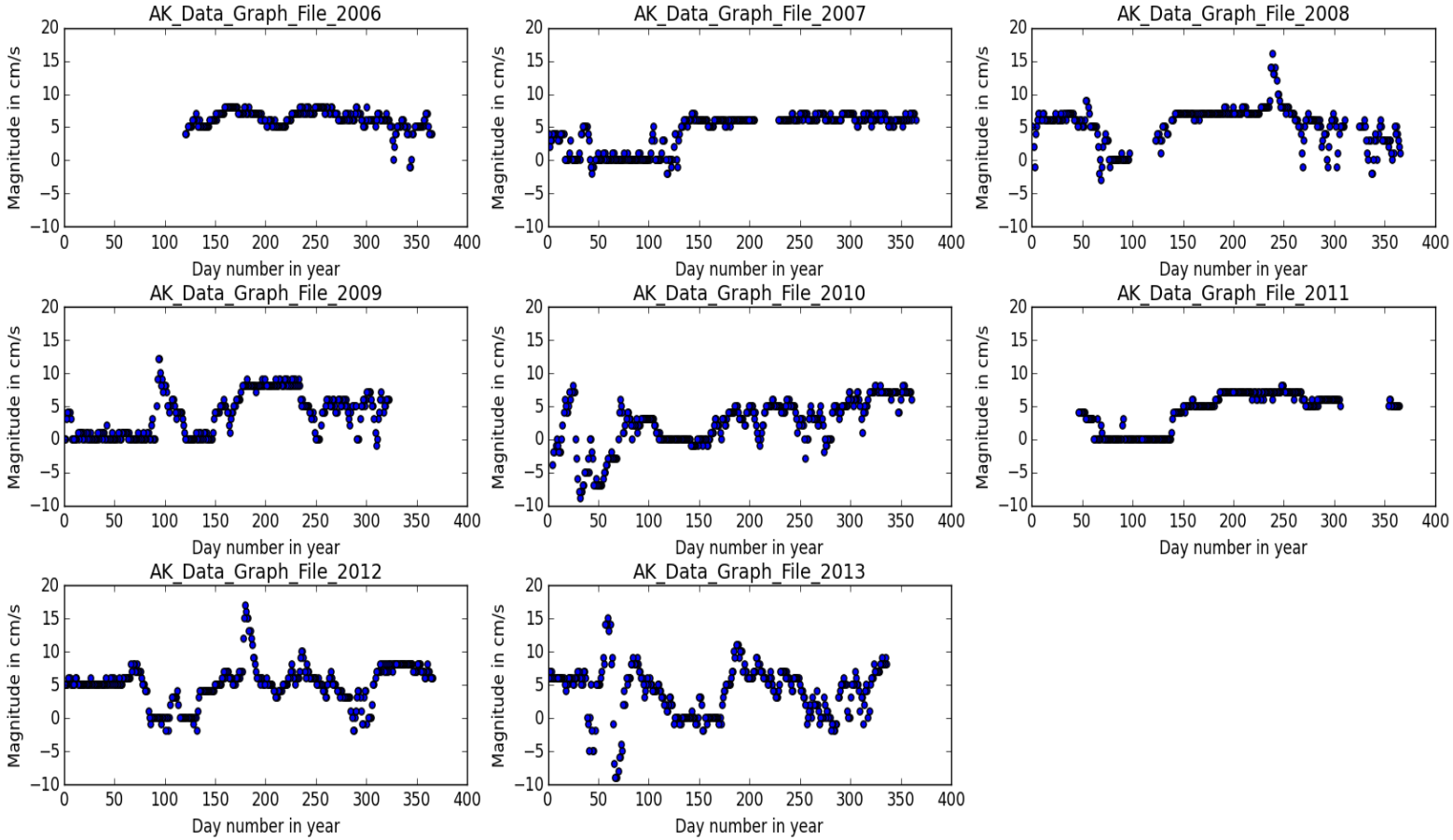


Figure 3.5 Days in the year vs Flow vector of AK sensor from 2006 to 2013

Each sensor internally can hold three different values ranging from zero to two. As we have mentioned earlier about Hidden Markov Model libraries, we want to have first order emissions, which implies we cannot have emissions having internal emissions. Therefore, we should concise the 5-value array into a unique number, which could be supplied to Baum-Welch algorithm and which can be used to distinguish between the sensors. To achieve such requirement, we can choose to convert the 5-value number into a base_3 number.

As each value in a base_3 number can hold values from 0-2, this helps us obtain a unique number which can be supplied to the Baum-Welch algorithm. The calculated observations are supplied to the Baum-Welch algorithm and the below model is obtained.

3.2 Understanding and Validating the Model

In the model, we have obtained, the probability of State – 0 moving to State – 1 is 0.03 and the probability of State – 1 moving to State – 0 is 0.08. The probability of State – 1 staying in its own state is 0.92 and the probability of State – 0 staying in its own state is 0.97.

The table depicts the emission probabilities of the hidden states. As mentioned earlier the flow at each sensor has been quantized into three ranges. If the flow is less than -2cm /s, it is called Inward flow. If the flow less than or equal to 2cm/s, and greater than or equal to -2cm/s it is called Stagnant flow. If the flow is greater than 2cm/s, it is called Outward flow.

In the model that I have created, outward flows at each sensor exists in State – 0 rather than State – 1, except for sensor B that always has outward outflow. Sensor C always has stagnant flow and this can be observed in State – 0, whereas in State – 1 the probability is

STATE – 0				STATE - 1			
Sensor	F < -2cm/s Inward Flow	F>= -2cm/s F<= 2cm/s Stagnant Flow	F>2cm/s Outward Flow	Sensor	F < -2cm/s Inward Flow	F>= -2cm/s F<= 2cm/s Stagnant Flow	F> 2cm/s Outward Flow
AD	5	14	81	AD	12	21	66
AK	2	9	89	AK	0	30	70
K	0	11	88	K	0	70	30
B	0	0	100	B	0	0	100
C	0	97	3	C	0	78	22

Figure 3.6: Final Hidden Markov Model, here F means Flow Vector

divided between outflow and stagnant flow. In State – 1 the flow at the sensors is divided between stagnant flow and outward flow.

The hidden states that I have obtained help us understand that when the flow at the sensors is outward, the sensors stay in State – 0. In this state sensors AD, AK, K and B sensors have high probability to have outward flow. Similarly, sensor C has high probability to have stagnant flow in this state. Whereas when the flow at sensors AD, AK, C and K shifts between stagnant flow and outward flow, they tend to stay in State – 1.

To validate the generated model, I created four test cases. I removed a year data from all the data and used this data to generate the model. The removed year's data is then supplied to the Viterbi algorithm, which creates a set of states for all the observations I have provided. The below figures depict the generated states over the year data, and the scatter data of AD sensor during the years 2007, 2008, 2009 and 2012.

As can be seen when the flow pushes outward, the model, tends to stay in State – 0 whereas when the flow decreases the model moves to State – 1. In the 2009 graphs, when the flow magnitude is less than 4cm/s during the days 1- 100 (January, February and March months) the model stayed in State – 1. When the flow increases suddenly, the model moved to State – 0, as shown in the graph. This phenomenon continued throughout the year.

The graphs for 2007, 2008, and 2012 also agree with the above-mentioned phenomena. When the flow increases, the model moves to State – 0 and when it decreases it moves to State – 1. These figures help us validate our created model. The comparisons between the states generated by the graph and the real-time data from AD sensor for the years 2007, 2008, 2009 and 2012 is shown below.

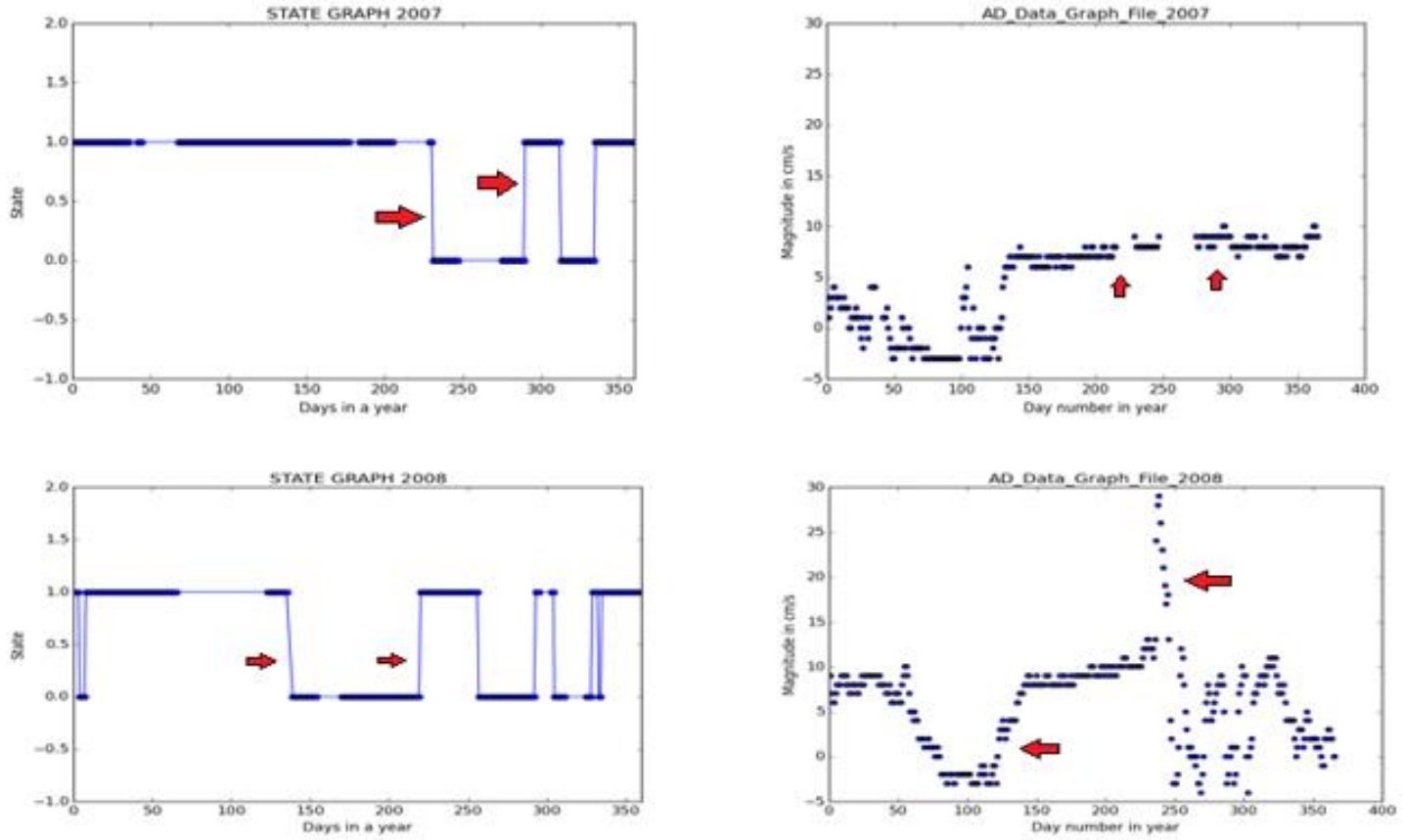


Figure 3.7 Comparison between predicted state graphs and actual data for years 2007, 2008

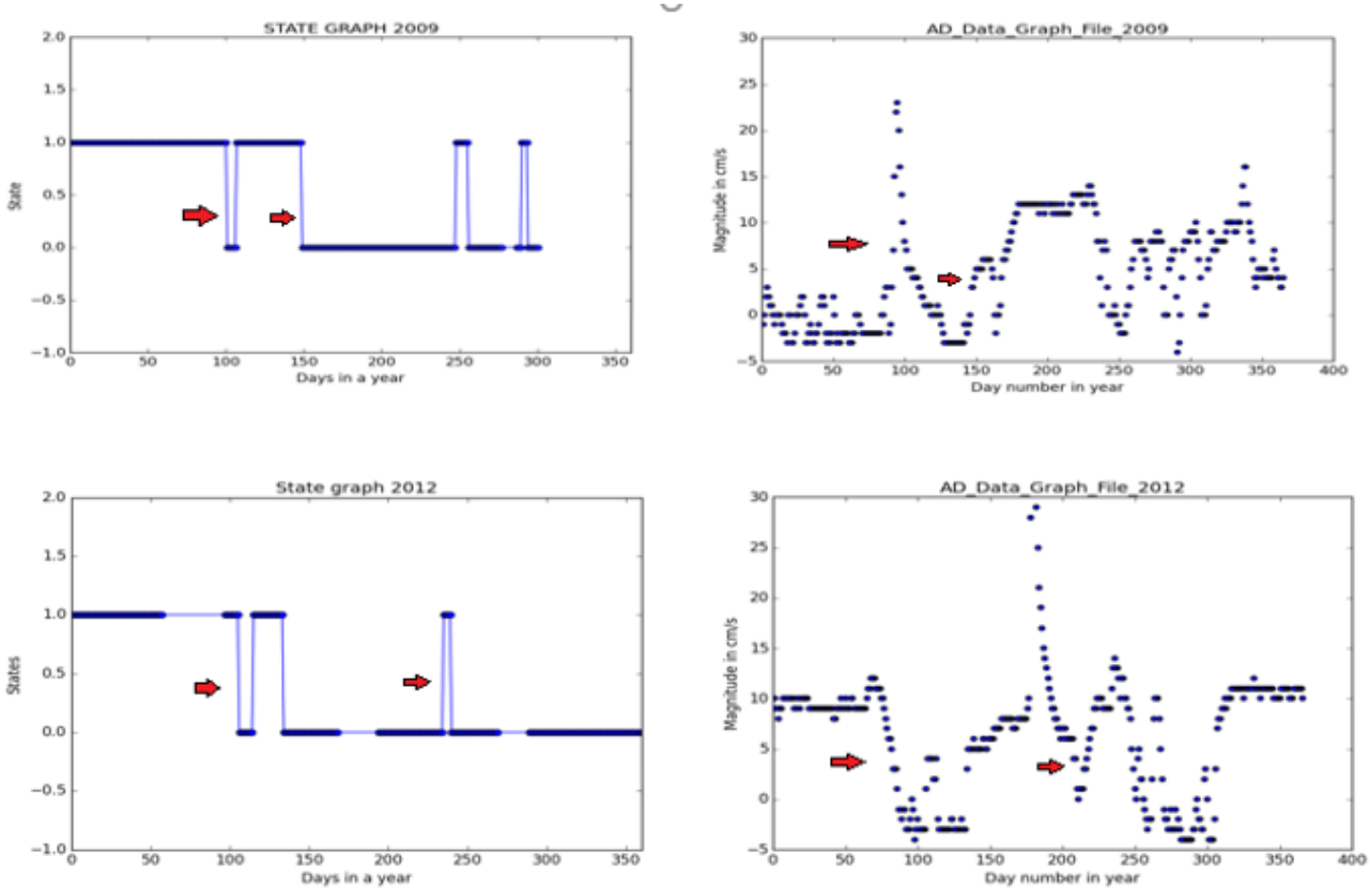


Figure 3.8 Comparison between predicted state graphs and actual data for years 2009, 2012

3.3 Conclusion

We have created flow vectors for each data point at each sensor and used these flow vectors to identify the hidden states for our model. The created model can be used to study the flow rate changes at each sensor with sudden changes reflected at different sensors. We tested this model using sensor data from years 2007,2008,2009 and 2012 and observed the model showing good validation for all the years. In the comparison graphs above you, the produced state graphs match well with the real-time data. One of the limitations to our model is that it has good match with real time data when there is a flow change with time at any given location.

REFERENCES

1. Werner, Christopher. "Determination of Groundwater Flow Patterns from Cave Exploration in the Woodville Karst Plain, Florida." Determination of Groundwater Flow Patterns from Cave Exploration in the Woodville Karst Plain, Florida. Web. 21 Mar. 2016.
2. Lawrence R., Rabiner. "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition." IEEE. Web. 21 Mar. 2016.
3. "Karst in Florida (FGS: Special Publication 29)." Karst in Florida (FGS: Special Publication 29). Ed. Elton J. Gissendanner. Florida Geologocal Surveys Publications, n.d. Web. 21 Aug. 2016.
4. Hu, Zexuan, Seth Willis Bassett, Bill Hu, and Scott Barrett Dryer. "Long Distance Seawater Intrusion through a Karst Conduit Network in the Woodville Karst Plain, Florida." Nature.com. Nature Publishing Group, 25 Aug. 2016. Web. 06 Sept. 2016.
5. Alexander, Schliep, Wasinee Rungsarityotin, Benjamin Georgi, Alexander Schonhuth. “The General Hidden Markov Model Library: Analyzing Systems with Unobservable States.” 06 Nov. 2016. Web. .