Clustering

Hierarchical, *k*-Means, DBSCAN

Matthias Fuchs and Wolfram Höpken

Learning Objectives

- Learn typical applications of clustering within the tourism domain
- Explain the conceptual foundations of widely used clustering approaches
- Illustrate a step-by-step application of major clustering approaches on real tourism data using the data science platform *RapidMiner*®
- Demonstrate a tourism case that applies clustering approaches to identify points of interest based on uploaded photo data from the platform Flickr

1 Introduction and Theoretical Foundations

Clustering represents one of the most commonly used quantitative analysis techniques in tourism, typically applied to the task of market segmentation (Baggio & Klobas, 2017; Dolnicar, 2021). Cluster analysis aims to identify classes, also labeled as clusters, of the most similar cases within a dataset. Clusters may represent any type of object, which is represented as a statistical *case*, such as individuals (e.g., travelers or tourism entrepreneurs) but also tourism products, firms, etc. More formally speaking, a cluster analysis means grouping cases based on their similarity as given by the multivariate characteristics representing the cases of a particular

M. Fuchs (✉)
Department of Economics, Geography, Law and Tourism, Mid Sweden University, Östersund, Sweden
e-mail: matthias.fuchs@miun.se

W. Höpken
Institute for Digital Transformation, University of Applied Sciences, Weingarten, Germany
e-mail: wolfram.hoepken@rwu.de

R. Egger (ed.), *Applied Data Science in Tourism*, Tourism on the Verge,
https://doi.org/10.1007/978-3-030-88389-8_8

sample or the population itself (Baggio & Klobas, 2017). Before describing the theoretical foundations of the most widely applied clustering approaches in the tourism domain, the typical application areas of cluster analysis in contemporary tourism science and generally in practice will be touched upon.

As mentioned, market segmentation represents the classical application domain of cluster analysis in tourism. A good example is the segmentation study by Hudson and Ritchie (2002), who used cluster analysis to identify domestic tourist segments in the Alberta region of Canada. Firstly, 13 influential factors driving tourism decision-making, such as the quality of accommodation, the variety of tourism activities offered, holiday periods, and weather conditions, were identified through qualitative research. As a second step, 3000+ residents were interviewed by telephone to assess the importance of these influential factors. On the basis of these answers, and by additionally considering demographic characteristics, the cases of this representative sample were clustered into five market segments: the young urban outdoor market, the indoor leisure traveler market, the children-first market, the fair-weather friends-visiting market, and the older cost-conscious market. More recently, Neuburger and Egger (2020) employed cluster analysis to identify segments of travelers at two different points in time based on their perceived risk of COVID-19, perceived risk of traveling during the pandemic, and travel behavior regarding a change, cancellation, or avoidance of travel (plans). The study identified three clusters (i.e., the anxious, the nervous, and the reserved) with distinctive characteristics. In addition, results revealed a significant increase in risk perception of COVID-19, travel risk perception, and travel behavior over a short period of time.

As suggested by Dolnicar (2021), in the future, market segmentation in tourism will harvest its primary strength by using web-based data, such as online search data (Fuchs et al., 2014; Höpken et al., 2015), web navigation data (Pitman et al., 2010), or online feedback data (Dietz et al., 2020) as opposed to relying on surveys or interview data. In line with this claim, Höpken et al. (2020) recently examined the suitability of different clustering techniques to identify points of interest based on uploaded photo data extracted from the photo-sharing platform Flickr. We will discuss the details of this work in more depth below. As the latter example shows, tourism studies can apply clustering to means beyond solely market segmentation, for instance, to meaningfully group tourism suppliers, such as lifestyle entrepreneurs in nature-based tourism (Fuchs et al., 2021). In this vein of analysis, Scholochow et al. (2010) employed cluster analysis to group 700 Austrian hotel managers based on their behavioral pattern of having adopted e-Business technologies to improve their companies' efficiency and effectiveness.

As a vast array of cluster analysis techniques exists (Everitt et al., 2011; Liu, 2011), our discussion will be restricted to the three most widely applied clustering approaches within the tourism domain, namely, *hierarchical* clustering, *k*-means, and DBSCAN (*Density-Based Spatial Clustering of Applications with Noise*) (Tan et al., 2018). In terms of an optimal mathematical solution, clusters exhibit high internal homogeneity (i.e., minimum *within-cluster* variation) and high external heterogeneity (i.e., maximum *between-cluster* variation), as shown in the cluster diagram in Fig. 1 (Hair et al., 2014).

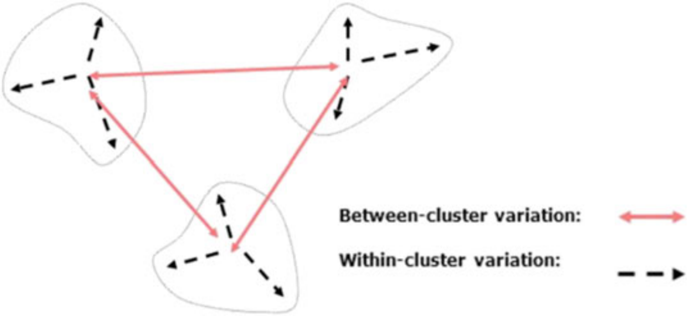
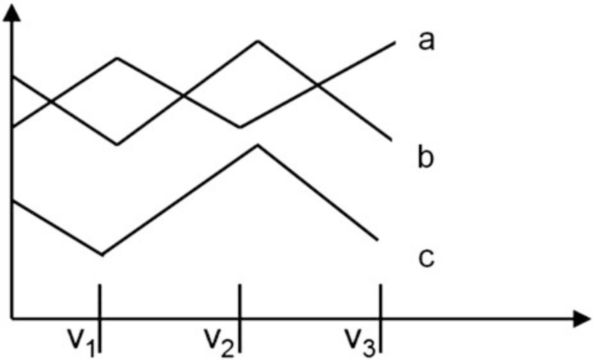


Fig. 1 Between- and within-cluster variation (see: Hair et al., 2014, p. 439)

Fig. 2 Inter-object similarity as difference or correlation (see: Hair et al., 2014, p. 430)



In fact, cluster analysis tries to minimize the differences between cases within each cluster by simultaneously maximizing the differences between clusters. Thus, one issue that is common to all clustering techniques is the representation of similarity (or difference) between pairs of cases or pairs of clusters, respectively (Baggio & Klobas, 2017). Differences are typically derived by a *similarity* measure (Everitt et al., 2011). Yet, the choice of a particular similarity measure does not only depend on the scale level of the cluster variables (i.e., binary, categorical, or metric), but, rather, it is mainly influenced by the fact that inter-object similarity (i.e., case/case, case/cluster, or cluster/cluster) can either be detected by *correlation* or *distance*-based measures of similarity. More precisely, by focusing on cluster variables' patterns, correlation measures (e.g., *Tanimoto*) can interpret the correlation of patterns as a similarity. By contrast, distance measures regard the magnitude of the distance. Accordingly, distance measures ponder an object-pair as similar if its variables show a low difference in magnitude (Hair et al., 2014). Looking at Fig. 2, object-pair a–b is viewed as similar based on a *distance*-based measure, while object pair b–c is viewed as similar by means of a *correlation*-based measure.

Notably, in clustering practice, distance-based measures of similarity tend to dominate. A prominent measure is the *Mahalanobis distance*, a standardized form

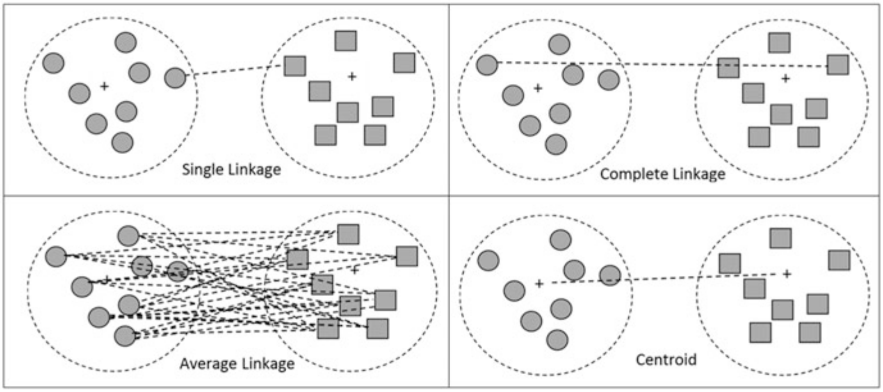


Fig. 3 Bases for calculating differences between cases (see: Baggio & Klobas, 2017, p. 78)

of the *Squared Euclidean distance*, which takes the co-variances between cluster variables into account (Baggio & Klobas, 2017).

Clustering methods typically allow, or require, the specification of the *points* within clusters between which distances are calculated, or, in other words, on which base clusters are subsequently *formed* (ibid, 2017). In the case of *nearest neighbor*, or *single linkage*, distances are defined as the distance between the closest elements in the clusters. The approach is most convenient if clusters are poorly delineated and tend to build long and slender chains. By contrast, *farthest neighbor*, or *complete linkage*, calculates distances between the farthest elements in the clusters, which is useful if clusters are compact and have consistent diameters (Hair et al., 2014). *Average linkage* computes distances as the average of all pairwise distances, thereby tending to combine clusters with similar variance, and, finally, *centroid distance* calculates distances between the geometric centers (*centroids*) of the clusters (see Fig. 3).

1.1 Hierarchical Cluster Analysis

Cluster analysis techniques are typically divided into *hierarchical* and *partitioning* categories (Everitt et al., 2011; Tan et al., 2018). *Hierarchical cluster analysis* is subdivided even further into *divisive* and *agglomerative*. The former starts by placing all objects into one single large cluster and progressively subdividing the one cluster into two, thereby maximizing the differences between the clusters obtained from each division (Baggio & Klobas, 2017). In contrast to this *top-down* approach, *agglomerative* techniques operate in the opposite *bottom-up* direction; they begin by defining each case as a single cluster and then by continuously combining the pairs that are most similar until all cases and clusters are conjoined in one cluster. A popular hierarchical *agglomeration* algorithm, in the case that all

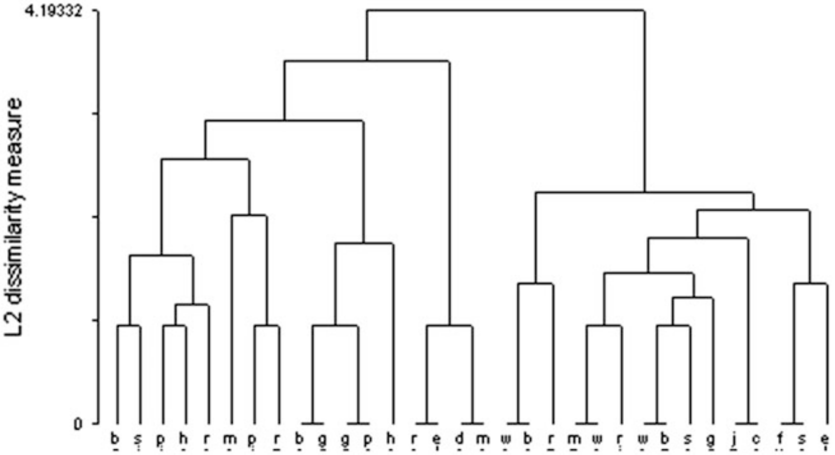


Fig. 4 Dendrogram (Source: Authors' illustration)

clustering variables are metric, is *Ward's linkage* (Ward, 1963). Considered a particular case of the single linkage algorithm, it combines clusters that minimize the within-cluster sum of squares across the complete set of clusters (Hair et al., 2014). Thus, the combined clusters are those that minimize the increase of the total sum of squares across all cluster variables. This popular algorithm tends to generate clusters of similar sizes. An alternative to *Euclidean distance*, typically employed with *Ward's linkage*, is *cophenetic* distance (Tan et al., 2018).

As highlighted, the process of joining clusters continues until the most distant (i.e., different) clusters are united. Notably, the further apart clusters join in later agglomeration stages, the more likely they are to form meaningfully distinct clusters (Baggio & Klobas, 2017). Therefore, when identifying an optimal number of the most meaningful clusters, an analyst normally refers to both the *agglomeration schedule* as well as a particular plot, the *dendrogram*. The former shows at which (typically late) stages the relative increase of the *agglomeration coefficient* appears to be particularly large, thus pointing at a merge of quite distinct, characteristic, and meaningful clusters. The latter provides a *rescaled* graphical illustration of the distance between clusters joined at each stage (Hair et al., 2014; see Fig. 4). Additional aid in identifying the optimal number of clusters can be provided by coefficients like *Silhouette* scores, *Calinski–Harabasz* index, and *Davies-Bouldin* (Tan et al., 2018).

1.2 Partitioning

In contrast to hierarchical clustering methods, non-hierarchical procedures do not involve the treelike construction process of clusters (Everitt et al., 2011; Baggio &

Klobas, 2017). Instead, they assign objects to clusters once the number of clusters is pre-specified (Hair et al., 2014). *Partitioning* procedures work based on a simple principle. As shown in Fig. 1, they seek to simultaneously maximize the distance between clusters and minimize the differences between in-group objects (ibid, 2014). In order to identify this optimum, partitioning procedures typically apply a series of *iterative* computations. The most widely used partitioning technique is known as *k*-means clustering (Kanungo et al., 2002). The *k*-means clustering algorithm partitions a dataset into a predefined number of clusters in which each data point belongs to the cluster with the nearest (i.e., most similar) cluster mean, or *centroid* (Lloyd, 1982). As finding the optimal clustering solution is computationally difficult, *k*-means is a heuristic algorithm, starting with a randomly chosen partition and then iteratively optimizing the solution by re-calculating the means (or centroids) and re-assigning data points to clusters accordingly (Larose & Larose, 2014; Tan et al., 2018). Cases continue to be moved until the sum of within-group variances is minimized (Baggio & Klobas, 2017). Through this iterative procedure, a potentially optimal solution can be identified. More concretely, in contrast to hierarchical cluster analysis, a 3-cluster solution is not merely a combination of 2 clusters from a 4-cluster solution; rather, it can be considered the "best possible" 3-cluster solution. Despite this advantage, one limitation of *k*-means involves that it requires the number of clusters to be explicitly specified and it can only partition a dataset into hyper-spherical or hyper-ellipsoid clusters, which, in turn, tend to be of similar size (Liu, 2011). Moreover, as *k*-means is sensitive to outliers, a distance-based outlier detection is also typically needed and is, therefore, recommended as a data preparation step (Pyle, 1999).

1.3 Density-Based Spatial Clustering of Applications with Noise

Developed by Ester et al. (1996), *Density-Based Spatial Clustering of Applications with Noise* (*DBSCAN*) enables the identification of especially spatial clusters based on the density of the data points. This approach differentiates between *core points*, lying in a *high*-density region, *border points*, lying at the edge of a *high*-density region, and *noise points* (i.e., outliers) lying in a *low*-density region. Accordingly, core points need to be surrounded by a minimal number of other points within proximity of their close neighborhood, while border points need to closely neighbor a core point. Noise points fulfill none of these requirements. *DBSCAN* clusters are built by starting with any core point that has yet to belong to a cluster and then successively adding core points or border points that lie in close proximity to a core point already belonging to a cluster (Tan et al., 2018). The *DBSCAN* algorithm can be parameterized by the minimal density within a cluster (*minPts*), in other words, the minimal number of data points that have to lie within the neighborhood of a core

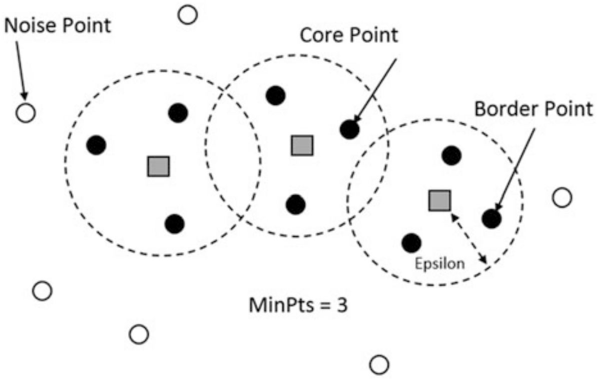


Fig. 5 Density-based spatial clustering of applications with noise (Source: Authors' illustration)

point as well as the size of the neighborhood (ε, epsilon); thus, the maximal distance between two neighboring points (Fig. 5).

As *DBSCAN* uses a density-based definition of a cluster, it is relatively resistant to noise and can handle clusters of arbitrary shapes and sizes. Therefore, *DBSCAN* can find many clusters that would otherwise be undetectable when using *k*-means. *DBSCAN*, however, shows weaknesses when the clusters have widely varying densities or are built on the basis of high-dimensional data due to density being difficult to define for such data. In the case of high-dimensional data, *DBSCAN* can be of low performance since the computation of the nearest neighbors requires computing all pairwise proximities (ibid, 2018). Additionally, defining the neighborhood size (*epsilon*) and the number of neighbors (*minPts*) appropriately can be difficult in certain application areas. Here, the hierarchical extension *HDBSCAN* may be more suitable.

1.4 Cluster Evaluation and Profiling

Typically, in the final stages of cluster analysis, the obtained clusters need to be interpreted (Larose & Larose, 2014). On the one hand, *profiling* identifies and describes the most typical characteristics of each cluster usually by means of investigating the maximal score values of the cluster variables. On the other hand, through *labeling*, a tag, which most accurately describes its nature, is assigned to each cluster. For the final validation step, significance tests between cluster variables (e.g., ANOVA) along with a multiple discriminant analysis, which estimates the share of cases correctly classified to cluster membership on the basis of a discriminant function composed by the cluster variables, is recommended (Hair et al., 2014). As previously mentioned, to identify the optimal number of clusters, an analyst may refer to agglomeration schedules and the dendrogram as well as to coefficients, such as Silhouette scores, Calinski-Harabasz index, and Davies-Bouldin (Tan et al., 2018). Ultimately, however, it remains the analyst's judgment, based on theoretical and practical knowledge, to decide which variable sets should be used to build the clusters and to determine the final number of clusters that best represent a set of cases (Baggio & Klobas, 2017). Cluster analysis is, therefore, considered both a "science and an art" (Hair et al., 2014).

2 Practical Demonstration

In this section, we will explain step-by-step how the clustering approaches presented above can be executed on real tourism data by using the data science platform *RapidMiner*® (www.rapidminer.com).

2.1 k-*Means Clustering*

Considered the most prominent clustering task, customer segmentation is a concrete example that can be solved by *k*-means. Here, we used real customer data from a winter destination's booking system as our dataset (cf. Table 1). The dataset contains customer information including guest's age, first year of a guest's arrival at the destination, number of past bookings, preferred duration of the trip, booking channel (i.e., 0 = web, 1 = phone), average cancellation rate, days between booking and arrival, price per booking, and the average number of booked products, rooms, ski passes, ski equipment, and ski school services per booking. The dataset consists of 5172 instances/customers (rows). Partitional clustering, like *k*-means, intends to divide the complete dataset into groups of customers that are as similar as possible in relation to the characteristics listed above.

A dataset typically requires certain steps of preprocessing in order to comply with the specific prerequisites of the selected data mining algorithm and to reach optimal and reliable results (Pyle, 1999; Tan et al., 2018). In the case of clustering, the first task is to select an appropriate subset of available attributes (cluster variables), which should serve as ideal characteristics to group similar examples (cases) together. Customer segmentation can be restricted, for example, to demographic characteristics or past booking behavior only (Dolnicar, 2021). In our case, however, we used all the attributes listed above. Additionally, an attribute's data type must also be checked. Although the *k*-means algorithm is capable of handling any of the usual data types, like numeric or nominal attributes, it is recommended to transform nominal attributes into numeric dummy attributes as this transformation simplifies the visualization of the results (Pyle, 1999). Accordingly, we transformed the attribute *TripDuration* into *TripDuration* = *Week*, *TripDuration* = *ShortWeek*, etc. with numerical values of 0/1. Another critical step regarding preprocessing is the normalization of the attributes' value ranges (Everitt et al., 2011; Tan et al., 2018). As the similarity of the examples corresponds to the distance between the examples in an *n*-dimensional space, the size of the value range of an attribute determines its influence on the similarity calculation. In our case, for example, the

Table 1 Dataset with customer data from a winter destination

Row no.	Age	FirstArrivalYear	NoBookings	TripDuration	Booking Channel	Cancellation Rate	DaysBook 2Arrival	PricePer Booking	NoBooked Products	NoRooms	NoSki Pass	NoSki Equipment	NoSki School
1	31	2005	5	Weekend	0	0	148,800	3281	1400	1	0	0	0
2	44	2004	4	N/A	0.500	0.500	273,250	152,500	2500	1	0	0	0
3	47	2004	10	N/A	1	0	247,100	57	1100	0.900	0	0.200	0
4	52	2004	29	N/A	1	0.034	192,207	37,931	1655	1.586	0	0	0
5	39	2005	2	Week	1	0.500	136	9170	9	0.500	1	3,500	3
6	54	2004	26	Week	1	0.269	242,269	0	1077	1	0	0	0

Source: Authors’ illustration

booking price would have a much stronger influence on the calculation of a similarity measure than the cancellation rate. As this influential power is entirely accidental, we normalized all value domains via *Z*-score standardization, setting the average of each attribute to zero and the standard deviation to one (Everitt et al., 2011). In the final step of preprocessing, we eliminated outliers by removing all instances (i.e., cases) with attribute values outside the range of −4 to 4 (in our case, 231 cases) since such values can be viewed as extreme values after having executed the Z-score standardization. Outliers have to be eliminated because the *k*-means algorithm is particularly sensitive to extreme values (ibid, 2011; Hair et al., 2014).

As the central part of our analysis, a *k*-means clustering can now be executed. First, since they are regarded as the most important parameters, the number of clusters *k* and a similarity measure, such as the *Euclidean distance*, the *Chebychev distance*, or the *cosine similarity* must be chosen (Tan et al., 2018). Optimizing these parameters should, on the one hand, be viewed from a mathematical perspective, that means, reaching optimal quality measures, like the within-cluster variation or the Davies Bouldin measure, which we calculate on the determined cluster model. On the other hand, the found clusters should be easily interpretable and make sense either from a business perspective or as input for consecutive steps of analysis, for example, as a dimension reduction technique or as input for a classification task. In our case, a *k*-means clustering with $k = 3$ along with the similarity measure *cosine similarity* reached good results with an average within-cluster variation of 0.573 and a Davies Bouldin measure of 0.164 (both measures should be minimized when comparing different clustering solutions). The resulting cluster model is depicted in Fig. 6.

As can be seen, cluster 0 (1472 customers) represents older and long-standing customers, staying for a short week or weekend, booking mainly via phone, showing a high cancellation rate, booking quite far in advance, having a low overall booking price, and are booking mainly accommodation services (thus, labeled as "*Older*

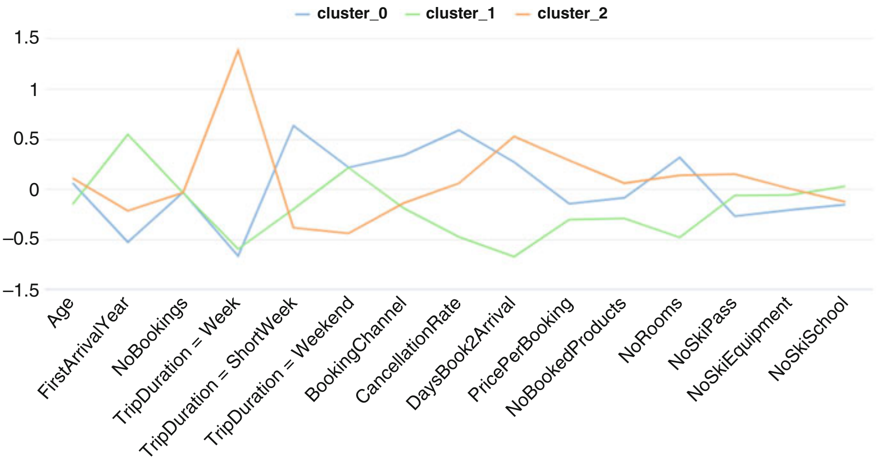


Fig. 6 Centroid plot *k*-means clustering (Source: Authors' illustration)

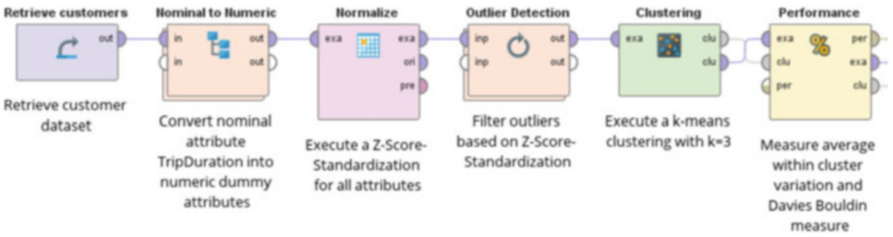

Retrieve customers
out
Retrieve customer dataset
Nominal to Numeric
in
in
out
out
Convert nominal attribute TripDuration into numeric dummy attributes
Normalize
exa
exa
ori
pre
Execute a Z-Score-Standardization for all attributes
Outlier Detection
inp
inp
out
out
Filter outliers based on Z-Score-Standardization
Clustering
exa
clu
clu
Execute a k-means clustering with k=3
Performance
exa
clu
per
per
exa
clu
Measure average within cluster variation and Davies Bouldin measure

Fig. 7 Rapidminer process for *k*-means clustering (Source: Authors' illustration)

weekend customers"). By contrast, cluster 2 (1538 customers) represents, again, older and long-standing customers, but this time staying for a full week, booking mainly via the Web far in advance with a high booking price, and booking mainly accommodation services as well as ski passes and ski equipment (thus, labeled as "*Older full-week skiing customers*"). Finally, cluster 1 (1931 customers) represents young and relatively new customers, staying over the weekend, booking mainly via the Web in a quite short-term/last-minute manner with a low cancellation rate and a low booking price, and mainly booking ski passes, ski equipment, and ski school services (thus, labeled as "*Young and spontaneous weekend skiing customers*").

All the steps described above were executed using the data mining toolset *Rapidminer*. Figure 7 shows the overall analysis process consisting of a chain of operators, each receiving some input (e.g., the dataset), incorporating preprocessing or analysis steps, producing some output (e.g., the transformed dataset or a learned model), and passing the output on to the next operator. In this case, the first operator read the dataset, the second transformed the nominal attribute TripDuration into numeric flag attributes, the third normalized all the attributes, the fourth detected and deleted outliers, the fifth executed the *k*-means clustering, the sixth calculated the performance measures, and the seventh calculated the cluster centroids for all the clusters.

2.2 *Hierarchical Clustering*

As highlighted in the introductory section, in contrast to partitional clustering, hierarchical clustering successively divides (top-down) or groups (bottom-up) cases into clusters at different stages, resulting in a cluster hierarchy. Consequently, cases belong to a cluster on each stage or level of this hierarchy (Everitt et al., 2011; Tan et al., 2018).

2.2.1 Top-Down Clustering

In this subsection, a *top-down* clustering is applied to the dataset above (and the same preprocessing steps are executed). On the left, Fig. 8 shows the results of a

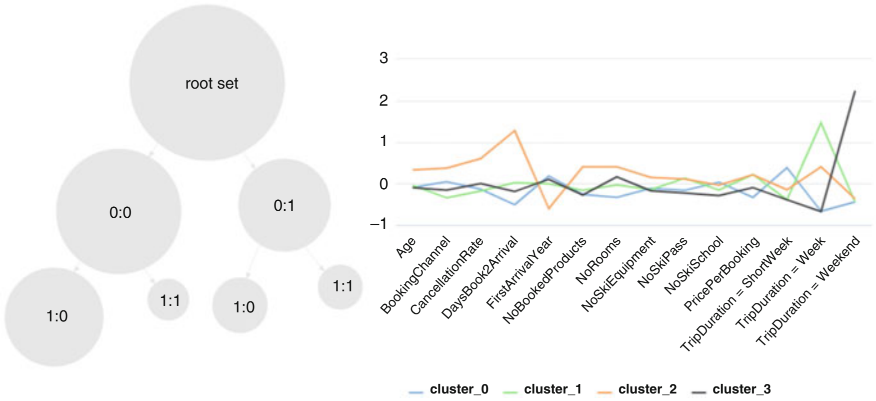


Fig. 8 Top-down clustering: Cluster hierarchy (left) and cluster centroid plot (right) (Source: Authors' illustration)

top-down clustering for two levels, thereby illustrating the successful division of the complete dataset into clusters. The number of clusters created in each split and the number of levels of the hierarchy can be specified as parameters of the top-down clustering approach. The clustering itself is, once again, executed by a partitional clustering approach; in our case, a *k*-means clustering. Although each example belongs to a different cluster on each level of the hierarchy, all clusters on the same level constitute an exact partitioning. Figure 8 on the right shows a cluster centroid plot for the four clusters located on the lowest level of the hierarchy. Through this approach, hierarchical clustering allows for the thorough analysis of clusters at different granular levels. The *Rapidminer* process remained the same as in Fig. 7 apart from the *k*-means operator being replaced by the *top-down clustering* operator.

2.2.2 Agglomerative (Bottom-Up) Clustering

For the second hierarchical clustering approach, we now apply *agglomerative* (*bottom-up*) clustering to the dataset described above (for demonstration purposes, 20 examples from the total dataset were randomly selected). Agglomerative clustering iteratively joins the two clusters that are most similar, where similarity is measured based on the two most similar cases of the two clusters (*single linkage*), the instances that are most dissimilar (*complete linkage*), or the cluster centroids (*average linkage*). On the left, Fig. 9 shows the cluster hierarchy for the similarity mode *single linkage*, while the right illustrates the one for *complete linkage*. As can easily be observed, the single linkage mode (left) tends to form slender clusters by

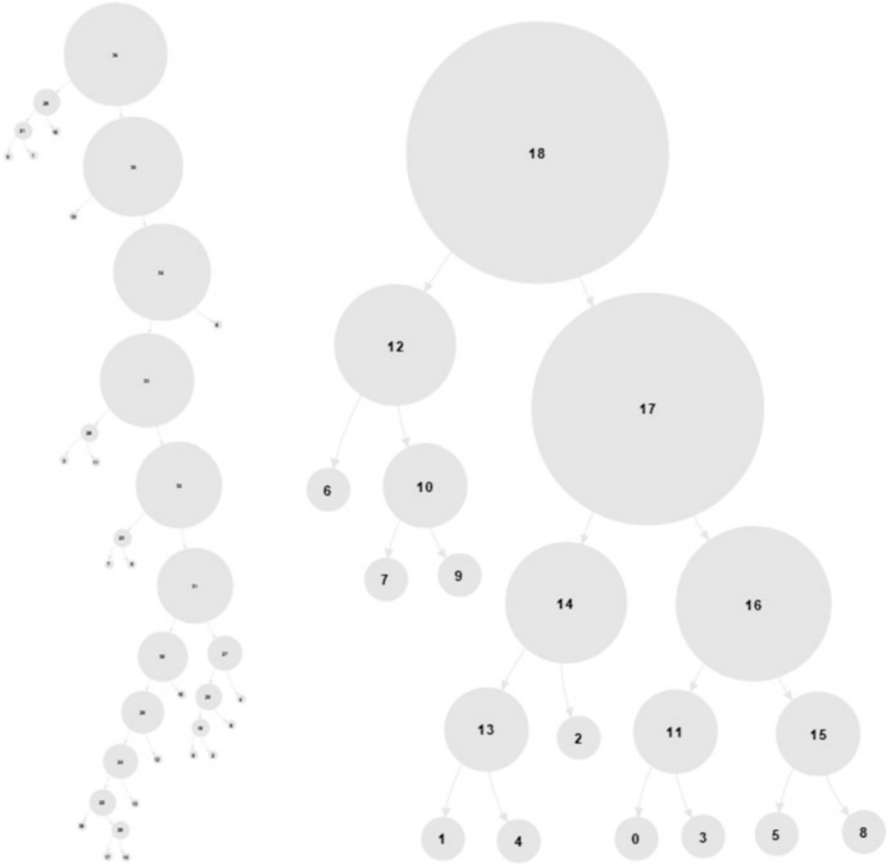


Fig. 9 Agglomerative clustering with single linkage (left) and complete linkage (right) (Source: Authors' illustration)

successively adding single examples or small clusters to one main cluster, getting bigger and bigger. On the other hand, the complete linkage mode (right) creates a much more balanced cluster hierarchy and, thus, represents the preferred approach in our case.

Figure 10 shows the *Rapidminer* process for the agglomerative clustering approach. The operator *Sample* creates a subsample by randomly selecting 20 examples for demonstration purposes. All preprocessing steps were the same as in Fig. 7, although clustering is executed via the operator *Agglomerative Clustering*.

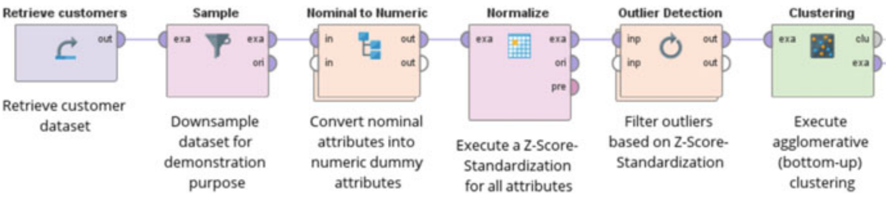

Retrieve customers
out
Retrieve customer dataset
Sample
exa
exa
ori
Downsample dataset for demonstration purpose
Nominal to Numeric
in
in
out
out
Convert nominal attributes into numeric dummy attributes
Normalize
exa
exa
ori
pre
Execute a Z-Score-Standardization for all attributes
Outlier Detection
inp
inp
out
out
Filter outliers based on Z-Score-Standardization
Clustering
exa
clu
exa
Execute agglomerative (bottom-up) clustering

Fig. 10 *Rapidminer* process for agglomerative (bottom-up) clustering (Source: Authors' illustration)

2.2.3 DBSCAN

As noted earlier, *DBSCAN* is a density-based spatial clustering algorithm. In this example, we will apply *DBSCAN* to geo-coded Flickr photo uploads collected for the region of Lake Constance, Germany. The dataset contains 4121 Flickr photo uploads specified by the latitude and longitude of the photo's geographic position. The intention of clustering is to group together photo uploads that are located quite close together in order to identify points of interest (POIs). In this specific case, no data preprocessing steps are necessary as both attributes have a numeric format and a normalized value domain. Additionally, *DBSCAN* identifies outliers automatically; thus, no separate outlier detection is required (Tan et al., 2018).

In contrast to *k*-means, the number of clusters is not specified by the user, but, rather, it is identified automatically. Instead, the user can define the minimal number of examples (*minPts*) that should exist in the direct neighborhood of a cluster member and the size, in other words, radius (*epsilon*), of the neighborhood. Often, optimizing these parameters is, unfortunately, not an easy task (Tan et al., 2018). In our case, *epsilon* = *0.03* (i.e., 3.3 km) and *minPts* = *13* led to satisfying results. As a similarity measure, we used the *Euclidean distance*. Figure 11 shows the geo-coded Flickr photo uploads on the left and the clusters (i.e., the POIs) identified by the *DBSCAN* clustering algorithm on the right.

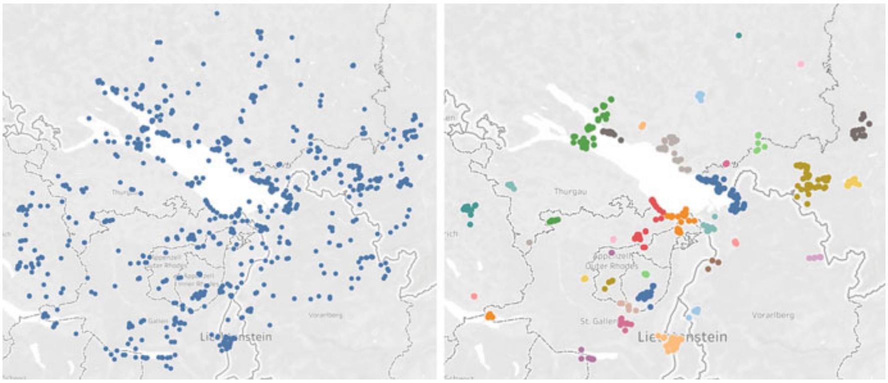


Fig. 11 Flickr photo uploads (left) and *DBSCAN* clustering (right) (Source: Authors' illustration)

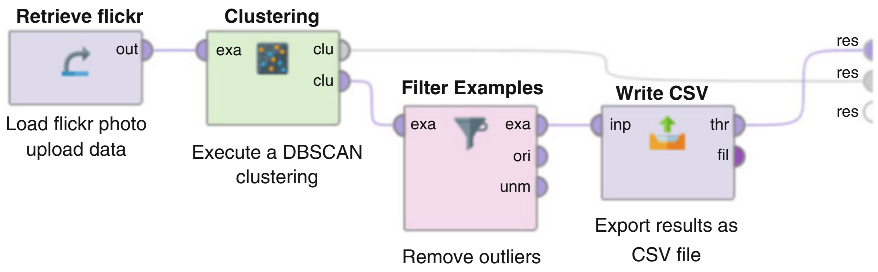


Fig. 12 Rapidminer process for *DBSCAN* clustering (Source: Authors' illustration)

Figure 12 shows the *Rapidminer* process for executing a *DBSCAN* clustering on Flickr photo uploads. The process, first, loaded the Flickr photo upload data and then, as with *DBSCAN* no preprocessing steps are necessary, directly executed the *DBSCAN* clustering. Outliers were automatically identified by the *DBSCAN* algorithm and removed from the resulting clustered dataset by means of the operator *Filter Examples*. Finally, the resulting dataset was stored as a CSV file and was used as input for tableau (www.tableau.com) in order to create the map-based visualizations shown in Fig. 11.

3 Research-Case

Traditional data sources, like guest surveys, visitor censuses, or on-site observations, impose a high amount of manual work and, thus, do not enable data gathering and analysis automatically and in real-time (Fuchs et al., 2014; Höpken et al., 2015; Önder et al., 2016). A study by Höpken et al. (2020) presents an approach that uses uploaded photos on the social media platform *Flickr* to analyze tourists' movement patterns when visiting points of interest (POIs), such as sights or attractions, in the destination city of Munich, Germany. By employing and comparatively assessing *DBSCAN* and *k*-means clustering for differing use scenarios, photo uploads on Flickr were clustered to POIs (Tan et al., 2018). Resulting POI visitation trajectories then served as input to analyze tourists' spatial behavior by association rule analysis and sequential pattern mining (ibid, 2020).

Data extraction was executed based on Flickr's application programming interface (API), such as *flickr.photo.search* to extract photo meta-data and *flickr.people.getInfo* to extract user information (e.g., user location). For each photo within the relevant geographic area, the following meta-data was extracted: photo id, owner id, owner name, latitude, longitude, location, date taken, and date uploaded. Following previous studies, users who continued to upload photos within an overall time period of more than 21 days as well as users who specified Munich as their home location were viewed as non-tourism users. Therefore, their photo uploads were removed from the dataset. Data was extracted from Flickr for the year 2015, resulting in 13,545 photo uploads from 971 tourists (ibid, 2020).

Next, clustering was employed to aggregate Flickr photo uploads to POIs based on their physical location (i.e., geo-coordinates) (ibid, 2020). The two clustering algorithms, *DBSCAN* and *k-means*, were both evaluated concerning their suitability to identify meaningful clusters corresponding to relevant POIs. Compared to the *k-means* algorithm, *DBSCAN* is known to have the capacity to identify clusters of arbitrary shape without any restrictions; thus, there is no need to specify the number of clusters. Also, as *DBSCAN* identifies noise points explicitly, no outlier detection is necessary. Identified clusters were filtered based on *minimal popularity*, in other words, the number of tourists within a cluster (Hu et al., 2015). A cluster is considered popular if at least 2% of all the tourists involved with the photo uploads within the respective time period and area belong to the cluster.

First, a *k-nearest neighbor* distance plot showing the average distance of each point to its k nearest neighbors was employed to identify optimal *DBSCAN* parameter values with $minPts = 3$ and $\varepsilon = 0{,}0009$ (i.e., 99 m), respectively (Höpken et al., 2020). Moreover, the number k of clusters found by *DBSCAN* was used for *k-means* to guarantee comparability of the two clustering approaches. The *DBSCAN* clustering model was characterized by one big cluster (containing 5909 photos), representing the city center of Munich and a high amount of relatively small and non-popular clusters (cf. 15 *DBSCAN* vs. 70 *k-means* popular clusters). In fact, *DBSCAN* grouped together all closely connected photo uploads, tending to generate large and often slender clusters (e.g., the cluster representing the city center of Munich). Put differently, *DBSCAN* was able to identify widespread and arbitrarily shaped clusters (not bound to hyper-ellipsoid or hyper-spherical clusters), which is an advantage in the case of identifying POIs on a larger geographic scale, for instance, for the *entire* urban region of Munich. On a smaller geographic scale, however, and in our case for the city center of Munich, *DBSCAN* identified all closely connected POIs as one single big cluster. In contrast, on such a small-scale granular level, like the city center of Munich, *k-means* clustering identified closely connected POIs correctly. This is mainly due to the fact that POIs in a city center environment tend to have a point-like form rather than a slender structure, constituting ideal conditions for the partitioning clustering approach of *k-means*, which requires the clusters to be of equal size and of hyper-ellipsoid or globular form (Liu, 2011). More concretely, while *DBSCAN* grouped all photo uploads of the central area into one big cluster, *k-means* identified 11 different clusters and, thus, correctly recognized corresponding POIs. One POI (Marienplatz) was separated into different clusters due to *k-means* well-known limitation of not being able to identify slender clusters properly (Höpken et al., 2020). Another POI (Feldherrenhalle) was merged with a neighboring POI (Odeonsplatz) due to *k-means* characteristic of trying to build clusters of similar size. In general, however, the results demonstrate that *k-means'* limitations do not substantially come into effect in regards to POIs in a city center environment as they mostly have a point-like structure and are typically of similar sizes. Overall, the assignment proves that clusters of Flickr photo uploads correspond to tourism-relevant POIs and, therefore, photo-sharing platforms like

Flickr can be constituted as a valuable source for analyzing tourists' POI visitation behavior.

In a final step, popular POIs identified through cluster analysis served as input for *association rule analysis* and *sequential pattern mining*. *Association rule analysis* aims at identifying which items or characteristics often "go together" within a dataset (Larose & Larose, 2014). Items, in the case of this study, correspond to clusters, or, POIs visited by tourists and an association rule $X \rightarrow Y$ meaning that a tourist visiting POI X will often visit POI Y as well (Höpken et al., 2020). In contrast to association rule analysis, *sequential pattern mining* considers the temporal order of items within a transaction (Larose & Larose, 2014). Thus, while a frequent item set represents items co-occurring, a frequent sequence represents a specific order in which items often occur (ibid, 2014). To identify sequential patterns, the Generalized Sequential Pattern (GSP) algorithm was employed, while the FP-Growth algorithm was applied to find frequent item sets (Liu, 2011). An exemplarily strong rule found, for instance, that tourists visiting the POIs "Kaufhaus der Sinne" (206) and "Altes Rathaus" (178) would most likely also visit POI "Heilig-Geist-Kirche" (190). This rule is supported by 1.4% of all transactions and holds true for 100% of all transactions containing the antecedents 206 and 178. The particularly high lift of 16.09 means that, when visiting POIs 206 and 178, it is over 16 times more likely that a tourist will also visit POI 190 when compared to the average likelihood of visiting POI 190 (Höpken et al., 2020).

Finally, when comparing the two clustering approaches *DBSCAN* and *k-means*, it can be summarized that *DBSCAN* identified 15 popular clusters, leading to 45 frequent item sets, 60 association rules, and 370 frequent sequences, while *k-means* identified 70 popular clusters, leading to 534 frequent item sets, 1432 association rules, and 4760 frequent sequences (ibid, 2020). Sequential pattern mining identified frequent visitation sequences of short (1-h) and medium (4-days) duration with support between 0.6% and 1.7%, respectively. Figure 13 displays the most frequent tourist routes in the old town of Munich identified via the *k*-means method.

The proposed approach demonstrates its ability to analyze tourists' spatial behavior and movement patterns based on uploaded photo data from *Flickr*. Compared to traditional data gathering techniques, the approach offers the advantage of being fully automatic and, thus, executable in a real-time environment (Kolas et al., 2015). The identified POIs visitation behavior allows for more detailed explanations regarding the attractiveness of various POIs depending on visitor characteristics including gender or country of origin. Furthermore, it also determines visitation time as an important input for tourism planning and marketing activities (Höpken et al., 2020).

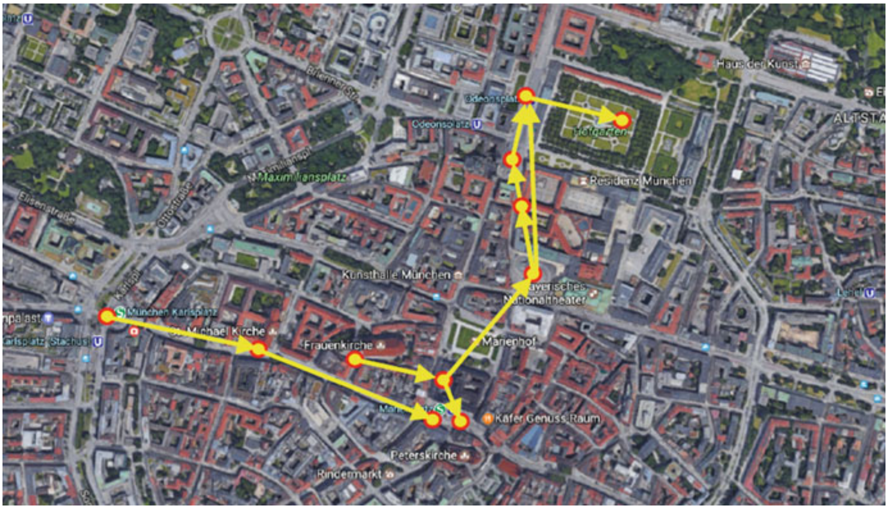


Fig. 13 Frequent tourist routes in the old town of Munich identified via *k*-means (The authors thank Marcel Müller for Fig. 13 extracted from the MA thesis "*Big Data als Quelle für die Forschung im Tourismus unter Verwendung personenbezogener Geodaten von Fotos*" (2017, p. 51) supervised by Prof. W. Höpken, University of Applied Science Ravensburg-Weingarten, and co-examined by Prof. M. Fuchs.)

Service Section

Main Application Fields: Cluster analysis is an unsupervised machine learning technique aiming to build groups of similar cases. Its most popular field of application lies in customer segmentation, for example, for customer relationship management or for the analysis of web usage behavior (e.g., as input for website adaptation, targeting, or recommender services). Additionally, cluster analysis can solve classification tasks if no pre-classified training data are available (e.g., segmenting financial behavior into benign and suspicious categories). In the area of text mining, cluster analysis is often used in the form of keyword clustering so as to find topics within a natural language text. Finally, cluster analysis can be applied within the field of network analysis in order to identify, for instance, groups of people closely connected on a social network.

Limitations and Pitfalls: As clustering is an unsupervised machine learning technique, no concrete definition of what is right and what is wrong exists. Consequently, there are no absolute quality metrics, such as accuracy in the case of classification; thus, one can only judge whether one clustering

(continued)

approach performs better or worse on a given dataset than another. Besides applying such a mathematical evaluation, a cluster model also has to be evaluated from a semantic or business perspective. This makes it difficult to judge whether a cluster model constitutes a meaningful and reliable result.

Additionally, each clustering approach has its own set of limitations, especially when it comes to the form of found clusters. On the one hand, *k*-means can only identify hyper-ellipsoid clusters, while, on the other hand, *DBSCAN* can identify clusters of any shape but tends to form long and slender clusters, which might be inappropriate in certain application domains. Furthermore, *k*-means requires the process of predefining the number of clusters; thus, an inappropriate cluster number might lead to inappropriate results. *DBSCAN*, contrarily, requires careful specification of the neighborhood size, which, in some application domains, may be difficult to define.

Similar Methods and Methods to Combine with: Numerous alternative clustering techniques have been invented over time and are used in certain application domains. One thereof, which is quite well-known, is a specific form of *artificial neural networks*, so-called *self-organizing maps* (SOMs), for example, *Kohonen networks* (Bloom, 2004). Moreover, *k*-medoids or *x*-means are used as specific extensions of the *k*-means algorithm. Finally, the *Louvain algorithm* for community detection is a method to extract clusters (communities) from large networks (Blondel et al., 2008).

In general, cluster analysis is used on its own as an unsupervised machine learning technique, for instance, in the case of customer segmentation. Additionally, cluster analysis can serve as a dimension reduction technique (similar to factor analysis) or as a reduction of the search space as input for a downstream analysis, such as a classification or an association rule analysis.

Code: The RapidMiner workflows are available at: https://github.com/DataScience-in-Tourism/Chapter-8-Clustering

References

Baggio, R., & Klobas, J. (2017). *Quantitative methods in tourism: A handbook* (2nd ed.). Chanel View Publications.

Blondel, V., Guillaume, J.-L., Lambiotte, R., & Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment, 10*(P10008), 1–12.

Bloom, J. (2004). Tourist market segmentation with linear and non-linear techniques. *Tourism Management, 25*(6), 723–733.

Dietz, L. W., Sen, A., Roy, R., & Wörndl, W. (2020). Mining trips from location-based social networks for clustering travelers and destinations. *Journal of Information Technology and Tourism, 22*(1), 131–166.

Dolnicar, S. (2021). Market segmentation for e-Tourism. In Z. Xiang, M. Fuchs, U. Gretzel, & W. Höpken (Eds.), *Handbook of e-tourism*. Springer Nature. https://doi.org/10.1007/978-3-030-05324-6_53-1

Ester, M., Kriegel, H.-P., Sander, J., & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In E. Simoudis, J. Han, & U. Fayyad (Eds.), *Proceedings of the second international conference on knowledge discovery and data mining, KDD-96* (pp. 226–231). AAAI Press.

Everitt, B., Landau, S., Leese, M., & Stahl, D. (2011). *Cluster analysis* (5th ed.). Arnold Publishers.

Fuchs, M., Höpken, W., & Lexhagen, M. (2014). Big data analytics for knowledge generation in tourism destinations: A case from Sweden. *Journal of Destination Marketing and Management, 3*(4), 198–209.

Fuchs, M., Fossgard, K., Stensland, S., & Chekalina, T. (2021). Innovation and creativity in nature-based tourism: A critical reflection and empirical assessment. In V. Haukeland & P. Fredman (Eds.), *Nordic perspectives on nature-based tourism* (pp. 175–193). Edward Elgar Publishing.

Hair, J. F., Black, B., Black, W. C., Babin, B. J., & Aderson, R. (2014). *Multivariate data analysis* (7th ed.). New International Edition, Pearson Education.

Höpken, W., Fuchs, M., Keil, D., & Lexhagen, M. (2015). Business intelligence for cross-process knowledge extraction at tourism destinations. *Journal of Information Technology and Tourism, 15*(2), 101–130.

Höpken, W., Müller, M., Fuchs, M., & Lexhagen, M. (2020). Flickr data for analyzing tourists' spatial behavior and movement patterns: A comparison of clustering techniques. *Journal of Hospitality and Tourism Technology, 11*(1), 69–82.

Hu, Y., Gao, S., Janowicz, K., Yu, B., Li, W., & Prasad, S. (2015). Extracting and understanding urban areas of interest using geotagged photos. *Computers, Environment and Urban Systems, 54*, 240–254.

Hudson, S., & Ritchie, B. (2002). Understanding the domestic market using Cluster Analysis: A case study of the marketing efforts of Travel Alberta. *Journal of Vacation Marketing, 8*(3), 263–276.

Kanungo, T., Mount, D., Netanyahu, N., Piatko, C., Silverman, R., & Wu, A. (2002). An efficient *k*-means clustering algorithm: Analysis and implementation. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 24*, 881–892.

Kolas, N., Höpken, W., Fuchs, M., & Lexhagen, M. (2015). Information gathering by ubiquitous services for CRM in tourism destinations: An explorative study from Sweden. In I. Tussyadiah & A. Inversini (Eds.), *Information and communication technologies in tourism* (pp. 73–85). Springer.

Larose, D. T., & Larose, C. D. (2014). Discovering knowledge in data: An introduction to data mining, Chapter 10. In *Hierarchical & k-means clustering* (2nd ed., pp. 209–227). Wiley.

Liu, B. (2011). Web data mining: Exploring hyperlinks, contents and usage data, Chapter 4. In *Unsupervised learning* (2nd ed., pp. 133–168). Springer.

Lloyd, S. (1982). Least squares quantization in PCM. *Journal IEEE Transactions on Information Theory, 28*(2), 129–137.

Neuburger, L., & Egger, R. (2020). Travel risk perception and travel behavior during the COVID-19 pandemic 2020: A case study of the DACH region. *Current Issues in Tourism*. https://doi.org/10.1080/13683500.2020.1803807

Önder, I., Koerbitz, W., & Hubmann-Haidvogel, A. (2016). Tracing tourists by their digital footprints. *Journal of Travel Research, 55*(5), 566–573.

Pitman, A., Zanker, M., Fuchs, M., & Lexhagen, M. (2010). Web usage mining in tourism: A query term analysis and clustering approach. In U. Gretzel, R. Law, & M. Fuchs (Eds.), *Information and communication technologies in tourism* (pp. 393–403). Springer.

Pyle, D. (1999). *Data preparation for data mining*. Morgan Kaufmann Publisher.

Scholochow, C., Fuchs, M., & Höpken, W. (2010). ICT-efficiency and effectiveness in the hotel sector: A three stage DEA approach. In U. Gretzel, R. Law, & M. Fuchs (Eds.), *Information and communication technologies in tourism* (pp. 13–24). Springer.

Tan, P.-N., Steinbach, M., Karpatne, A., & Kumar, V. (2018). Introduction to data mining, Chapter 7. In *Cluster analysis: Basic concepts and algorithms* (2nd ed., pp. 525–612). Pearson Education.

Ward, J. H., Jr. (1963). Hierarchical grouping to optimize an objective Function. *Journal of the American Statistical Association, 58*, 236–244.