Chapter 11 - Spatial Outlier Detection

1 Introduction

Spatial data measures how some behavioral attribute (e.g. wind speed) varies over a contextual attribute (e.g. latitude and longitude). Sometimes, a spatial quantity, like latitude/longitude, can be the behavioral attribute and time is the contextual attribute (this is called trajectory analysis). Sometimes, we have space AND time as contextual attributes, which we call spatiotemporal data.

We leverage two characteristics in our data. First is spatial autocorrelation, which simply says nearby spaces are more similar than distant ones. Second is spatial heteroscedasticity, which says the variance of the behavioral attribute depends on its location in space. The former is more used than the latter. We can find contextual outliers (points in space with weird behavior compared to their surroundings) and collective outliers (weird patterns that can cover many points in space). Supervision is useful here.

The key difference between spatial data and time series data is that spatial is multidimensional and not unidirectional. Also, space may be discrete (e.g. zip code).

2 Spatial Attributes are Contextual

Neighborhood algorithms look for abrupt changes in a neighborhood. The neighborhoods can be defined via distances or via a edges in a graph.

You could use LOF, but that doesn't incorporate the difference between the behavioral and contextual attributes. For example, k nearest neighbors in spatial data must just look for points nearby in space. We can find the nearest neighbors and average (or take median) of their behavioral values. If the point in question has very different behavioral attributes than the neighborhood, it is an outlier. You should normalize the difference by the standard deviation of the neighborhood behavioral attributes.

We may represent space with a graph. Let o be our point, $o_1, ..., o_k$ be the k-nearest neighbors, $w(o, o_i)$ be the edge weight between o and o_i, and $f(o)$ be one of the behavioral attributes. We can compute a neighborhood mean with $g(o) = \frac{\sum_{i=1}^{k} w(o,o_i)f(o_i)}{\sum_{i=1}^{k} w(o,o_i)}$. You can then do extreme value analysis on the normalized difference of each point to its neighborhood mean as described above. If you have multiple behavioral attributes, compute the normalized differences independently, use the Mahalanobis method, and then do extreme values on the Mahalanobis outlier scores.

An autoregressive model assumes space is represented with coordinates in a grid and that most of space has behavioral attributes (i.e. not sparse). Our model is then $X_{t_1,t_2} = \sum_{i=-p}^{p} \sum_{j=-p}^{p} a_{ij} X_{t_1-i,t_2-j} + c + \epsilon_{t_1,t_2}$ where we force $a_{00} = 0$ to avoid using a value to predict itself. You can solve this with least squares. We have ARMA and ARIMA variants of this like we did with time series. Autoregressive models are not used often in the literature because data is usually sparse.

Consider a pair of points and measure the distance between their spatial attributes and measure another distance between their behavioral attributes. If you plot behavioral distance vs. spatial distance for all pairs of points, you get a variagram cloud. High behavioral distance with small spatial distance represents an outlier. Processing all pairs of points is expensive, so you can simply discretize space into a grid and make each grid cell a point.

How do we detect anomalous shapes in spatial data? One option is contour detection. A contour is a boundary of a shape. Notice that this can be turned into a time series if the x-axis is given by a clockwise sweep of the contour and the y-axis is the distance from the centroid. If you do not normalize the time series, that means you care about how large the contour is. If you scale to unit mean, that means you do not care about size but you do care about relative local variation. If you do zero-mean and unit variance, it means you do not care about size or relative local variation (circular shapes can cause problems here). None of these account for rotation, however, which shifts the time series. To account for rotation, we can use the rotation invariant Euclidean distance between two time series $RIDist(T_1, T_2) = \min_{i=1} n \sum_{j=1}^{n} (a_j - b_{1+(j+i)\mathrm{mod}_n})^2$. With this in hand, an outliers can be detected with the k-nearest neighbors method (use early inner loop termination to avoid doing too much computation).

You can also split space into $p \times p$ grids, turn each grid into a vector, and then use multidimensional methods for outlier detection. You can also use the Haar wavelet transform to turn the grids into features.

Supervision helps a lot. Just learn some shapes from the normal class (and the outlier class, if you have it) and then see if the test shape is more similar to the normal class or outlier class (using the shape to time series method discussed earlier).

3 Spatiotemporal Outliers with Spatial and Temporal Context

You can extend previous approaches to spatiotemporal data. Just define the neighborhood to span time as well (be careful about how you normalize here so you can compare across both space and time).

4 Spatial Behavior with Temporal Context: Trajectories

You can think of this as multiple time series and use the time series analysis methods. You can use kernel methods to make a similarity matrix, use the Mahalanobis method on the matrix, and then do extreme value analysis on the outlier scores.

In a streaming situation, you can process each time series separately. Then you can sum their outlier scores (assumed to come from a normal distribution) and model it with a χ^2 distribution.

For unusual shape detection, ignore the time dimension and use some of the techniques described earlier. Another option is to break a trajectory into pieces and classify each piece into one of a set of fixed trajectories (the symbols). Now you get a symbol frequency vector and you can find k-nearest neighbors. This also lets you turn the trajectory into a discrete sequence and you can use the techniques in chapter 10 for this.

You may have some supervised trajectories. Turning these trajectories into a sequence of symbols and then using a supervised discrete sequence technique is helpful here. If you do not know good symbols beforehand, you can do clustering to infer them.

5 Conclusions and Summary

Spatial outlier dimension is similar to temporal outlier detection (time series). We can also handle spatiotemporal data and trajectories. This chapter also showed how to convert a shape into a time series.