Predictive Analysis of Weather Data via Diffusion Maps, Spectral Clustering, and SOMETHING

Branden Keck, Johns Hopkins University

**Abstract**—Weather analysis

**Index Terms**—Stochastic Differential Equations, Diffusion Maps, Spectral Clustering, Weather, Data

—————————— ◆ ——————————

# 1 Introduction

Weather analysis and forecasting is a discipline that has been at the forefront of human curiosity since the time of the Babylonians. Due to their wide array of applications and importance in daily life, weather forcasting models are continually improved and tested by researchers around the globe [1]. Many of these models are very complicated given the number of variables necessary to make an accurate prediction.

## 1.1 Diffusion Maps

Diffusion maps are a dimensionality reduction technique which is beneficial in data analysis due to its applicability to nonlinear data. Additionally, diffusion maps are commonly used to discover underlying properties of a data set that would not necessarily be revealed by other dimensionality reduction techniques [2]. **More here?**

The first step in implementing a diffusion map is to develop a kernel. The kernel acts as a measure of similarity between two data points and is often denoted as follows [3]:

The measure of “similarity” between two data points can be defined in a variety of ways. A common similarity measure is Euclidean distance in instances where data points can be described in terms of their coordinates in a Euclidean space . However, sometimes overall similarity should account for other data attributes such as color of pixels in an image [4].

The first step in constructing the Diffusion Map is to create an affinity matrix based on the kernel function. Given a set of “n” data points, the affinity matrix will be of size “n by n” with the kernel being calculated for every possible set of points in the dataset. Using the notation proposed by Farbman et. al., the affinity matrix will be denoted as matrix “W” with elements given by:

with xi and xj being data points.

The primary function of the affinity matrix is as a metric for measuring the level of similarity between each data point and the other points in the dataset.

## 1.2 Nyström Approximation

Diffusion Map calculations require an abundant amount of computational resources, especially for large image data sets. It is possible to limit the resource requirements by applying the Nyström approximation to the pairwise affinity matrix, W, instead of explicitly calculating kernel values for each element of the matrix. Several computational pathways for this approximation are presented by Fowlkes et. al. [4], the most basic of these pathways being an estimation of W using smaller matrices. This is written in the form:

Given an affinity matrix of size “n by n”, A is an “m by m” matrix of “sample” affinities and B is an “(n-m) by m” matrix of affinities between the remaining data points and those in the sample. W can then be completed by estimating the remaining portion of the matrix, C, via the following formula [5]:

Thus, a very large problem (“n2“ kernal computations) is reducable to something more manageable (“nm” kernel computations).

## 1.3 K-Means Clustering

[6?]

## 1.4 Predictive Analysis

asdf

# 2 Methods

## 2.1 Data Set

## 2.2 Development of Python Model

A python script was developed to p

The kernel developed for this model is the same as that developed by Farbman et. al. [4]:

*)* (1)

where is a scaling parameter, and are data points.

## 2.3 Efficiency Improvement

Also drawn from the Farbman et. al paper [4], is a well-known efficiency improvement that can be implemented via the Nyström approximation.

Additionally, the ‘numpy’ and ‘tensorflow’ python libraries were used to handle all linear algebra computations. Given

# 3 Results

Additionally

# 4 Conclusion

Although a conclusion may review the main points of the paper, do not replicate the abstract as the conclusion. A conclusion might elaborate on the importance of the work or suggest applications and extensions. Authors are strongly encouraged not to reference multiple figures or tables in the conclusion—these should be referenced in the body of the paper.

**Acknowledgment**

I’d like to thank Ms. Kathryn Tadduni for her continued support and contribution to the overall integrity of the python code.

**References**

1. Steve Graham, et al. “Weather Forecasting Through the Ages.” *NASA, Earth Observatory.* 25 February 2002. Web. 18 August 2018. https://earthobservatory.nasa.gov/Features/WxForecasting/wx2.php
2. J. de la Porte, et. al. “An Introduction to Diffusion Maps.”
3. Ronald R. Coifman and Stephane Lafon. “Diffusion Maps.”
4. Zeev Farbman, et. al. “Diffusion Maps for Edge-Aware Image Editing.”
5. Charless Fowlkes, et al. “Spectral Grouping Using the Nyström Method.” *IEEE Transactions on Pattern Analysis and Machine Intelligence.* February 2004.
6. Mubaris NK. “K-Means Clustering in Python.” 1 October 2017. Web. 16 August 2018. https://mubaris.com/2017/10/01/kmeans-clustering-in-python/



Fig. 1. Magnetization as a function of applied field. Note that “Fig.” is abbreviated. There is a period after the figure number, followed by one space. It is good practice to briefly explain the significance of the figure in the caption.