AM207 Final Project
A dive into Bayesian Weather Modelling

Victor Lei

Institute for Applied Computational Science

52 Oxford Street

Cambridge, MA

`vlei@g.harvard.edu`

Charles Liu

Institute for Applied Computational Science

52 Oxford Street

Cambridge, MA

`cliu02@g.harvard.edu`

Leonhard Spiegelberg

Institute for Applied Computational Science

52 Oxford Street

Cambridge, MA

`spiegelberg@g.harvard.edu`

Vinay Subbiah

Institute for Applied Computational Science

52 Oxford Street

Cambridge, MA

`vinaysubbiah@g.harvard.edu`

May 5, 2016

Abstract

Statistical weather modeling has a long tradition. In this project we apply different methods in a Bayesian context on historical weather data. Using a simplistic approach we implement and test Bayesian models based on a network topology (Bayesian Networks) and time series based models on weather measures like temperature, pressure or precipitation. In another modeling approach we investigate a hidden Markov model on weather events like rain, snow or fog. Statistical Inference based on these models has a wide range of possible applications. In this project we try to infer values for either missing data points (i.e. when measurements are not sufficient) or for missing stations / points.

Contents

1   Introduction

Traditional local and global weather modeling hugely relies on numerical simulations as well as statistical techniques. With the development and increasing popularity of Machine Learning algorithms in a Bayesian context, this project aims at serving as a study of techniques demonstrated along examples on how Bayesian techniques could be used for a data based approach in weather modeling.

Usually, weather data is collected at fixed weather stations that do not follow any particular pattern in terms of position or location. Hence, data is inherently spatially aware but not for every location is data available. Traditional approaches to infer missing data for specific locations rely either on numerical simulation results (and interpolating them) or on a local numerical simulation on a locally fine-grained grid. However, for many applications it is sufficient to use a statistical weather generator[1] which then can be used to infer a local distribution for weather parameters. One application of this could be for example to compute the risk of heavy rain for a location where no historical weather data is available. Another would be to actually use a statistical weather generator to construct initial boundary conditions of missing points that are then fed into a numerical model.

For this project we collected historical data of 18 stations from 1960-2015 around Boston from `wunderground.com`.

Figure 1: Illustration of the 18 stations which data has been used for this project with weather stations being located next to airports.

Since historical weather data is usually hard to get or hidden behind paywalls, though we secured a decent amount of variables we do not claim that our data base should be used for a real-world application. Instead, it serves more as a base for applying several techniques and investigating their feasibility. The first part of our study investigates the modeling of single weather measures demonstrated along precipitation whereas the second part focuses on a state-based approach for weather events.

2   Bayesian Networks

Due to its spatial awareness, a hierarchical structure based on independence is a natural candidate to model a single weather measure. Bayesian Networks have been successfully applied to model precipitation by using a combination of historical weather data and numerical simulation data [4].

2.1   Network construction

However, the question on how to construct the dependency graph which needs to be acyclic is non-trivial as it is a proven NP-hard problem. Thus, possible solutions often rely on greedy algorithms. In the context of weather modeling, dependency are usually inferred by a combination of manual dependencies obtained via domain knowledge and automated processing. As we do not possess any in-depth domain knowledge about weather dependencies we opted to push automatic network structure learning further by automating the whole process.

One of the most popular algorithms for structural learning of a Bayesian Network is the K2 algorithm. But to make it work, an initial ordering of the nodes (i.e. stations) is needed.

2.1.1   Initial Ordering

We assume that stations with a fewer distance are more likely to be dependent than stations that are farther away.

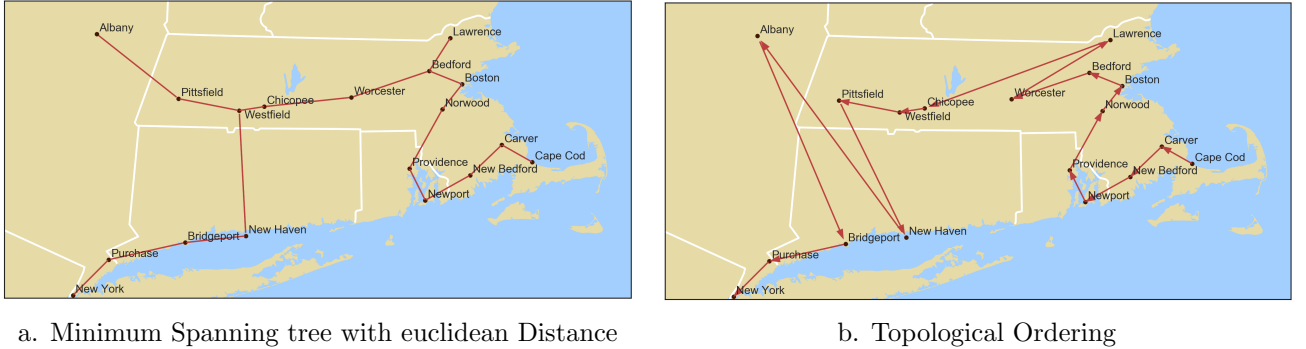a. Minimum Spanning tree with euclidean Distance   b. Topological Ordering

Figure 2: Construction of an initial node order. Arrows denote a child-parent relation.

In a first step we construct a Minimum Spanning Tree (MST) with the euclidean distance used as edge weight (i.e. via Kruskal's algorithm[8]). Using this MST we select a node (typically one that has many neighbors in the MST) and perform a Breadth First Search (BFS) to obtain a topological ordering which is then used as input for the K2 algorithm. It should be noted that neither the MST nor the topological ordering are unique. Running the algorithm several times to obtain a plausible ordering is therefore strongly encouraged. Furthermore, we see this ansatz as a practical approach to get an initial dependency layout which can then be manually refined if needed (cf. Figure 2).

2.1.2   K2 + Simulated Annealing

We created six buckets of precipitation levels - other than the first bucket which indicates no rain, the latter five were manually adjusted to have similar counts. K2 is a greedy algorithm based on our ordering heuristic.

We define the probability that a node i has parents π_i as:

$$f(i, \pi_i) \;=\; \prod_{j=1}^{q_i} \frac{(r-1)!}{(N_{ij}+r-1)!} \prod_{k=1}^{r} \alpha_{ijk}!$$

where

$$
\begin{aligned}
q_i &: \quad |\{\text{Permutations of values for } \pi_i\}| \\
r &: \quad |\{\text{Number of precipitation levels}\}| \\
\alpha_{ijk} &: \quad |\{\text{node i with value } k, \text{ parents } \pi_i \text{ with permutation j}\}| \\
N_{ij} &: \quad \sum_{k=1}^{r} \alpha_{ijk}
\end{aligned}
$$

Using this probability, we iterate through the ordering and for each node i do the following:

1. Calculate base probability with no parents

2. Select the node preceding i in the ordering that maximizes the probability

3. If probability is greater than base probability:

   - Update base probability
   - Add node to list of parents
   - Go back to 2

4. Otherwise return node's parents
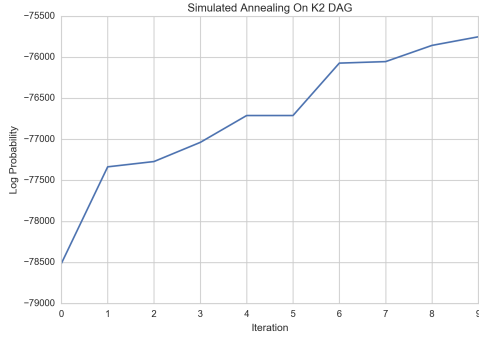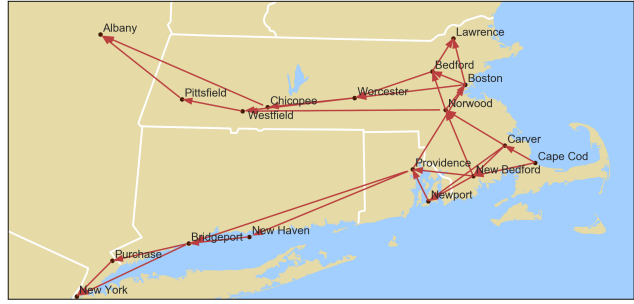
a. Simulated Annealing   b. Final Bayesian Network

Figure 3: Simulating Annealing on K2 algorithm and resulting structure.

As this algorithm is greedy, it has no guarantees on being optimal. We took the resulting structure from the K2 algorithm and ran Simulated Annealing, randomly selecting parents for a node based on temperature. Over 1000 iterations we saw only nine improvements, but the resulting dependency structure seems reasonable [cf. Figure 3].
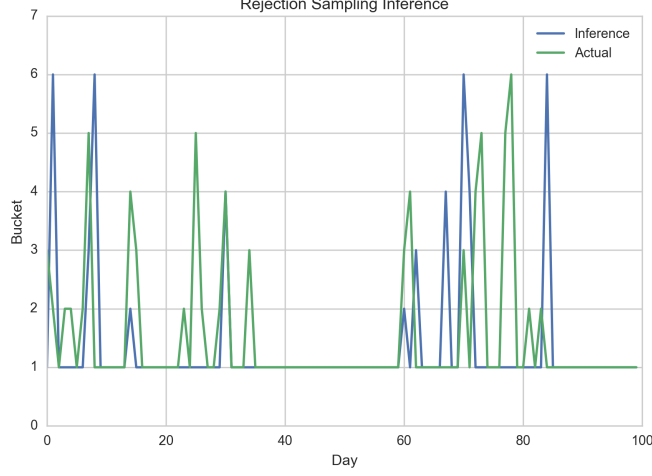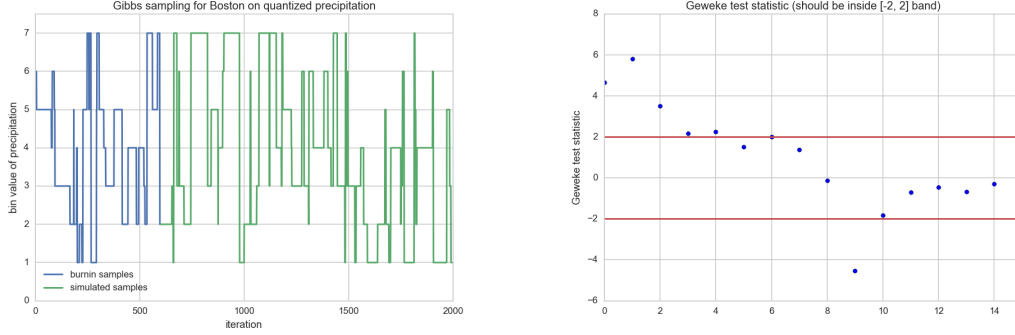
2.2 Inference

2.2.1 Direct Sampling

Figure 4: Rejection Sampling For Inferring Boston Precipitation.

We used rejection sampling as a simple method to infer historical precipitation levels. Using the empirical data, a sample is created by iterating through the node ordering and sampling from the conditional probability distribution based on the parent nodes. We tried to model the precipitation levels of Boston via it's parents in our network. For each historical data point we created many samples, rejecting those where the parents' precipitation levels didn't match with those from the historical data point. The resulting distribution was then sampled to obtain the precipitation level for Boston that day [cf. Figure 4].

2.2.2 Monte Carlo / Gibbs Sampling

Another approach to sample the Bayesian Network is to use Gibbs Sampling. Let $X^{(t)} := \begin{bmatrix} X_1^{(t)} & ... & X_n^{(t)} \end{bmatrix}^T$ a vector of random variables describing the current state of each of n nodes at time t. Then we can sample a new state vector by picking randomly one $X_i^{(t)}$ and sampling a new representation of it by conditioning on the other variables, i.e. draw $X_i^{(t+1)} \sim \mathbb{P}(X_i^{(t)}|X_1^{(t)}, ..., X_{i-1}^{(t)}, X_{i+1}^{(t)}, X_n^{(t)})$. For Bayesian networks we know that conditioning on $X_1^{(t)}, ..., X_{i-1}^{(t)}, X_{i+1}^{(t)}, X_n^{(t)}$ is equal to conditioning on the parents, the children, and the parents of the children of X_i. The set of these nodes is also called the markov blanket of X_i [7]. Repeating this process will eventually converge to the distribution as implied by the Bayesian Network's model (cf. Figure 5).

a. Inference using Gibbs Sampling for Boston   b. Geweke test statistic

Figure 5: Exemplary stochastic precipitation generator for Boston using Gibbs Sampling. Due to complexity of data many samples are needed for convergence.

2.3 Next steps

The Bayesian Network assumes that weather properties for a particular region are derived from other local regions. While our model only relies on precipitation data, it can easily be extended to include models on atmospheric pressure and the like. The motivation for this would be to infer some properties of a region where weather data is hard to find - an ordering that ends on that region would then create its dependency structure. It can also be used for filling in missing historical data - in our case some regions such as Boston had nearly every data point between 1960 to 2015 (around 20K points), whereas others had far fewer. We found the intersection of common dates between all 18 regions was only around 5000 points, so one potential use case would be to look through the missing data and try to infer its precipitation levels through available data from the network.

3 Bayesian Hierarchical Modeling of Rainfall Volume

Rainfall models are typically focused on two issues:

1. the probability of any rainfall; and

2. the volume of rainfall when it does rain.

Here, we are focused on the second question. Traditional models for rainfall volume have found that the daily rainfall volume typically follows a Gamma distribution with shape parameter α and rate parameter $\frac{\alpha}{\mu}$ such that the expected volume of rainfall for each day is μ. [3]

Coe et al. [5] made the simplification that that the shape parameter for the Gamma distribution would remain constant, citing previous studies that made this same claim. Instead, they focused on the μ variable which could vary over time by the generalized linear model $\mu = h(g(t))$. h is some known function like an exponential, and g is some functional form that describes the variation in the mean rainfall volume over time, such as a polynomial or Fourier series. Based on these ideas, we make two contributions that extend the discussion from the paper: producing a two-period annual model for rainfall volume with a variable switching-point, and assessing whether the shape parameter remains constant using this two-period model as has been claimed in previous works.

3.1 The case of New Bedford, MA

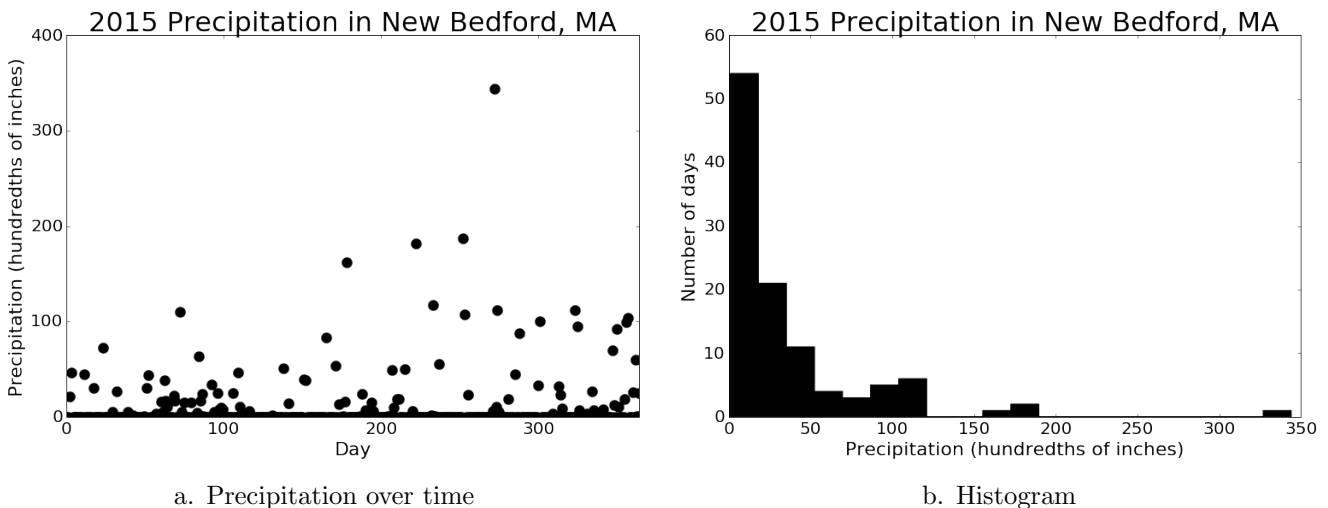a. Precipitation over time   b. Histogram

Figure 6: Precipitation in New Bedford, MA during 2015.

After analyzing plots of rainfall volume over time, we found at times, there appeared to be a lower volume of daily rainfall in the early part of the calendar year, followed by a marked increase towards the middle of the year. We use the idea from Coe et al. of a time-varying μ, but rather than fitting GLMs, we used the observation of this apparent increase in daily rainfall volume somewhere during the middle of the year to create a 'switching point' s that breaks the annual rainfall volume into 2 separate Gamma distributions. We selected a rainy city that exhibited this behavior and focused on the 2015 data set as an example.

Logically, this idea makes sense since we might expect some seasons to produce a greater volume of rain than other seasons. We could also have done four switch points to coincide with the seasons, or 12 switching points to account for each month, but from the plots for New Bedford it appears two is sufficient to describe most of the variation and still retain a reasonable amount of data for the fitting process (there are around 100-120 rain days per year.) This may vary depending on the city and the year.

3.2 Core intuition

The key idea here is that the switching point segments the year so we retain some of the underlying time-based variability over the different seasons. Using pyMC, we can expose this switch point as another variable that we estimate on, and can then produce two separate models for rainfall volume. Compared to a fixed model that looks at the changing point for each season, this has the advantage of accounting for unusually short or long seasons, as is becoming increasingly common in recent years, plus we get uncertainty estimates for the switch point s.

This idea of a switching point is not necessarily going to apply well to all cities in all years; the climate is very variable and the underlying assumptions that exist for this model may not be present. However, as demonstrated here, we believe this technique of creating a switching point could be an interesting avenue used to explore and extend existing rainfall models.

3.3 Results

We fitted a standard Gamma distribution against the daily rainfall volume for the entire year, and then compared that to the results we obtained from fitting a two-period model with a variable switching point. We tried estimating both the shape and rate parameter separately, as well fixing the shape parameter across both models. In order to keep it brief, the results below are for fitting both the switching point, and the shape and rate parameters separately; further results are available in the notebook.

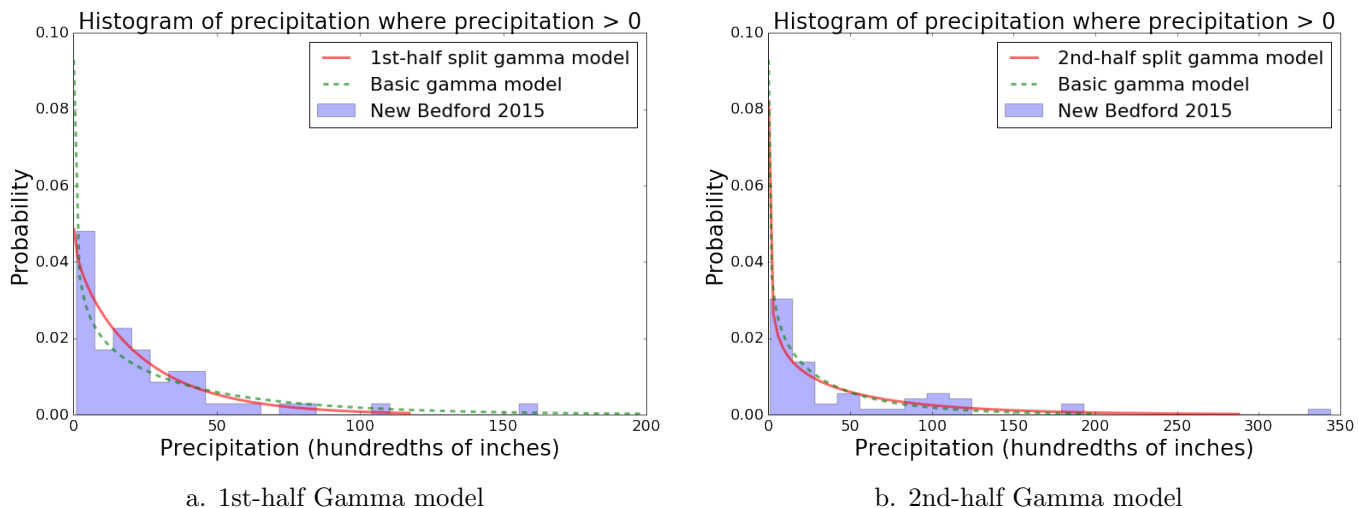a. 1st-half Gamma model b. 2nd-half Gamma model

Figure 7: Fitting a 2-period Gamma model with variable switching point.

As the results show, the Gamma model for the first part of the year does a better job of increasing the likelihood of lower rain volumes while decreasing the likelihood of higher rain volumes. The second part does the opposite, which is what we expect to see since it appears rainfall volume is higher during the second part of the year. So on the whole, we get a greater level of granularity and accuracy from creating this split model versus having a single model for the entire year.

The differences between the μ variable across the two Gamma models appear to be statistically significant, but the difference between the α variable appears to be only weakly significant. So the previous studies on the constant α variable are correct, at least in this instance.

3.4 Next steps

The key idea of the switching point can clearly be extended further, such as creating additional switching points. We can also make the 2-period model more robust by making the start and end points variable so that they do not need to coincide with the start and end of the calendar year; this would require an additional switching point. We might also consider the circumstances under which this model might work and when it might not, for example the geographical location of cities this would apply to, the types of climate in those cities, and which years.

4 Hidden Markov Model

Weather is often very hard to predict through purely statistical methods even when we have a sense of the weather measures. Nevertheless, we have attempted to model weather events based on weather measures using a second order Hidden Markov Model. This then provides a hierarchical framework through which to predict and study weather starting from Bayesian methods to model a specific weather measure like precipitation (as discussed above) all the way through to modeling and predicting specific weather events like Rain or Snow. Specifically, we are interested in seeing how well we can predict extreme weather events or other specific events like Fog days that have an adverse impact on people and society. In a similar study carried out by Betro, Bodini and Cossu [2], it has been demonstrated that HMMs are a reasonable model to attack this problem with the key underlying assumption that there exists discrete weather states corresponding to the hidden states of our model.

4.1 Data

For our study, we have decided to focus on Boston, MA as we have reliable data going back more than 50 years across a wide range of weather measures. We have 19 discrete labels on our data representing weather events that correspond to a condensed 9 hidden states in our model. This mapping was chosen to ensure each hidden state had a minimum number of data points for statistical analysis. The bucketing chosen also ensures that only neighboring events in terms of similarity were bucketed together. For example, Rain-Snow-Thunderstorm, Fog-Rain-Snow-Thunderstorm and Rain-Hail-Thunderstorm were all considered to simply be Thunderstorms. The observed states in our model correspond to 7 continuous real-valued weather measures like Mean Humidity, Mean Wind Speed etc.

The model was trained on 50 years of data (1st Jan 1960 to 31st Dec 2010) and tested on the following five years (Jan 1st 2011 to Dec 31st 2015).

4.2 Model

The architecture of the model which is a second order Hidden Markov Model [cf. Figure 8]. Given the data available, our model does not require an expectation maximization framework to estimate the emission probabilities or transition probabilities. Instead, to model the emission probabilities, we modeled each of the observations as either a Gaussian distribution or a Log-normal distribution. We inferred the corresponding distribution parameters from the historical data. We then constructed a multivariate Gaussian representing the probability of a vector of observations corresponding to a particular hidden state. To model the transition probabilities, we simply counted the state transitions between the discrete weather events available in our labeled data. We also used a naive estimate of the initial state probability for our model.

To infer the weather events on each day over our test period, we ran an extended Viterbi algorithm to infer the most probable sequence of hidden states given the observations over the five year test period. Using the recursive tracing approach, introduced by He [6], evaluating the maximum a posteriori estimation of the second order HMM is easily extendable to higher order models to improve the inference accuracy further.
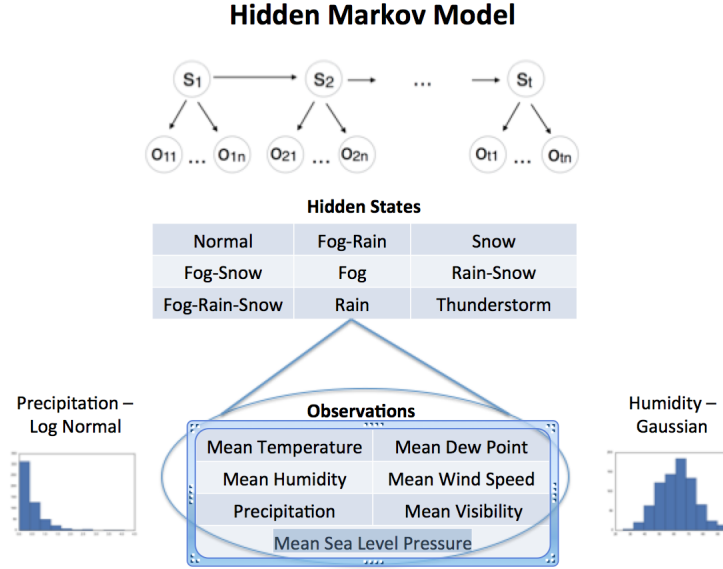
Hidden Markov Model

Figure 8: Second order HMM with 9 hidden states and 7 observations

4.3 Results

Over the five year test period, the model had an overall accuracy of over 65% [cf. Figure 9]. Particularly, the model did well to infer extreme events like thunderstorms and specific events like Fog days as can been seen in Figure 9. The model did less well on highly overlapping events like Rain, Rain-Snow or Fog-Snow. However, still considerably better than a nave baseline. Furthermore, even for the worst performing states, from an end-user standpoint, this is not as bad as it seems. In this problem, there is an inherent ordering of the hidden states, with some states being closer to each other than others. For example, Rain-Snow, Rain and Fog-Rain-Snow are more similar to each other than to Fog days or Normal days which implies that such a misclassification is less impactful than would seem initially. The confusion matrix indicates the same and is discussed with further detail in the accompanying notebook.
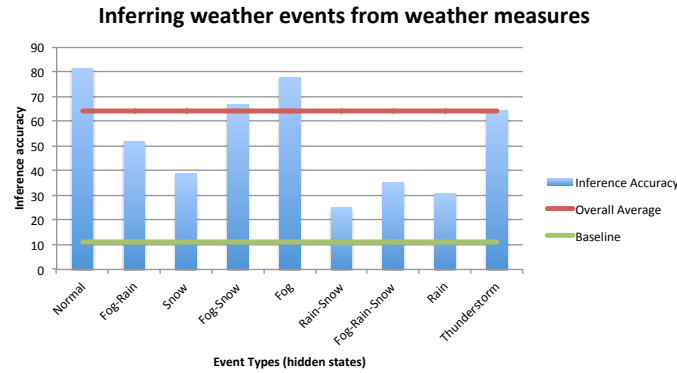

Figure 9: Inference accuracy over the five year test period

4.4   Next Steps

Some extensions to the above model can also be studied further including the stationarity of the transition probabilities over the 50 year training period and a possible third order model for improved inference. Also, as next steps, this model can be further extended by including more observation parameters that would express different aspects of the hidden states or more appropriate parameters like the max/min instead of the mean. Fundamentally, this analysis helps us infer weather events in places/times where we can't directly experience it as in the past/future, or on a different planet for which we only have weather measures.

References

[1] Pierre Ailliot, Denis Allard, Valérie Monbet, and Philippe Naveau. Stochastic weather generators: an overview of weather type models. *Journal de la Société Française de Statistique*, 156(1):101–113, 2015.

[2] Antonella Bodini Bruno Betro and Q.Antonio Cossu. Using a hidden markov model to analyse extreme rainfall events in central-east sardinia. *Environmetrics*, 19(7):702–713, 2008.

[3] Tjerk Adriaan Buishand. *Stochastic modelling of daily rainfall sequences*. Veenman, 1977.

[4] Rafael Cano, Carmen Sordo, and José M. Gutiérrez. *Advances in Bayesian Networks*, chapter Applications of Bayesian Networks in Meteorology, pages 309–328. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004. ISBN 978-3-540-39879-0. doi: 10.1007/978-3-540-39879-0_17. URL http://dx.doi.org/10.1007/978-3-540-39879-0_17.

[5] R Coe and RD Stern. Fitting models to daily rainfall data. *Journal of Applied Meteorology*, 21(7): 1024–1031, 1982.

[6] Yang He. Extended viterbi algorithm for second order hidden markov process. *9th International Conference on Pattern Recognition*, pages 718–720, 1988.

[7] Reimar Hofmann and Volker Tresp. Discovering structure in continuous variables using bayesian networks. In D. S. Touretzky and M. E. Hasselmo, editors, *Advances in Neural Information Processing Systems 8*, pages 500–506. MIT Press, 1996. URL http://papers.nips.cc/paper/1098-discovering-structure-in-continuous-variables-using-bayesian-networks.pdf.

[8] Joseph B. Kruskal. On the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem. *Proceedings of the American Mathematical Society*, 7(1):48–50, February 1956. URL http://www.jstor.org/stable/2033241.