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

Statistcal weather modelling 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 modelling approach we investigate a hidden markov model on weather events like rain, snow or fog. Statistical Inference based on this 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 **2**

2 Bayesian Networks **2**
2.1 Network construction . . . 2
2.1.1 Initial Ordering . . . 3
2.1.2 K2 + Simulated Annealing . . . 3
2.2 Inference . . . 4
2.2.1 Direct Sampling . . . 4
2.2.2 Monte Carlo / Gibbs Sampling . . . 4
2.3 Application . . . 5

3 Hidden Markov Models **5**

1 Introduction

Traditional local and global weather modelling 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 modelling.

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 data is 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 modelling of single weather measures demonstrated along precipitation whereas the second part focuses on a state-based approach for weather events.

2 Bayesian Networks

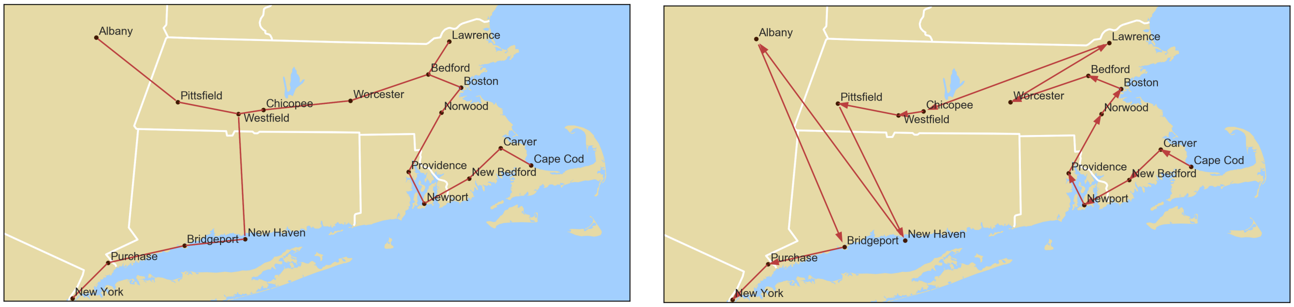
Due to its spatial awareness, a hierarchical structure based on independence is a natural candidate to model a single weather measure. Bayesian Networks have been successfully applied to model precipitation by using a combination of historical weather data and numerical simulation data [2].

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 modelling, 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 farer 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[4]). 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).

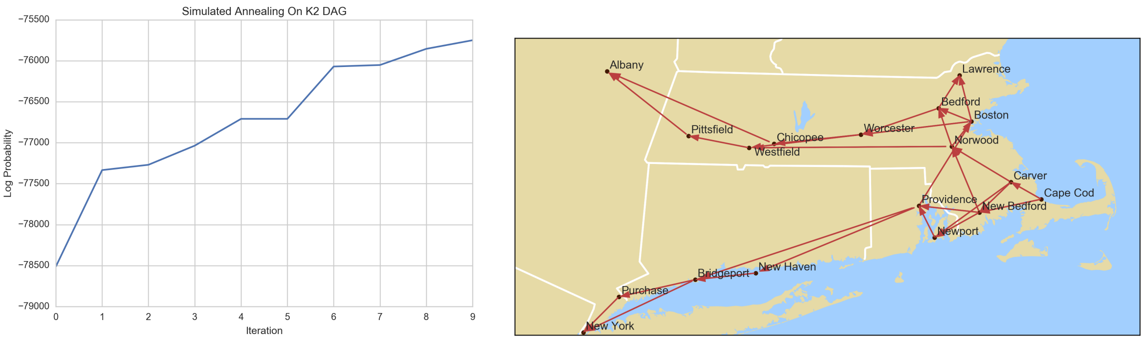
2.1.2 K2 + Simulated Annealing

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:

$$
\begin{aligned}
f(i, \pi_i) &= \prod_{j=1}^{q_i} \frac{(r-1)!}{(N_{ij}+r-1)!} \prod_{k=1}^{r} \alpha_{ijk}! \\
\text{where} & \\
q_i &: |\{\text{Permutations of values for } \pi_i\}| \\
r &: |\{\text{Number of precipitation levels}\}| \\
\alpha_{ijk} &: |\{\text{node i with value } k\text{, parents } \pi_i \text{ with permutation j}\}| \\
N_{ij} &: \sum_{k=1}^{r} \alpha_{ijk}
\end{aligned}
$$

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

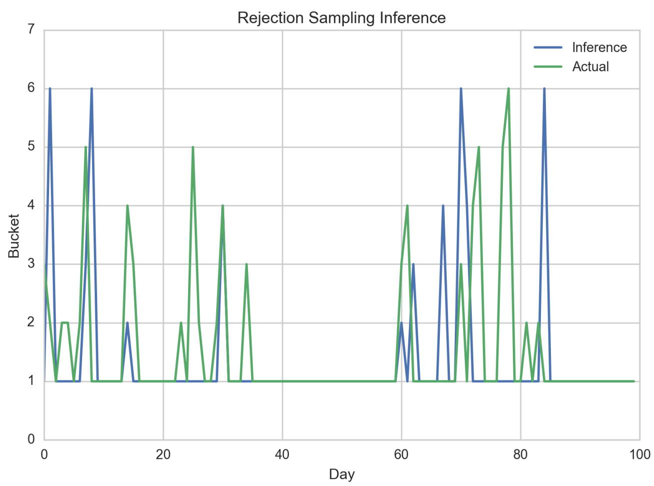
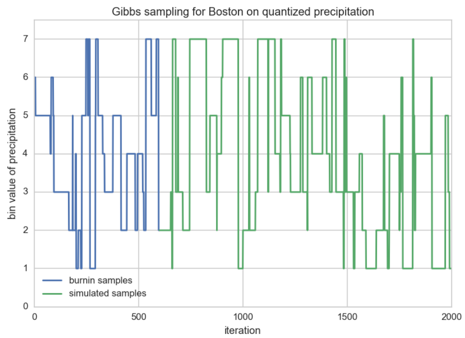


Figure 4: Rejection Sampling For Inferring Boston Precipitation.

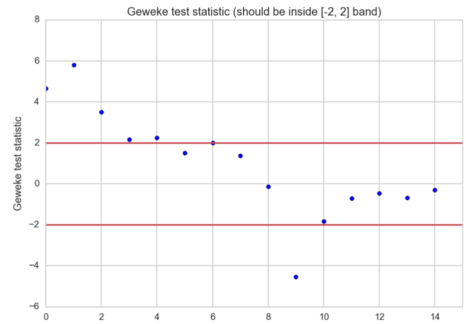
We used rejection sampling as a simple method to infer historical precipitation levels. Using the empirical data, a sample is created by iterating through the node ordering and sampling from the conditional probability distribution based on the parent nodes. We tried to model the precipitation levels 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 $X^{(t)} := \begin{bmatrix} X_1^{(t)} & \dots & X_n^{(t)} \end{bmatrix}^T$ 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 [3]. 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 Application

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 Hidden Markov Models

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] 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`.

[3] 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`.

[4] 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`.