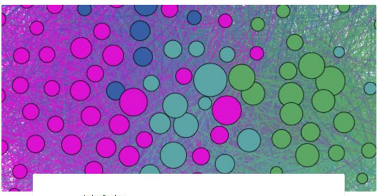
20. MAY 2014 / ANALYTICS TIPS

Introducing Clustering II: Clustering Algorithms



Anders Drachen

Anders Drachen, Ph.D. is a veteran Data Scientist, Game Analytics consultant and Professor at the DC Labs, University of York (UK).

Facebook LinkedIn Twitter

[This post was written in collaboration with Christian Bauckhage and Rafet Sifa.]

Clustering is imminently useful for finding patterns in gameplay data. In this second post in the clustering series, we briefly outline several classes of algorithms and discuss the types of contexts they are useful in.

Cluster algorithms can be categorized based on how the underlying models operate. Not all cluster algorithms are easily classified, but the following four categories provide a baseline overview, which can be used to identify the overall provenance of the algorithms mentioned and/or used in the sections below. Note that the four categories below do not form the only way to typify cluster algorithms.

Hierarchy clustering: Also called connectivity based clustering, this category of models is based on the idea that objects are more related to nearby objects than those further away. Clusters are thus developed based on distance between objects in the data space. Hierarchy clustering models were among the earliest techniques developed, and care must be taken with respect to outliers that can cause cluster merging (chaining). Furthermore, these models do not scale well to big datasets. In general, the models can be either agglomerative (beginning with individual objects and aggregating them) or divisive (beginning with all observations in the dataset and partitioning them). Hierarchical models are further differentiated based on the distance function used. When applying these models, the analyst needs to decide which distance function to use, as well as decide which linkage criterion to employ, and finally which distance to use. Given the hierarchical nature of the algorithms, there is no single partitioning provided, but rather a hierarchy of clusters which expand or decrease in number based on distance, and the choice of distance function and linkage criterion.

Centroid clustering: Centroid based clustering is often used in game analytics, primarily due to popularity and widespread use of k-means clustering (Lloyd's algorithm), which forms the basis for centroid clustering techniques, and is conceptually easy to understand. Centroid based models represent clusters by a central vector, which does not need to be an actual object (the term k-medoids denote when centroids are restricted to objects in the dataset, k-medians when medians are used. Fuzzy assignment of clusters is also common, i.e. fuzzy c-means). The analyst needs to define the number of clusters in advance (the number can be defined via initial exploration of the dataset), and k-means clustering then targets the optimization problem of finding k centers and assigning objects to the nearest center, in a way that minimizes the squared distances. K-means finds local optima, not global optima, and is therefore typically run multiple times with randomized initializations. Modifications of the k-means algorithm such as k-medoids and spherical k-means allow for using other distance measures than Euclidian distance.

Distribution clustering: Distribution-based clustering directly relates to the use of distribution models (e.g. Gaussian/Normal) in statistics. Fundamentally, clusters are defined based on how likely the objects included are likely to belong to the same distribution. Distribution-based models can provide information beyond the cluster assignments of objects, for example correlation of object attributes, but suffer from overfitting problems if the complexity of the model used is not constrained – for example defining a specific number of Gaussian distributions (Gausian mixture models). Importantly, these models do not work if there is no mathematical model inherent in the dataset for the model to optimize, and assuming that data adhere to Gaussian distribution models is inherently dangerous.

Density clustering: In this group of models, clusters are defined based on identifying areas of higher density than what can be found in the remainder of the data space. These approaches apply a local cluster criterion, and the resulting clusters (regions in data space) can have an arbitrary shape, and the points within can be arbitrarily distributed. Density clustering is able to handle noise if the result of the noise is objects in areas of the dataspace that is sparse. Density-based models can discover clusters of arbitrary shape and are optimized in one scan; however, they require the analyst to define density parameters as termination condition. Among the commonly used methods are DBSCAN, OPTICS, DENCLUE and CLIQUE, which vary substantially in how they operate and come in numerous variations. Density-based models have rarely, if at all, been used on behavioral data from games, but might find use in the future due to the ability to handle noise, which is a common feature in behavioral game telemetry.

Validation of models

> **The validation of clustering structures is the mode difficult and frustrating part of cluster analysis. Without a strong effort in this direction, cluster analysis will remain a black art accessible only to those true believers who have experience and great courage.**
>
> Jain and Dubes

A cluster model needs to be validated, or evaluated, before the results are implemented. A good cluster model consists of high-quality clusters with high intra-cluster similarity and low inter-cluster similarity. The quality of a result however depends both on the similarity measure used by the method of choice, and how it was implemented. Furthermore, the definition of distance functions is usually different for vector, ratio, ordinal, categorical, Boolean and interval-scaled features. Additionally, the quality of a clustering method can be measured based on its ability to discover some or all the hidden patterns in a dataset. Regardless, it is hard to define a "similar enough" or 'good enough" criterion – the answer is often subjective.

When running a classification algorithm on a dataset, there are a variety of measures available to evaluate how well the objects in the dataset fit the model. However, in cluster analysis there are no prior classes defined, and thus validation and evaluation of models risks becoming an 'eye of the beholder"-exercise, unless proper methods are employed.

A variety of different measures of similarity between two cluster results have been proposed, with the overall goal of comparing how well different algorithms perform on a specific dataset, i.e. for evaluating which method provides the best result. Two common approaches to cluster validation are internal and external evaluation. Internal evaluation operates on the dataset itself. External validation evaluates clustering results based on data that were not used in the cluster analysis, such as benchmarks or known classes. Both approaches have their strengths and weaknesses towards evaluating the results of different algorithms. For example, methods that work by assigning high scores to algorithms that produce high intra-cluster similarity and low inter-cluster similarity are biased towards algorithms that use the same cluster model, e.g. k-means clustering, leading to potential overrating of the result. For external validation methods, the key limitation is that clustering is often used in situations where there are no known classes or other external knowledge to compare with.

A full review of cluster validation techniques is out of scope here (see e.g. here and here), the key point being that cluster validation is a necessary step of any cluster analysis.

In the next post, we take a look at the specific challenges involved when using clustering for game analytics.

Acknowledgements
We are indebted to several colleagues for sharing their insights and feedback on this post, including but not limited to Christian Thurau, Fabian Hadiji and Shawn Connor.