

# Clustering of Animal Migration Paths

Keith Strickling, 260674699, keith.strickling@mail.mcgill.ca  
Alba Xifra Porxas, 260678352, alba.xifraporxas@mail.mcgill.ca  
Liqiang Ding, 260457392, Liqiang.ding@mail.mcgill.ca

## I. INTRODUCTION

As machine learning grows in popularity and use, more and more interesting applications are discovered. In this paper, we present machine learning based methodologies for clustering animal migration paths both temporally and spatially. The methods we propose are not limited to any given migrating species, however for the sake of demonstration we present the results for their application to turkey vulture migration in North and South America. Turkey vultures were selected as they are known to migrate along several distinct spatial paths, as shown in Fig. 1, making them a good choice for demonstrating the efficacy of our clustering methods, at least as far as spatial clustering goes. In addition, nearly 10 years of migration data is available for the species on the online Movebank database [www.movebank.org].

Our clustering objectives are two-fold. Firstly, we would like to cluster migration paths for many individuals over the course of a single year. Such clustering allows researchers to easily identify different groups within a species on a year-to-year basis. Without ever looking at points plotted on a map, one can answer questions such as: Do all individuals within a species tend to migrate along the same routes or different routes? Do they migrate in groups or individually? Do they migrate at the same time or at different times? Secondly, we wish to cluster the migration paths for a specific individual over the course of many years, thus allowing researchers to trivially discover how the migration of an individual may be changing (or not changing) over many years. Though such questions are rather simple in nature, their answers can have more significant implications, e.g. if a species is forced to alter its migration due to climate change or new human activities in a region.

As a fringe benefit, clustering paths also allows for the easy identification of outliers.

In order to solve these two problems efficiently and accurately, we designed a model which is able to cluster migration paths for multiple animals of one species over multiple years, given the dataset of a particular species. Then, we analyzed the results and plots in a quantitative and qualitative manners, to answers the two questions asked. The model utilizes two clustering methodologies, both centered around dynamic time warping. After discussing related work (Section 2) and our formulation of the clustering problem (Section 3), we outline these methods (Section 3), before applying them to the turkey vulture data and presenting the results (Section 4). Section 5 consists of a discussion of the

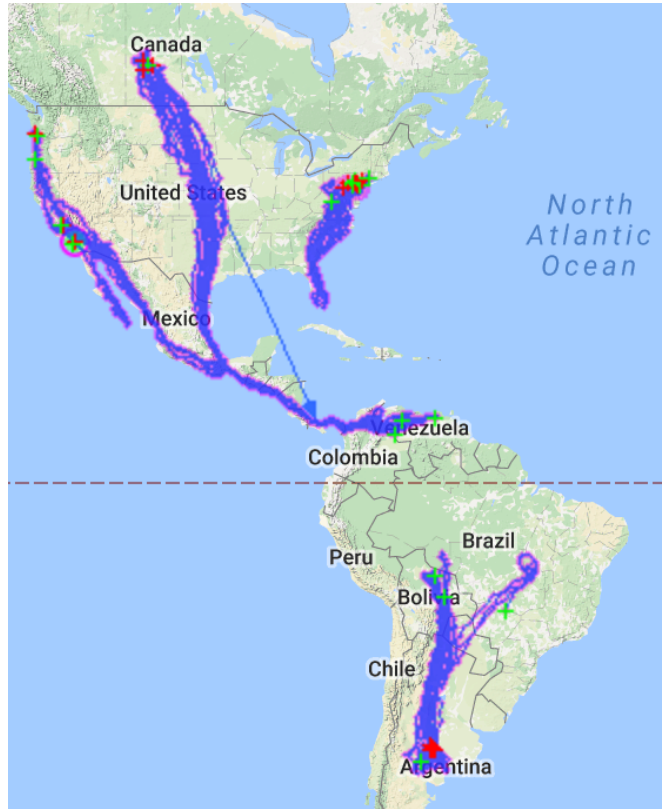


Fig. 1: Turkey vulture movement data from Movebank.

results. Section 6 concludes the report with limitations of our methods and future improvements.

## II. RELATED WORK

Migration routes of animals are recorded as numerous plots in an organized time sequence. In order to cluster their migration paths temporally and spatially and find any possible outliers, we would like to utilize the concept of dynamic time warping method. As far as we know, there have been no other attempts to cluster migration paths in this way.

Dynamic time warping (DTW) is a dynamic programming algorithm capable of detecting similarities between two temporal sequences which may vary in speed [2]. Its advantage over other, more naive, similarity measures such as Euclidean distance is that it is time independent. In other words, if two sequences share similarities in their “shapes” but these similarities occur at different times, dynamic time

warping will still recognize them as similar. In essence, it comes in two forms: full and windowed DTW. In either case, it identifies an optimal point-wise matching between two sequences such as to minimize their “distance” from one another. In full DTW, the entirety of each sequence is scanned for a match, whereas in windowed DTW matches are limited to a window around the current point. Any data which can be modeled as a sequence can be analyzed and modeled by this method. In practice, DTW is perhaps well known for its use in speech recognition, though it has also been applied to video and audio analysis, as well as used in a variety of data mining applications.

In speech recognition studies and applications [1], the use of DTW has been widespread, because the time scales of a sample and a reference pattern in speech are generally not perfectly aligned. In some cases, some simple linear compression and expansion can be performed to align them based on the time scales. Unfortunately, in other cases, a non-linear time warping is required for local compression and expansion of the time scale. Although extra computation is required, the non-linear (DTW) approach undeniably outperforms the linear approach for such series. It was this scenario that inspired the development of dynamic time warping.

We chose to utilize dynamic time warping due to the fact that it is able to accurately perform time series comparisons independent of time, which is precisely what we need in order to cluster migration paths spatially: two individuals of the same species may migrate along similar paths, yet during different time periods. Thus, DTW allows us to compare, cluster and analyze migration time series in a spatial manner, while ignoring the dimension of time. As we will see in a later section, it can also assist us in clustering temporally.

### III. PROBLEM FORMULATION

As mentioned in the introduction section, we wish to be able to cluster both all individuals for any given year as well as all years for any given individual. Doing so provides us a straightforward and intuitive way to visualize how migrations of multiple individuals compare on a year-to-year basis, as well as how migrations of a single individual vary over time. This leads us to the following idea: why don’t we split the data into subsets of data by year and of data by individual and cluster each separately?

However, there are a few limitations of such a method. The first is perhaps best evinced by Figure 2. As the figure depicts, although the turkey vulture data spans approximated 10 years in total, for any given individual we may only have a few years or less worth of data. Thus, even though the dataset contains data for 19 different individuals, at any given point in time we have locations for at most nine of them. This leaves us very few series with which to cluster in any given year, making clustering prone to overfitting (Why? Though two paths may be relatively similar in the universe of possible paths, without an appreciable number of other paths to compare them to we might conclude that they are different). Furthermore, we would be even less inclined to attempt clustering paths for years for which we have data

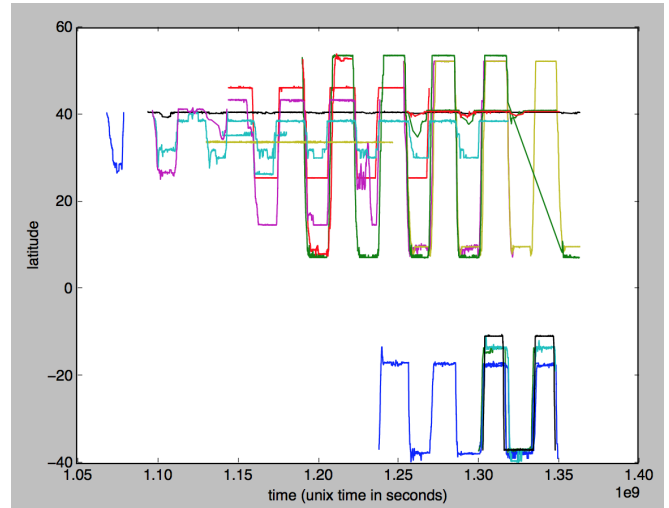


Fig. 2: All turkey vulture data, plotted as latitudinal location vs. time. Vertical sections represent migrations and flat sections represent the periods in between.

for even fewer individual, resulting in the discarding of data for these years - this is a significant waste of our already-limited data resource! As for clustering migration paths for a single individual over many years, for any given turkey vulture we have at most six years worth of data. Again, this leaves us prone to overfitting. Secondly, the method lacks in simplicity. It results in many different clusters for many different individuals and years, making comparing any two or more sets of clusters inconvenient and inefficient.

Therefore, we “migrated” the two methods into one and designed a single method to meet our two objectives simultaneously. The method works as follows. For each individual, partition its data points into years. For example, if one turkey vulture has three years’ worth of migration data, the method would divide this data into three separate time series. Take all resulting time series and cluster them.

The result is a simple and efficient method that wastes no data and is able to cluster both horizontally (multiple years of a certain animal) and vertically (multiple animals over one year) at once. The following section provides greater detail on this method of treating the data, and introduces the clustering methods used.

### IV. METHODOLOGY

Our clustering methodologies require that the data is formatted in a specific way before they can be applied. This section details the pre-processing procedure, before discussing the distance metrics used and the two clustering algorithms

#### A. Data pre-processing

As discussed in the previous section, we cluster all years for all individuals together. Thus, pre-processing is, in essence, simply a splitting of the data for each individual by year, and pooling all the resulting time series together to be fed to the clustering algorithms. Because for some years we

may only have points spanning a small time range (perhaps just a month or two), and this is not enough time to be sure to capture a full migration, we check that the interval spanned by the points for a given year is at least eight months in duration. We throw out the time series for any year and individual for which we do not have points spanning at least eight months. For the turkey vulture dataset, this leaves us with a total of 53 time series to cluster, including time series for 18 out of the 19 individuals.

### B. Interpolation

k-means clustering, as we will see, uses time series as cluster centroids, rather than individual points. In order for this to work, a centroid must have the same number of data points as each of the time series being clustered, and hence each time series must have the same number of data points. Thus, for each time series we interpolate at a constant time interval in order to achieve an equal number of data points in each series. For our dataset, we chose to interpolate at a three-day interval. Seeing as we used year-long time series, this resulted in each series consisting of 122 data points. You may recall that we kept a time series as long as it had points for at least eight months out of a year. Hence, a given time series did not necessarily span a full twelve months, making it impossible to interpolate for the entire series. However, by examining the data we noticed that if points ceased to be recorded for a given year, it meant that the individual had already completed its migration cycle for the year. Thus, for any times for which we could not interpolate, we assumed the individual was at either end of its migration path, and extrapolated with the closest point in time that we *did* have for the individual.

Note that, in addition to being necessary for k-means clustering, interpolating at a constant time interval allows us to compare two paths temporally, even if they were taken during different years. For example, as we interpolated at three-day intervals, we know that the first point in each time series corresponds to January 1<sup>st</sup>, the second to January 4<sup>th</sup>, the third to January 7<sup>th</sup>, etc., all irrespective of the year.

### C. Distance metrics

As discussed in the Related Work section, dynamic time warping permits the detection of similarities in temporal sequences regardless of their temporal differences. Hence, it is the clear choice for clustering paths spatially. However, we also wish to cluster temporally. How can we do this? As we've already made sure that each time series has the same number of points, and the  $n^{\text{th}}$  point in each series correspond as to the same relative time in any given year, one option is to use Euclidean distance as a measure of the temporal difference between two sequences deemed spatially similar. Note we must first determine that two series are spatially similar in order for Euclidean distance to be a good indicator of temporal similarity, i.e. if two series are spatially dissimilar, then the Euclidean distance between them will be large irrespective of any temporal similarities or differences. A drawback of Euclidean distance, however, is that it the

distance between two series will be appreciable even if their points are misaligned by only a small time shift. For example, if two vultures follow migrate along precisely the same path at precisely the same speed, but one leaves a week after the other, the Euclidean distance between the two paths may be large, even though the two paths are temporally quite similar. In order to capture small, yet acceptable, temporal differences such as this example, we can again use DTW. However, rather than full DTW, we can use windowed DTW - in the above example, using a window of one week would result in a dynamic time warping distance of zero between the two paths, which is what we want.

Thus, we first use full dynamic time warping to cluster series spatially. Once spatial clusters are established, we can use windowed DTW to cluster temporally within each spatial cluster. For our dataset, we use a window of five data points. At a three-day interval, this corresponds to a window size of 15 days. Hence, we consider two spatially similar paths to also be temporally similar as long as they differ in time by at most two weeks - a fairly arbitrary choice but useful for demonstration.

### D. Clustering approach: k-means

k-means clustering is perhaps the most common clustering method due to its intuitive and simple nature. Thus, we use it as our first clustering methodology. The only moderately interesting aspect of this approach, as it applies to time series, is that each cluster centroid is a time series of its own, rather than a single point. To learn  $n$  clusters in the data, we randomly initialize  $n$  centroidal time series by selecting  $n$  of the time series being clustered as initial centroids. We then use DTW to get the distance between a given time series and each centroidal time series, and cluster the time series with the centroid that it is closest to. We repeat for all time series. Once all time series have been clustered, we update each cluster's centroid as the average of all time series in the cluster. We then re-cluster each time series according to the new centroids, and repeat until convergence or oscillation, returning the final cluster assignments. We measure the error of a set of assignments as the average distance between each time series and the centroid it is clustered with.

Yet, how do we know what the best choice of  $n$  is? In clustering, this is a bit of a tricky task because we know that the more clusters we use, the lower the error will be. In particular, if we have  $k$  time series, then the clustering error will be zero if we use  $k$  clusters. But clearly we gain no useful insight into the data at hand in doing this. Thus, we try a range of number of clusters,  $n$ , and select the number of clusters at which the error begins to plateau.

The final convergence (or oscillation) of centroids, and thus the assignments for a given number of clusters, is contingent upon the initialization of the centroids. To account for this variability, we use random restarts for each candidate value of  $n$ , and take the error for the given  $n$  to be the average assignment error over all random restarts. As full dynamic time warping is relatively slow,  $O(m^2)$  where  $m$  is the size

of the input series, we limit the number of random restarts to three. Thus, this is three-fold cross-validation.

In all, we perform the following steps:

- 1) For each candidate  $n$ , we cluster all time series using full DTW three times. We take the average assignment error over these three restarts as the error when using  $n$  clusters.
- 2) We examine the errors and select the value of  $n$  when the error begins to plateau. Call this value  $n^*$ .
- 3) For  $n^*$ , we retrieve the best (lowest error-yielding) assignment of the three random restarts used. This constitutes the “best” spatial clustering.
- 4) For each of the  $n^*$  clusters, we repeat steps 1-3 to obtain the temporal clusters. However, this time we use windowed DTW and a range of  $n$  from 1 to the number of time series in the cluster.

#### E. Clustering approach: hierarchical clustering

The second method we implemented is hierarchical clustering. We apply an agglomerative approach, in which each example starts in its own cluster (in our case a migration path for a given animal and year). Subsequently, as you move up the tree, clusters are combined together in pairs. Finally, when only one cluster remains, this cluster becomes the root of the tree. The final rooted tree is usually called a dendrogram. It is a diagram that illustrates the cluster arrangement constructed by hierarchical clustering.

Hierarchical clustering uses as input a similarity matrix. Dynamic time warping is used as the metric of distance to measure similarity between clusters. Regarding the linkage criteria, two clusters are combined together into a higher-level cluster using the UPGMA method [3]. At each clustering step, this method computes the distance between two clusters by taking the average of all distances between examples of the two clusters that will potentially be merged.

Concerning the choice of how to select the number of clusters  $n$ , there is no gold standard procedure. We decided to use the silhouette score, which is computed as follows:

$$\frac{b - a}{\max(a, b)}$$

where  $a$  is the average intra-cluster distance, and  $b$  is the average distance to the nearest cluster for each example within a cluster. The highest silhouette score is 1 and the lowest is -1. Values around 0 indicate overlapping clusters.

We used the Scipy package for hierarchical clustering to implement this method (`scipy.cluster.hierarchy`) [4]. We computed the silhouette score using Scikit Learn’s implementation of it (`sklearn.metrics.silhouette_score`) [5].

## V. RESULTS

We applied our methodology to the dataset: “Turkey vultures in North and South America” [6] [7]. This dataset includes migrations from 19 birds within 2005 and 2012. However, data is not available for all years from all individuals.

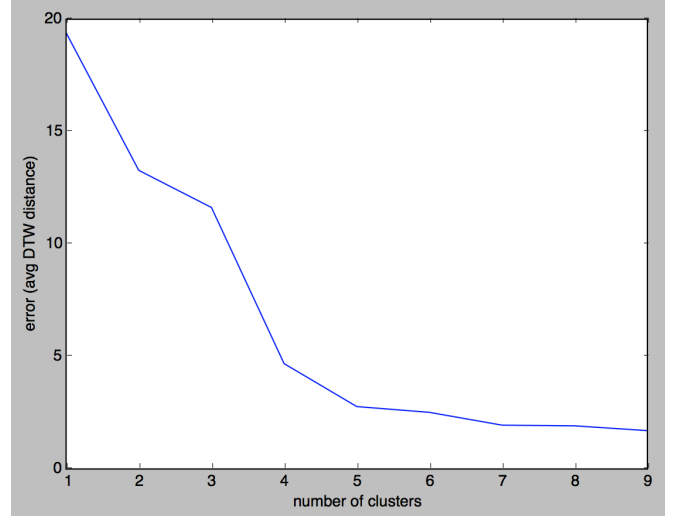


Fig. 3: Average DTW error vs. number of cluster for spatial clustering.

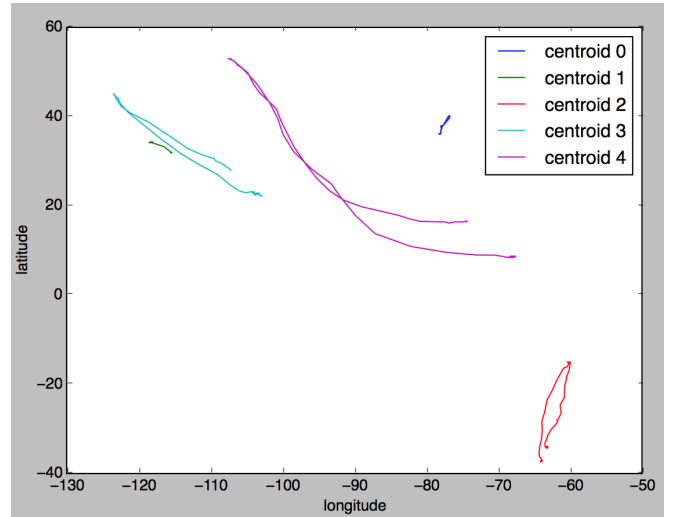


Fig. 4: Centroids for five clusters.

#### A. k-means clustering

The first task, as described in the previous section, is to cluster all paths spatially. After visual inspection of all the data points (see Fig. 1), there are ostensibly four or five spatial clusters. However, we want the algorithm to discover this for us, and perhaps even prove us wrong. Thus, we attempted clustering with one to nine clusters using random restarts, and recorded the errors for each. The results are displayed in Fig. 3. As seen, the error effectively plateaus at five clusters, hence we select five clusters as the optimal number. The centroids for these five clusters are plotted in Fig. 4, and the clusters themselves are plotted in Fig. 5.

The spatial clusters have the following sizes (in number of time series per cluster):

- Cluster 0: 21
- Cluster 1: 4
- Cluster 2: 11

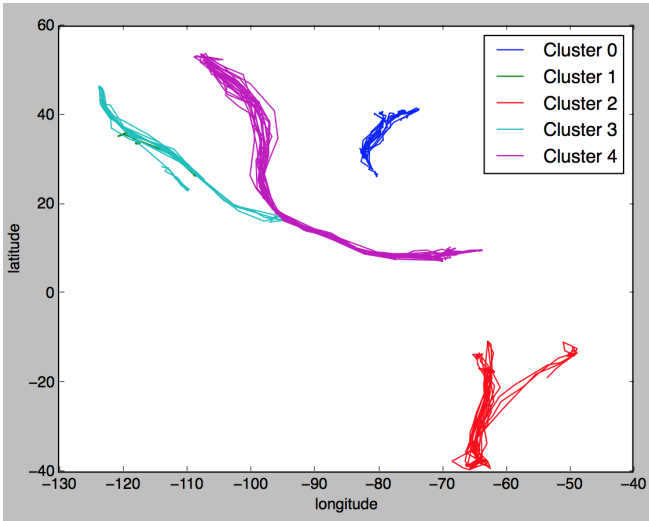


Fig. 5: Clusters.

- Cluster 3: 7
- Cluster 4: 10

For each spatial cluster, we then cluster temporally using windowed DTW, with a window size of five points (15 days). For each spatial cluster, we try using 1 to *size-of-cluster* temporal clusters. Clearly, when using *size-of-cluster* temporal clusters, error will be zero. Hence, the best number of clusters must occur somewhere in this range. Error as a function of the number of temporal clusters is plotted in Fig. 6, for each of the five spatial clusters. For spatial clusters 0 and 1, there is a large drop in clustering error when using two temporal clusters as opposed to one. This suggests that spatial clusters 0 and 1 can each be further divided into two temporal clusters. The other three clusters are less obvious, but we could argue for three temporal clusters of spatial cluster 3 and two temporal clusters of spatial cluster 4, as we also see appreciable error drops up to these divisions. Temporal division of spatial cluster 2 is by far the most unclear, but using an error threshold of 1.5, which is fairly low, we can argue for two temporal clusters of spatial cluster 2 as well. The final clustering results are given in the Appendix.

To further examine temporal clustering, we take plot two paths that are in the same spatial and temporal cluster, followed by two paths which are in the same spatial, yet different temporal, clusters. The results are given in Figs. 7 and 8. As we can see in Fig. 7, both series are rather similar both spatially and temporally, as expected. Fig. 8 is not exactly what we expect, however. The two paths were clustered in the same spatial cluster, yet spatially they look somewhat different. One (Disney\_2007) includes a migration, while the other does not. Observing the axes, however, we see that the the distance covered is relatively small, explaining why these two series were placed in the same spatial cluster. There is no obvious justification for different temporal clustering of these two series - we would expect two temporally different series to be time-shifted away from

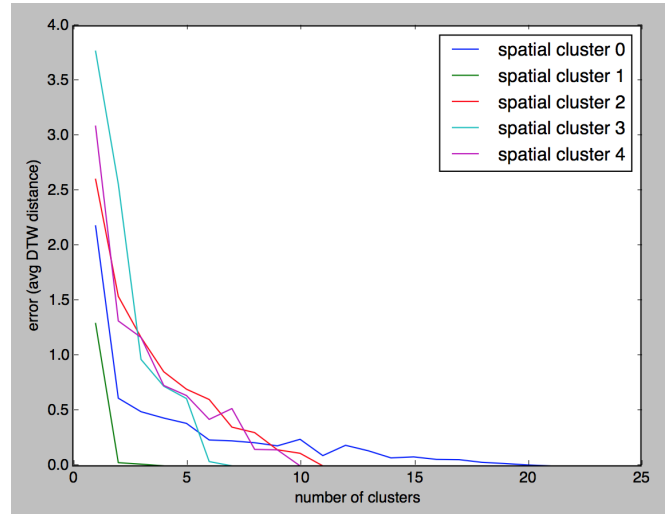


Fig. 6: Error vs. number of clusters for temporal clustering of each spatial cluster.

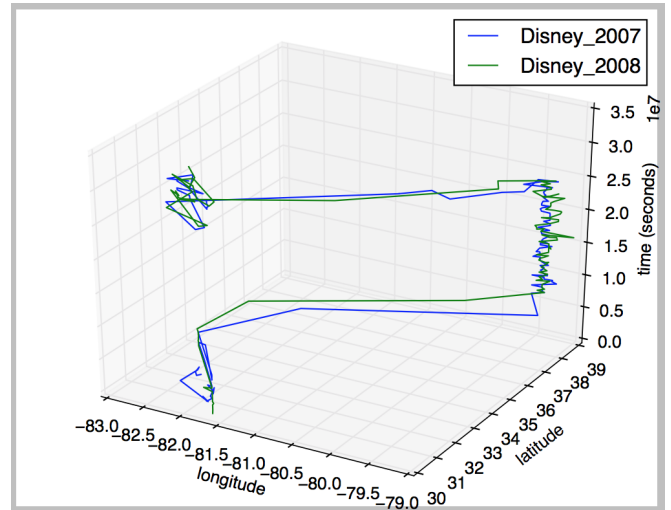


Fig. 7: Two time series in the same spatial and temporal cluster.

each other (i.e. one above the other in the figure). It seems the windowed DTW may have been influenced by spatial differences in this example, rather than just temporal ones.

### B. Hierarchical clustering

As explained in the methodology section, first we computed the similarity matrix using dynamic time warping. The result is illustrated in Fig. 9. It can be observed already in the similarity matrix that clusters exist. Further, we applied hierarchical clustering using the similarity matrix as input. The constructed dendrogram is shown in Fig. 10. The calculated silhouette score as the number of clusters increases can be seen in Fig. 11. As expected, the score reaches its maximum when there are as many clusters as examples. However, there is a local maximum when the number of clusters is 4 which has a relatively high silhouette score: 0.8. Hence, the algorithm selected four as the optimal number of clusters.



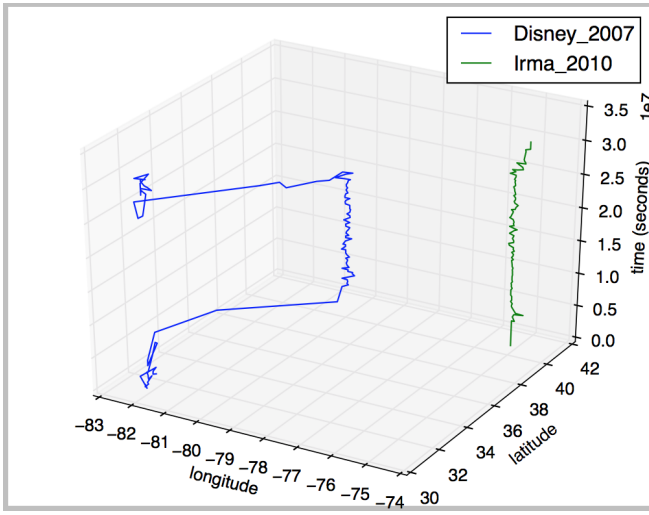


Fig. 8: Two time series in the same spatial cluster but different temporal clusters.

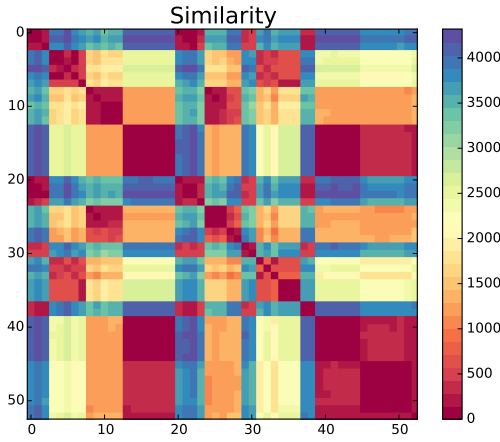


Fig. 9: Similarity matrix computed using dynamic time warping.

The clusters have the following number of time series per cluster: 11, 11, 21, and 10. As can be appreciated, migration paths from the same animal but from different years are clustered together. We later clustered the migration paths temporally, which leads to the same clusters. In other words, there are no temporal differences found, hence each spatial cluster is also a temporal cluster. This result is plausible, since turkey vultures tend to migrate in groups.

## VI. DISCUSSION

We have shown the ability of two methods to cluster animal migration paths. Both clustering approaches led to similar results. The best  $n$  (number of clusters) differed slightly: k-means chose 5 whereas hierarchical clustering chose 4. However, the spatial clustering results were identical if we used the same  $n$  between both methodologies. Temporal clustering results were somewhat different - k-means clustering found each spatial clusters could be partitioned

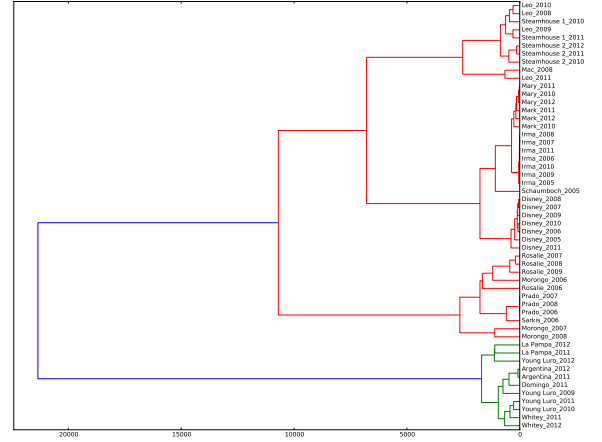


Fig. 10: Dendrogram constructed using hierarchical clustering.

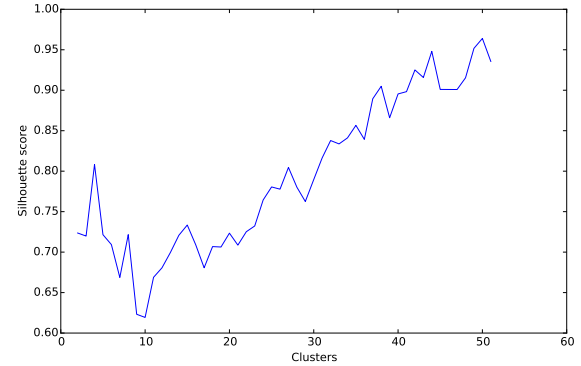


Fig. 11: Silhouette score as the number of clusters increases.

into two or three temporal clusters, while hierarchical clustering found the spatial and temporal clusters to be equivalent. Realistically, these results are not too far off as one temporal cluster per spatial cluster vs. two or three is not a substantial difference. Moreover, although the windowed DTW used to cluster temporally in the k-means algorithm suggested the existence of temporal clusters, it was not overwhelmingly clear that one temporal cluster per spatial cluster was insufficient, and making this decision was difficult given the monotonically decreasing nature of the error with an increasing number of clusters.

One advantage of hierarchical clustering is that it only requires a matrix of pairwise similarity measures in order to construct the tree. In other words, once the similarity matrix is computed, the individual time series are no longer needed. Conversely, k-means clustering must re-use each time series in each iteration of the algorithm, in order to update the centroids. Hierarchical clustering also does not require the number of clusters as input nor initialization. Further, it provides different levels of partitioning, which

allows you to select the level of resolution. Although agglomerative clustering is computationally expensive for large datasets ( $O(n^2 \log(n))$ ), it is still more efficient than k-means clustering which requires repeated application of DTW to each possible time series and centroid combination in each iteration (making it  $O(pm^3n)$ , where  $m$  is the size of the time series, an  $p$  is the maximum number of iterations allowed. Thus, hierarchical clustering is the winner in terms of running time and efficiency.

Additionally, hierarchical clustering is more robust to outliers, since those examples can significantly affect the mean of a cluster using k-means. Both, however, are equally capable of identifying outliers. In hierarchical clustering, an outlier is a series whose parent is relatively high in the dendrogram. Meanwhile, k-means clustering will discover outliers as relatively small or single-series clusters.

This does not leave k-means clustering without its advantages, however. With hierarchical clustering, when a new example arrives, ideally the hierarchical tree should be computed again. This may be time consuming, especially if one is interested in online tracking and clustering in real time. On the other hand, having a centroid time series is advantageous to cluster future queries. Thus, k-means clustering is more convenient if one is interested in clustering future migration paths - i.e. it is better for the related classification problem: given a new series, classify it according to the cluster of series it is closest to.

Regarding the choice of the number of clusters, both methods as they are presented here will not necessarily find the optimal configuration. However, k-means clustering pursues a less heuristic procedure, which may lead to a more reliable estimate. Finally, both methods have proven useful to obtain exploratory results of animal migration paths.

We thus recommend using hierarchical clustering if training time is a limitation and if outliers are a concern. However, if one wishes to classify time series that they don't yet have but will obtain at some point in the future, k-means clustering is the better method. k-means clustering is also a better choice for online learning.

#### A. Future Improvements

While the results obtained using both methodologies were consistent and revealed useful information about the underlying data, there are certainly a few improvements that can be made in order to improve performance further.

Firstly, our choice to split the data for each individual by year was fairly naive. Clearly, as we are clustering migration paths, it makes more sense to split the data by migrations. In a typical year, a migrating individual makes one northward and one southward migration. For example, the vultures mainly migrate from October to November and from March to May each year. Thus, it would be fruitful to parse out each of these two migrations per individual per year, and then cluster all the northward and all the southward migrations separately. Revisiting Fig. 8, we see that one of the two paths contains two migrations, while the other does not. Nevertheless, these two paths were clustered the

same spatially - although they are fairly close in latitude and longitude, they would be less similar without the extraneous data between the two migration periods.

We limited ourselves to splitting by year for the purposes of this report, as learning when an individual begins and ends a migration and splitting the data at the right times could be formulated as an interesting machine learning problem on its own - we leave this for future work. Additionally, by partitioning the migration data into southward and northward migrations, we create the opportunity to learn even more interesting information about a species. It is highly possible that a migrating species takes slightly different paths in either direction due to some external factors, such as climate change or human activity. By clustering this way, one could gain a much better understanding of the overall migration behavior of the species of interest.

Secondly, temporal clustering needs improvement, as it did not produce clear divisions in the spatial clusters when used with k-means clustering. This may have been due to the specific data used, or due to the distance function (windowed DTW), or a combination of the two. Further investigation is needed, and other distance functions should be tried.

Lastly, the task of selecting the best number of clusters is difficult, particularly in the k-means clustering method, since clustering error monotonically decreases as the number of clusters is increased. In the methods presented in this report, we relied on heuristics to select the best number of clusters, but this approach lacks in robustness. One possible solution would be to create a supervised learning task that would learn how to select the optimal number of clusters from error-vs-number-of-clusters data. By collecting this data for a large number of species and labeling each dataset with the optimal number of clusters, we could train a learner to select the optimal number of clusters without relying on heuristics.

## VII. STATEMENT OF CONTRIBUTIONS

Liqiang worked on searching and collecting the dataset from the MoveBank Database. Also, he helped visualize the dataset by plotting individual migration paths of turkey vultures. Keith wrote the code for data pre-processing and k-means clustering, and wrote the corresponding sections of the report. Alba implemented the hierarchical clustering, and wrote the corresponding sections of the report. We all discussed the methodology during several meetings. We hereby state that all the work presented in this report is that of the authors.

## REFERENCES

- [1] C. Myers, L. Rabiner and A. Rosenberg, "Performance tradeoffs in dynamic time warping algorithms for isolated word recognition", *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 28, no. 6, pp. 623-635, 1980.
- [2] T. Vintsyuk, "Speech discrimination by dynamic programming", *Cybernetics*, vol. 4, no. 1, pp. 52-57, 1972.
- [3] Sokal, RR and Michener, CD. A statistical method for evaluating systematic relationships. *Scientific Bulletin*. 38(22): pp. 140938. 1958.
- [4] Jones E, Oliphant E, Peterson P, et al. *SciPy: Open Source Scientific Tools for Python*, 2001-, <http://www.scipy.org/> [Online; accessed 2016-11-24].

- [5] Pedregosa, F. et al., Scikit-learn: Machine Learning in Python, JMLR 12, pp. 2825-2830, 2011.
- [6] Dodge S, Bohrer G, Bildstein K, Davidson SC, Weinzierl R, Mecharh MJ, Barber D, Kays R, Brandes D, Han J (2014) Environmental drivers of variability in the movement ecology of turkey vultures (*Cathartes aura*) in North and South America. *Philosophical Transactions of the Royal Society B* 20130195. doi:10.1098/rstb.2013.0195
- [7] Bildstein K, Barber D, Becharh MJ (2014) Data from: Environmental drivers of variability in the movement ecology of turkey vultures (*Cathartes aura*) in North and South America. Movebank Data Repository. doi:10.5441/001/1.46ft1k05

## APPENDIX

Final k-means clustering output:

- Spatial cluster 0
  - Temporal cluster 0
    - \* Disney\_2005, Disney\_2006, Disney\_2007, Disney\_2008, Disney\_2009, Disney\_2010, Disney\_2011
  - Temporal cluster 1
    - \* Irma\_2005, Irma\_2006, Irma\_2007, Irma\_2008, Irma\_2009, Irma\_2010, Irma\_2011, Mark\_2010, Mark\_2011, Mark\_2012, Mary\_2010, Mary\_2011, Mary\_2012, Schaumboch\_2005
- Spatial cluster 1
  - Temporal cluster 0
    - \* Sarkis\_2006
  - Temporal cluster 1
    - \* Prado\_2008, Prado\_2006, Prado\_2007
- Spatial cluster 3
  - Temporal cluster 0
    - \* La Pampa\_2011, La Pampa\_2012
  - Temporal cluster 1
    - \* Whitey\_2011, Whitey\_2012, Domingo\_2011, Young Luro\_2009, Young Luro\_2010, Young Luro\_2011, Young Luro\_2012, Argentina\_2011, Argentina\_2012
- Spatial cluster 3
  - Temporal cluster 0
    - \* Morongo\_2008, Morongo\_2007
  - Temporal cluster 1
    - \* Rosalie\_2008, Rosalie\_2009, Rosalie\_2007
  - Temporal cluster 2
    - \* Rosalie\_2006, Morongo\_2006
- Spatial cluster 4
  - Temporal cluster 0 Steamhouse 1\_2010, Steamhouse 1\_2011, Steamhouse 2\_2010, Steamhouse 2\_2011, Steamhouse 2\_2012, Leo\_2008, Leo\_2009, Leo\_2010
  - Temporal cluster 1 Leo\_2011, Mac\_2008