Hurricane Characterization with Common Inferential Statistics and Machine Learning Techniques

Charlie Edelson

April 28, 2017

Abstract

PUT ABSTRACT HERE!!!!!! This is where the abstract will eventually go. However, currently we only have a place holder.

Keywords: PUT KEY WORDS HERE!!!!, these, are, important, words

1 Introduction

UPDATE BACKGROUND HERE!!!! In this section I will talk about hurricanes and how they're very interesting model systems. Additionally, I'll go on to talk about a little bit of the recent history into modeling them and why people use them.

The purpose of this paper is to investigate the properties of named storms in the Atlantic ocean basin using inferential statistics and machine learning techniques. Quadratic and Cubic polynomial regressions will be used to parameterize and model storm windspeed as a function of time, with coefficients binned across storms. ARIMA models will then be fit to each storm, and the most common models will be further investigated. Additionally, the average number of storms per year, λ, will be investigated using Bayesian statistics to get a distribution on probable values of λ. Finally, points in windspeed vs pressure phase space will be clustered using two classical machine learning technique, k means clustering and hierarchical clustering. These results will serve as a starting point for further investigation into storm characterization with statistical methods.

Unisys 2000-2010 Hurricane/Tropical storm data was used throughout this paper[1]. This data consists of timestamped measurements of storm pressure, temperature, windspeed, latitude, and longitude for all major tropical depressions, tropical storms, and hurricanes between 2000 and 2010. Measurements were taken at approximately six hour intervals. Furthermore, storm name is included for all tropical storms and hurricanes.

This data was selected for two reasons. The first is the data is consistent, with few missing time intervals. The second is the data tracks the storms directly, as opposed to buoy data. A researcher using this data will not have to judge where the storm is and which buoys are representative of the parameter. This makes the current study repeatable with the original data sample.

2 Methods

All computational analysis was performed using the python programming language with the Pandas, Scipy, NumPy, Scikit-Learn, and Stats-Models libraries[2, 3, 4, 5, 6]. Graphics and visualizations were created using the Matplotlib and Seaborn libraries[7, 8].

Unnamed storms were removed from the data, as they are not the focus of this study. This leaves 157 named storms over the 10 year period.

2.1 Polynomial Regression

To understand the general shape of the windspeed profile, quadratic and cubic regression of the following forms

$$w_i = \beta_0 + \beta_1 t_i + \beta_2 t_i^2 \tag{1}$$

$$w_i = \beta_0 + \beta_1 t_i + \beta_2 t_i^2 + \beta_3 t_i^3, \tag{2}$$

where w_i is the windspeed at time t_i, were fit to each storm. The resulting coefficients, $\hat{\beta}_i$, were then binned according to values to create empirical distribution.

2.2 ARIMA Modeling

Each storm was cast into a time series and missing time steps were linearly interpolated. The minimum AIC ARIMA was then computed for $p, q, d \leq 2$. The order tuple was recorded and tallied for each unique occurrence.

2.3 Estimation of λ

Since named storms are an example of a poisson process, an estimate of the shape parameter λ, the average number of storms per year, can be computed using Bayesian analysis. The likelihood function for the evidence D given λ would then be

$$P(D|\lambda) = \frac{\lambda^k e^{-n\lambda}}{k!}, \tag{3}$$

where k is the number of storms observed in n years. It is well known that the gamma distribution, $Gamma(\alpha, \beta)$, is a conjugate prior for the poisson distribution, where α and β are the shape and inverse scale parameter, respectively.

References

[1] Unisys, "Hurricane/tropical data." http://weather. unisys.com/hurricane/. Accessed: 04-20-2017.

[2] W. McKinney *et al.*, "Data structures for statistical computing in python," in *Proceedings of the 9th Python in Science Conference*, vol. 445, pp. 51–56, van der Voort S, Millman J, 2010.

[3] E. Jones, T. Oliphant, P. Peterson, *et al.*, "SciPy: Open source scientific tools for Python," 2001–. [Online; accessed ¡today¿].

[4] S. v. d. Walt, S. C. Colbert, and G. Varoquaux, "The numpy array: a structure for efficient numerical computation," *Computing in Science & Engineering*, vol. 13, no. 2, pp. 22–30, 2011.

[5] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[6] J. Seabold and J. Perktold, "Statsmodels: Econometric and statistical modeling with python," in *Proceedings of the 9th Python in Science Conference*, 2010.

[7] J. D. Hunter, "Matplotlib: A 2d graphics environment," *Computing In Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007.

[8] M. Waskom, O. Botvinnik, P. Hobson, J. B. Cole, Y. Halchenko, S. Hoyer, A. Miles, T. Augspurger, T. Yarkoni, T. Megies, L. P. Coelho, D. Wehner, cynddl, E. Ziegler, diego0020, Y. V. Zaytsev, T. Hoppe, S. Seabold, P. Cloud, M. Koskinen, K. Meyer, A. Qalieh, and D. Allan, "seaborn: v0.5.0 (november 2014)," Nov. 2014.