Time series clustering and the analysis of film style

Nick Redfern

Introduction

Time series clustering provides a simple solution to the problem of searching a database containing time series data – such as the Cinemetrics database – for those series that display similar behaviour. In undertaking such a search we move away from the conception of statistical analysis of film style used to date and into *data mining*. Cluster analysis is used to segment large databases into homogenous groups and to visualize the structure of large databases in order to aid analysts in identifying meaningful groups and subgroups based on the similarities between data objects (Jain, Murty, & Flynn 1999).[1]

In this paper I describe a simple method for applying time series clustering to the shot length data of motion pictures based on the standardized shot density for a normalized point process. In the next section I provide a brief overview of cluster analysis. Shot length data must be prepared prior to cluster analysis and in the third section I describe a simple method of normalizing and standardizing shot length data based on the kernel density of the point process of cuts in a motion picture. Finally, I demonstrate the application of time series clustering to the analysis of shot length data.

Cluster analysis

Cluster analysis is the process of sorting a set of unlabeled data sets into groups so that the within-group similarity is minimized and the between-group dissimilarity is maximized. The groups are not predefined and their meaning must be interpreted once constructed. Cluster analysis is therefore part of exploratory data analysis.

Cluster analysis is applied to an $n \times p$ matrix

$$\mathbf{X} = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,p} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n,1} & x_{n,2} & \cdots & x_{n,p} \end{bmatrix},$$

where the n rows are the data objects to which we wish to apply cluster analysis and the p columns are the variables. The entry $x_{i,j}$ in $\mathbf{X}$ is the value of the *j*th variable for the *i*th data object. In this paper the data objects are time series of shot length data and so each row is an individual film; while the variables are the values taken by the time series of a film at a given point (after pre-processing the data – see below).

The first stage in cluster analysis is determining the similarity between data objects. A commonly used measure is the Euclidean distance:

[1] We could use other methods (such as principal component analysis or multidimensional scaling) to look for similarities between data sets in the Cinemetrics database, but for now I will focus only on cluster analysis.

$$d(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{\sum_{k=1}^{p} (x_{i,k} - x_{j,k})^2}.$$

The Euclidean distance measure is simply an extension of Pythagoras's theorem to *p*-dimensional space, where *p* is the number of variables in **X**. The distance matrix can be computed using statistical software: for example, in `R` the Euclidean distance matrix for an input matrix **X** is given by `dist(X, method="euclidean")`. The Euclidean distance measure has some drawbacks: (1) it cannot cope with different baselines; (2) it cannot cope with different scales; and (3) it cannot cope with series of different lengths (i.e. the number of variables *p* has to be the same for each data object). In order to apply time series clustering it is therefore necessary to pre-process the data and the next section describes a simple data processing procedure based on the kernel density estimate of a point process.

Once the distance matrix between data objects has been calculated it is necessary to choose a clustering algorithm. In this paper I use a*gglomerative hierarchical clustering,* which proceeds by placing each object in its own cluster and then merging clusters into larger and larger clusters until all the data objects are contained within a single cluster or some stopping criteria (such as a specified number of clusters) is reached. There are several methods for determining which data objects should clustered together based on the minimum distance between data objects (single linkage), the maximum distance between data objects (complete), or the minimum variance (Ward's method). These are implemented as standard by statistical software packages. In `R` this is done using `hclust(d, method="ward")`, where `d` is a distance matrix.

The resulting cluster analysis can then be plotted as a dendrogram. A dendrogram is a tree diagram representing the nested grouping of data objects. The individual data objects are placed at the end of the branches of a dendrogram and are called *leaves*. A group of data objects is called a *clade*, and may contain a single leaf. Data objects within a clade are more similar to one another than objects in another clade. The point at which the branches of a dendrogram join represent the distance between clusters merged together. Items similar to one another are combined at low heights, while dissimilar items are combined at greater heights. In `R` the dendrogram is produced by `plot(h)`, where `h` is an object produced by `hclust`.

If you want to use PAST, cluster analysis is found under the multivariate menu and will calculate the distance matrix, cluster, and plot the dendrogram all in one go.

There are several distance measures and clustering algorithms to choose from and they tend to produce slightly different results, though many features will be consistent whichever methods are used. In this paper I apply Ward's method to a Euclidean distance matrix to illustrate time series clustering, but it is only through more detailed research comparing the performance of various distance measures and clustering methods that we will discover the most useful approach.

Data pre-processing

Analysis of multiple time series of shot length data is challenging because (1) the running time of motion pictures varies so that the time series are of different lengths; (2) the number of cuts in a motion picture varies so that a different number of events is recorded for each time series; and (3) there is no uniform editing pattern so events are recorded at

irregular intervals. To make our analyses simpler we need a way to represent these time series that solves these problems.

As I have argued elsewhere (Redfern 2013), fitting a kernel density to the normalized point process is a simple way of analysing the time series of a motion picture. A *point process* is a set of points in time, which for a motion picture is simply the set of times at which the cuts occur. The time at which the *j*th cut occurs is equal to the sum of the duration of the prior shots, and to construct a point process of cuts in a motion picture we add the duration of each shot to cumulative sum of the preceding shots. To normalize this vector to a unit length we divide each value by the total running time of the film. Table 1 demonstrates this process using data for *Halloween* (1978): the raw shot length data is in the second column, the third column is the cumulative running time (i.e. point process), and the fourth column is the cumulative duration divided by the total running time of the film (5397.9s).

Table 1. Calculating the normalized point process for *Halloween* (1978)

Shot Number	Shot length (s)	Point process (s)	Normalized point process
1	8.3	8.3	0.0015
2	8.3	16.6	0.0031
3	258.9	275.5	0.0510
…	…	…	…
604	5.3	5386.5	0.9979
605	2.9	5389.4	0.9984
606	8.5	5397.9	1.0000

The kernel density estimate at x is

$$y = \hat{f}(x) = \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right),$$

where h is the bandwidth of the kernel function (K). Because the kernel density can be calculated at any point on the x-axis rather than just those locations where cuts occur we can produce a density trace with a specified number of data points. By setting the number of points at which we calculate the density the same for each film we produce a set of time series where each series has the same number of events (thus making the number of variables in the input matrix the same for each series). Furthermore, the points at which the density is calculated will be evenly spaced. Using this method we can therefore solve all of the above problems: by converting the raw data to density we can take two time series of different lengths with different numbers of irregularly occurring events and replace them with two new time series that cover the same time interval with equal numbers of evenly spaced data points.

Before we apply time series clustering we need to standardize the density of each film in order to remove baseline differences and differences of scale. One of the most overlooked aspects of shot length data is the variation of shot lengths in a motion picture. For example,

Figure 1 presents the time series for two versions of *Halloween* – John Carpenter's original 1978 version and Rob Zombie's 2007 remake. The densities for these films were calculated by using a bandwidth of 0.0125 and n = 500. The high density sections occur when shots are close together and indicate rapid editing, while low density sections result when shots are distant from one another and shows those sections when the cutting rate is slow. Particularly noteworthy in Figure 1 is the series of peaks in the latter fifth of each film where the density is greatest. In both versions this is associated with the final girl sequence, and the multiple peaks in these series all represent violent confrontations between Michael and Laurie. These films therefore present the same sequence in the same way. There are strong similarities between these time series, but there are also key differences. The density of the most rapidly edited section of the 2007 version of *Halloween* is lower than the corresponding section in the original version. This is a product of the difference in the variation in shot lengths of these films: for the original version Q_n = 3.3s and for the remake Q_n = 1.8s, while the ranges for the two versions are 258.6s and 34.5s, respectively. The difference in the variation of shot lengths in these films is obviously of interest and tells us much about changes in Hollywood film style over the past four decades. However, if we were to calculate the Euclidean distance between these two time series we would find they do not cluster together because they have different scales and what we understand as 'high density' in the remake of *Halloween* is not the same as high density in the original version of the film, even though they may be associated with the same narrative event and occur at roughly the same point in time. It is therefore necessary to standardize the shot densities before applying the cluster analysis.

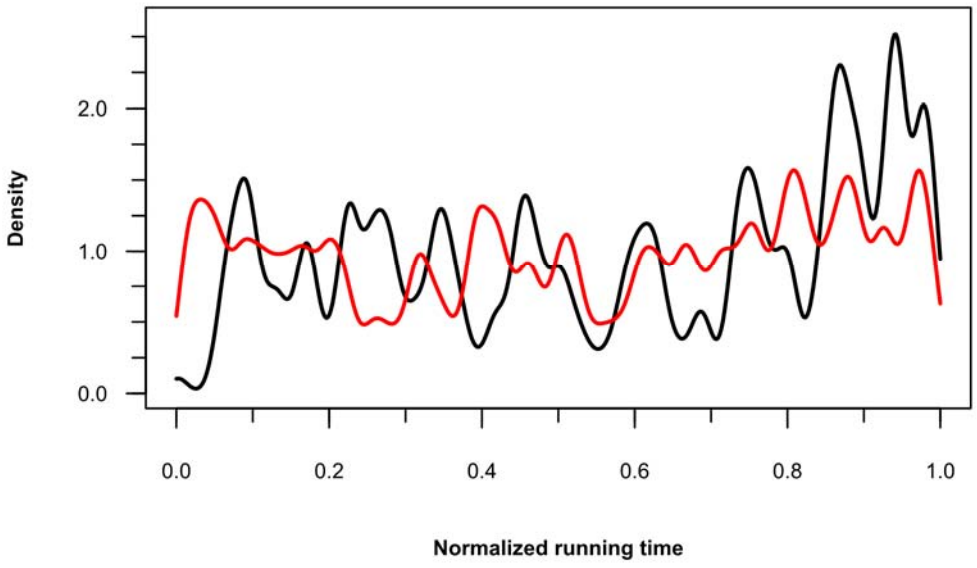

Figure 1 Shot densities for John Carpenter's *Halloween* (1978, black) and Rob Zombie's 2007 remake (red)

Standardizing the shot density of each film to the interval [0, 1] is a simple solution to this problem. To do this we apply the following transformation to each value in the shot density (y) we produced by the method described above:

$$F(y) = \frac{y_i - y_{min}}{y_{max} - y_{min}}.$$

That is, we subtract the smallest from each value and divide this difference by the range of the density. The result is the standardized shot density, in which the point at which the density is highest has a value of 1 and the point at which the density is lowest has a value of 0, with the rest of the values scaled within this interval. Figure 2 shows the re-scaled densities for the two versions of *Halloween*. Standardizing data in this way allows us to compare like with like by reducing semantic variation so that concepts like 'high density' and 'low density' have a single interpretation.

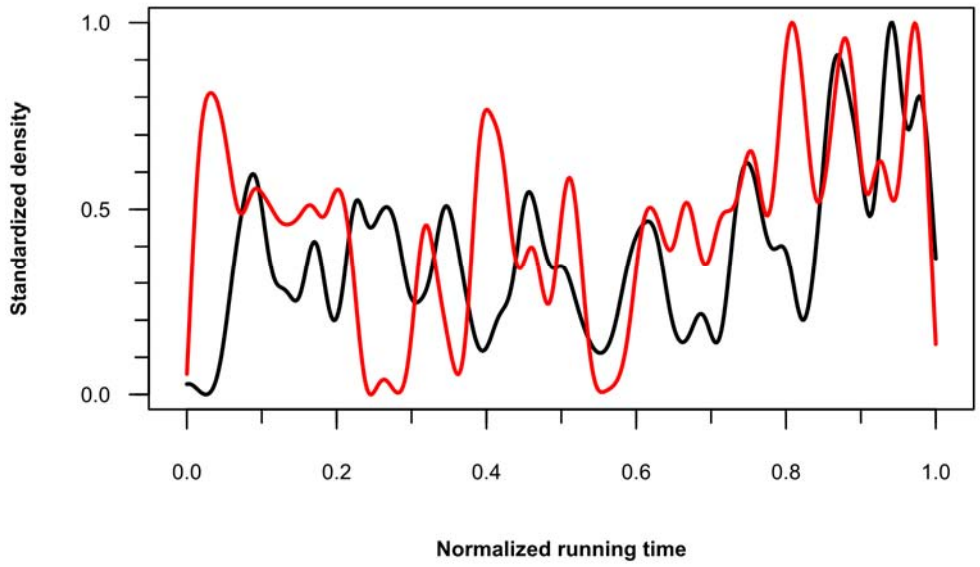

Figure 2 Standardized shot densities of John Carpenter's *Halloween* (1978, black) and Rob Zombie's 2007 remake (red)

If we think back to the matrix **X** described above this means that, after normalizing the point process and fitting and standardizing the kernel density, we now have a matrix with n films all of which have p variables, where every value in the matrix is the jth value of the standardized density for the ith film. After pre-processing the data to produce this matrix we are ready to perform the cluster analysis.

Time series clustering of Hollywood films

To illustrate time series clustering I compare a sample of twenty Hollywood films from the data set produced by Cutting, De Long, and Nothelfer (2010). The data was prepared according to the method described above and the clustering performed by applying Ward's method to the Euclidean distance matrix. Table 2 lists the sample of films, and Figure 3 presents the resulting dendrogram.

Table 2. Sample of twenty Hollywood films

Title	Year	Genre	Title	Year	Genre
Foreign Correspondent	1940	Drama	Airplane	1980	Comedy
Grapes of Wrath	1940	Drama	Coal Miner's Daughter	1980	Drama
Pinocchio	1940	Animation	The Empire Strikes Back	1980	Action
Santa Fe Trail	1940	Action	Nine To Five	1980	Comedy
The Great Dictator	1940	Comedy	Ordinary People	1980	Drama
Inherit the Wind	1960	Drama	Castaway	2000	Adventure
The Magnificent Seven	1960	Adventure	Charlie's Angels	2000	Action
Ocean's 11	1960	Comedy	Dinosaur	2000	Animation
Peeping Tom	1960	Drama	Erin Brockovich	2000	Drama
Spartacus	1960	Action	The Grinch Who Stole Christmas	2000	Comedy

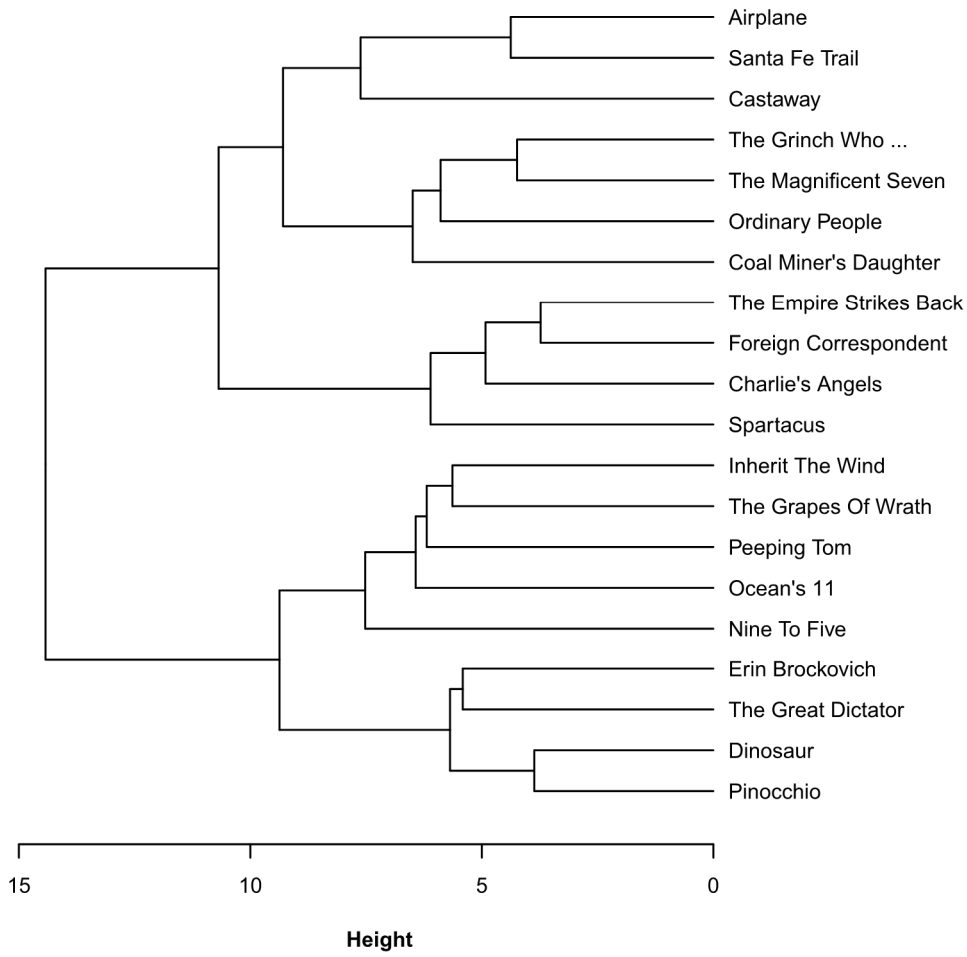

Figure 3 Cluster analysis of twenty films released between 1940 and 2000

From this dendrogram we see that there is no overall pattern in terms of year or release or genre. Films from the 1940s do not group together; nor do comedies, drama, or adventure films. Within this sample, the film with an editing structure most similar to *Foreign Correspondent* is *The Empire Strikes Back*. The standardized densities in Figure 4 show that both these films have high density sections and lower shot density. *Charlie's Angels* and *Spartacus* both have similar features to the films in Figure 4, though as the dendrogram illustrates the distance between these films increases. These four films can be said to represent a single cluster based on these features. Figure 3 indicates five clusters among the films in the sample and provides a jumping off point for exploring the formal similarities and differences between these films.

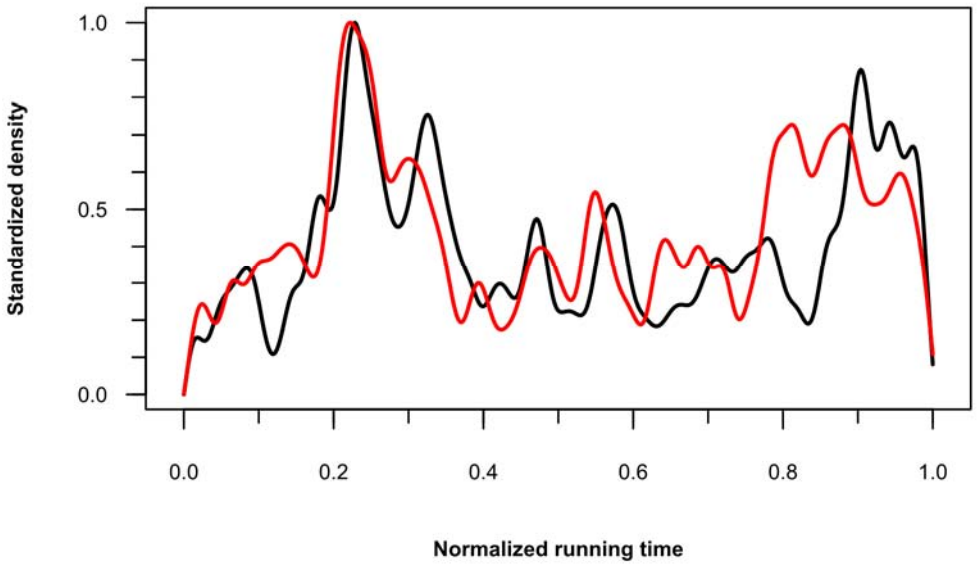

Figure 4 Standardized shot densities of *Foreign Correspondent* (1940, black) and *The Empire Strikes Back* (1980, red)

From Figure 3 we see that the standardized shot densities of *Foreign Correspondent* and *The Empire Strikes Back* are distant from *Pinocchio* and *Dinosaur*. The standardized densities for the latter films in Figure 5 are clearly different from those in Figure 4, with higher shot density across the running time of each film. These are the only two animated films in the dendrogram in Figure 3 and so we cannot say that this type of editing pattern is typical of animated films in general, but it is an avenue worth exploring.

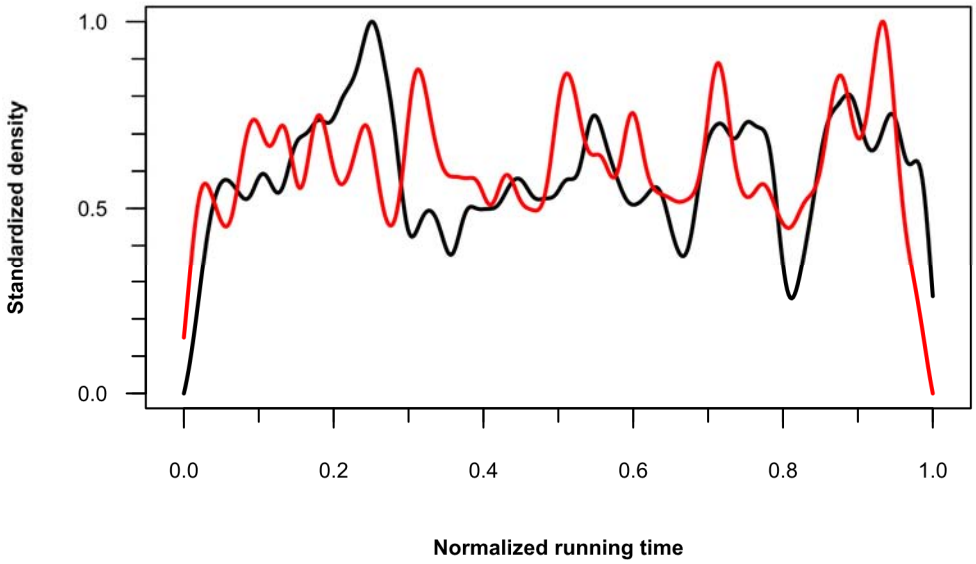

Figure 5 Standardized shot densities of *Pinocchio* (1940, black) and *Dinosaur* (1980, red)

There are some important caveats to bear in mind when applying cluster analysis to shot length data in the manner described above. First, you can only apply cluster analysis to the data you include in the original input matrix; and, second, just because cluster analysis indicates two data objects are similar as measured by the Euclidean distance between two vectors does not mean they are stylistically similar. Of the twenty films in the sample used here, *Foreign Correspondent* is most similar to *The Empire Strikes Back*; but this is not the same as saying the editing of these two films is the same. They are just the most similar objects among the samples to which we applied cluster analysis; and, while they do appear to have similar features at similar points, it is up to the researcher to take this information and go back to the films to discover which narrative events are associated with those features. Cluster analysis does not interpret data, and though we may talk of 'meaningful clusters' it is up to the analyst – and not the algorithm – to determine the meaning of any particular grouping.

Conclusion

In this paper I demonstrated a simple approach to data mining shot length data for a set of films using cluster analysis. The value of cluster analysis is that it allows us to search through large amounts of data and alert us to the possibilities of relationships between the data objects we wish to analyse. Cluster analysis is subjective, but given the size and complexity of data sets that record the style of motion pictures it is a useful tool that can make life much simpler for the researcher.

References

Cutting JE, De Long JE, and Nothelfer CE 2010 Attention and the evolution of Hollywood film. *Psychological Science* 21: 432-439.

Jain AK, Murty M, and Flynn PJ 1999 Data clustering: a review, *ACM Computing Surveys* 31 (3): 264-323.

Liao TW 2005 Clustering of time series data: a survey, *Pattern Recognition* 38 (11): 1857-1874.

Redfern N 2013 An introduction to using graphical displays for analysing the editing of motion pictures, http://www.cinemetrics.lv/dev/on_statistics.php, accessed 18 April 2013.