Information Systems and e-Business Management (2023) 21:527–570
https://doi.org/10.1007/s10257-023-00640-4

ORIGINAL ARTICLE

A review on customer segmentation methods for personalized customer targeting in e-commerce use cases

Miguel Alves Gomes[1] · **Tobias Meisen**[1]

Received: 25 April 2022 / Revised: 16 February 2023 / Accepted: 9 May 2023 /
Published online: 9 June 2023
© The Author(s) 2023

Abstract

The importance of customer-oriented marketing has increased for companies in recent decades. With the advent of one-customer strategies, especially in e-commerce, traditional mass marketing in this area is becoming increasingly obsolete as customer-specific targeting becomes realizable. Such a strategy makes it essential to develop an underlying understanding of the interests and motivations of the individual customer. One method frequently used for this purpose is segmentation, which has evolved steadily in recent years. The aim of this paper is to provide a structured overview of the different segmentation methods and their current state of the art. For this purpose, we conducted an extensive literature search in which 105 publications between the years 2000 and 2022 were identified that deal with the analysis of customer behavior using segmentation methods. Based on this paper corpus, we provide a comprehensive review of the used methods. In addition, we examine the applied methods for temporal trends and for their applicability to different data set dimensionalities. Based on this paper corpus, we identified a four-phase process consisting of information (data) collection, customer representation, customer analysis via segmentation and customer targeting. With respect to customer representation and customer analysis by segmentation, we provide a comprehensive overview of the methods used in these process steps. We also take a look at temporal trends and the applicability to different dataset dimensionalities. In summary, customer representation is mainly solved by manual feature selection or RFM analysis. The most commonly used segmentation method is k-means, regardless of the use case and the amount of data. It is interesting to note that it has been widely used in recent years.

Keywords Customer segmentation · Feature engineering · Customer targeting · Customer relationship management · RFM-analysis

Extended author information available on the last page of the article

1 Introduction

"*As the Internet emerges as a new marketing channel, analyzing and understanding the needs and expectations of their online users or customers are considered as prerequisites to activate the consumer-oriented electronic commerce. Thus, the mass marketing strategy cannot satisfy the needs and expectations of online customers. On the other hand, it is easier to extract knowledge out of the shopping process under the Internet environment. Market segmentation is one of the ways in which such knowledge can be represented and make it new business opportunities.*" (Kim and Ahn 2004). Already in 2004, Kim and Ahn (2004) described an essential paradigm shift that online marketing was encountering in a time in which the world wide web was rising. The statement focused on the limitation of mass marketing in a period where data-driven technological possibilities arose to analyze web-users footprints and enable personalized-oriented marketing. About two decades later personalized-oriented marketing is still a key challenge that many researchers conduct in their work (Chen et al. 2018; Apichottanakul et al. 2021; de Marco et al. 2021; Nguyen 2021; Sokol and Holy 2021). Not only has it been shown that personalized customer targeting is more profitable for companies, but also that knowledge about customer behavior is a decisive factor for success and failure (Mulhern 1999; Zeithaml et al. 2001; Kumar et al. 2008). In this respect, it is essential to understand the customers and their needs, and to be aware of their behavioral changes over time (Liu et al. 2009; Ding et al. 2019; Griva et al. 2021; Apichottanakul et al. 2021). In addition to technological changes and increasing functional requirements, legal regulations are also subject to constant change. This results in further non-functional requirements, as these regulations firstly describe local conditions and secondly can counteract the functional objectives (Burri and Schär 2016; European-Parliament 2016). From a functional perspective, companies that want to analyze customer behavior need (1) the capacity to record customer data, (2) an algorithm to characterize similar user behavior, and (3) strategies or processes that use the extracted information to achieve the business goal.

Regarding the first requirement, it is necessary to collect data that enable algorithm-based characterization of user behavior. Thereby, we distinguish between customer behavior data that is collected explicitly and implicitly. As the names suggest, explicit data collection is intentional to collect customers' information. In implicit data collection, the main purpose is not to collect information about customers, but to collect information about the process in which the customer appears as the interactant, such as purchase information for accounting purposes. Explicitly collected data such as demographic information, on the other hand, is difficult to collect and maintain for several reasons. Not all customers are willing to share demographic data or they browse anonymously on the web. In addition, information collected in this way is subject to change over time and, accordingly, is always subject to uncertainty that is difficult to quantify (Chan et al. 2011; Chen et al. 2018). Accordingly, implicitly gathered data is easier to collect. This data can be tracked with every user interaction. E.g. information about products

that are purchased together or the amount of money spent for a purchase. Nonetheless, data collected in such an implicit manner requires deeper analytical skills to exploit.

For the second requirement, the gathered interaction data is used. A frequently used approach for managing different customers with diverse preferences is segmentation (Hong and Kim 2012; Hsieh 2004; Chen et al. 2018). Customer segmentation is an unsupervised-learning process and utilizes different clustering approaches which have the goal to separate aforementioned customer data based on similarity. Hereby, similarity is measured by an objective function such as euclidean distance. It should be noted that customer behavior is a continuous process, with customer needs, wants and satisfaction changing over time. Accordingly, the processes and underlying procedures implemented in companies must be flexible in order to accommodate this high level of dynamism (Liu et al. 2009; Ding et al. 2019; Griva et al. 2021).

The last requirement is to utilize the analyzed customer information. Domain experts like marketers can tailor appropriate marketing strategies for individual customer groups based on segmentation. As Birtolo et al. (2013) already stated and showed, instead of domain experts, more and more automated methods to extract and to learn underlying patterns in customer behavior allow to target customers in advance.

The aforementioned dynamics are not only reflected in the respective target market but also can be observed in the underlying segmentation methods. Therefore, the goal of our survey is to provide an overview of digital and autonomous customer targeting processes for customer relationship management (CRM) based on historical data. The main objective of the literature research lies in the customer segmentation process for different e-commerce related use cases like retailing or services in the banking sector. Our study is structured by three guiding questions, to which we provide answers in this work.

1. Which clustering processes and methods are frequently used to understand customer behavior and targeting afterward?
2. Are there methodological limits with regard to data dimensionality?
3. Do methodological trends exist that can be observed over a period of two decades?

The main difference between our survey and former ones is that we focus on the process of customer targeting and behavior analysis in the e-commerce domain. The most recent literature review with a related topic is from 2016 (Sari et al. 2016). However, six years have passed since then, which makes an updated view necessary. Besides that in our study, we conduct a more extensive literature review that leads to a different classification of segmentation methods and more use case examples. In addition, we recognized a more extensive e-commerce process for customer targeting. Our contribution and main finding are:

- We provide an overview with examples from the literature of how customer behavior analysis is used.
- We determine a customer targeting process with four phases.
- We could not identify a consensus in metrics to evaluate and compare the quality of the segmentation algorithm and therefore it cannot be said which of the methods is "best".
- Based on the frequency in publications and ability to handle large amounts of data, we recommend a process that uses RFM-analysis as a feature representation and k-means for segmentation.
- We identify open questions and possible research gaps regarding embeddings for customer representation and deep learning-based segmentation for customer analysis and customer targeting strategies

Our study is structured as follows: In Sect. 2, we present and explain our research methodology. In Sect. 3, we present a literature overview of the identified works. Hence, in this section, we address the first guiding question accordingly and provide an answer. Moreover, we present the survey literature more in-depth. Based on the identified process, we notice that feature selection (be it manual or computerized) is an essential preprocessing step of customer behavior segmentation. Therefore, we explain the different segmentation and feature selection methods that are used. Additionally, the methods in the surveyed literature are described regarding the applied use cases for customer targeting and data volume. Section 3 ends with an overview of the publications' evaluation metrics for customer segmentation. We analyze and discuss our findings in Sect. 4 which is further divided into two subsections. The first subsection is about the feature selection. In terms of feature selection methods, we present an answer to guiding question two and three. Similarly, in the second subsection we analyzes, discusses, and answers guiding questions two and three regarding the reviewed segmentation methods. In each subsection, we state open research questions that are not covered by our survey but have future potential. Finally, we conclude the survey in Sect. 5 with a brief summary of the findings and new open research questions and potential.

2 Literature research methodology

As already encouraged in the introduction we want to scientifically investigate which processes exist for personalized customer marketing approaches. Especially, to get an overview of commonly used customer segmentation methods in the context of CRM in e-commerce, we have conducted an extensive literature review. Thereby, Vom Brocke et al. (2015) published a recommendation on how to conduct such a search in an effective and highly qualified way. Hence, we followed their recommendation for the most part. Figure 1 illustrates our review process. We started our literature research by reading survey papers to derive an integrated and consolidated understanding of the conceptualization of the subject. Thereafter, we started the literature search. Therefore, we defined our search scope. Vom Brocke et al. (2015)

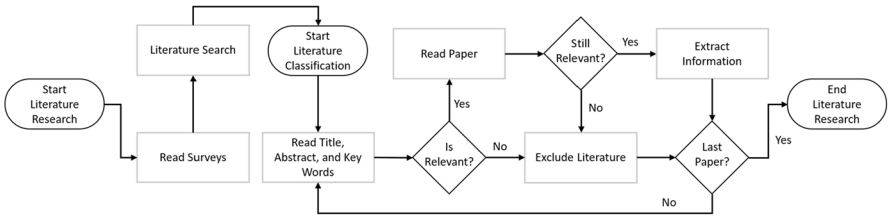


Fig. 1 Flow chart of the literature research process

refer to Cooper (1988) which states four steps on how to define a search scope: (1) process, (2) sources, (3) coverage, and (4) technique. Leaning on these four steps we choose a sequential search process. As a publication source, we used the Web of Science[1] (WoS) online research tool as it is one of the leading scientific citation and analytical platforms and provides scientific publications across a wide amount of knowledge domains (Li et al. 2018). To keep the focus on the customer segmentation methods we used the following search term:

- "Customer segmentation" or "customer clustering" or "user segmentation" or "user clustering"

Herein, we chose to use the word "user" as a synonym for "customer" and "clustering" for "segmentation". We wanted the search to be as less restrictive as possible to not miss relevant publications. Therefore, we expected works that are not relevant to our research. After having a corpus of hundreds of publications, we started reading the title, abstract, and keywords of the publications. We filtered out all publications that did not deal with customer behavior in commerce, especially in the context of e-commerce. The next step was to read all remaining papers and excluded all publications that did not deal with customer segmentation in an e-commerce use case and it became apparent that customers were segmented based on their information and actions. We extracted all wanted information from publications we classified as relevant. Specifically, we retrieved bibliometric information, information about the use case, the used methods, information about the used data, and the results.

3 Literature overview

As aforementioned in Sect. 2, we started our literature review with reading related surveys. Plenty of research surveys in the field of segmentation prioritize the underlying methodology or class of methods but not their usage in specific domain (Gennari 1989; Rokach 2010; Hiziroglu 2013; Ben Ayed et al. 2014; Firdaus and Uddin 2015; Reddy and Vinzamuri 2018; Shi and Pun-Cheng

[1] https://www.webofscience.com/.

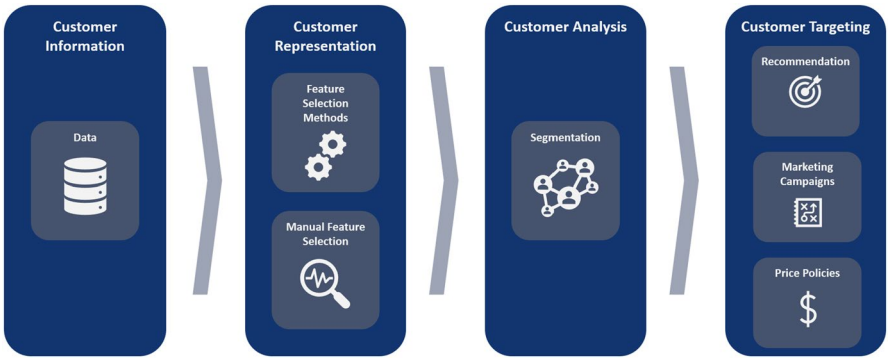


Fig. 2 Process of customer targeting based on behavioral information gathered from data

2019). For example, Shi and Pun-Cheng (2019) review clustering methods for spatiotemporal data which are collected in diverse domains like social media, human mobility, or transportation analysis. Another survey example is brought by Hiziroglu (2013). The author reviews segmentation approaches for applications of soft computing techniques. Other surveys or studies focus on specific methods like k-means or RFM-analysis (Sarvari et al. 2016; Deng and Gao 2020). The most related literature review we found in our literature search is from Sari et al. (2016) which reviews customer and marketing segmentation methods and the necessary data. They identify different segmentation approaches and e-commerce process which coincides in some parts with our outcomes. However, as already mentioned before, six years have already passed and their paper corpus consist of less than 20 publications. From this, we deduce the need for an up-to-date and more detailed review in the area of customer segmentation in e-commerce.

The WoS search from 2023/01/01 led to 852 publications, of which not all were related to our research as assumed. As described we excluded all publication that did not deal with customer behavior in e-commerce. The major domain that was not related to our research objective dealt with user segmentation in non-orthogonal multiple access (NOMA) techniques. Over half (66%) of the publications were not related to our research topic and we had 289 publications left that were somehow e-commerce related. From the 289 publications, we classified 149 publications as “not relevant” and 140 publications as “relevant” based on the title, abstract and keywords with the aforementioned criteria.

Reading the remaining literature (140 publications), we paid particular attention to recurring processes. We identified a process that is constantly used to determine customer behavior with segmentation approaches. Figure 2 illustrates the identified process that depicts the answer to our first question. It illustrates the customer targeting process and it can be divided into four steps: (1) customer information, (2) customer representation, (3) customer analysis, and (4) customer targeting.

In the first step, the customer information is stored in form of data and is made available for further processing. In the literature, this step is usually given by

provided datasets. Nevertheless, in some publications, the information is collected by the researcher. Especially, when data is collected explicitly which is for example done by Hong and Kim (2012), Nakano and Kondo (2018), and Wu (2011).

Based on the collected information a customer representation is built as the second step. The customers are represented by their features which are selected manually or with a feature selection method. In nearly half of the cases (47.6%), features are selected manually and in the other half (52.4%) feature selection methods are used. Both feature selection approaches have their advantages and disadvantages. For example, feature selection methods are utilized to eliminate features with less information content or to aggregate and extract additional knowledge out of the customer data. The most used method in our literature review is the Recency, Frequency, Monetary (RFM) analysis that aggregates additional information about the customers' behavior and value to a company (Hughes 1994) which we show in Sect. 4. Manual feature selection usually is performed by extracting information like item view or click events, purchased items, and item information such as the associated category. In some other cases, mostly for recommendation, the authors additionally use ratings and reviews for the behavior analysis. Otherwise, demographic data is collected through membership or similar programs. Another approach to get demographic or psychographic information is by user surveys.

The third step of the found process is customer analysis which is the key component of the process and is done by applying segmentation methods. Customers are split into more homogeneous groups of similar behavior. This is done by different segmentation approaches, like methods that compare the similarity between the customer representation or other methods that partition the customers by given thresholds. In Sect. 3.4, we further explain the interaction of customer representation with feature selection methods and the customer analysis on found case studies.

The fourth and last step, customer targeting, uses the behavioral information from the customer analysis to target the right user with the right CRM decision. In the literature, we identified different targeting approaches which includes recommendation, marketing campaigns, and pricing strategies. The main difference in the literature is that recommenders are evaluated against others with evaluation metrics like hit-rate, accuracy, etc, and marketing campaigns or pricing strategies focus on the plausibility of the customer segmentation and try to explain the outcomes over the performance.

We decided to consider only literature that mostly adheres to this characteristic process because it fulfills all necessary conditions for personalized customer marketing which is our defined investigation scope. The work of Coussement et al. (2014) is an example of a scientific publication which we did not consider in our work because it is not in our scope. In their research, they investigated the impact of data quality on different segmentation methods and showed which methods are more robust to inaccuracies.

Based on this aforementioned method, we further filtered our corpus to obtain a final corpus of 105 scientific papers. The literature is distributed between the years 2000 and 2022 over different use cases and journals. The reviewed publications are not equally distributed over the years. Figure 3 illustrates the distribution of the paper's publication year. We see that there are more publications over time in

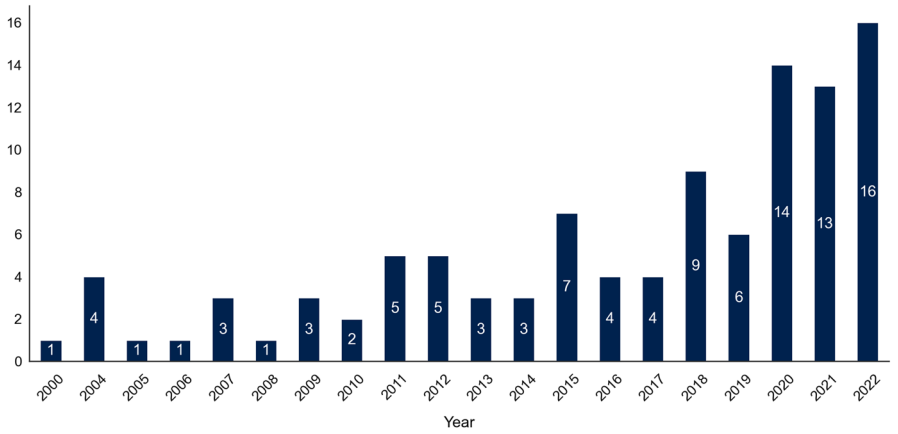


Fig. 3 Distribution of surveyed publications from 2000 until 2022

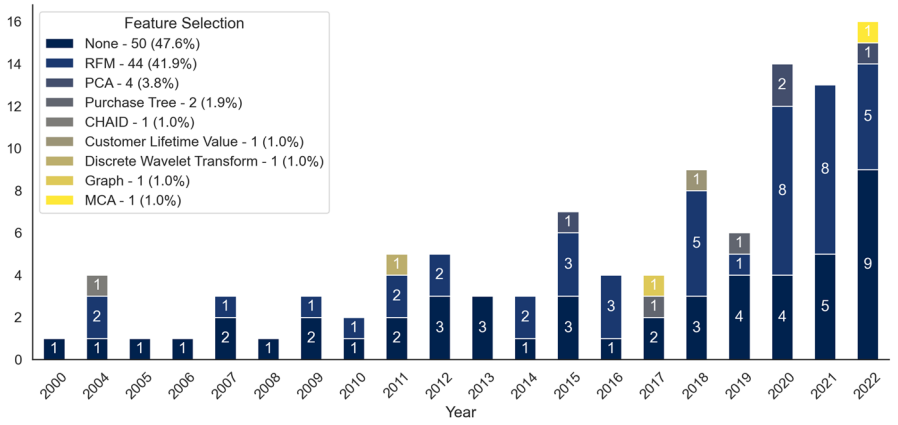


Fig. 4 Distribution of the surveyed feature selection methods over the years

the field of e-commerce considering customer analysis with segmentation methods. Before 2010, we usually find one publication per year. In the period from 2001 to 2003, however, there is no publication in the paper corpus at all. In total, there are 16 publications in the period from 2000 to 2010. After 2010, there are at least three publications per year with an increasing tendency. 43 out of 105 publications (about 41%) are published in 2020, 2021 and 2022.

Table 4 gives an overview of the 105 publications containing title, author, and year.[2]

[2] Table 4 can be found in the Appendix 1.

3.1 In-depth feature selection methods for customer representation

We identify customer representation as a fundamental step in the customer targeting process. Therefore, before applying segmentation methods for customer analysis an appropriated customer representation is needed. As mentioned earlier, this is achieved by applying feature selection methods. In the following, we will refer to manual feature selection as "none" feature selection method. Figure 4 displays the distribution of the used feature selection methods over the years as well as the total amount in percentage. In 50 publications, the authors decide to use handcrafted features to represent the customers.

The *RFM-analysis* is by far the most popular feature selection method with 44 (80%) of 55 publications that use feature selection methods and 41.9% in total. In the RFM-analysis three features are extracted from customer data. The features are recency, frequency, and monetary. Recency relates to the time of the last user activity, like a purchase. Frequency describes how often a customer interacts in a given period and monetary measures how much money a customer spends in that period (Hughes 1994). In some works, e.g., Stormi et al. (2020), Chang and Tsai (2011) the RFM-analysis is extended by additional features.

Principal component analysis (PCA) is applied in four publications. In 2015 and 2022 once and in 2020 twice. PCA is a dimensionality-reduction method in which the information content of the features is determined and features with low information content can be removed (Pearson 1901; Hotelling 1933).

Purchase Tree are used in two publications and were proposed by Chen et al. (2018). The fundamental idea is to represent purchased products by a tree in which products are the leafs and the product category the nodes of the tree.

The remaining five feature selection methods are each only used once. *Chi-square Automatic Interaction Detectors (CHAID)* is based on decision trees to handle categorical variables (Kass 1980). *Customer Lifetime Value* is a popular economic key performance indicators which describes the profit of the customer for the entire lifetime. *Discrete wavelet transform* captures location and frequency information. In *Graph* representation, the customer interaction is encoded in such. *Multiple correspondence analysis (MCA)* allows the representation of categorical features in lower-dimension.

3.2 In-depth customer segmentation methods

The authors of the reviewed publications utilize different customer segmentation methods for the customer targeting process. Figure 5 shows the distribution of segmentation methods among all publications and over the years.

K-means is the most frequently used customer segmentation method in our surveyed literature (41 of 105). The goal of the k-means algorithm is to partition a set of data points into k segments which minimize the distance between the data. Usually, the euclidean distance is used. Solving the underlying optimization problem is NP-hard and therefore, various approximation algorithms are used (MacQueen 1967;

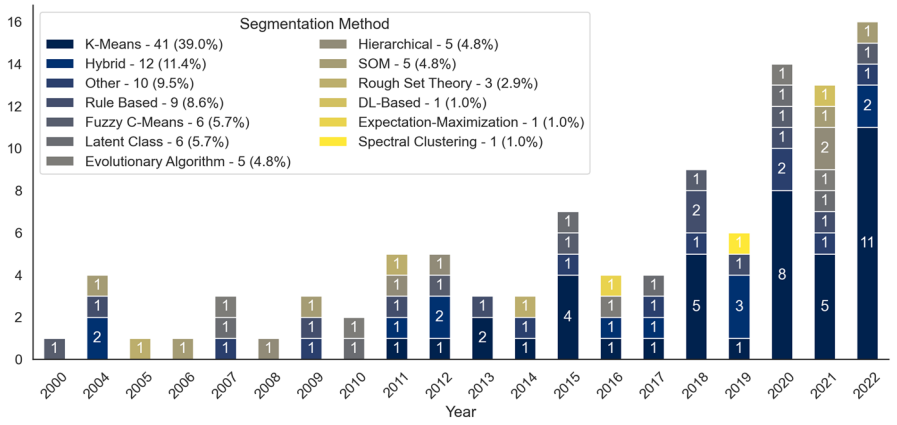


Fig. 5 Distribution of the surveyed clustering methods over the years of publication

Lloyd 1982). The usage of k-medoids and other k-means variations is included in our k-means classification.

The second most used segmentation algorithms are *Hybrid* approaches that are used twelve times (11.4%), followed by *Other* approaches that are used ten times (9.5%). Hybrid clustering refers to the application of two or more clustering approaches to segment the customers. As "other" clustering, we define the clustering methods which don't fit the previous cluster definitions. For example, Abbasimehr and Shabani (2021) propose a time series segmentation approach to get knowledge from customer behavior or Chen et al. (2018) proposed an segmentation an algorithm which they call PurTreeClust. Hsu and Chen, Y.-g.C. (2007) propose an algorithm to cluster mixed data which is named CAVE and An et al. (2018) proposes a segmentation algorithm based on non-negative matrix factorization.

Nine publications use *Rule-based clustering* to segment their customers into different behavioral groups. In rule-based approaches, data points are assigned to predefined segments by value thresholds.

In our surveyed literature, five publications utilize a *Fuzzy C-Means (FCM)* approach. In a fuzzy clustering algorithm, data points can be assigned to different clusters at the same time. The fuzzy c-means (FCM) clustering algorithm is a fuzzy version of the k-means algorithm (Dunn 1973; Bezdek et al. 1984).

Latent class models are used for the latent class analysis to classify discrete variables (Lazarsfeld 1950). This segmentation approach is used six times in the surveyed literature.

Evolutionary Algorithm (EA) are inspired by the biological evolution of living things. EAs are a class of optimization methods to find an approximate solution to a problem which also includes clustering. Simplified, the algorithm can be described as follows. In the first step, a random solution is initialized. The second step is to determine the quality of the solution using a fitness function. In the third step, the best solutions are selected and these are randomly changed, which is also referred to as mutation in this context. This process is repeated until a stopping criterion is

met (Eiben and Smith 2003; De Jong 2016). Genetic algorithms (GA) like particle swarm optimization (PSO) (Kennedy and Eberhart 1995) or chaotic ant swarms (CAS) (Zhu et al. 2007) also belong to the family of EAs. Our survey contains five publications that utilize EAs.

Hierarchical clustering is utilized five times by the authors of the surveyed litterateur. The basic idea of hierarchical clustering is to bring similar data points close to each other regardless of the distribution. There are two approaches, known as the agglomerative and divisive approaches. In the agglomerative approach, the algorithm starts with each data point being in its own cluster. At each iteration step, the most similar clusters are merged until a distance criterion is met. The divisive approach works similarly, except that it starts with one cluster and splits in each iteration (Maimon and Rokach 2005).

Self-organizing map (SOM) is also used five times in our paper corpus and are based on neural networks. Neural networks are mainly used for supervised learning tasks. However, it is also possible to use neural networks in an unsupervised manner for clustering by pushing fully connected neurons towards the data points that are closest to them (Kohonen 1982).

The *Rough Set Theory* was introduced by Pawlak (1982) and is a data mining method to extract knowledge of databases. Besides the use for segmentation, the rough set theory can also be used for feature selection, data reduction, and other applications. In our research, we found three publication utilizing rougth sets to segment the customers.

Deep learning (DL)-based clustering, *spectral clustering*, and clustering via *expectation-maximization* are only used once. Similar to SOMs, deep learning-based clustering methods are based on neural networks. Nguyen (2021) presents a deep learning-based clustering approach named Deep Embedding Clustering that combines a deep neural network and a self-supervised probabilistic clustering technique. They state that their approach produces explainable customer segments. In the first step, they determine the optimal number of clusters with a spectral clustering approach and the elbow method. Then they encode their manually selected variables and apply the deep embedding clustering which is a deep autoencoder that is trained with the mean squared error (MSE) loss. The expectation-maximization (EM) algorithm performs a maximum likelihood estimation on given data points which consists of latent variables. It is an iterative approach that optimizes the mean and variance of the cluster distribution until it converges (Dempster et al. 1977). Spectral Clustering is a graph-based clustering approach in which distances between data points are represented by the edges. With the resulting graph's Laplacian-matrix segments can be computed (Fiedler 1973; Donath and Hoffman 1973).

In the first decade (2000–2010) rule-based, Evolutionary Algorithms (EA), latent class, hybrid, and "other" clustering approaches were used twice. Both hybrid approaches were published in 2004. One hybrid approach combines k-means with a EA and the other combines a hierarchical approach with k-medoids. Hierarchical, fuzzy C-means, and rough set theory segmentation approaches are used once in the years between 2000 and 2010. Self-organizing map (SOM)-based segmentation was used three times which makes it the most applied method in this decade in our survey.

In the second decade (2011–2022), 89 of 105 (84.76%) relevant papers were published. K-means is used for the first time in 2011 (disregarding hybrid approaches). Since then, k-means has been used at least once a year. In 2014 k-means is used in two, 2015 in four, 2018 in five, 2020 in eight, and 2022 in eleven publications. Statistically, this indicates an upward trend. Also rule-based approaches are used repeatedly in the last years.

3.3 Overview customer targeting use cases

The underlying customer targeting process applies to a large amount of business and e-business use cases. In this section, we present an overview of which segmentation methods are used on which use case. Therefore, we briefly introduce the found e-commerce use cases.

The first category of use cases we want to introduce is *Retailing*. It is the sale of different goods that are not further specified and don't belong to any other use case category. We also assign use cases to this category if it is not further specified. This means that a pure sports retailer is classified under the *Sports* use case, or a retailer that sells only clothing is classified under *Fashion*. Different to retailing, fashion is a dynamic industry (Brito et al. 2015). Like the fashion branch, *Electronic* is considered as a branch of e-commerce retailing. In the literature, some customer behavior segmentation use cases are related to *Banking*. Use cases in this category naturally have more information about the customer. In addition, the products and services don't change as quickly as in retailing. In *Mobile operators'* use cases the authors deal with data from mobile network providers. With Youtube, Netflix, and other companies, *Video & music* streaming platforms and services become very popular, and forecasts show that sales will also grow strongly in the coming years (statista.com 2022). In our literatur search we found some *Book* use cases that deal with book retailing or renting services. Nowadays, there a plenty of online services to plan a trip. In *travel* use cases we consider case studies that deal with trip-related action like hotel booking, reviewing, or trip and location recommendation. In our survey *food* use cases get their own category because in some cases it is difficult to distinguish between food retailing, restaurant reviews, or food production or manufacturing. *Manufacturing* in e-commerce comes with some benefits and new opportunities. One is product customization (Fan and Huang 2007). Another one is manufacturing-related services. In *others* use cases, we classify use cases that we could not determine explicitly or don't fit in one of the other groups, e.g. online news or email campaigns of charitable organizations.

Table 1 shows in the rows all clustering method used in the literature. Each column represents one use case. A check mark indicates that we were able to identify an example for at least one use case. The number of checkmarks indicates the number of use cases we identify for the segmentation methods. Note, that in some publications, the utilized method is showcased on multiple use cases which leads to a mismatch between the number of publications and the number of use cases.

Retail is the most occurring use case in the surveyed litterateur with 43 case studies. The authors show with their publications that every segmentation method is

Table 1 Use cases and utilized segmentation approaches for customer targeting of commerce business from the survey literature

Segmentation methods	Retail	Bank	Mobile operator	video & music	Book	Sports	Automobile	Fashion	Travel	Food	Electronic	Manufacturing	Other
Rule-based	✓✓✓✓		✓					✓	✓			✓	✓
K-means	✓✓✓✓✓✓ ✓✓✓✓✓✓ ✓✓✓✓	✓✓✓	✓	✓✓✓	✓✓	✓	✓✓	✓✓	✓	✓✓	✓		✓✓✓
Fuzzy C-means	✓		✓	✓			✓		✓	✓	✓		
Hierarchical	✓✓	✓			✓				✓			✓	
Rough Set Theory	✓												✓
Latent class	✓✓✓✓								✓	✓			
Expectation-maximization	✓												
Spectral clustering	✓												
Evolutionary Algorithm	✓						✓✓		✓				✓✓
Self-organizing map	✓✓	✓							✓				✓
Deep learning-based	✓												
Hybrid clustering	✓✓✓✓✓	✓✓		✓					✓				✓✓
Other approaches	✓✓✓✓	✓			✓					✓			✓

usable to approach retail use cases. The retail use case is the only one that have examples for each segmentation method. Besides retail use cases, only travel and "other" use cases are approach by most of the segmentation methods for the customer analysis. The remaining use case categories have at least five different segmentation methods as an application example.

Regarding use case coverage, we found that k-means clustering are used to approach all use cases expect manufacturing. Thereby, k-means is utilized 16 times to approach retail and three times in bank, video and music, and "other" use cases each. Our literature review show that FCM is applied to seven different use cases. Rule-based, hierarchical, hybrid, and "other" segmentation approaches are applied on five different use cases.

3.4 Overview and examples of the interplay between customer representation and analysis for customer targeting use cases

The authors of the identified publications utilize different customer segmentation methods with different feature selection methods for the customer targeting process. In this section, we further investigate and describe these approaches to give a better insight into the interaction of the feature selection and segmentation methods. Table 2 provides an overview of the different segmentation methods with the corresponding feature selection approaches used. It also lists the number of times such a pair of segmentation method and feature selection was used in the paper corpus. The last column of the table shows the publication's reference. In the following, we present some examples on how the different segmentation and feature selection methods are used in the found literature to approach customer targeting in e-commerce.

In nine publications rule-based clustering is used to segment the customers into different behavioral groups. Therefrom, seven use the RFM-analysis to represent their customers. An example retail use case that combines RFM-analysis and k-means is provided by Hsu and Huang (2020). In their research they want to identify VIP customers. VIP customers are buyers of critical products which are not purchased by the average customer. In their approach, they apply the RFM-analysis on over 600,000 transactions from around 3800 customers. The segmentation is based on the 20%-quantile of th RFM-values. Another example which utilzes rule-based segmentation with RFM-analysis is from Jonker et al. (2004). In their publication, the authors want to find the best marketing policy out of a set of policies for a customer. The data are from a mailing scenario of a charitable organization. They first utilise an on the email data adapted RFM-analysis and segment the customers based on defined thresholds. To identify the the best policy for a segment the authors used a markov decision process.

Two authors applied rule-based segmentation without applying a feature selection method. Hjort et al. (2013) want to investigate the impact of product returns in a fashion use case provided by Nelly.com which is a Scandinavian online fashion retailer. For the research, the scientists selected six features for each customer which are total sales, average sales per order, total contribution margin, average contribution margin, the total number of orders, and the total number of returns. Based on

Table 2 Segmentation and used feature selection methods with corresponding references of the survey literature

Segmentation method	Feature selection	Count	References
Rule-based	None	2	Boettcher et al. (2009), Hjort et al. (2013)
	RFM-analysis	7	Jonker et al. (2004), Chang and Tsai (2011), Hiziroglu et al. (2018), Wong and Wei (2018), Stormi et al. (2020), Hsu and Huang (2020), Sokol and Holy (2021)
K-means	None	16	Zhang et al. (2014), Abdolvand et al. (2015), Brito et al. (2015), Liu et al. (2015), Hafshejani et al. (2018), Bai et al. (2019), Deng and Gao (2020), Griva et al. (2021), Zhang et al. (2020), Alghamdi (2022b), Araujo et al. (2022), Chalupa and Petricek (2022), Zhang and Huang (2022), Gautam and Kumar (2022), Griva (2022), Tabianan et al. (2022)
	RFM-analysis	21	Chan et al. (2011), Peker et al. (2017), Akhondzadeh-Noughabi and Albadvi (2015), Ravasan and Mansouri (2015), Sarvari et al. (2016), Dogan et al. (2018), Alberto Carrasco et al. (2019), Christy et al. (2018), Guney et al. (2020), Lam et al. (2021); Pratama et al. (2020), Sivaguru and Punniyamoorthy (2021), Rahim et al. (2021), Wu et al. (2020), Wu et al. (2021), Zhao et al. (2021), Bellini et al. (2022), Mensouri et al. (2022), Mosa et al. (2022), Wu et al. (2022), Kanchanapoom and Chongwatpol (2022)
	PCA	3	Nie et al. (2021), Tsai et al. (2015), Umuhoza et al. (2020)
	Graph	1	Ding et al. (2019)
Fuzzy C-means	None	1	Ozer (2001)
	RFM-analysis	3	Wang (2010), Safari et al. (2016), Munusamy and Murugesan (2020)
	Customer lifetime value	1	Nemati et al. (2018)
	PCA	1	Alghamdi (2022a)
Hierarchical clustering	None	3	Li et al. (2009), Hsu et al. (2012), Wang and Zhang (2021)
	Discrete wavelet transform	1	Aghabozorgi et al. (2012)
	RFM-analysis	1	Zhou et al. (2021)
Rough Set Theory	None	2	Song and Shepperd (2006), Wu (2011)
	RFM-analysis	1	Dhandayudam and Krishnamurthi (2014)
Latent class	None	4	Teichert et al. (2008), Goto et al. (2015), Nakano and Kondo (2018), Valentini et al. (2020)
	RFM-analysis	2	Wu and Chou (2011), Apichottanakul et al. (2021)

Table 2 (continued)

Segmentation method	Feature selection	Count	References
Expectation-maximization	RFM-analysis	1	Rezaeinia and Rahmani (2016)
Spectral clustering	Purchase Tree	1	Chen et al. (2019)
Evolutionary Algorithm	None	3	Wan et al. (2010), Sivaramakrishnan et al. (2020), Krishna and Ravi (2021)
	RFM-analysis	2	Chan (2008), Chan et al. (2016)
Self-organizing map	None	2	Nilashi et al. (2021); Verdu et al. (2006)
	RFM-analysis	3	Hsieh (2004), Liu et al. (2009), Liao et al. (2022)
Deep learning-based	None	1	Nguyen (2021)
Hybrid clustering	None	10	Wang and Shao (2004), Kang et al. (2012), Bian et al. (2013), Hong and Kim (2012), Ma et al. (2016), Ramadas and Abraham (2018), Logesh et al. (2020), Wang et al. (2020), Barman and Chowdhury (2019), Griva et al. (2022)
	CHAID	1	Kim and Ahn (2004)
	MCA	1	Jadwal et al. (2022)
Other approaches	None	6	Hsu and Chen, Y.-g.C. (2007), Jiang and Tuzhilin (2009), Rapecka and Dzemyda (2015), An et al. (2018), Madzik and Shahin (2021), Dogan et al. (2022)
	Purchase Tree	1	Chen et al. (2018)
	RFM-analysis	3	Hu and Yeh (2014), Abbasimehr and Shabani (2021), Simoes and Nogueira (2021)

the feature information, they assign each customer to one of four groups. The groups are based on the buying and returning habits of the customers. The authors conclude from the customer analysis, that customers who tend to return goods are also the more valuable for the company.

In 16 publications the authors decide to not use a feature selection method but select features by hand before applying k-means clustering to the customer data. Authors of 21 publications use the value of RFM-analysis for the segmentation with k-means. Three research groups use a principal component analysis (PCA) for feature selection before clustering with k-means. Only Ding et al. (2019) use a graph representation before segmentation. The graph is built based on user-item interactions.

Griva (2022) analysis the customer of 140 e-commerce stores in European countries with k-means and hand crafted features. The features are extracted from 270,000 responses from a customer satisfaction survey and 1 million orders from 800,000 customers. They propose a framework which is capable to build automated marketing actions based on the created customer satisfaction segments. Example for such marketing actions are social media sharing strategies for the satisfied segments or discounts for the less satisfied customer segments.

Guney et al. (2020) are looking for the best campaign in movie rental use case (video on demand). In a first step they apply an modified RFM-approach which extract two additional features from the data. The two features are the number of days between the first and last rental and the standard deviation of the days between two rented movies. These five features are clustered via a k-means algorithm. The clustering results in four customer groups. An apriori algorithm namely an association rule mining approach is than used to assign the best marketing campaign to the customer segment.

In our selected literature six publications utilize an FCM approach. Ozer (2001) collects the data from customers of an online music service via a customer survey and doesn't use a feature selection method before applying FCM on the features.

Nemati et al. (2018) search for the most appropriated marketing strategy for the customers of a telecommunication industry use case. First, they compute the customer lifetime value (CLV) for each customer and group them with FCM. To assign the right marketing strategy to the right segment they utilize a fuzzy TOPSIS technique.

For hotel businesses, customers' satisfaction is crucial. Alghamdi (2022a) investigate customers' satisfaction of hotel visitors in Mecca and Medina (Saudi Arabia). Therefore, they apply PCA on data collected from TripAdvisor and segment the resulting features via FCM.

Hierarchical clustering is used in five publications. Three authors handcraft their features. Aghabozorgi et al. (2012) calculate the necessary features by applying a discrete wavelet transformation (DWT) on customer data of a bank use case. In their research, DWT is an appropriate approach because they consider customer activities as a time series which is not the norm. After using DWT on the data, the data is initially segmented with a hierarchical clustering method. The cluster is updated incrementally in a given period with new data. Zhou et al. (2021) combines hierarchical clustering with an extended the RFM-analysis for a retail use case. The

RFM-analysis is extended by the interpurchase time which results in four different features. The interpurchase time is defined as the time gap between two consecutive purchases in the same location (same website). Afterwards, the customers are clustered by the calculated features.

In our research, we have one publication from Dhandayudam and Krishnamurthi (2014) that combines RFM-analysis for feature selection with rough sets for clustering. In addition, they add another feature to the RFM-values that describes the average time between purchase and payment. They categorize all four features in their 20%-quantiles and then utilize a slightly modified rough set theory approach for the clustering. Song and Shepperd (2006), Wu (2011) don't use feature selection methods before segmenting the customers with a rough set approach.

Clustering based on latent class models is used six times in the surveyed literature. Four of them manually select the features and therefore, don't use feature selection methods. Nakano and Kondo (2018) use psychographic, demographic, online store, social media, and device touchpoint data. The information is clustered with a latent class analysis approach which results in seven segments. Goto et al. (2015) propose a method based on latent class analysis that clusters items and customers. They assume valuable users purchase more often only browsing and valuable products are bought more often. They use the latent class model to cluster the customers into "good users" and "other users". To analyse the resulting segments they use the Classification and Regression Tree (CART) Algorithmus.

Wu and Chou (2011), Apichottanakul et al. (2021) use RFM-analysis for the feature selection and apply a latent class approach for the clustering. Apichottanakul et al. (2021) use the proposed GRFM approach from Chang and Tsai (2011) to analyse the customers of a pork processing use case. First, the RFM scores are calculated for nine product categories and each feature is categorized in one of five categories based on the 20%-quantile. The features are clustered with a probabilistic latent class model. Apriori the optimal number of k is unknown therefore, a suitable number of clusters is determined with the Akaike Information Criterion (Akaike 1974). In the last step, the clusters are analyzed with the help of the RFM-values.

The only publication that uses the EM algorithm for clustering is from Rezaeinia and Rahmani (2016). The goal of their work is to recommend products in a retail use case. Therefore, they first compute the features via RFM-analysis and cluster them with an EM approach for customer targeting.

Spectral clustering is used by Chen et al. (2019) to segment customers buying behavior. Therefore, they use a Purchase Tree representation for customers transactions which was proposed earlier by Chen et al. (2018). For the customer segmentation, they propose a two-level subspace weighting spectral clustering algorithm. Spectral clustering approaches are used only once in our literature.

Our survey contains five publications that utilize EAs for customer clustering of which two use RFM-analysis and three don't use a feature selection method on the available data. Both publications using RFM-analysis are published by or with Chu Chai Henry Chan. In his publication from 2007, the task is to determine an appropriated strategy for each customer of an Nissan automobile retailer. Therefore, Chan (2008) computes the features from the RFM-analysis and categorizes the values in one of five 20%-quantiles. Then the features are binary

encoded with four bits. Based on the binary features a GA is used with the customer lifetime value (CLTV) as the fitness function. In 2016, Chan et al. (2016) apply the same feature preprocessing and PSO with CLTV as fitness function on a similar use case but with more data.

SOMs are used in five publications in total. Verdu et al. (2006); Nilashi et al. (2021) utilize handcrafted features to represent the customers. In the remaining three publications customers are represented by their RFM-values. For example, Hsieh (2004); Liu et al. (2009) combine an RFM-analysis feature extraction with a SOM clustering to segment the customers in their case study. A recent example of an SOM approach is proposed by Liao et al. (2022). They develop different marketing strategies for each segment for a retail use case. Therefore, they use an extended RFM-analysis approach to represent the customers. The extension is not only using RFM-analysis on customer purchase information but also on other behavioral information like clicks, add-to-cart, or add-to-favorite. For this, they utilize 2 million customer interaction records. The SOM approach is than applied on the different RFM-values of the customers to segment them in similar behavioral groups.

Nguyen (2021) presents a deep learning-based clustering approach named Deep Embedding Clustering that combines a deep neural network and a self-supervised probabilistic clustering technique. They state that their approach produces explainable customer segments. In the first step, they determine the optimal number of clusters with a spectral clustering approach and the elbow method. Then they encode their manually selected variables and apply the deep embedding clustering which is a deep autoencoder that is trained with the mean squared error (MSE) loss.

In our literature review, we found twelve research papers that use hybrid clustering methods. In ten publications no feature selecteion method is used. For example, Kang et al. (2012) don't utilize a feature selection. They split the dataset into two sets of answering customers and not answering customers. The data points are clustered with a k-means and CSI Algorithm with different criteria. Kim and Ahn (2004) use (CHAID) as a feature preprocessing. The clustering is performed by a GA based on k-means clustering. Jadwal et al. (2022) use MCA as feature preprocessing and segment the customers of a bank use case with an segmentation approach based on k-means and hierarchical clustering.

In our survey, we classified ten publications as "other clustering". Six authors have manually selected features. In three publications the RFM-analysis is used as feature selection method. For example, Abbasimehr and Shabani (2021) propose a time series clustering approach to get knowledge from customer behavior. First, they split the dataset into predefined time intervals. As a second step, they apply RFM-analysis on each interval and use the monetary value of the customer for the time series. On the resulting time series, a time series clustering approach is applied. Also, Hu and Yeh (2014) utilize RFM-analysis based features for the clustering. Therefore, they propose an RFM-pattern-tree to represent customers which also is used to approximate customers with less information. They can use this to detect similar customers with similar behavior. Simoes and Nogueira (2021) uses RFM-features and segment the customers with an ABC curve segmentation. Chen et al. (2018). represent the data as a Purchase Tree and propose for the segmentation

an algorithm which they call PurTreeClust and is based on a partitional clustering algorithm.

3.5 An overview of the data dimensionality in the publications' experiments

An essential component for the behavior analysis and customer targeting process is the information that is collected by the companies. In this section, we describe which methods are used for which data in respect to the order of magnitude. We distinguish between two different types of data amount. The first is the number of data points e.g. transactions and it describes the amount of data an algorithm can handle at least. The second type is the number of customers in a dataset.

The number of customers may indicate how much data an algorithm can process because customers and not data points are segmented. Therefore, it is important to consider the number of customers when analyzing the data dimensionality. Depending on the number of customers the number of data points can be reduced after a feature selection method. For example, in RFM-analysis the information of a user is aggregated for one period which leads to fewer data points the clustering algorithm needs to process. Other feature selection approaches like PCA doesn't affect the number of data points or user but the number of features.

Table 3 shows which feature selection methods and clustering algorithms are used with which data dimensionality regarding the number of data points and the number of customers in the use case. The number of data points is described by six columns of which each has a different order of magnitude. We choose a similar representation for the number of customers in a dataset but only have five columns. We annotate the methods that deal with this amount of data with checkmarks. Note, that not all publications describe the data in a way it is possible to extract the information of the data dimensionality. In some cases, only the number of data points are given, in others, we only know about the number of customers, and sometimes we don't have information at all. How often a method is used, is indicated by the number of checkmarks. In some publications, different datasets with different sizes are utilized. If two datasets have different orders of magnitude, we indicated it by using checkmarks in the appropriated cells. However, if the datasets in the same publication have the same order of magnitude, we indicated it only once per publication.

In terms of feature selection, we see that RFM-analysis is applied up to 10^8 data points, but above this number of data points it is not used anymore. For example, Akhondzadeh-Noughabi and Albadvi (2015) apply RFM-analysis on 35,537,276 customer activities from 14,772 customers.

Based on the survey literature, PCA and DWT can be applied to data with up to 1 million data points. The graph approach utilized by Ding et al. (2019) is used on around 50,000 user activities. Chen et al. (2018, 2019) propose a purchase Tree approach which is tested on several datasets with different sizes in a range of a few thousand and 350 million transactions with customer numbers between 800 and 300,000.

The clustering method rows only refer to the clustering algorithms where no feature selection methods are applied. Regarding the number of users in datasets, we

Table 3 Data dimensionality that the feature selection methods and clustering approaches handle in the experiments of the surveyed literature

Methods		Data points						Number of customers				
		$< 10^3$	10^3–10^4	10^4–10^5	10^5–10^6	10^6–10^8	$> 10^8$	$< 10^3$	10^3–10^4	10^4–10^5	10^5–10^6	$> 10^6$
Feature selection	RFM-analysis	✓	✓	✓✓✓	✓✓✓✓✓✓	✓✓✓✓		✓✓	✓✓✓✓✓	✓✓✓	✓✓✓	✓
	PCA		✓	✓	✓				✓		✓	
	MCA									✓		
	Graph			✓								
	Purchase Tree		✓	✓	✓	✓	✓	✓	✓	✓	✓	
	Discrete wavelet trasformation				✓				✓			
Clustering method	Rule-based	✓		✓	✓							
	K-means clustering	✓	✓✓	✓✓	✓✓	✓✓✓			✓	✓✓	✓	
	Hierarchical clustering		✓						✓			
	Rough Set Theory	✓						✓				
	Latent class					✓			✓✓	✓		
	Evolutionary Algorithm	✓	✓	✓				✓	✓			
	Self-organizing map			✓				✓				
	Deep learning-based	✓										
	Hybrid clustering			✓	✓✓✓	✓		✓	✓✓	✓	✓	
	Other clustering			✓						✓		

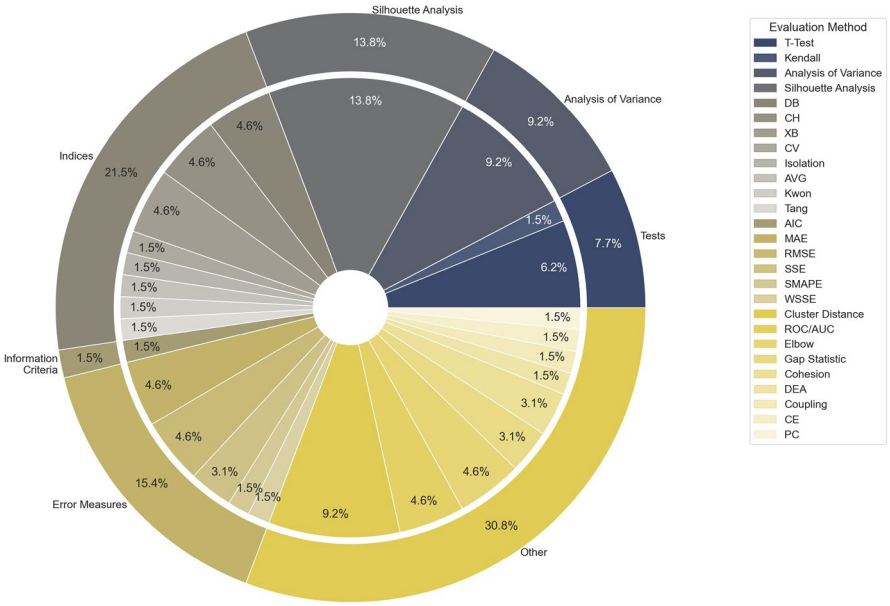


Fig. 6 Distribution of the segmentation methods used evaluation methods

see that usually, their number doesn't exceed 10,000. An expectation is provided by Kang et al. (2012). They test their hybrid approach on two datasets in which one dataset contains information about 101,532 customers. Another one comes from Goto et al. (2015) where they apply a latent class model on 37,278,907 browsing actions from 99,924 users. Abdolvand et al. (2015) apply k-means on 25,000 bank customers. Investigating which clustering methods are used for data with at least one million entries, we identify that it is k-means clustering, latent class models, and hybrid clustering approaches. K-means is used twice on over a million data points by Liu et al. (2015); Zhang et al. (2014). Liu et al. (2015) have access to 3 million transaction data from taoboa.com. Zhang et al. (2014) use the MovieLens datasets in which one has 100,000 movie ratings of 1682 different movies rated by 943 different users and the other has 1 million ratings for 3952 movies made by 6040 users.

3.6 Evaluation metrics

Usually, a clustering model learns in an unsupervised manner and the ground truth is unknown. Therefore different criteria need to be used to evaluate their performance. In the following the frequently used evaluation measures are described and briefly analyzed.

Clustering evaluation or cluster validation is an essential step in verifying the discovered groups in a data set. The fundamental challenge of evaluation lies in the missing ground truth, which can be a reason that we have not found a consensus between the evaluation methods in our literature research. Figure 6 presents the

distribution of used evaluation methods for segmentation methods in the literature that also shows the missing consensus. We classified the evaluation criteria into seven different groups which are indicated by a color. In the following, we briefly introduce the evaluation criteria and give some examples. It should be noted, that in some publications the authors don't apply evaluation metrics. Instead, they analyze the segments based on their plausibility. Chan et al. (2011) for example, measures the performance of the proposed method by comparing the company's sails before and after the using the approach. Guney et al. (2020), Nie et al. (2021), Wu et al. (2020) evaluate the segments with the help of the RFM-values.

Statistical significance test The underlying concept of a Statistical significance test is to determine whether the data points are randomly distributed or not. Krishna and Ravi (2021) have used a statistical t-test to evaluate their genetic algorithm approach on five different datasets. Another approach is the Kendall coefficient (Kendall 1938) that is used by An et al. (2018).

Analysis of variance The basic idea behind the analysis of variance (ANOVA) is to analyze whether the expected values of variables differ in distinct groups. By testing, if the variance of a variable is larger or smaller between the groups than within the groups, a statement about the meaningfulness of the group can be determined. ANOVA tests are used by Li et al. (2009), Hong and Kim (2012), Hjort et al. (2013), Hiziroglu et al. (2018).

Silhouette analysis The silhouette analysis is a (visual) validation method that is independent of the number of clusters and determines the consistency within a cluster. In addition to validation, this method can also be used to find the optimal number of clusters (Rousseeuw 1987). For example, the silhouette analysis is used by Akhondzadeh-Noughabi and Albadvi (2015), Peker et al. (2017), Christy et al. (2018).

Indices As shown in Fig. 6 many different index metrics were used to validate the clustering performance. The most used indices in our literature review are Davies-Bouldin (DB) index, Calinski-Harabasz (CH) index, and Xie-Beni (XB) index. The DB index describes the average similarity of each cluster with its most similar cluster. The DB index is to be interpreted in such a way that the lower the value is, the better the clustering (Davies and Bouldin 1979). The CH index, is the ratio of intra-cluster dispersion and inter-cluster dispersion (Caliński and Harabasz 1974). The XB index is used for fuzzy segmentation approaches and describes the separation and compactness of the clusters. The optimal number of clusters has the lowest XB value (Xie and Beni 1991). Chan et al. (2016) evaluate their proposed EA clustering with the DB index. Munusamy and Murugesan (2020) evaluate their fuzzy c-means clustering approach with XB index but also with the Kwon index, and the Tang index. They also use error measures for the cluster evaluation.

Information criteria These measures are used to select the models that fit the given data best but also take the number of parameters into account to prevent overfitting. One popular information criterion is the Akaike information criterion (AIC) which describes the model's information based on the number of parameters and the model's log-likelihood (Akaike 1974). Apichottanakul et al. (2021) utilize the AIC for evaluation to determine the optimal number of clusters in their latent class model.

Error measures Another evaluation method that is used in the surveyed literature is based on error measures like the mean absolute error (MAE), sum of squared error (SSE), root mean squared error (RMSE), or symmetric mean absolute percentage error (SMAPE). Abbasimehr and Shabani (2021) measure the cluster performance with SMAPE. Aghabozorgi et al. (2012) evaluate their proposed hierarchical clustering with SSE. Also, Lam et al. (2021) evaluate their clustering approach with SSE.

Others Some authors combine several evaluation metrics to express the usefulness and quality of their clustering models or use methods which donot fit in the six categories above. The mostly used "other" metric is cluster distance. We classify all inter and intra-cluster distance metrics as cluster distance if they are not further explained by the authors. For example, Wan et al. (2010) utilize an inter and intracluster distance to show that their CAS clustering approach has better distances and is more stable than k-means. Sivaguru and Punniyamoorthy (2021) apply a within/total clustering error index (which we consider as a cluster distance metric) to evaluate their k-means approach. In addition, they utilize DB index and t-test too. Umuhoza et al. (2020) utilize the elbow method, silhouette score, and CH index to determine the optimal number of segments. Another metric is the concordance (C) statistic (C-index) also known as receiver operating characteristic (ROC) and associated area under curve (AUC) score is for example used by Hsu et al. (2012) (also use SVM, isolation, and AVG index) or Barman and Chowdhury (2019). Dhandayudam and Krishnamurthi (2014) uses cohesion and coupling to evaluate the cluster quality for their rough set theory approach. Griva et al. (2021) use cohesion, inter and intracluster distance, similarity, and separation for cluster validation and gap statistic plus silhouette analysis to determine the optimal number for their latent class model clustering. Ramadas and Abraham (2018) validate the hybrid clustering which combines GA and fuzzy c-means with a partition coefficient (degree of intersection of clusters), classification entropy (the fuzziness of clusters), XB index, separation index, and partition index. Abdolvand et al. (2015) utilize the DB index to determine the optimal number of segments for their k-means approach and data envelopment analysis (DEA) for the evaluation.

4 Analysis and discussion

As previously shown in Fig. 3, the reviewed publications were not equally distributed over the years. An upward trend in the number of publications can be recognized which indicates the importance of customer behavior analysis and therefore, their segmentation even after twenty decades of research. Especially, in the years 2020, 2021, and 2022, we have found more publications than the years before. There may be several reasons for this. The first reason that comes to mind is the current covid pandemic. This has increased the growth in e-commerce services. This could have prompted less digitalized companies to digitalize more and offer their services online. In many publications the company remains unknown. However, in some other publications the companies are named. Two examples are taobao.com or nelly.com that are established online companies which is an indication against our

statement. From the literature conducted experiments did not show the state of digitization of the companies. Therefore, whether this connection exists remains open, and is not further investigated by us. Another reason, and in our opinion a more decisive one, is the increasing availability of the internet regardless of location. This means that a user can access the available online services at any time and from any place. For example, watching a series during a train ride or buying a new product at the online retailer of choice. With new requirements and necessities, the topic is also becoming more relevant in science and thus more is being published.

4.1 Analysis of feature selection methods

Based on our research, feature selection to represent customers is a fundamental step in the customer targeting process. For feature selection, customer information is indispensable. It is a challenge to get customers' demographic information, physiographic information, or information about their preferences. As already stated, there are two possible ways to collect such data. Explicit information collection is done by questionnaires or user surveys that require customers' accommodation to participate. Another, more implicit way is to collect demographic information via registration. Information can be collected by setting them as mandatory. Nevertheless, collecting data via registration is often limited to the usual information like age, gender, or address. In some use cases, like fashion, additional information about height and weight can be collected. It needs to be considered, that some users don't want to provide any information and wish to remain anonymous. They either give false information or leave the website (service). In both cases, it is not possible to gather useful information and in the worst case, the former leads to false conclusions regarding the customers. Furthermore, user groups that don't participate in a survey or are signed up are not represented in the data which makes the acquisition of unknown and new customers harder.

It is possible to gather customers' preferences with the aforementioned method. Nevertheless, this comes with a huge disadvantage. The information is outdated soon and needs to be constantly updated which increases the maintaining effort. Constantly asking the customer for an information update can also cause him to quit as a consequence. Therefore, customer preference should be estimated based on their recent behavior. Customer behavior information can be recorded implicitly. Usually, purchase information with product information, timestamp, etc., is stored for a company's financial overview. In addition, online touchpoints with the customer can be logged by the system. These logs can include various touchpoints like product views, click events, reviews, (dis)like, and many more. The advantages are that the customers do not disclose any personal information. Also, they are likely not interrupted on their shopping journey by unwanted questions. Nevertheless, disadvantages exist too. Predicting customer information from their behavior is not always correct that is for example caused by customers' heterogeneity. Additionally, a large amount of data is required to make such predictions. Another challenge of implicit data collection is that the information needs to be

linked to the customer. However, there are plenty of tracking-techniques to link the data with customers by using cookies or the browser identifier to name two examples.

As shown in Fig. 4, for the customer process as a whole, it makes no difference whether a feature selection method is used or the features are selected or handcrafted by an expert. However, manual feature selection and feature selection methods have their pros and cons.

One advantage of manual feature selection is that no additional computation is required. However, it requires expertise and domain knowledge to select customer information that is meaningful and representative. Feature selection methods are designed to automate the selection of features. One advantage is that domain knowledge is no longer required. However, this doesn't mean that domain knowledge should generally be dispensed with. Another argument in favor of feature selection methods is that information redundancy can be removed. Redundancies come in hand with the amount of data collected. Removing unnecessary and redundant information can speed up the customer analysis algorithms. This information is hard to determine and select manually even with domain knowledge. Regarding Table 3, we notice that feature selection methods have processed larger amounts of data in our literature. Considering our second question from the introduction, we can state that feature selection methods allow larger amounts of data for customer behavior analysis. Particularly, the RFM-analysis and Purchase Tree have no limitation concerning the data dimensionality based on our research.

Our literature research shows that the RFM-analysis is by far the most popular feature selection method. Therefore, we analyze the RFM-analysis method in more detail hereafter and discuss the advantages and disadvantages. During the literature research, several points caught our attention. The RFM-analysis could be applied to almost any type of purchase or activity data since only three features need to be calculated. Furthermore, the calculation is very simple and requires only the basic arithmetic operations. So there is valuable customer representation in only three values. These values can be represented either numerically or categorically. For the categorical representation, the values were typically divided into five categories, each with 20%-quantiles. Thus, the obtained features are used for any clustering method. In addition, we notice that the RFM-analysis is often extended with additional features. The feature extension is usually use case-specific. Besides adding new features, the RFM-features are extended on different activity levels. For example, the RFM-values are calculated for all product categories or different customer activities. This provides additional information about the customer's product preference at the category or activity level. Another advantage of RFM-analysis is that it can handle all sizes of data sets without having a scalability problem. This has been sufficiently demonstrated in the publications and is illustrated by Table 3. We also like to note that in some publications, the RFM-analysis is used to explain the resulting clusters and helps with the customer behavior analysis which shows that decision makers can easily understand and interpret the RFM-values. Based on our findings to feature selection methods, we can answer the third question as follows. For feature selection methods no time-depended methodological trend could be determined. However, the most popular feature selection method is the RFM-analysis.

These versatile properties of the RFM-analysis are the reason for its popularity which is also stated by Chan et al. (2011), Alberto Carrasco et al. (2019). Despite it being the most used feature selection method, we also identified weaknesses in the RFM-analysis that all found customer representation has in common. The RFM-analysis, other feature selection methods like PCA, or manual feature selection don't consider the whole information content of the accessible data. However, to represent more information, more features and therefore, more memory is required, which also increases the computation time for the segmentation methods. Another issue is that there is information in the data that cannot be extracted using feature selection methods or expertise. Recently, embeddings become a popular approach for representations. Embeddings are capable to represent words as shown by Mikolov et al. (2013), time series (Nalmpantis and Vrakas 2019), or products (Vasile et al. 2016) but are not limited to them. With embeddings, it could be possible to encode additional behavioral information that could improve the customer targeting process. This was already demonstrated for product recommendation (Vasile et al. 2016; Tercan et al. 2021; Alves Gomes et al. 2021; Srilakshmi et al. 2022) or customers' purchase behavior prediction (Alves Gomes et al. 2022). Despite the popularity in several e-commerce tasks, no author used an customer embedding representation in the reviewed literature. From our perspective, the reason is that embeddings are less interpretable, and therefore, non-automated customer targeting is more difficult.

4.2 Analysis of segmentation methods

We found 13 different types of segmentation methods. K-means is by far the most used approach. Especially, in the last years from 2020 to 2022 k-means is used 24 times. In regard to the third guiding question, we can conclude that besides a k-means upwards trend no other trend can be spotted. The question that now arises is "why is k-means becoming so popular recently"? One answer is that k-means is simple to implement and an established approach. In contrast, other approaches like EAs, hierarchical clustering, or SOMs are more complex according to how the run time or space requirements grow as the input size grows (Bachmann-Landau notation) and it needs more effort to implement them (Firdaus and Uddin 2015). The ever-increasing amounts of data in e-commerce amplifies this trend because simple methods can be used more quickly, and thus, results can be obtained faster. However, if this is the reason, then the question that follows is why are rule-based approaches not popular as well? As shown by Fig. 5 the density of rule-base approaches increased in the years between 2018 and 2021 but some other influencing factors play a major role on the methods popularity. While we can only make assumptions at this point, rule-based segmentation approaches have significant drawbacks. For example, they require domain knowledge to set appropriate thresholds for separating customer segments. The increasing and heterogeneous amount of data complicates this setting of appropriate thresholds or requires a higher dynamic, which in turn results in more rules and complex relationships. Our assumption is supported by the aggregated information in Table 3 that shows that k-means is applicable on 100 million data points.

Considering the data dimensionality which is used in the publications we see that k-means approaches can handle a larger amount of data and is in pair with latent class approaches. As we mentioned, the hybrid approach that uses the largest amount of data is a combination of the latent class model. However, concerning the number of customers in the data which are the objective of the clustering, the numbers rarely exceed the 10,000. This indicates that clustering approaches need an appropriate feature selection method to deal with a larger amount of data. All this doesn't mean that the methods cannot be applied to larger data sets. Our argumentation is based solely on the paper corpus we saw. Based on the findings concerning the data dimensionality, we can state for guiding question number three, that k-means and latent class models can process the largest amount of data among all segmentation methods. However, as already stated this applies only in case of manual feature selection. We recommend using a feature selection methods namely the RFM-analysis that allows to process any kind of data dimensionality. Note that we don't address the time or memory complexity of the segmentation methods, which is also a performance indicator, but evaluate them based solely on the amount of data used in the literature.

In terms of use cases, we can state that each clustering method is usable in retailing use cases. We cannot make such a generalized statement for other domains. However, it is not unlikely that all segmentation methods can be used independently of the domain. Especially with k-means, we can see that it has the largest variant of different use cases. Nonetheless, the reason for being used in different domains can be because k-means is applied in most publications.

Apart from a quantitative analysis of the segmentation method, we would like to make a qualitative analysis. Unfortunately, there is no way to determine which segmentation method performs best. The major issue in our opinion is that there is no ground truth for the customer segments to determine a score. Therefore, there is no unified method for qualitative evaluation which is necessary to state which segmentation method is superior to the other. We noticed that there are a vast amount of different evaluation methods as presented in Sect. 3.6. Different evaluation approaches are required for different clustering approaches, i.e. fuzzy (soft) clustering has different properties than hard clustering. It would simplify qualitative segmentation analysis if the scientific community agree on a small set of evaluation methods. The urge is there which we can see in the number of different evaluation metrics and the considered publication where the authors try to show that their approach is superior to others. If everyone would use the same metrics, the authors' efforts would have more significance and the performance of the method could be compared over different publications which are usually done in other scientific disciplines. Nevertheless, due to the absence of ground truth, correctness can never be shown, and therefore, the purpose of unified evaluation methods may be questioned. Another aspect we want to consider is evaluation metrics with semantic interpretability. Such metrics would have the advantage to show which segmentation algorithm partitions the customers in a desirable way. Furthermore, it would create comparability between multiple segmentation methods for identical use cases. However, the challenge is to define evaluation metrics that have the capacity to be semantic interpretable and, at the same time, can be applied to different segmentation methods and use cases. In

numerous publications, evaluation methods are used to find the optimal number of segments. Therefore, even if there is no defined uniform way to compare clustering approaches, they still have their reason of existence and are necessary methods for determining an optimal number of segments.

Before the study, we would not have expected such a distribution, as we thought that a relatively old method like k-means (first proposed around 1960 and published in 1982) is not so often used especially not so often in the last years of the considered literature. In addition, we assumed that there would be newer and more innovative approaches like deep learning-based approaches. The reason for our assumption is that deep learning techniques archived great results in a broad range of applications such as computer vision and natural language processing and we expected to see these methods transferred to customer segmentation and analysis. However, deep learning-based segmentation only appeared once in the literature. Regarding our initial assumption, an open question still remains. Will deep learning methods be used for customer segmentation in the future? As with embeddings used as feature representations, one advantage might be that the feature representation phase can be omitted, and thus less information is lost. However, a disadvantage and probably the reason why we did not find more than one deep learning-based segmentation method is that the customer segmentation needs to be formalized as a learning problem. Furthermore, segmentation is by design an unsupervised process and no ground truth exists. Another point that speaks against deep learning segmentation is that deep learning models are black boxes and therefore, interpretation, explainability, and reasoning for decision making are no longer achievable.

Based on our findings and analysis, we recommend using k-means or rule-based segmentation approaches which are easy to use and implement, to partition different customers for e-commerce use cases. In addition, if massive transaction data is available, we recommend RFM-analysis for the customer representation that can be extended with additional features.

5 Conclusion and future research

In this survey, we provided an extensive literature review on customer targeting process for e-commerce use cases whose main focus lies in the segmentation methods for customer behavior analysis. Our goal was to provide an overview of segmentation methods used in the literature and to determine best-practice approaches and their limitations. We introduced the steps of the research and key criteria for the paper selection and analyzed as well as discussed our findings afterward. In our work, we considered 105 publications with different case studies that focused on customer analysis with segmentation methods.

Summarizing the approaches examined, the identified four-step process emerges as the current gold standard for personalized customer targeting in e-commerce. For the customer representation, either hand-crafted features or an RFM analysis adapted to the use case are generally used. Subsequently, for customer analysis, the generated customer representation is segmented using a k-means approach.

Based on our research and literature analysis we made several findings regarding our investigated topic.

- We identified a common process for personalized customer targeting which includes feature selection methods, customer segmentation, and customer targeting. This process is illustrated by Fig. 2 and can be utilized to plan customer targeting campaigns. Each of the four steps has its own requirements and its a discipline of its own worth to be investigated. We focused on the customer analysis and customer representation part.
- Over the years, the number of publication that deals with customer targeting in e-commerce are continuously increasing. This supports the preceding assumption that it is a time-relevant subject.
- Feature selection methods enable the usage of larger datasets and among the utilized methods the RFM-analysis is by far the most popular one. There are many reasons for this: first, the method is easy to use, and second, it is based on features that can be extracted and understood. Another advantage of RFM analysis is the possibility of its easy adaptation to specific use cases by adding further or changing existing features.
- In approximately half of the publications (47.6%), manual feature selection was used.
- Among all the used clustering methods, k-means has emerged as the most popular approach (39% in total). Since 2011, it was repeatably used. Besides that, no other over-time trend was identified. The popularity of k-means can be explained by its simplicity and applicability to large scale datasets.
- We were not able to define the best clustering approach based on its performance because many different evaluation methods exist and were used to evaluate the cluster quality.
- Some evaluation methods can be used to determine the optimal number of segments which is unknown from the beginning and is often a tunable hyperparameter.
- The literature review doesn't show that a segmentation method exists that is applicable to every e-commerce use case that involves customer analysis. This could only be suggested, if at all, for the retail use case. In terms of method, k-means has been used in every use case identified, with the exception of the manufacturing use case.

New insights always come with new challenges and opportunities. Based on our research and findings we propose future research ideas which should be investigated. Especially with regard to recent developments in the field of Deep Learning, there are many approaches that can be adapted and, according to the our assessment, display a lot of potential.

- Deep learning introduced innovations in many domains such as natural language processing and computer vision. Nevertheless, we only found one DL-based segmentation approach in our research. Therefore, we see potential and a research gap in DL techniques for segmentation.
- The process steps in the identified four-phase process for customer targeting are essentially based on a high level of understanding of the customers, i.e. their

needs and behavior. This is necessary for marketing and domain expert to tailor personalized marketing strategies for the customers. However, with the advent of deep learning-based approaches personalized customer targeting can be done fully automated e.g. end-to-end model and therefore, the customer analysis step which includes customer segmentation can be omitted. This development can be seen for example in deep learning-based recommendation systems which make personalized recommendation without the need of the customer analysis. This leads to the question; *How customizable are the individual phases of this process and can individual steps be omitted to increase efficiency or are all steps so fundamental that a deviation from these procedures would have a negative impact on the goal, customer targeting?*

- Manual feature selection is still frequently used. The feature quality is thereby highly depended on the underlying expertise to select or define important features for clustering. Progressive digitization is leading to growing challenges, especially in dealing with data volumes and data diversity. To meet these challenges, manual feature selection is reaching its limits as it is not able to tap the insight potential within this data. Hence, the question arises *if approaches exist that can help experts to create meaningful and representative features for customer representation?*

  In this regard a look outside the box to other e-commerce research, e.g. click-through rates prediction can yield new approaches. There researchers and professionals have started using feature embeddings on manual selected features with the underlying assumption that the learning models will learn meaningful representations from the data. This would simplify the manual feature selection process. However, these learning models are usually based on deep neural networks which are unfortunately black boxes and not interpretable. The question rises, *if segmentation methods can be used as a post-processing to provide interpretability for the embedded features and therefore, an insight over the customers?* (Which got lost by not using the customer analysis step).
- In our research, we identified many different evaluation metrics to evaluate the performance of segmentation methods. Nevertheless, we could not find a consensus on evaluation metrics as in other domains. The reason is the missing ground-truth. This circumstance makes it difficult to determine the effectiveness and transferability of a segmentation approach from one use case to another. The open question that remains is, *is it necessary, to develop evaluation metrics with semantic meaning and is it possible to transfer such metrics to different experiments to enable comparision of the segmentation methods?*

In our literature review, we covered the usage of feature selection and segmentation method for personalized customer targeting. E-commerce is a dynamic environment with ever new challenges and therefore, new research opportunities.

A Table of reviewed literature

Table 4 Literature with title, author and date which are the object of investigation sorted by year of publication/acceptance

Title	References	Years
User segmentation of online music services using fuzzy clustering	Ozer (2001)	2000
An integrated data mining and behavioral scoring model for analyzing bank customers	Hsieh (2004)	2004
Using a clustering genetic algorithm to support customer segmentation for personalized recommender systems	Kim and Ahn (2004)	2004
Joint optimimization of customer segmentation and marketing policy to maximize long-term profitability	Jonker et al. (2004)	2004
Effective personalized recommendation based on time-framed navigation clustering and association mining	Wang and Shao (2004)	2004
Mining web browsing patterns for E-commerce	Song and Shepperd (2006)	2005
Classification, filtering, and identification of electrical customer load patterns through the use of self-organizing maps	Verdu et al. (2006)	2006
Mining of mixed data with application to catalog marketing	Hsu and Chen, Y.-g.C. (2007)	2007
Intelligent value-based customer segmentation method for campaign management: A case study of automobile retailer	Chan (2008)	2007
Customer segmentation revisited: the case of the airline industry	Teichert et al. (2008)	2007
Chameleon based on clustering feature tree and its application in customer segmentation	Li et al. (2009)	2008
Improving personalization solutions through optimal segmentation of customer bases	Jiang and Tuzhilin (2009)	2009
Mining changing customer segments in dynamic markets	Boettcher et al. (2009)	2009
A hybrid of sequential rules and collaborative filtering for product recommendation	Liu et al. (2009)	2009
Customer segmentation of multiple category data in e-commerce using a soft-clustering approach	Wu and Chou (2011)	2010
CAS based clustering algorithm for Web users	Wan et al. (2010)	2010
Segmenting and mining the ERP users' perceived benefits using the rough set approach	Wu (2011)	2011
Group RFM analysis as a novel framework to discover better customer consumption behavior	Chang and Tsai (2011)	2011
Pricing and promotion strategies of an online shop based on customer segmentation and multiple objective decision making	Chan et al. (2011)	2011
Incremental clustering of time-series by fuzzy clustering	Aghabozorgi et al. (2012)	2011
User action interpretation for online content optimization	Bian et al. (2013)	2011
Improved response modeling based on clustering, under-sampling, and ensemble	Kang et al. (2012)	2012

Table 4 (continued)

Title	References	Years
Segmenting customers in online stores based on factors that affect the customer's intention to purchase	Hong and Kim (2012)	2012
Segmenting customers by transaction data with concept hierarchy	Hsu et al. (2012)	2012
Apply robust segmentation to the service industry using kernel induced fuzzy clustering techniques	Wang (2010)	2012
LRFMP model for customer segmentation in the grocery retail industry: a case study	Peker et al. (2017)	2012
Customer segmentation based on buying and returning behaviour	Hjort et al. (2013)	2013
Online purchaser segmentation and promotion strategy selection: evidence from Chinese E-commerce market	Liu et al. (2015)	2013
Information filtering via collaborative user clustering modeling	Zhang et al. (2014)	2013
Discovering valuable frequent patterns based on RFM analysis without customer identification information	Hu and Yeh (2014)	2014
Rough set approach for characterizing customer behavior	Dhandayudam and Krishnamurthi (2014)	2014
Customer segmentation in a large database of an online customized fashion business	Brito et al. (2015)	2014
A new recommendation method for the user clustering-based recommendation system	Rapecka and Dzemyda (2015)	2015
Customer segmentation issues and strategies for an automobile dealership with two clustering techniques	Tsai et al. (2015)	2015
Performance management using a value-based customer-centered model	Abdolvand et al. (2015)	2015
A fuzzy ANP based weighted RFM model for customer segmentation in auto insurance sector	Ravasan and Mansouri (2015)	2015
Customer lifetime value determination based on RFM model	Safari et al. (2016)	2015
Mining the dominant patterns of customer shifts between segments by using top-k and distinguishing sequential rules	Akhondzadeh-Noughabi and Albadvi (2015)	2015
A new latent class model for analysis of purchasing and browsing histories on EC sites	Goto et al. (2015)	2015
Performance evaluation of different customer segmentation approaches based on RFM and demographics analysis	Sarvari et al. (2016)	2016
Recommender system based on customer segmentation (RSCS)	Rezaeinia and Rahmani (2016)	2016
An exploration of improving prediction accuracy by constructing a multi-type clustering based recommendation framework	Ma et al. (2016)	2016
Marketing segmentation using the particle swarm optimization algorithm: a case study	Chan et al. (2016)	2016
PurTreeClust: a clustering algorithm for customer segmentation from massive customer transaction data	Chen et al. (2018)	2017
Data clustering using eDE, an enhanced differential evolution algorithm with fuzzy c-means technique	Ramadas and Abraham (2018)	2017

Table 4 (continued)

Title	References	Years
Customer segmentation with purchase channels and media touchpoints using single source panel data	Nakano and Kondo (2018)	2017
User-centered recommendation using US-ELM based on dynamic graph model in E-commerce	Ding et al. (2019)	2017
An empirical assessment of customer lifetime value models within data mining	Hiziroglu et al. (2018)	2018
Customer segmentation by using RFM model and clustering methods: a case study in retail industry	Dogan et al. (2018)	2018
RFM ranking—an effective approach to customer segmentation	Christy et al. (2018)	2018
Improving sparsity and new user problems in collaborative filtering by clustering the personality factors	Hafshejani et al. (2018)	2018
Customer online shopping experience data analytics—integrated customer segmentation and customised services prediction model	Wong and Wei (2018)	2018
A fuzzy linguistic RFM model applied to campaign management	Alberto Carrasco et al. (2019)	2018
A CLV-based framework to prioritize promotion marketing strategies: a case study of telecom industry	Nemati et al. (2018)	2018
Customer segmentation using online platforms: isolating behavioral and demographic segments for persona creation via aggregated user data	An et al. (2018)	2018
A study on e-commerce customer segmentation management based on improved K-means algorithm	Deng and Gao (2020)	2018
RFM customer analysis for product-oriented services and service business development: an interventionist case study of two machinery manufacturers	Stormi et al. (2020)	2019
Hybrid bio-inspired user clustering for the generation of diversified recommendations	Logesh et al. (2020)	2019
A novel approach for the customer segmentation using clustering through self-organizing map	Barman and Chowdhury (2019)	2019
Effective user preference clustering in web service applications	Wang et al. (2020)	2019
A hybrid two-phase recommendation for group-buying E-commerce applications	Bai et al. (2019)	2019
Spectral clustering of customer transaction data with a two-level subspace weighting method	Chen et al. (2019)	2019
An effective user clustering-based collaborative filtering recommender system with grey wolf optimisation	Sivaramakrishnan et al. (2020)	2020
Product recommendation in offline retail industry by using collaborative filtering	Pratama et al. (2020)	2020
IECT: a methodology for identifying critical products using purchase transactions	Hsu and Huang (2020)	2020
Modified dynamic fuzzy c-means clustering algorithm—application in dynamic customer segmentation	Munusamy and Murugesan (2020)	2020
Alleviating the data sparsity problem of recommender systems by clustering nodes in bipartite networks	Zhang et al. (2020)	2020

Table 4 (continued)

Title	References	Years
Identifying omnichannel deal prone segments, their antecedents, and their consequences	Valentini et al. (2020)	2020
A combined approach for customer profiling in video on demand services using clustering and association rule mining	Guney et al. (2020)	2020
A new framework for predicting customer behavior in terms of RFM by considering the temporal aspect based on time series techniques	Abbasimehr and Shabani (2021)	2020
Data analytics and the P2P cloud: an integrated model for strategy formulation based on customer behaviour	Lam et al. (2021)	2020
A methodology for classification and validation of customer datasets	Nie et al. (2021)	2020
Performance-enhanced rough k-means clustering algorithm	Sivaguru and Punniyamoorthy (2021)	2020
Using unsupervised machine learning techniques for behavioral-based credit card users segmentation in Africa	Umuhoza et al. (2020)	2020
Customer categorization using a three-dimensional loyalty matrix analogous to FMEA	Madzik and Shahin (2021)	2020
An empirical study on customer segmentation by purchase behaviors using a RFM Model and K-means algorithm	Wu et al. (2020)	2020
Customer segmentation by web content mining	Zhou et al. (2021)	2021
The role of shopping mission in retail customer segmentation	Sokol and Holy (2021)	2021
Research and implementation of the customer-oriented modern hotel management system using fuzzy analytic hiererchical process (FAHP)	Wang and Zhang (2021)	2021
High utility itemset mining using binary differential evolution: An application to customer segmentation	Krishna and Ravi (2021)	2021
Factors affecting customer analytics: evidence from three retail cases	Griva et al. (2021)	2021
Customer behaviour analysis based on buyingdata sparsity for multi-category products in pork industry: A hybrid approach	Apichottanakul et al. (2021)	2021
An extended regularized K-means clustering approach for high-dimensional customer segmentation with correlated variables	Zhao et al. (2021)	2021
Online Reviews Analysis for Customer Segmentation through Dimensionality Reduction and Deep Learning Techniques	Nilashi et al. (2021)	2021
RFM-based repurchase behavior for customer classification and segmentation	Rahim et al. (2021)	2021

Table 4 (continued)

Title	References	Years
User value identification based on improved RFM model and K-means++ algorithm for complex data analysis	Wu et al. (2021)	2021
Deep customer segmentation with applications to a Vietnamese supermarkets' data	Nguyen (2021)	2021
Learning about the customer for improving customer retention proposal of an analytical framework	Simoes and Nogueira (2021)	2021
A hybrid method for big data analysis using fuzzy clustering, feature selection and adaptive neuro-fuzzy inferences system techniques: case of Mecca and Medina hotels in Saudi Arabia	Alghamdi (2022a)	2022
A hybrid method for customer segmentation in Saudi Arabia restaurants using clustering, neural networks and optimization learning techniques	Alghamdi (2022b)	2022
A novel approach for send time prediction on email marketing	Araujo et al. (2022)	2022
Multi clustering recommendation system for fashion retail	Bellini et al. (2022)	2022
Understanding customer's online booking intentions using hotel big data analysis	Chalupa and Petricek (2022)	2022
A precision marketing strategy ofe-commerce platform based on consumer behavior analysis in the era of big data	Zhang and Huang (2022)	2022
Customer behavior analysis by intuitionistic fuzzy segmentation: comparison for two major cities in Turkey	Dogan et al. (2022)	2022
Customer segmentation using k-means clustering for developing sustainable marketing strategies	Gautam and Kumar (2022)	2022
"I can get no e-satisfaction". What analytics say? Evidence using satisfaction data from e-commerce	Griva (2022)	2022
A two-stage business analytics approach to perform behavioural and geographic customer segmentation using e-commerce delivery data	Griva et al. (2022)	2022
Analysis of clustering algorithms for credit risk evaluation using multiple correspondence analysis	Jadwal et al. (2022)	2022
Integrated customer lifetime value (CLV) and customer migration model to improve customer segmentation	Kanchanapoom and Chongwatpol (2022)	2022
Multi-behavior RFM model based on improved SOM neural network algorithm for customer segmentation .	Liao et al. (2022)	2022
K-means customers clustering by their RFMT and score satisfaction analysis	Mensouri et al. (2022)	2022
A novel hybrid segmentation approach for decision support: a case study in banking	Mosa et al. (2022)	2022
K-means clustering approach for intelligent customer segmentation using customer purchase behavior data	Tabianan et al. (2022)	2022
Research on segmenting E-commerce customer through an improved K-medoids clustering algorithm	Wu et al. (2022)	2022

Author contributions The author MAG had the idea for the article, performed the literature search and data analysis, and drafted the article. The author TM mentored the process with his expertise and critically revised the article.

Funding Open Access funding enabled and organized by Projekt DEAL.

Declarations

Conflict of interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

Abbasimehr H, Shabani M (2021) A new framework for predicting customer behavior in terms of RFM by considering the temporal aspect based on time series techniques. J Ambient Intell Hum Comput 12(1):515–531. https://doi.org/10.1007/s12652-020-02015-w

Abdolvand N, Albadvi A, Aghdasi M (2015) Performance management using a value-based customer-centered model. Int J Prod Res 53(18):5472–5483. https://doi.org/10.1080/00207543.2015.1026613

Aghabozorgi S, Saybani MR, Teh YW (2012) Incremental clustering of time-series by fuzzy clustering. J Inf Sci Eng 28(4):671–688

Akaike H (1974) A new look at the statistical model identification. IEEE Trans Autom Control 19(6):716–723

Akhondzadeh-Noughabi E, Albadvi A (2015) Mining the dominant patterns of customer shifts between segments by using top-k and distinguishing sequential rules. Manag Decis 53(9):1976–2003. https://doi.org/10.1108/MD-09-2014-0551

Alberto Carrasco R, Francisca Blasco M, Garcia-Madariaga J, Herrera-Viedma E (2019) A fuzzy linguistic RMF model applied to campaign management. Int J Interact Multimed Artif Intell 5(4):21–27. https://doi.org/10.9781/ijimai.2018.03.003

Alghamdi A (2022) A hybrid method for big data analysis using fuzzy clustering, feature selection and adaptive neuro-fuzzy inferences system techniques: case of Mecca and Medina hotels in Saudi Arabia. Arab J Sci Eng. https://doi.org/10.1007/s13369-022-06978-0

Alghamdi A (2022) A hybrid method for customer segmentation in Saudi Arabia restaurants using clustering, neural networks and optimization learning techniques. Arab J Sci Eng. https://doi.org/10.1007/s13369-022-07091-y

Alves Gomes M, Tercan H, Bodnar T, Meisen T, Meisen P (2021) A filter is better than none: improving deep learning-based product recommendation models by using a user preference filter. In: 2021 IEEE 23rd int conf on high performance computing and communications; 7th int conf on data science and systems; 19th int conf on smart city; 7th int conf on dependability in sensor, cloud and big data systems and application (hpcc/dss/smartcity/dependsys) (pp 1278–1285). https://doi.org/10.1109/HPCC-DSS-SmartCity-DependSys53884.2021.00195

An J, Kwak H, Jung S-g, Salminen J, Jansen BJ (2018) Customer segmentation using online platforms: isolating behavioral and demographic segments for persona creation via aggregated user data. Soc Netw Anal Mining. https://doi.org/10.1007/s13278-018-0531-0

Apichottanakul A, Goto M, Piewthongngam K, Pathumnakul S (2021) Customer behaviour analysis based on buying-data sparsity for multicategory products in pork industry: a hybrid approach. Cogent Eng. https://doi.org/10.1080/23311916.2020.1865598

Araujo C, Soares C, Pereira I, Coelho D, Rebelo MA, Madureira A (2022) A novel approach for send time prediction on email marketing. Appl Sci. https://doi.org/10.3390/app12168310

Bai L, Hu M, Ma Y, Liu M (2019) A hybrid two-phase recommendation for group-buying e-commerce applications. Appl Sci. https://doi.org/10.3390/app9153141

Barman D, Chowdhury N (2019) A novel approach for the customer segmentation using clustering through self-organizing map. Int J Bus Anal 6(2):23–45. https://doi.org/10.4018/IJBAN.2019040102

Bellini P, Palesi LAI, Nesi P, Pantaleo G (2022) Multi clustering recommendation system for fashion retail. Multimed Tools Appl. https://doi.org/10.1007/s11042-021-11837-5

Ben Ayed A, Ben Halima M, Alimi AM (2014) Survey on clustering methods: Towards fuzzy clustering for big data. In: 2014 6th international conference of soft computing and pattern recognition (SoCPaR) (pp 331-336). https://doi.org/10.1109/SOCPAR.2014.7008028

Bezdek JC, Ehrlich R, Full W (1984) FCM: The fuzzy c-means clustering algorithm. Comput Geosci 10(2–3):191–203. https://doi.org/10.1016/0098-3004(84)90020-7

Bian J, Dong A, He X, Reddy S, Chang Y (2013) User action interpretation for online content optimization. IEEE Trans Knowl Data Eng 25(9):2161–2174. https://doi.org/10.1109/TKDE.2012.130

Birtolo C, Diessa V, De Chiara D, Ritrovato P (2013) Customer churn detection system: identifying customers who wish to leave a merchant. In: International conference on industrial, engineering and other applications of applied intelligent systems (pp 411–420)

Boettcher M, Spott M, Nauck D, Kruse R (2009) Mining changing customer segments in dynamic markets. Expert Syst Appl 36(1):155–164. https://doi.org/10.1016/j.eswa.2007.09.006

Brito PQ, Soares C, Almeida S, Monte A, Byvoet M (2015) Customer segmentation in a large database of an online customized fashion business. Robot Comput-Integr Manuf 36:93–100. https://doi.org/10.1016/j.rcim.2014.12.014

Burri M, Schär R (2016) The reform of the EU data protection framework: outlining key changes and assessing their fitness for a data-driven economy. J Inf Policy 6(1):479–511

Caliński T, Harabasz J (1974) A dendrite method for cluster analysis. Commun Stat-Theory Methods 3(1):1–27

Chalupa S, Petricek M (2022) Understanding customer's online booking intentions using hotel big data analysis. J Vacat Mark. https://doi.org/10.1177/13567667221122107

Chan CCH (2008) Intelligent value-based customer segmentation method for campaign management: a case study of automobile retailer. Expert Syst Appl 34(4):2754–2762

Chan C-CH, Cheng C-B, Hsien W-C (2011) Pricing and promotion strategies of an online shop based on customer segmentation and multiple objective decision making. Expert Syst Appl 38(12):14585–14591. https://doi.org/10.1016/j.eswa.2011.05.024

Chan CCH, Hwang Y-R, Wu H-C (2016) Marketing segmentation using the particle swarm optimization algorithm: a case study. J Ambient Intell Humaniz Comput 7(6):855–863. https://doi.org/10.1007/s12652-016-0389-9

Chang H-C, Tsai H-P (2011) Group RFM analysis as a novel framework to discover better customer consumption behavior. Expert Syst Appl 38(12):14499–14513. https://doi.org/10.1016/j.eswa.2011.05.034

Chen X, Fang Y, Yang M, Nie F, Zhao Z, Huang JZ (2018) Purtreeclust: a clustering algorithm for customer segmentation from massive customer transaction data. IEEE Trans Knowl Data Eng 30(3):559–572. https://doi.org/10.1109/TKDE.2017.2763620

Chen X, Sun W, Wang B, Li Z, Wang X, Ye Y (2019) Spectral clustering of customer transaction data with a two-level subspace weighting method. IEEE Trans Cybern 49(9):3230–3241. https://doi.org/10.1109/TCYB.2018.2836804

Christy AJ, Umamakeswari A, Priyatharsini L, Neyaa A (2018) RFM ranking—an effective approach to customer segmentation. J King Saud Univ-Comput Inf Sci 32(10):1215. https://doi.org/10.1016/j.jksuci.2018.09.004

Cooper HM (1988) Organizing knowledge syntheses: a taxonomy of literature reviews. Knowl Soc 1(1):104–126. https://doi.org/10.1007/BF03177550

Coussement K, van den Bossche FAM, de Bock KW (2014) Data accuracy's impact on segmentation performance: benchmarking RFM analysis, logistic regression, and decision trees. J Bus Res 67(1):2751–2758. https://doi.org/10.1016/j.jbusres.2012.09.024
Davies DL, Bouldin DW (1979) A cluster separation measure. IEEE Trans Pattern Anal Mach Intell PAMI 1(2):224–227. https://doi.org/10.1109/TPAMI.1979.4766909
De Jong K (2016) Evolutionary computation: a unified approach. In: Proceedings of the 2016 on genetic and evolutionary computation conference companion (pp 185–199)
de Marco M, Fantozzi P, Fornaro C, Laura L, Miloso A (2021) Cognitive analytics management of the customer lifetime value: an artificial neural network approach. J Enterp Inf Manag 34(2):679–696. https://doi.org/10.1108/JEIM-01-2020-0029
Dempster AP, Laird NM, Rubin DB (1977) Maximum likelihood from incomplete data via the EM algorithm. J R Stat Soc Ser B (Methodol) 39(1):1–22
Deng Y, Gao Q (2020) A study on e-commerce customer segmentation management based on improved k-means algorithm. Inf Syst E-Bus Manag 18(4):497–510. https://doi.org/10.1007/s10257-018-0381-3
Dhandayudam P, Krishnamurthi I (2014) Rough set approach for characterizing customer behavior. Arab J Sci Eng 39(6):4565–4576. https://doi.org/10.1007/s13369-014-1013-y
Di Zhang, Huang M (2022) A precision marketing strategy of e-commerce platform based on consumer behavior analysis in the era of big data. Math Prob Eng. https://doi.org/10.1155/2022/8580561
Ding L, Han B, Wang S, Li X, Song B (2019) User-centered recommendation using US-ELM based on dynamic graph model in ecommerce. Int J Mach Learn Cybern 10(4):693–703. https://doi.org/10.1007/s13042-017-0751-z
Dogan O, Aycin E, Bulut ZA (2018) Customer segmentation by using RFM model and clustering methods: a case study in retail industry. Int J Contemp Econ Admin Sci 8(1):1–19
Dogan O, Seymen OF, Hiziroglu A (2022) Customer behavior analysis by intuitionistic fuzzy segmentation: comparison of two major cities in turkey. Int J Inf Technol Decis Mak 21(02):707–727. https://doi.org/10.1142/S0219622021500607
Donath W, Hoffman A (1973) Lower bounds for the partitioning of graphs. IBM J Res Dev 17(5):420–425
Dunn JC (1973) A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters. J Cybern. https://doi.org/10.1080/01969727308546046
Eiben AE, Smith JE (2003) Introduction to evolutionary computing, vol 53. Springer, Berlin
European-Parliament (2016) Regulation (eu) 2016/679 of the european parliament and of the council. https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32016R0679. Accessed 7 June 2023
Fan Y, Huang GQ (2007) Networked manufacturing and mass customization in the ecommerce era: the Chinese perspective. Taylor & Francis, Milton Park
Fiedler M (1973) Algebraic connectivity of graphs. Czechoslov Math J 23(2):298–305
Firdaus S, Uddin MA (2015) A survey on clustering algorithms and complexity analysis. Int J Comput Sci Issues (IJCSI) 12(2):62
Gautam N, Kumar N (2022) Customer segmentation using k-means clustering for developing sustainable marketing strategies. Biznes Inf-Bus Inf 16(1): 72–82. https://doi.org/10.17323/2587-814X.2022.1.72.82
Gennari JH (1989) A survey of clustering methods
Gomes MA, Meyes R, Meisen P, Meisen T (2022) Will this online shopping session succeed? predicting customer's purchase intention using embeddings. In: Proceedings of the 31st ACM international conference on information & knowledge management (p. 2873–2882). Association for Computing Machinery, New York, NY, USA. Retrieved from https://doi.org/10.1145/3511808.3557127
Goto M, Mikawa K, Hirasawa S, Kobayashi M, Suko T, Horii S (2015) A new latent class model for analysis of purchasing and browsing histories on EC sites. Ind Eng Manag Syst 14(4):335–346. https://doi.org/10.7232/iems.2015.14.4.335
Griva A (2022) "I can get no e-satisfaction". what analytics say? evidence using satisfaction data from e-commerce. J Retail Consum Serv. https://doi.org/10.1016/j.jretconser.2022.102954
Griva A, Bardaki C, Pramatari K, Doukidis G (2021) Factors affecting customer analytics: evidence from three retail cases. Inf Syst Front. https://doi.org/10.1007/s10796-020-10098-1
Griva A, Zampou E, Stavrou V, Papakiriakopoulos D, Doukidis G (2022) A two-stage business analytics approach to perform behavioural and geographic customer segmentation using e-commerce delivery data. J Decis Syst. https://doi.org/10.1080/12460125.2022.2151071

Guney S, Peker S, Turhan C (2020) A combined approach for customer profiling in video on demand services using clustering and association rule mining. IEEE Access 8:84326–84335. https://doi.org/10.1109/ACCESS.2020.2992064

Hafshejani ZY, Kaedi M, Fatemi A (2018) Improving sparsity and new user problems in collaborative filtering by clustering the personality factors. Electron Commer Res 18(4):813–836. https://doi.org/10.1007/s10660-018-9287-x

Hiziroglu A (2013) Soft computing applications in customer segmentation: state-of-art review and critique. Expert Syst Appl 40(16):6491–6507. https://doi.org/10.1016/j.eswa.2013.05.052

Hiziroglu A, Sisci M, Cebeci HI, Seymen OF (2018) An empirical assessment of customer lifetime value models within data mining. Baltic J Modern Comput 6(4): 434–448. https://doi.org/10.22364/bjmc.2018.6.4.08

Hjort K, Lantz B, Ericsson D, Gattorna J (2013) Customer segmentation based on buying and returning behaviour. Int J Phys Distrib Logist Manag 43(10):852–865. https://doi.org/10.1108/IJPDLM-02-2013-0020

Hong T, Kim E (2012) Segmenting customers in online stores based on factors that affect the customer's intention to purchase. Expert Syst Appl 39(2):2127–2131. https://doi.org/10.1016/j.eswa.2011.07.114

Hotelling H (1933) Analysis of a complex of statistical variables into principal components. J Educ Psychol 24(6):417

Hsieh NC (2004) An integrated data mining and behavioral scoring model for analyzing bank customers. Expert Syst Appl 27(4):623–633. https://doi.org/10.1016/j.eswa.2004.06.007

Hsu P-Y, Huang C-W (2020) IECT: a methodology for identifying critical products using purchase transactions. Appl Soft Comput. https://doi.org/10.1016/j.asoc.2020.106420

Hsu C-C, Y-gC Chen (2007) Mining of mixed data with application to catalog marketing. Expert Syst Appl 32(1):12–23. https://doi.org/10.1016/j.eswa.2005.11.017

Hsu F-M, Lu L-P, Lin C-M (2012) Segmenting customers by transaction data with concept hierarchy. Expert Syst Appl 39(6):6221–6228. https://doi.org/10.1016/j.eswa.2011.12.005

Hu Y-H, Yeh T-W (2014) Discovering valuable frequent patterns based on RFM analysis without customer identification information. Knowl-Based Syst 61:76–88. https://doi.org/10.1016/j.knosys.2014.02.009

Hughes AM (1994) Strategic database marketing: the masterplan for starting and managing a profitable, customer-based marketing program. Irwin Professional, USA

Jadwal PK, Pathak S, Jain S (2022) Analysis of clustering algorithms for credit risk evaluation using multiple correspondence analysis. Microsyst Technol-Micro- Nanosystemsinf Storage Process Syst 28(12):2715–2721. https://doi.org/10.1007/s00542-022-05310-y

Jiang T, Tuzhilin A (2009) Improving personalization solutions through optimal segmentation of customer bases. IEEE Trans Knowl Data Eng 21(3):305–320. https://doi.org/10.1109/TKDE.2008.163

Jonker JJ, Piersma N, van den Poel D (2004) Joint optimization of customer segmentation and marketing policy to maximize long-term profitability. Expert Syst Appl 27(2):159–168. https://doi.org/10.1016/j.eswa.2004.01.010

Kanchanapoom K, Chongwatpol J (2022) Integrated customer lifetime value (CLV) and customer migration model to improve customer segmentation. J Mark Anal. https://doi.org/10.1057/s41270-022-00158-7

Kang P, Cho S, MacLachlan DL (2012) Improved response modeling based on clustering, under-sampling, and ensemble. Expert Syst Appl 39(8):6738–6753. https://doi.org/10.1016/j.eswa.2011.12.028

Kass GV (1980) An exploratory technique for investigating large quantities of categorical data. J R Stat Soc Ser C (Appl Stat) 29(2):119–127

Kendall MG (1938) A new measure of rank correlation. Biometrika 30(1/2):81–93

Kennedy J, Eberhart R (1995) Particle swarm optimization. In: Proceedings of ICNN'95-international conference on neural networks (vol 4, pp 1942–1948)

Kim KJ, Ahn H (2004) Using a clustering genetic algorithm to support customer segmentation for personalized recommender systems. In: Kim TG (eds) Artificial intelligence and simulation (vol 3397, pp 409–415)

Kohonen T (1982) Self-organized formation of topologically correct feature maps. Biol Cybern 43(1):59–69

Krishna GJ, Ravi V (2021) High utility itemset mining using binary differential evolution: an application to customer segmentation. Expert Syst Appl. https://doi.org/10.1016/j.eswa.2021.115122
Kumar V, Venkatesan R, Reinartz W (2008) Performance implications of adopting a customer-focused sales campaign. J Mark 72(5):50–68
Lam HY, Tsang YP, Wu CH, Tang V (2021) Data analytics and the P2P cloud: an integrated model for strategy formulation based on customer behaviour. Peer-to-Peer Netw Appl 14(5):2600–2617. https://doi.org/10.1007/s12083-020-00960-z
Lazarsfeld PF (1950) The logical and mathematical foundation of latent structure analysis. Stud Soc Psychol World War II Vol. IV Meas Predict 362–412
Li J, Wang K, Xu L (2009) Chameleon based on clustering feature tree and its application in customer segmentation. Ann Oper Res 168(1):225–245. https://doi.org/10.1007/s10479-008-0368-4
Li K, Rollins J, Yan E (2018) Web of science use in published research and review papers 1997–2017: a selective, dynamic, cross-domain, content based analysis. Scientometrics 115(1):1–20
Liao J, Jantan A, Ruan Y, Zhou C (2022) Multi-behavior RFM model based on improved SOM neural network algorithm for customer segmentation. IEEE Access 10:122501–122512. https://doi.org/10.1109/ACCESS.2022.3223361
Liu D-R, Lai C-H, Lee W-J (2009) A hybrid of sequential rules and collaborative filtering for product recommendation. Inf Sci 179(20):3505–3519. https://doi.org/10.1016/j.ins.2009.06.004
Liu Y, Li H, Peng G, Lv B, Zhang C (2015) Online purchaser segmentation and promotion strategy selection: evidence from Chinese e-commerce market. Ann Oper Res 233(1):263–279. https://doi.org/10.1007/s10479-013-1443-z
Lloyd S (1982) Least squares quantization in PCM. IEEE Trans Inf Theory 28(2):129–137
Logesh R, Subramaniyaswamy V, Vijayakumar V, Gao X-Z, Wang G-G (2020) Hybrid bio-inspired user clustering for the generation of diversified recommendations. Neural Comput Appl 32(7):2487–2506. https://doi.org/10.1007/s00521-019-04128-6
Ma X, Lu H, Gan Z, Zhao Q (2016) An exploration of improving prediction accuracy by constructing a multi-type clustering based recommendation framework. Neurocomputing 191:388–397. https://doi.org/10.1016/j.neucom.2016.01.040
MacQueen J (1967) Some methods for classification and analysis of multivariate observations. In: Proceedings of the fifth Berkeley symposium on mathematical statistics and probability (vol 1, pp 281–297)
Madzik P, Shahin A (2021) Customer categorization using a three-dimensional loyalty matrix analogous to FMEA. Int J Qual Reliab Manag 38(8):1833–1857. https://doi.org/10.1108/IJQRM-05-2020-0179
Maimon O, Rokach L (2005) Data mining and knowledge discovery handbook. Springer, Berlin
Mensouri D, Azmani A, Azmani M (2022) K-means customers clustering by their RMFT and score satisfaction analysis. Int J Adv Comput Sci Appl 13(6): 469–476. https://doi.org/10.14569/IJACSA.2022.0130658
Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J (2013) Distributed representations of words and phrases and their compositionality. Adv Neural Inf Process Syst 26
Mosa M, Agami N, Elkhayat G, Kholief M (2022) A novel hybrid segmentation approach for decision support: a case study in banking. Comput J. https://doi.org/10.1093/comjnl/bxac009
Mulhern FJ (1999) Customer profitability analysis: measurement, concentration, and research directions. J Interact Mark 13(1):25–40
Munusamy S, Murugesan P (2020) Modified dynamic fuzzy c-means clustering algorithm—application in dynamic customer segmentation. Appl Intell 50(6):1922–1942. https://doi.org/10.1007/s10489-019-01626-x
Nakano S, Kondo FN (2018) Customer segmentation with purchase channels and media touchpoints using single source panel data. J Retail Consum Serv 41:142–152. https://doi.org/10.1016/j.jretconser.2017.11.012
Nalmpantis C, Vrakas D (2019) Signal2vec: time series embedding representation. In: International conference on engineering applications of neural networks (pp 80–90)
Nemati Y, Mohaghar A, Alavidoost MH, Babazadeh H (2018) A CLV-based framework to prioritize promotion marketing strategies: a case study of telecom industry. Iran J Manag Stud 11 (3): 437–462https://doi.org/10.22059/ijms.2018.242492.672837
Nguyen SP (2021) Deep customer segmentation with applications to a Vietnamese supermarkets' data. Soft Comput 25(12):7785–7793. https://doi.org/10.1007/s00500-021-05796-0

Nie D, Cappellari P, Roantree M (2021) A methodology for classification and validation of customer datasets. J Bus Ind Mark 36(5):821–833. https://doi.org/10.1108/JBIM-02-2020-0077
Nilashi M, Samad S, Minaei-Bidgoli B, Ghabban F, Supriyanto E (2021) Online reviews analysis for customer segmentation through dimensionality reduction and deep learning techniques. Arab J Sci Eng 46(9):8697–8709. https://doi.org/10.1007/s13369-021-05638-z
Ozer M (2001) User segmentation of online music services using fuzzy clustering. OMEGA-Int J Manag Sci 29(2):193–206. https://doi.org/10.1016/S0305-0483(00)00042-6
Pawlak Z (1982) Rough sets. Int J Comput Inf Sci 11(5):341–356
Pearson K (1901) LIII. On lines and planes of closest fit to systems of points in space. Lond Edinb Dublin Philos Mag J Sci 2(11):559–572
Peker S, Kocyigit A, Eren PE (2017) LRFMP model for customer segmentation in the grocery retail industry: a case study. Mark Intell Plan 35(4):544–559. https://doi.org/10.1108/MIP-11-2016-0210
Pratama BY, Budi I, Yuliawati A (2020) Product recommendation in offline retail industry by using collaborative filtering. Int J Adv Comput Sci Appl 11(9):635–643
Rahim MA, Mushafiq M, Khan S, Arain ZA (2021) RFM-based repurchase behavior for customer classification and segmentation. J Retail Consum Serv. https://doi.org/10.1016/j.jretconser.2021.102566
Ramadas M, Abraham A (2018) Data clustering using eDE, an enhanced differential evolution algorithm with fuzzy c-means technique. Turk J Electr Eng Comput Sci 26(2):867–881. https://doi.org/10.3906/elk-1706-104
Rapecka A, Dzemyda G (2015) A new recommendation method for the user clustering-based recommendation system. Inf Technol Control 44(1):54–63. https://doi.org/10.5755/j01.itc.44.1.5931
Ravasan AZ, Mansouri T (2015) A fuzzy ANP based weighted RFM model for customer segmentation in auto insurance sector. Int J Inf Syst Serv Sect 7(2):71–86. https://doi.org/10.4018/ijisss.2015040105
Reddy CK, Vinzamuri B (2018) A survey of partitional and hierarchical clustering algorithms. In: Data clustering (pp 87–110). Chapman and Hall, London. https://doi.org/10.1201/9781315373515-4
Rezaeinia SM, Rahmani R (2016) Recommender system based on customer segmentation (RSCS). Kybernetes 45(6):946–961. https://doi.org/10.1108/K-07-2014-0130
Rokach L (2010) A survey of clustering algorithms. In: Data mining and knowledge discovery handbook (pp 269–298). Springer US, Boston. https://doi.org/10.1007/978-0-387-09823-414
Rousseeuw PJ (1987) Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. J Comput Appl Math 20:53–65. https://doi.org/10.1016/0377-0427(87)90125-7
Safari F, Safari N, Montazer GA (2016) Customer lifetime value determination based on RFM model. Mark Intell Plan 34(4):446–461. https://doi.org/10.1108/MIP-03-2015-0060
Sari JN, Nugroho LE, Ferdiana R, Santosa PI (2016) Review on customer segmentation technique on ecommerce. Adv Sci Lett 22(10):3018–3022
Sarvari PA, Ustundag A, Takci H (2016) Performance evaluation of different customer segmentation approaches based on RFM and demographics analysis. Kybernetes 45(7):1129–1157. https://doi.org/10.1108/K-07-2015-0180
Shi Z, Pun-Cheng LS (2019) Spatiotemporal data clustering: a survey of methods. ISPRS Int J Geoinf 8(3):112
Simoes D, Nogueira J (2021) Learning about the customer for improving customer retention proposal of an analytical framework. J Mark Anal. https://doi.org/10.1057/s41270-021-00126-7
Sivaguru M, Punniyamoorthy M (2021) Performance-enhanced rough k-means clustering algorithm. Soft Comput 25(2):1595–1616. https://doi.org/10.1007/s00500-020-05247-2
Sivaramakrishnan N, Subramaniyaswamy V, Ravi L, Vijayakumar V, Gao X-Z, Sri SLR (2020) An effective user clustering-based collaborative filtering recommender system with grey wolf optimisation. Int J Bio-Inspir Comput 16(1):44–55. https://doi.org/10.1504/IJBIC.2020.108999
Sokol O, Holy V (2021) The role of shopping mission in retail customer segmentation. Int J Mark Res 63(4):454–470. https://doi.org/10.1177/1470785320921011
Song Q, Shepperd M (2006) Mining web browsing patterns for E-commerce. Comput Ind 57(7):622–630. https://doi.org/10.1016/j.compind.2005.11.006
Srilakshmi M, Chowdhury G, Sarkar S (2022) Two-stage system using item features for next-item recommendation. Intell Syst Appl 14:200070. https://doi.org/10.1016/j.iswa.2022.200070
Statista.com (2022) Video-streaming (SVOD). Retrieved 12-02-2022, from https://www.statista.com/outlook/dmo/digital-media/videoon-demand/video-streaming-svod/worldwide

Stormi K, Lindholm A, Laine T, Korhonen T (2020) RFM customer analysis for product-oriented services and service business development: an interventionist case study of two machinery manufacturers. J Manag Gov 24(3):623–653. https://doi.org/10.1007/s10997-018-9447-3

Sun F, Liu J, Wu J, Pei C, Lin X, Ou W, Jiang P (2019) Bert4rec: sequential recommendation with bidirectional encoder representations from transformer. In: Proceedings of the 28th ACM international conference on information and knowledge management (pp 1441–1450)

Tabianan K, Velu S, Ravi V (2022) K-means clustering approach for intelligent customer segmentation using customer purchase behavior data. Sustainability. https://doi.org/10.3390/su14127243

Teichert T, Shehu E, vonWartburg I (2008) Customer segmentation revisited: the case of the airline industry. Transp Res Part A Policy Pract 42(1):227–242. https://doi.org/10.1016/j.tra.2007.08.003

Tercan H, Bitter C, Bodnar T, Meisen P, Meisen T (2021) Evaluating a session-based recommender system using prod2vec in a commercial application. In: Proceedings of the 23rd international conference on enterprise information systems (vol 1: Iceis, pp 610–617). SciTePress. https://doi.org/10.5220/0010400706100617

Tsai C-F, Hu Y-H, Lu Y-H (2015) Customer segmentation issues and strategies for an automobile dealership with two clustering techniques. Expert Syst 32(1):65–76. https://doi.org/10.1111/exsy.12056

Umuhoza E, Ntirushwamaboko D, Awuah J, Birir B (2020) Using unsupervised machine learning techniques for behavioral-based credit card users segmentation in Africa. SAIEE Afr Res J 111(3): 95–101. https://doi.org/10.23919/SAIEE.2020.9142602

Valentini S, Neslin SA, Montaguti E (2020) Identifying omnichannel deal prone segments, their antecedents, and their consequences. J Retail 96(3):310–327. https://doi.org/10.1016/j.jretai.2020.01.003

Vasile F, Smirnova E, Conneau A (2016) Meta-prod2vec: product embeddings using side-information for recommendation. In: Proceedings of the 10th ACM conference on recommender systems (pp 225–232)

Verdu SV, Garcia MO, Senabre C, Marin AG, Garcia Franco FJ (2006) Classification, filtering, and identification of electrical customer load patterns through the use of self-organizing maps. IEEE Trans Power Syst 21(4):1672–1682. https://doi.org/10.1109/TPWRS.2006.881133

Vom Brocke J, Simons A, Riemer K, Niehaves B, Plattfaut R, Cleven A (2015) Standing on the shoulders of giants: challenges and recommendations of literature search in information systems research. Commun Assoc Inf Syst 37(1):9

Wan M, Li L, Xiao J, Yang Y, Wang C, Guo X (2010) CAS based clustering algorithm for Web users. Nonlinear Dyn 61(3):347–361. https://doi.org/10.1007/s11071-010-9653-2

Wang C-H (2010) Apply robust segmentation to the service industry using kernel induced fuzzy clustering techniques. Expert Syst Appl 37(12):8395–8400. https://doi.org/10.1016/j.eswa.2010.05.042

Wang FH, Shao HM (2004) Effective personalized recommendation based on time-framed navigation clustering and association mining. Expert Syst Appl 27(3):365–377. https://doi.org/10.1016/j.eswa.2004.05.005

Wang Q, Zhang B (2021) Research and implementation of the customer-oriented modern hotel management system using fuzzy analytic hiererchical process (FAHP). J Intell Fuzzy Syst 40(4):8277–8285. https://doi.org/10.3233/JIFS-189650

Wang Y, Zhou J-T, Li X, Song X (2020) Effective user preference clustering in web service applications. Comput J 63(11):1633–1643. https://doi.org/10.1093/comjnl/bxz090

Wong E, Wei Y (2018) Customer online shopping experience data analytics integrated customer segmentation and customised services prediction model. Int J Retail Distrib Manag 46(4):406–420. https://doi.org/10.1108/IJRDM-06-2017-0130

Wu W-W (2011) Segmenting and mining the ERP users' perceived benefits using the rough set approach. Expert Syst Appl 38(6):6940–6948. https://doi.org/10.1016/j.eswa.2010.12.030

Wu R-S, Chou P-H (2011) Customer segmentation of multiple category data in e-commerce using a soft-clustering approach. Electron Commer Res Appl 10(3):331–341. https://doi.org/10.1016/j.elerap.2010.11.002

Wu J, Shi L, Lin W-P, Tsai S-B, Li Y, Yang L, Xu G (2020) An empirical study on customer segmentation by purchase behaviors using a RFM model and k-means algorithm. Math Probl Eng. https://doi.org/10.1155/2020/8884227

Wu J, Shi L, Yang L, Niu X, Li Y, Cui X, Zhang Y (2021) User value identification based on improved RFM model and k-means plus plus algorithm for complex data analysis. Wirel Commun Mob Comput. https://doi.org/10.1155/2021/9982484

Wu Z, Jin L, Zhao J, Jing L, Chen L (2022) Research on segmenting e-commerce customer through an improved K-medoids clustering algorithm. Comput Intell Neurosci. https://doi.org/10.1155/2022/9930613

Xie XL, Beni G (1991) A validity measure for fuzzy clustering. IEEE Trans Pattern Anal Mach Intell 13(8):841–847

Zeithaml VA, Rust RT, Lemon KN (2001) The customer pyramid: creating and serving profitable customers. Calif Manag Rev 43(4):118–142

Zhang C-X, Zhang Z-K, Yu L, Liu C, Liu H, Yan X-Y (2014) Information filtering via collaborative user clustering modeling. Phys A Stat Mech Appl 396:195–203. https://doi.org/10.1016/j.physa.2013.11.024

Zhang S, Yao L, Sun A, Tay Y (2019) Deep learning based recommender system: a survey and new perspectives. ACM Comput Surv (CSUR) 52(1):1–38

Zhang F, Qi S, Liu Q, Mao M, Zeng A (2020) Alleviating the data sparsity problem of recommender systems by clustering nodes in bipartite networks. Expert Syst Appl. https://doi.org/10.1016/j.eswa.2020.113346

Zhao H-H, Luo X-C, Ma R, Lu X (2021) An extended regularized K-means clustering approach for high-dimensional customer segmentation with correlated variables. IEEE Access 9:48405–48412. https://doi.org/10.1109/ACCESS.2021.3067499

Zhou J, Wei J, Xu B (2021) Customer segmentation by web content mining. J Retail Consum Serv. https://doi.org/10.1016/j.jretconser.2021.102588

Zhu H, Jia Z, Peng H, Li L (2007) Chaotic ant swarm. In: Third international conference on natural computation (ICNC 2007) (vol 3, pp 446–450). https://doi.org/10.1109/ICNC.2007.296

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Authors and Affiliations

Miguel Alves Gomes[1] · **Tobias Meisen**[1]

✉ Miguel Alves Gomes
alvesgomes@uni-wuppertal.de

Tobias Meisen
meisen@uni-wuppertal.de

[1] Chair for Technologies and Management of Digital Transformation, University of Wuppertal, Rainer-Gruenter-Str. 21, 42119 Wuppertal, North Rhine-Westphalia, Germany

CUSTOMER SEGMENTATION USING MACHINE LEARNING

AMAN BANDUNI[*1], Prof ILAVENDHAN A.[2]
[*1,2]School of Computing Science & Engineering,
Galgotias University, Greater Noida, U.P.
[*1]abanduni.5@gmail.com,[2]ilavendhan@galgotiasuniversity.edu.in

Abstract-The emergence of many competitors and entrepreneurs has caused a lot of tension among competing businesses to find new buyers and keep the old ones. As a result of the predecessor, the need for exceptional customer service becomes appropriate regardless of the size of the business.[2] Furthermore, the ability of any business to understand the needs of each of its customers will provide greater customer support in providing targeted customer services and developing customized customer service plans. This understanding is possible through structured customer service. Each segment has customers who share the same market features.[5] Big data ideas and machine learning have promoted greater acceptance of automated customer segmentation approaches in favor of traditional market analytics that often do not work when the customer base is very large. In this paper, the k-means clustering algorithm is used for this purpose.[8] The Sklearn liabrary was developed for the k-Means algorithm (found in the Appendix) and the program is trained using a 100-pattern two-factor dataset derived from the retail trade. Characteristics of average number of customer purchases and average number of monthly customers.

Keywords- data mining; machine learning; big data; customer segment; k-Mean algorithm; sklearn; extrapolation;

I. Introduction

Over the years, increased competition among businesses and the availability of large-scale historical data has resulted in widespread use of data mining techniques to find critical and strategic information that is hidden in organizations' information.[1] Data mining is the process of extracting logical information from a dataset and presenting it in a human-accessible manner for decision support. Data mining techniques distinguish fields such as statistics, artificial intelligence, machine learning, and data systems. Data mining applications include, but are not limited to bioinformatics, weather forecasting, fraud detection, financial analysis and customer segmentation. The key to this paper is to identify customer segments in a commercial business using the data mining method. Customer segmentation is a group of business customer base called customer segment such that each customer segment has customers who share the same market characteristics.[5] These differences are based on factors that directly or indirectly affect the market or business such as product preferences or expectations, location, behavior and so on. The importance of customer segmentation includes, inter alia, the ability of a business to customize market plans that would be appropriate for each segment of its customers;[6] Support for business decisions based on risky environments such as credit relationships with its customers; Identify products related to individual components and how to manage demand and supply power; Interdependence and interaction between consumers, between products, or between customers and products are revealed, which the business may not be aware of; The ability to predict customer declines, and which customers are likely to have problems and raise other market research questions and provide clues to find solutions.

Buried in a database of integrated data proved to be effective for detecting subtle but subtle patterns or relationships. This mode of learning is classified under supervised learning. Integration algorithms include the K-Means algorithm, K-nearest algorithm, sorting map (SOM), and more.[4] These algorithms, without prior knowledge of the data, are able to identify groups in them by repeatedly comparing input patterns, as long as static aptitude in training examples is achieved based on subject matter or process. Each set has data points that have very close similarities but differ greatly from the data points of other groups. Integration has great applications in pattern recognition, image analysis, and bioinformatics and so on.[15] In this paper the k-means clustering algorithm was implemented in the customer segment. The scalar library (Appendix) of the K-Means algorithm was developed, and training was started using a standard silhouette -score with two feature sets of 100 training patterns found in the retail trade. After several indications, four stable intervals or customer segments were identified. Two factors are considered in combination with the number of items a customer purchases per month and the average number of customers per month. From the dataset, four customers or categories are classified and labeled as follows: cluster_metrics_1, cluster_metrics_2, cluster_metrics_3, cluster_metrics_4.

II. Literature Survey

A. Customer Classification

Over the years, the commercial world has become more competitive, as organizations such as these have to meet the needs and desires of their customers, attract new customers, and thus improve their businesses.[6] The task of identifying and meeting the needs and requirements of every customer in the business is very difficult. This is because customers can vary according to their needs, wants, demographics, size, taste and taste, features etc. As it is, it is a bad practice to treat all customers equally in business. This challenge has adopted the concept of customer segmentation or market segmentation, where consumers are divided into subgroups or segments, where members of each subcategory exhibit similar market behaviors or characteristics.[9] Accordingly, customer segmentation is the process of dividing the market into indigenous groups.

B Big Data

Recently, Big Data research has gained momentum. Defines big data - a term that describes a large number of formal and informal data, which cannot be analyzed using traditional methods and algorithms. Companies include billions of data about their customers, suppliers, and operations, and millions of internally connected sensors are sent to the real world on devices such as mobile phones and cars, sensing, manufacturing and communications data.[10] Ability to improve forecasting, save money, increase efficiency and improve various areas such as traffic control, weather forecasting, disaster prevention, finance, fraud control, business transactions, national security, education and healthcare. Big data is mainly seen in three Vs: volume, variability, and speed. Other 2Vs are available - authenticity and price, thus making it 5V.

C. data repository

Data collection is the process of collecting and measuring information against targeted changes in an established system, which enables one to answer relevant questions and evaluate the results.[12] Data collection is part of research in all fields of study including physical and social sciences, humanities and business. The purpose of all data collection is to obtain quality evidence that leads the analysis to construct concrete and misleading answers to the questions presented. We collected data from the UCI machine learning repository.

D. Clustering data

Clustering is the process of grouping information into a dataset based on some commonalities. There are several algorithms, which can be applied to datasets based on the provided condition.[7] However, no universal clustering algorithm exists, hence it becomes important to choose the appropriate clustering techniques. In this paper, we have implemented three clustering algorithms using the Python scalar library.

E. K-mein

K-means that an algorithm is one of the most popular classification algorithms. This clustering algorithm relies on centro, where each data point is placed in one of the overlapping ones, which is pre-sorted in the K-algorithm. Clusters are created that correspond to hidden patterns in the data that provide the necessary information to help decide execution. process. There are many ways to make assembling K-means, we will use the elbow method.

III. Methodology

The data used in this paper were collected from the UCI Machine Learning Repository. It is a set of geographic data, including all transactions that occur between 1/1/2/10 and 9/12/2011 in an unregistered and unregistered UK broker. The company mainly sells unique gifts to everyone at once. Many of the company's customers are shopkeepers.[10] The database has 8 attributes. These features include:

"Invoice: invoice number. By default, a 6-digit total number is assigned separately for each transaction. If this code starts with the letter 'c', it indicates a cancellation. "

Stockcode Code: Product (Item). Name, a 5-digit number assigned only to each unique product. "

"Definition: Product Name (Item). By Name."

“Price: The price of each product (item). Number. "

"Invoice: The date and time of the invitation. In terms of numbers, the date and time of each transaction. "

"UnitPrice: Price is one unit. Price, product price per unit of measure. "

"Customer: Customer Number. Name, 5-digit number to each customer. "

Country: Country name. Name, the name of the country where each customer resides. "

In this paper several steps were taken to obtain an accurate result. It includes a feature with Centro's first stage, allocation phase and update phase, which are the most common phase k-means algorithms.

A. Collect data

This is a data preparation phase. The feature usually helps to refine all data items at a standard rate to improve the performance of clustering algorithms.[12] Each data point varies from grade 2 to +2. Integration techniques that include min-max, decimal, and z-point are the standard z-signing strategy used to make things uneven before the dataset algorithm applies the k-Means algorithm.

B. Methods of customer classification

There are many ways to partition, which vary in severity, data requirements, and purpose. The following are some of the most commonly used methods, but this is not an incomplete list.[13] There are papers that discuss artificial neural networks, particle determination and complex types of ensemble, but are not included due to limited exposure. In future articles, I may go into some of these options, but for now, these general methods should suffice.

Each subsequent section of this article will include a basic description of the method, as well as a code example for the method used. If you do not have the expertise, well, just skip the code and you have to get a good handle on each of the 4 sub-sections included in this article.[14]

C. Group analysis

Group analysis is an integration or unification, approach to consumers based on their similarity.

There are 2 main types of categorical group analysis in market policy: hierarchical group analysis, and classification (Miller, 2015). In the meantime, we will discuss how to classify groups, called k-methods.

D. K. Means encounter

The K-means clustering algorithm is an algorithm often used to draw insights into formats and differences within a database.[13] In marketing, it is often used to build customer segments and understand the behavior of these unique segments. Let's try to build an assembly model in Python's environment.

E. Centroids initiation

Selected cents or initials were selected. Figure 1 introduces the beginning of graduate centers. The four selected centers, shown in different sizes, were selected using the Forgi method. In Forgy's method, data points are randomly selected as cluster centroids using k (k = 4 in this case).

Technical introduction: -

The code below was created in the Jupiter manual using Python 3.x and some Python packages for editing, processing, analyzing, and visualizing information.[11]

Most of the codes below come from the Github package of a book called Hands-on Data Science for Marketing. The book is available on Amazon or OilReilly if you are a customer.

The open source data cost used in the following code comes from Irwin's machine learning repository.

IV. Proposed Model

A) Import packages and data:

To begin, we import the necessary packages to do our analysis and then the xlsx (Excel spreadsheet) data file.[12] If you want to follow up with the same data, you have to download it from UCI. For this example, I place the xlsx file in the folder (directory) where I present Jupiter's notebook.

B) Data cleaning:

After importing the package and data, we will see that the data is not as helpful as that, so we need to clean and organize this data in a way that we can create more actionable insights.

C) Normalize the data:

The K-means area unit is sensitive to the scale of the information used, such as clustering algorithms, so we would like to generalize the information.[15]

A screenshot of the StackExchange answer below discusses why standardization or normalization is necessary for data used in K-means clustering. The screenshot is linked to the StackExchange question, so you can click on it and read the entirety of the discussion if you want more information.[10]

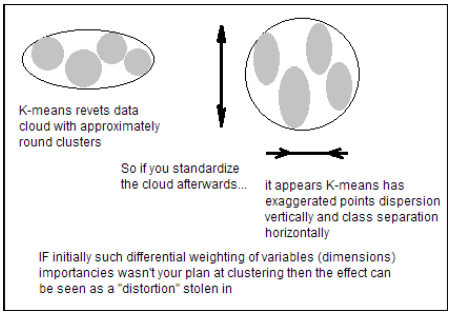


Fig. 1 Standardistion and Normalisation

D) Select the optimal number of groups:

Okay, we are ready to run cluster analysis. But first, we need to find out how many groups we want to use. There are several approaches to selecting the number of groups to use, but I am going to cover two in this article: (1) the silhouette coefficient, and (2) the elbow method.[7]

E) Silhouette (clustering):

The silhouette refers to how to interpret and validate consistency within data structures. This method shows a diagram of how well each item is organized. [1]

The value of a silhouette is a measure of how something is more similar in its collection (combination) than other groups (partitions). The silhouette goes from -1 to +1, where a higher value indicates that an object matches its collection properly and is compared to neighboring groups. If several objects have a high value, the integration configuration is appropriate. If most points have a value or a negative value, the coordinate system may have too many or too few clusters.

The silhouette can be calculated with any distance metric, such as the Euclidean distance or the Manhattan distance.

Now that we know a whole lot of silhouettes, we use code to find the right number of groups.

```
Silhouette Score for 4 Clusters: 0.4114
Silhouette Score for 5 Clusters: 0.3773
Silhouette Score for 6 Clusters: 0.3785
Silhouette Score for 7 Clusters: 0.3913
Silhouette Score for 8 Clusters: 0.3810
```

Fig. 2 Silhoette Score

Cluster 4 had the most complete silhouette fit, indicating that 4 may be the best number of clusters. But we'll see twice the way to the elbow.

F) Elbow criterion method (with the sum of squared errors) (SSE):

The idea behind the elbow method is to run a k-mean correlation in the data given for the k value (num_clusters, e.g. k = 1 to 10), and for each k value, calculate the sum of the squared errors (SSE). is.

Then, adjust the SSE line for each k value. If the line graph looks like a hand - a red circle (in the form of an angle) below the line of the line, the "elbow" on the hand is the correct value (collection value).[6] Here, we want to reduce SSE. SSE usually falls to 0 as we go up k (and SSE is 0 where k is equal to the number of data points, because where each data point has its own set, and there is no error between it and its trunk) .

The objective is therefore to select a smaller value of k, which still has a lower SSE, and the cone usually represents where it begins to return negatively with increasing.

Well, with the correct understanding of the elbow mechanism at hand, let's use the elbow method to see if it agrees with our previous results suggesting 4 sets.

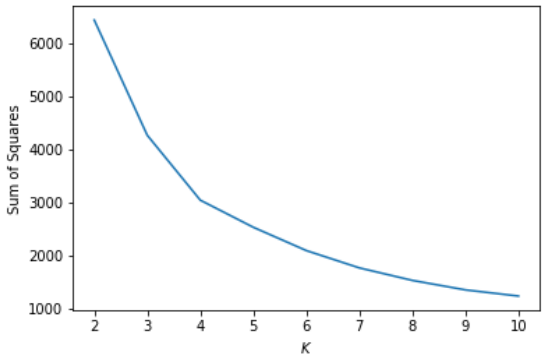


Fig. 3 Elbow Graph exported from my working Jupyter notebook

Based on the graph above, it looks like K = 4, or 4 clusters is the correct number of clusters in this analysis. Now translates the customer segments provided by these components.

G) Explaining customer segment

CustomerID	TotalSales	OrderCount	AvgOrderValue	Cluster
12346.0	1.724999	-1.731446	1.731446	0
12347.0	1.457445	1.064173	1.401033	2
12348.0	0.967466	0.573388	0.929590	2
12349.0	0.944096	-1.730641	1.683093	0
12350.0	-0.732148	-1.729835	0.331622	0
12352.0	1.193114	1.309162	0.169639	2
12353.0	-1.636352	-1.729029	-1.570269	3
12354.0	0.508917	-1.728223	1.612981	0
12355.0	-0.386422	-1.727417	0.970690	0
12356.0	1.268868	0.158357	1.557375	2

Fig. 4 Customer table

Now we have to combine the matrix of integration and see what we can gather from the standard data for each cluster.

	TotalSales	OrderCount	AvgOrderValue
0	0.244056	0.740339	-0.640559
1	-0.137750	-0.851493	0.792034
2	1.203710	0.996813	0.879446
3	-1.235415	-0.784442	-1.056848

Fig. 5 Clusters

In the following section, we need to visualize clustering by adding different columns in the x and y axes. Let's see what we say.

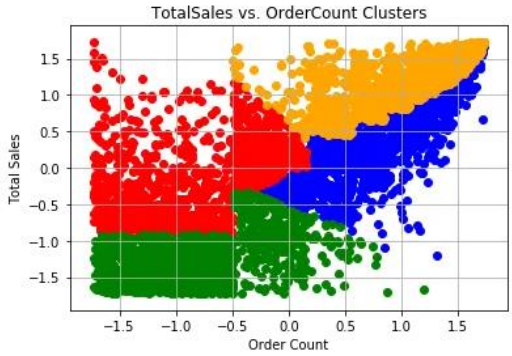


Fig. 6 TotalSales vs OrderCount Clusters

Green customers have the lowest price and lowest order count, meaning they are the lowest bidder. On the other hand, orange customers have the highest total sales and highest order count, indicating that they are the highest priced customers.

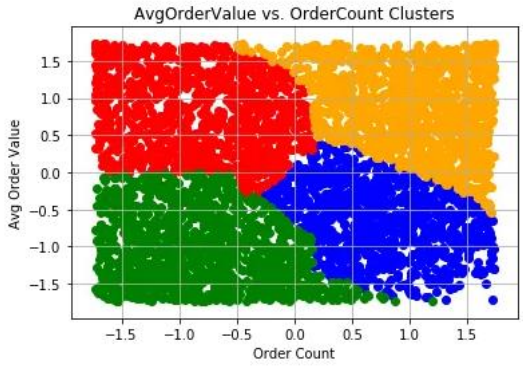


Fig. 7 AvgOrderValue vs OrderCount Clusters

In this structure, we consider the average order value versus the order value. Once again, green buyers have the lowest prices and orange has the highest customer prices.

You can see it this way. You can target customers in red graphics and try to find ways to increase your order count via email reminders or SMS notifications directed to other identification features. Maybe you can give them a discount when they come back within 30 days. Ideally, you can provide a delayed coupon (which will be used at some point) at checkout.

Similarly, with customers who are in the blue segment, you may want to try other sales and marketing strategies for the cart. Possibly the fastest offer based on market basket analysis (see section on market basket analysis below).

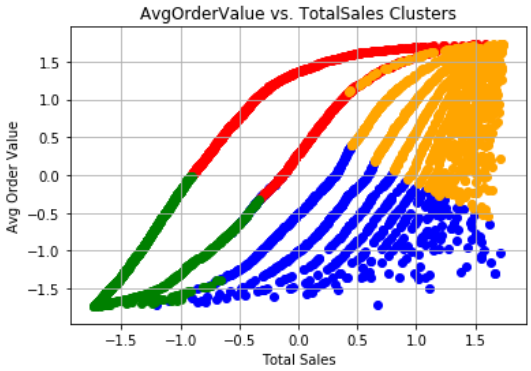


Fig. 8 AvgOrderValue vs TotalSales Clusters

In this building, it has an average price and order compared to the total retail price. This structure also reinforces the previous 2 sites in identifying the orange group as the highest value customer, green as the lowest priced customer, and blue and red as the high potential customers.

From a development perspective, I focus my attention on the blue and red collections. I try to better understand each encounter and their intelligent behavior on site as to which team to focus on first and introduce some test cycles.

H) Best-selling item by segment

We know that we have 4 categories and we know how much they spend on each purchase, their total usage and the number of their orders. The next thing we can do is to help customer segments better understand which items sell best in each segment.

Description	StockCode
JUMBO BAG RED RETROSPOT	1129
REGENCY CAKESTAND 3 TIER	1080
WHITE HANGING HEART T-LIGHT HOLDER	1062
LUNCH BAG RED RETROSPOT	924
PARTY BUNTING	859

Fig. 9 StockCode

V. Result

Here, the result suggests that the orange cluster as the highest value customers, green as the lowest value customers, and blue and red as the high opportunity customers.

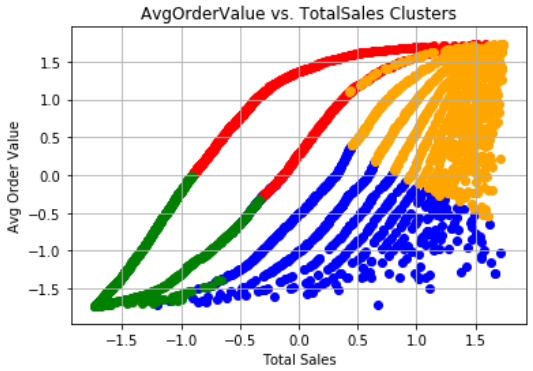


Fig. 8 AvgOrderValue vs ToatalSales Clusters

Result also concludes that the Jumbo Bag Red Retrospot is the best-selling item.

Description	StockCode
JUMBO BAG RED RETROSPOT	1129
REGENCY CAKESTAND 3 TIER	1080
WHITE HANGING HEART T-LIGHT HOLDER	1062
LUNCH BAG RED RETROSPOT	924
PARTY BUNTING	859

Fig. 9 StockCode

VI. Conclusion

As our dataset was unbalanced, in this paper we opted for internal clustering validation rather than external clustering verification, which relies on some external data such as labels. Internal cluster validation can be used to choose the clustering algorithm that best suits the dataset and vice versa can correctly cluster the data in the cluster.

Customer segmentation can have a positive impact on business if done properly.

So we can give people of orange bunches special discounts or gift vouchers to keep them for a long time and we can give discounts to people in blue and red clusters and advertise highly sold items to attract them , And for those of lower value who are in green clusters, we can organize feedback columns to find out what we can change to attract them.

Based on the above information, we now know that the Jumbo Bag Red Retrospot is the best-selling item by our most expensive team. With that information available, we can make recommendations for other potential customers in this section.

VII. References

[1] Blanchard, Tommy. Bhatnagar, Pranshu. Behera, Trash. (2019). Marketing Analytics Scientific Data: Achieve your marketing objectives with Python's data analytics capabilities. S.l: Packt printing is limited

[2] Griva, A., Bardaki, C., Pramatari, K., Papakiriakopoulos, D. (2018). Sales business analysis: Customer categories use market basket data. Systems Expert Systems, 100, 1-16.

[3] Hong, T., Kim, E. (2011). It separates consumers from online stores based on factors that affect the customer's intention to purchase. Expert System Applications, 39 (2), 2127-2131.

[4] Hwang, Y. H. (2019). Hands-on Advertising Science Data: Develop your machine learning marketing strategies… using python and r. S.l: Packt printing is limited

[5] Puwanenthiren Premkanth, - Market Classification and Its Impact on Customer Satisfaction and Special Reference to the Commercial Bank of Ceylon PLC.‖ Global Journal of Management and Business Publisher Research: Global Magazenals Inc. (USA). 2012. Print ISSN: 0975-5853. Volume 12 Issue 1.

[6] Puwanenthiren Premkanth, - Market Classification and Its Impact on Customer Satisfaction and Special Reference to the Commercial Bank of Ceylon PLC.‖ Global Journal of Management and Business Publisher Research: Global Magazenals Inc. (USA). 2012. Print ISSN: 0975-5853. Volume 12 Issue 1.

[7] Sulekha Goyat. "The basis of market segmentation: a critical review of the literature. European Journal of Business and Management www.iiste.org. 2011. ISSN 2222-1905 (Paper) ISSN 2222-2839 (Online). Vol 3, No.9, 2011

[8] By Jerry W Thomas. 2007. Accessed at: www.decisionanalyst.com on July 12, 2015.

[9] T.Nelson Gnanaraj, Dr.K.Ramesh Kumar N.Monica. AnuManufactured cluster analysis using a new algorithm from structured and unstructured data. International Journal of Advances in Computer Science and Technology. 2007. Volume 3, No.2.

[10] McKinsey Global Institute. Big data. The next frontier is creativity, competition and productivity. 2011. Accessed at: www.mckinsey.com/mgi on July 14, 2015.

[11] Jean Yan. - Big Data, Big Opportunities- Domains of Data.gov: Promote, lead, contribute, and collaborate in the big data era. 2013. Retrieved from http://www.meritalk.com/pdfs/bdx/bdx-whitepaper-090413.pdf July 14, 2015.

[12] A.K. Jain, M.N. Murty and P.J. Flynn.‖ Data Integration: A Review‖. ACM Computer Research. 1999. Vol. 31, No. 3.

[13] Vishish R. Patel1 and Rupa G. Mehta. MpImpact for External Removal and Standard Procedures for JCSI International International Science Issues Issues, Vol. 8, Appeals 5, No 2, September 2011 ISSN (Online): 1694-0814

[14] Jayant Tikmani, Sudhanshu Tiwari, Sujata Khedkar "Telecom Customer Classification Based on Group Analysis of K-methods", JIRCCE, Year: 2015.
[15] Vaishali R. Patel and Rupa G. Mehta “Impact of Outlier Removal and Normalization Approach in Modified k-Means Clustering Algorithm”, IJCSI,Year: 2011.

Recent Trends in Intensive Computing
M. Rajesh et al. (Eds.)

© 2021 The authors and IOS Press.
This article is published online with Open Access by IOS Press and distributed under the terms of the Creative Commons Attribution Non-Commercial License 4.0 (CC BY-NC 4.0).
doi:10.3233/APC210200

Customer Segmentation Using Machine Learning

Prof. Nikhil Patankar [a,1], Soham Dixit [a], Akshay Bhamare [a], Ashutosh Darpel [a] and Ritik Raina [a]
[a] *Dept. Of Information Technology Sanjivani College of Engineering, Kopargaon-423601 (MH), India*

Abstract. Nowadays Customer segmentation became very popular method for dividing company's customers for retaining customers and making profit out of them, in the following study customers of different of organizations are classified on the basis of their behavioral characteristics such as spending and income, by taking behavioral aspects into consideration makes these methods an efficient one as compares to others. For this classification a machine algorithm named as k-means clustering algorithm is used and based on the behavioral characteristic's customers are classified. Formed clusters help the company to target individual customer and advertise the content to them through marketing campaign and social media sites which they are really interested in.

Keywords. Machine learning, Customer segmentation, K-means algorithm

1. Introduction

Today many of the businesses are going online and, in this case, online marketing is becoming essential to hold customers, but during this, considering all customers as same and targeting all of them with similar marketing strategy is not very efficient way rather it's also annoys the customers by neglecting his or her individuality, so customer segmentation is becoming very popular and also became the efficient solution for this existing problem. Customer segmentation is defined as dividing company's customers on the basis of demographic (age, gender, marital status) and behavioral (types of products ordered, annual income) aspects. Since demographic characteristics does not emphasize on individuality of customer because same age groups may have different interests so behavioral aspects is a better approach for customer segmentation as its focus on individuality and we can do proper segmentation with the help of it.

2. Literature Survey

[1] A solution is proposed as distinguish the customers group into two groups named as premium and standard with the help of machine learning methods named as NEM, LiRM and LoRM [2].

[1] Prof. Nikhil Patankar, Sanjivani College of Engineering, Kopargaon, India.
Email: patankarnikhilit@sanjivani.org.in

Tushar Kansal, Suraj Bahuguna, Vishal Singh, Tanupriya Choudhury. "Customer Segmentation using K-means Clustering", International Conference on Computational Techniques, Electronics and Mechanical Systems (CTEMS).2018, In this paper customer segmentation on Telecom customers is achieved by using information such as age, interest, etc. with the help of cluster analysis method.

3. Use Case Diagram

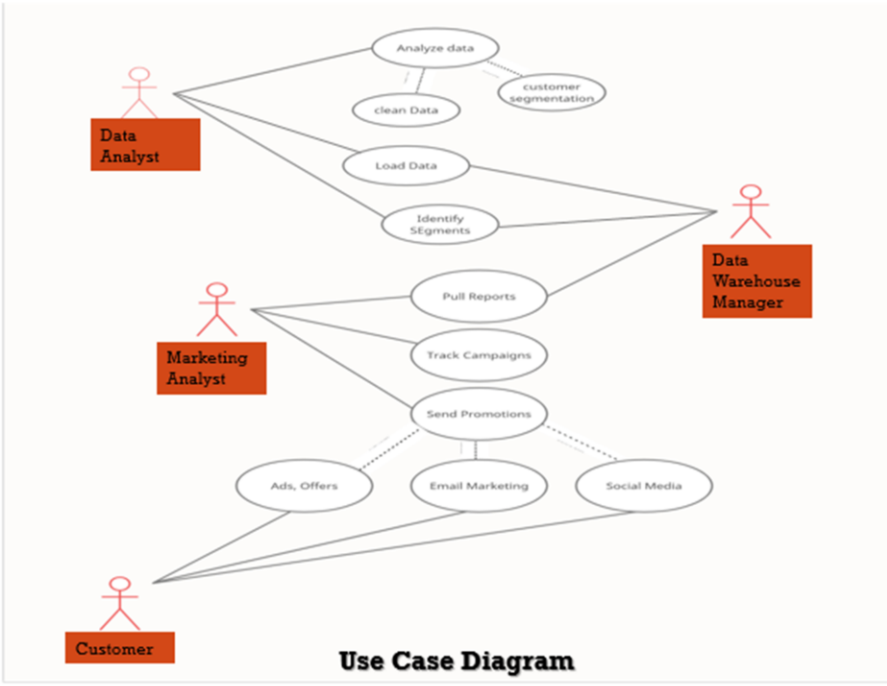


Figure 1. Use case Diagram

Use case diagram of proposed system consist of 4 users 1. Data Analyst 2. Marketing Analyst 3. Data Warehouse Manager 4. Customer in figure 1

And 6 use cases,

1. Analyze Data: analyst has the access to loaded data and analyst clean the data and perform analysis to form clusters.
2. Load Data: analyst log into database & view data & load into memory to work on it.
3. Identify Segments: analyst form report for segmented customer data and send to data warehouse and marketing analyst can access that data to form marketing strategies.
4. Pull Reports: marketing team can view & make edits on the reports, data for report is pulled from DW system.

5. Track Campaigns: The customer's interaction tracked by marketing team for success report.
6. Send Promotions: Marketing team send promotions through mail, social media ads, paid ads, coupons.

4. K-means Clustering Algorithm

K-means Clustering is a clustering Algorithm in which we are given with data points with its data set and features and the mechanism is to categories those data points into clusters as per their similarities.
The algorithm forms K clusters based on its similarity. To calculate the similarity K-means uses Euclidean distance measurement method.
Steps

i. In first step, we randomly initialize k points.
ii. K-means classifier categorizes each data point to its nearest mean and rewrite the mean's coordinates.
iii. Iteration is continuing up till all data points are classified.

5. Proposed System

In our system we including annual income and total spending as a feature for classification in figure2

1. **Data Gathering**: first, Data analyst fetch data required for analysis from database, format data i.e., remove all NA values from data & make data ready for processing.
2. **Feature Extraction**: Selects features which makes model more accurate, in our case features are annual income and spending score for efficient analysis.
3. **K-means Classifier**: After that, K means classifier performs clustering with respect to features provided to it,
4. **Hyper Parameter Tuning**: during forming groups to select optimal no of clusters we applied hyper parameter tuning which is achieved by Elbow method to choose optimal no of clusters.
   below graph is for elbow method which shows curve is getting flatter after 5 which indicates that 5 is optimal no of clusters we can form for better classification.

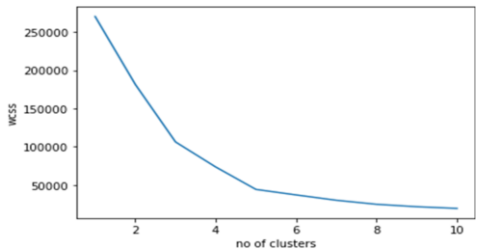


Figure 2. Elbow Method

5. **Data Visualization**: With the formed clusters marketing team can make different strategies for better targeting customers in figure 3.

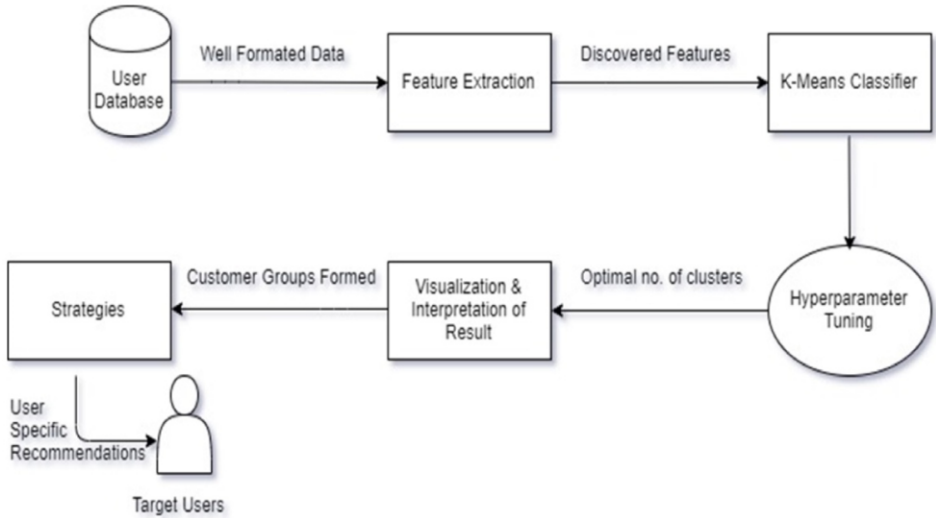


Figure 3. Flow of operation

6. Results

After analysis of data and classifying customers with features annual income and spending score, we got clusters of customers & with formed clusters marketing team form strategies for customers specific recommendation to make value out of them in figure 4.

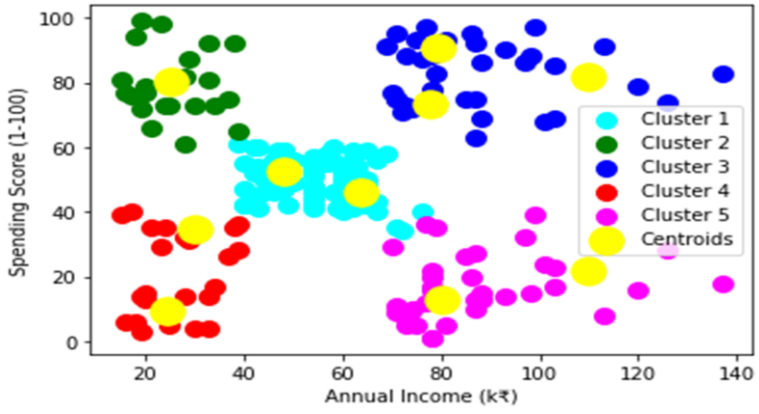


Figure 4. Final Cluster Formed

7. Drawback of System

1. Marketing will become expensive.
2. Because of having less no. of customers in a segment problem of limited production occurs.

8. Conclusions

Customer segmentation is performed on the company's customers data and with the help of K-means clustering machine learning algorithm customers are divided using features like total spending and annual income, this study also proves that the dividing customers on the basis of behavioral characteristics is a better solution for existing customer segmentation problem and K-means clustering algorithm is identified as a good choice for this approach.

References

[1] Sukru Ozan, "A Case Study on Customer Segmentation by using Machine Learning Methods", IEEE, Year: 2018.

[2] Jayant Tikmani, Sudhanshu Tiwari, Sujata Khedkar "Telecom customer segmentation based on cluster analysis An Approach to Customer Classification using k-means", IJIRCCE,Year: 2015.

[3] Chinedu Pascal Ezenkwu, Simeon Ozuomba, Constance kalu Electrical/Electronics and Computer Engineering Department, University of Uyo, Akwa Ibom State, Nigeria "Application of K-Means Algorithm for Efficient Customer Segmentation: A Strategy for Targeted Customer Services", IJARAI,Year: 2015.

[4] Potharaju, S. P., Sreedevi, M., Ande, V. K., & Tirandasu, R. K. (2019). Data mining approach for accelerating the classification accuracy of cardiotocography. Clinical Epidemiology and Global Health, 7(2), 160-164.

[5] Yogita Rani and Dr. Harish Rohil "A Study of Hierarchical Clustering Algorithm", IJICT, Year: 2013.

[6] Omar Kettani, Faycal Ramdani, Benaissa Tadili "An Agglomerative Clustering Method for Large Data Sets", IJCA, Year: 2014.

[7] Snekha, Chetna Sachdeva, Rajesh Birok "Real Time Object Tracking Using Different Mean Shift Techniques–a Review", IJSCE, Year: 2013.SulekhaGoyat"The basis of market segmentation: a critical review of literature", EJBM, Year: 2011.

[8] Potharaju, S. P., Sreedevi, M., & Amiripalli, S. S. (2019). An Ensemble Feature Selection Framework of Sonar Targets Using Symmetrical Uncertainty and Multi-Layer Perceptron (SU-MLP). In Cognitive Informatics and Soft Computing (pp. 247-256). Springer, Singapore.

[9] Tanupriya Choudhury, Vivek Kumar, Darshika Nigam, Intelligent Classification and Clustering Of Lung and Oral Cancer through Decision Tree and Genetic Algorithm, International Journal of Advanced Research in Computer Science and Software Engineering,2015

[10] Tanupriya Choudhury, Vivek Kumar, Darshika Nigam, An Innovative and Automatic Lung and Oral Cancer Classification Using Soft Computing Techniques, International Journal of Computer Science and Mobile Computing,2015

[11] H. Mehta, V.S. Dixit and P. Bedi," Refinement of recommendations based on user preferences".

[12] H. Mehta, S.K. Bhatia, V.S. Dixit and P. Bedi," Collaborative personalized web recommender system using entropy-based similarity measure".

[13] Rivedi, A., Rai, P., DuVall, S. L., and Daume III, H. (2010, October). ´Exploiting tag and word correlations for improved webpage clustering in Proceedings of the 2nd international workshop on Search and mining user-generated contents (pp. 3-12). ACM.

[14] Potharaju, S. P., & Sreedevi, M. (2017). A Novel Clustering Based Candidate Feature Selection Framework Using Correlation Coefficient for Improving Classification Performance. Journal of Engineering Science & Technology Review, 10(6).

[15] Domavicius, G., and Tuzhilin, A. (2015). Context-aware recommender systems. In Recommender systems handbook (pp. 191-226). Springer US.

[16] K. Windler, U. Juttner, S. Michel, S. Maklan, and E. K. ¨Macdonald, "Identifying the right solution customers: A managerial methodology," Industrial Marketing Management, vol. 60, pp. 173 –186, 2017.

[17] R. Thakur and L. Workman, "Customer portfolio management (cpm) for improved customer relationship management (crm): Are your customers platinum, gold, silver, or bronze?" Journal of Business Research, vol. 69, no. 10, pp. 4095 – 4102, 2016.

Least squares quantization in PCM

Stuart Lloyd

▶ **To cite this version:**

Stuart Lloyd. Least squares quantization in PCM. IEEE Transactions on Information Theory, 1982, 28 (2), pp.129-137. 10.1109/TIT.1982.1056489 . hal-04614938

HAL Id: hal-04614938
https://hal.science/hal-04614938v1

Submitted on 17 Jun 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Least Squares Quantization in PCM

STUART P. LLOYD

Abstract—**It has long been realized that in pulse-code modulation (PCM), with a given ensemble of signals to handle, the quantum values should be spaced more closely in the voltage regions where the signal amplitude is more likely to fall. It has been shown by Panter and Dite that, in the limit as the number of quanta becomes infinite, the asymptotic fractional density of quanta per unit voltage should vary as the one-third power of the probability density per unit voltage of signal amplitudes. In this paper the corresponding result for any finite number of quanta is derived; that is, necessary conditions are found that the quanta and associated quantization intervals of an optimum finite quantization scheme must satisfy. The optimization criterion used is that the average quantization noise power be a minimum. It is shown that the result obtained here goes over into the Panter and Dite result as the number of quanta become large. The optimum quantization schemes for 2^b quanta, $b = 1, 2, \cdots, 7$, are given numerically for Gaussian and for Laplacian distribution of signal amplitudes.**

I. Introduction

THE BASIC IDEAS in the pulse-code modulation (PCM) system [1], [2, ch. 19] are the Shannon–Nyquist sampling theorem and the notion of quantizing the sample values.

The sampling theorem asserts that a signal voltage $s(t)$, $-\infty < t < \infty$, containing only frequencies less than W cycles/s can be recovered from a sequence of its sample values according to

$$s(t) = \sum_{j=-\infty}^{\infty} s(t_j)K(t - t_j), \qquad -\infty < t < \infty, \quad (1)$$

where $s(t_j)$ is the value of s at the jth sampling instant

$$t_j = \frac{j}{2W}, \qquad -\infty < j < \infty,$$

and where

$$K(t) = \frac{\sin 2\pi Wt}{2\pi Wt}, \qquad -\infty < t < \infty, \quad (2)$$

is a "sin t/t" pulse of the appropriate width.

The pulse-amplitude modulation (PAM) system [2, ch. 16] is based on the sampling theorem alone. One sends over the system channel, instead of the signal values $s(t)$ for all times t, only a sequence

$$\cdots, s(t_{-1}), s(t_0), s(t_1), \cdots \quad (3)$$

of samples of the signal. The (idealized) receiver constructs the pulses $K(t - t_j)$ and adds them together with the received amplitudes $s(t_j)$, as in (1), to produce an exact reproduction of the original band-limited signal s.

PCM is a modification of this. Instead of sending the exact sample values (3), one partitions the voltage range of the signal into a finite number of subsets and transmits to the receiver only the information as to which subset a sample happens to fall in. Built into the receiver there is a source of fixed representative voltages—"quanta"—one for each of the subsets. When the receiver is informed that a certain sample fell in a certain subset, it uses its quantum for that subset as an approximation to the true sample value and constructs a band-limited signal based on these approximate sample values.

We define the *noise signal* as the difference between the receiver-output signal and the original signal and the *noise power* as the average square of the noise signal. The problem we consider is the following: given the number of quanta and certain statistical properties of the signal, determine the subsets and quanta that are best in minimizing the noise power.

II. Quantization

Let us formulate the quantization process more explicitly. A quantization scheme consists of a class of sets $\{Q_1, Q_2, \cdots, Q_\nu\}$ and a set of quanta $\{q_1, q_2, \cdots, q_\nu\}$. The $\{Q_\alpha\}$ are any ν disjoint subsets of the voltage axis which, taken together, cover the entire voltage axis. The $\{q_\alpha\}$ are any ν finite voltage values. The number ν of quanta is to be regarded throughout as a fixed finite preassigned number.

We associate with a partition $\{Q_\alpha\}$ a label function $\gamma(x)$, $-\infty < x < \infty$, defined for all (real) voltages x by

$$\begin{aligned} \gamma(x) &= 1 \quad \text{if} \quad x \quad \text{lies in} \quad Q_1, \\ \gamma(x) &= 2 \quad \text{if} \quad x \quad \text{lies in} \quad Q_2, \\ &\vdots \\ \gamma(x) &= \nu \quad \text{if} \quad x \quad \text{lies in} \quad Q_\nu. \end{aligned} \quad (4)$$

If $s(t_j)$ is the jth sample of the signal s, as in Section I, then we denote by a_j the label of the set that this sample falls in:

$$a_j = \gamma\big(s(t_j)\big), \qquad -\infty < j < \infty.$$

In PCM the signal sent over the channel is (in some code or another) the sequence of labels

$$\cdots, a_{-1}, a_0, a_1, \cdots, \quad (5)$$

each a_j being one of the integers $\{1, 2, \cdots, \nu\}$. The technology of this transmission does not concern us, except that

The author is with Bell Laboratories, Whippany Road, Whippany, NJ 07981.

we assume that such a sequence can be delivered to the receiver without error.

The receiver uses the fixed voltage q_α as an approximation to all sample voltages in Q_α, $\alpha = 1, 2, \cdots, \nu$. That is, the receiver, being given the value of a_j in the sequence (5), proceeds as if the jth sample of s had value q_{a_j} and produces the receiver-output signal

$$r(t) = \sum_{j=-\infty}^{\infty} q_{a_j} K(t - t_j), \qquad -\infty < t < \infty.$$

To put it another way, the system mutilates an actual sample voltage value x to the quantized value $y(x)$ given by

$$y(x) = q_{\gamma(x)}, \qquad -\infty < x < \infty, \tag{6}$$

and we may express the receiver output in terms of this as

$$r(t) = \sum_{j=-\infty}^{\infty} y\big(s(t_j)\big) K(t - t_j), \qquad -\infty < t < \infty. \tag{7}$$

Hence the noise signal, defined as

$$n(t) = r(t) - s(t), \qquad -\infty < t < \infty,$$

is given by

$$n(t) = \sum_{j=-\infty}^{\infty} z\big(s(t_j)\big) K(t - t_j), \qquad -\infty < t < \infty, \tag{8}$$

where

$$z(x) = y(x) - x, \qquad -\infty < x < \infty, \tag{9}$$

may be regarded as the quantization error added to a sample which has voltage value x.

Note that we assume that the receiver uses the nonrealizable pulses (2). If other pulses are used (e.g., step functions or other realizable pulses) there will be sampling noise, in general, even without quantization [3]. Our noise (8) is due strictly to quantization.

Finally we must emphasize that we assume that the $\{Q_\alpha\}$ and $\{q_\alpha\}$ are constant in time. In deltamodulation and its refinements the $\{Q_\alpha\}$ and $\{q_\alpha\}$ change from sampling instant to sampling instant, depending on the past behavior of the signal being handled. Such systems are very difficult to treat theoretically.

III. Noise Power

Instead of working with a particular band-limited signal, we assume that there is given a probabilistic family of such signals. That is, the s of the preceding sections and hence the various signals derived from it are to be regarded as stochastic processes [4]. We denote the underlying probability measure by $P\{\cdot\}$ and averages with respect to this measure (expectations) by $E\{\cdot\}$.

We use the following results of the probabilistic treatment. We assume that the s process is stationary, so that the cumulative probability distribution function of a sample,

$$F(x) = P\{s(t) \le x\}, \qquad -\infty < x < \infty,$$

is independent of t, $-\infty < t < \infty$, as indicated by the notation. Then the average power of the s process, assumed to be finite, is constant in time:

$$S = E\{s^2(t)\} = \int_{-\infty}^{\infty} x^2 \, dF(x), \qquad -\infty < t < \infty. \tag{10}$$

Moreover, the r and n processes have this same property; the average receiver-output power R is given by

$$R = E\{r^2(t)\} = \int_{-\infty}^{\infty} y^2(x) \, dF(x), \qquad -\infty < t < \infty, \tag{11}$$

where $y(x)$ is defined in (6), and the noise power N is

$$N = E\{n^2(t)\} = \int_{-\infty}^{\infty} z^2(x) \, dF(x), \qquad -\infty < t < \infty, \tag{12}$$

with $z(x)$ as in (9). (Detailed proofs of these statements, together with further assumptions used, are given in Appendix A.) The stochastic process problem is thus reduced to a problem in a single real variable: choose the $\{Q_\alpha\}$ and $\{q_\alpha\}$ so that the rightmost integral in (12) is as small as possible.

IV. The Best Quanta

We consider first the problem of minimizing N with respect to the quanta $\{q_\alpha\}$ when the $\{Q_\alpha\}$ are fixed preassigned sets.

The dF integral in (12) may be written more explicitly as

$$N = \sum_{\alpha=1}^{\nu} \int_{Q_\alpha} (q_\alpha - x)^2 \, dF(x). \tag{13}$$

(The sets $\{Q_\alpha\}$ must be measurable $[dF]$ if (11)–(13) are to have meaning, and we assume always that this is the case.) If we regard the given F as describing the distribution of unit probability "mass" on the voltage axis [5, p. 57], then (13) expresses N as the total "moment of inertia" of the sets $\{Q_\alpha\}$ around the respective points $\{q_\alpha\}$. It is a classical result that such a moment assumes its minimum value when each $\{q_\alpha\}$ is the center of mass of the corresponding $\{Q_\alpha\}$ (see, e.g., [5, p. 175]). That is,

$$q_\alpha = \frac{\int_{Q_\alpha} x \, dF(x)}{\int_{Q_\alpha} dF(x)}, \qquad \alpha = 1, 2, \cdots, \nu, \tag{14}$$

are the uniquely determined best quanta to use with a given partition $\{Q_\alpha\}$.

To avoid the continual mention of trivial cases we assume always that F is increasing at least by $\nu + 1$ points, so that the quantization noise does not vanish. Then none of the denominators in (14) will vanish, at least in an

optimum scheme. For if Q_α has vanishing mass it can be combined with some set Q_β of nonvanishing mass (discarding q_α) to give a scheme with $\nu - 1$ quanta and the same noise. Then one of the sets of this scheme can be divided into two sets and new quanta assigned to give a scheme with ν quanta and noise less than in the original scheme. (We omit the details.)

If the expression on the right in (14) is substituted for q_α in (13), there results

$$N = S - \sum_{\alpha=1}^{\nu} q_\alpha^2 \int_{Q_\alpha} dF(x),$$

where the $\{q_\alpha\}$ here are the optimum ones of (14). The sum on the right is the receiver-output power from (11). Hence when the $\{q_\alpha\}$ are centers of mass of the $\{Q_\alpha\}$, optimum or not, then $S = R + N$, which implies that the noise is orthogonal to the receiver output. One expects this in a least squares approximation, of course.

V. The Best Partition

Now we find the best sets $\{Q_\alpha\}$ to use with a fixed preassigned set of quanta $\{q_\alpha\}$. The considerations of this section are independent of those of the preceding section. In particular, the best $\{Q_\alpha\}$ for given $\{q_\alpha\}$ may not have the $\{q_\alpha\}$ as their centers of mass.

We assume that the given $\{q_\alpha\}$ are distinct since it will never happen in an optimum scheme that $q_\alpha = q_\beta$ for some $\alpha \neq \beta$. For if $q_\alpha = q_\beta$, then Q_α and Q_β are effectively one set $Q_\alpha \cup Q_\beta$ as far as the noise is concerned (13), and this set can be redivided into two sets and these two sets can be given distinct quantum values in such a way as to reduce the noise. (We omit the details.)

Consider the probability mass in a small interval around voltage value x. According to (13) any of this mass which is assigned to q_α (i.e., which lies in Q_α) will contribute to the noise at rate $(q_\alpha - x)^2$ per unit mass. To minimize the noise, then, any mass in the neighborhood of x should be assigned to a q_α for which $(q_\alpha - x)^2$ is the smallest of the numbers $(q_1 - x)^2$, $(q_2 - x)^2, \cdots, (q_\nu - x)^2$. In other words,

$$Q_\alpha \supset \left\{x : (q_\alpha - x)^2 < (q_\beta - x)^2 \text{ for all } \beta \neq \alpha\right\}, \quad \alpha = 1,, \cdots, \nu,$$

modulo sets of measure zero $[dF]$.[1] This simplifies to

$$Q_\alpha \supset \left\{x : (q_\beta - q_\beta)\left(x - \tfrac{1}{2}(q_\alpha + q_\beta)\right) < 0 \text{ for all } \beta \neq \alpha\right\}, \quad \alpha = 1, 2, \cdots, \nu. \quad (15)$$

It is straightforward that the best $\{Q_\alpha\}$ are determined by (15) as the intervals whose endpoints bisect the segments between successive $\{q_\alpha\}$, except that the assignment of the endpoints is not determined. To make matters definite we let the $\{Q_\alpha\}$ be left-open and right-closed, so that the best partition to use with the given quanta is

$$\begin{aligned} Q_1 &= \{x : -\infty < x \leq x_1\} \\ Q_2 &= \{x : x < x \leq x_2\} \\ &\vdots \\ Q_{\nu-1} &= \{x : x_{\nu-2} < x \leq x_{\nu-1}\} \\ Q_\nu &= \{x : x_{\nu-1} < x < \infty\}, \end{aligned} \quad (16)$$

where the endpoints $\{x_\alpha\}$ are given

$$\begin{aligned} x_1 &= \tfrac{1}{2}(q_1 + q_2) \\ x_2 &= \tfrac{1}{2}(q_2 + q_3) \\ &\vdots \\ x_{\nu-1} &= \tfrac{1}{2}(q_{\nu-1} + q_\nu). \end{aligned} \quad (17)$$

We have assumed, as we shall hereafter, that the indexing is such that $q_1 < q_2 < \cdots < q_\nu$.

VI. Quantization Procedures

From Sections IV and V we know that we may confine our attention to quantization schemes defined by $2\nu - 1$ numbers

$$q_1 < x_1 < q_2 < x_2 < \cdots < q_{\nu-1} < x_{\nu-1} < q_\nu, \quad (18)$$

where the $\{x_\alpha\}$ are the endpoints of the intervals $\{Q_\alpha\}$, as in (16), and the $\{q_\alpha\}$ are the corresponding quanta. We will regard such a set of numbers as the Cartesian coordinates of a point

$$\rho = (q_1, x_1, \cdots, q_\nu)$$

in $(2\nu - 1)$-dimensional Euclidean space $E_{2\nu-1}$. The noise as a function of ρ has the form

$$N(\rho) = \int_{-\infty}^{x_1} (q_1 - x)^2 \, dF(x) + \int_{x_1}^{x_2} (q_2 - x)^2 \, dF(x) + \cdots + \int_{x_{\nu-1}}^{\infty} (q_\nu - x)^2 \, dF(x). \quad (19)$$

In an optimum scheme the $\{q_\alpha\}$ will be centers of mass of the corresponding $\{Q_\alpha\}$, (14), and the $\{x_\alpha\}$ will lie midway between adjacent $\{q_\alpha\}$, (17). From the derivations these conditions are sufficient that $N(\rho)$ be a minimum with respect to variations in each coordinate separately and hence are necessary conditions at a minimum of $N(\rho)$. As it turns out, however, they are not sufficient conditions for a minimum of $N(\rho)$. Points at which (14) and (17) are satisfied, which we term *stationary points*, while never local maxima, may be saddle points of $N(\rho)$. Moreover, among the stationary points there may be several local minima, only one of which is the sought absolute minimum of $N(\rho)$. These complications are discussed further in Appendix B. The author has not been able to determine sufficient conditions for an absolute minimum.

The derivations suggest one trial-and-error method for finding stationary points. A trial point $\rho^{(1)}$ in $E_{2\nu-1}$ is

[1] If $C(x)$ is a condition on x, then $\{x: C(x)\}$ denotes the set of all x which satisfy $C(x)$.

chosen as follows. The endpoints

$$-\infty < x_1^{(1)} < x_2^{(1)} < \cdots < x_{\nu-1}^{(1)} < \infty$$

are chosen arbitrarily except that each of the resulting $\{Q_\alpha^{(1)}\}$ should have nonvanishing mass. Then the centers of mass of these sets are taken as the first trial quanta $\{q_\alpha^{(1)}\}$.

These values will not satisfy the midpoint conditions (17), in general, so that the second trial point $\rho^{(2)}$ is taken to be

$$q_\alpha^{(2)} = q_\alpha^{(1)}, \qquad \alpha = 1,2,\cdots,\nu$$
$$x_\alpha^{(2)} = \tfrac{1}{2}\left(q_\alpha^{(2)} + q_{\alpha+1}^{(2)}\right), \qquad \alpha = 1,2,\cdots,\nu-1,$$

with appropriate modifications if any of the resulting $\{Q_\alpha^{(2)}\}$ have vanishing mass. This step does not increase the noise, in view of the discussion in Section V; that is, $N(\rho^{(2)}) \leq N(\rho^{(1)})$.

The new $\{q_\alpha^{(2)}\}$, centers of mass (c.m.) of the old $\{Q_\alpha^{(1)}\}$, will not be centers of mass of the new $\{Q_\alpha^{(2)}\}$, in general; trial point $\rho^{(3)}$ is determined by

$$x_\alpha^{(3)} = x_\alpha^{(2)}, \qquad \alpha = 1,2,\cdots,\nu-1,$$
$$q_\alpha^{(3)} = \left(\text{c.m. of } Q_\alpha^{(3)}\right), \qquad \alpha = 1,2,\cdots,\nu.$$

For the resulting noise we have $N(\rho^{(3)}) \leq N(\rho^{(2)})$.

We continue in this way, imposing conditions (14) and (17) alternately. There results a sequence of trial points

$$\rho^{(1)}, \rho^{(2)}, \cdots \tag{20}$$

such that

$$N\left(\rho^{(1)}\right) \geq N\left(\rho^{(2)}\right) \geq \cdots.$$

The noise is nonnegative, so that $\lim_m N(\rho^{(m)})$ will exist, and we might hope that the sequence (20) had as a limit a local minimum of $N(\rho)$.

If the sequence (20) has no limit points then some of the $\{x_\alpha^{(m)}\}$ must become infinite with m; this corresponds to quantizing into fewer than ν quanta. Since we have assumed that F increases at least by $\nu+1$ points there will be quantizing schemes with ν quanta for which the resulting noise is less than the optimum noise for $\nu-1$ quanta, obviously. If $\rho^{(1)}$ is such a scheme then (20) will have limit points, using the property that $N(\rho^{(m)})$ is a decreasing sequence.[2]

Suppose $\rho^{(\infty)}$ is such a limit point. If each of the coordinate values $\{x_\alpha^{(\infty)}\}$ of $\rho^{(\infty)}$ is a continuity point of F then it is easy to see that the coordinates of $\rho^{(\infty)}$ will satisfy both (14) and (17). In particular, if $N(\rho)$ has a unique stationary point ρ_0 (which is the minimum sought), then the sequence (20), unless it diverges, will converge to ρ_0.

Note, by the way, that at a local minimum of $N(\rho)$ the numbers $\{x_\alpha\}$ are necessarily continuity points of F. Suppose to the contrary that there is a nonvanishing amount of mass concentrated at one of the endpoints $\{x_\alpha\}$, and that the adjacent sets Q_α and $Q_{\alpha+1}$ are as in (16), so that the mass at x_α belongs to Q_α. The centers of mass q_α and $q_{\alpha+1}$ will lie equidistant from x_α (17), and from (19) the noise will not change if we reassign the mass at x_α to $Q_{\alpha+1}$, retaining the given $\{q_\alpha\}$ as quanta. But q_α and $q_{\alpha+1}$ are definitely not centers of mass of the corresponding modified sets, and the noise will strictly decrease as q_α and $q_{\alpha+1}$ are moved to the new centers of mass. Thus the given configuration is not a local minimum, contrary to assumption. From this result and (19) we see that $N(\rho)$ is continuous in a neighborhood of a local minimum. We have proved also that there is no essential loss of generality in assuming the form (16) for the $\{Q_\alpha\}$.

We refer to the above trial-and-error method as Method I. Another trial-and-error method is the following one, Method II. To simplify the discussion we assume for the moment that F is continuous and nowhere constant. We choose a trial value q_1 satisfying

$$q_1 < \int_{-\infty}^{\infty} x\, dF(x).$$

The condition that q_1 be the center of mass of Q_1 determines x_1 as the unique solution of

$$q_1 = \frac{\int_{-\infty}^{x_1} x\, dF(x)}{\int_{-\infty}^{x_1} dF(x)}.$$

The quantities q_1 and x_1 now being known, the first of conditions (17) determines q_2 as

$$q_2 = 2x_1 - q_1.$$

If this q_2 lies to the right of the center of mass of the interval (x_1, ∞) then the trial chain terminates, and we start over again with a different trial value q_1. Otherwise, x_1 and q_2 being known, the second of conditions (14):

$$q_2 = \frac{\int_{x_1}^{x_2} x\, dF(x)}{\int_{x_1}^{x_2} dF(x)}$$

serves to determine x_2 uniquely. Now the second of conditions (17) gives

$$q_3 = 2x_2 - q_2.$$

We continue in this way, obtaining successively $q_1, x_1, \cdots, q_{\nu-1}, x_{\nu-1}, q_\nu$; the last step is the determination of q_ν according to

$$q_\nu = 2x_{\nu-1} - q_{\nu-1}. \tag{21}$$

However in this procedure we have not used the last of conditions (14):

$$q_\nu = \frac{\int_{x_{\nu-1}}^{\infty} x\, dF(x)}{\int_{x_{\nu-1}}^{\infty} dF(x)}, \tag{22}$$

and the q_ν obtained from (21) will not satisfy (22) in general. The discrepancy between the right members of (21) and (22) will vary continuously with the starting value q_1, and the method consists of running through such chains

[2] It seems likely that this condition $N(\rho^{(1)}) \leq$ (optimum noise for $\nu-1$ quanta) is stronger than necessary for the nondivergence of (20).

using various starting values until the discrepancy is reduced to zero.

This method is applicable to more general F, with some obvious modifications. When F has intervals of constancy the $\{x_\alpha\}$ may not be uniquely determined by conditions (14), and a trial chain may involve several arbitrary parameters besides q_1. Discontinuities of F will cause no real trouble, since we know that the $\{x_\alpha\}$ of an optimum scheme are continuity points of F; a trial chain that does not have this property is discarded. We note that Method II may be used to locate all stationary points of $N(\rho)$.

VII. Examples

In all of the examples we now consider, the distribution of sample values is absolutely continuous with a sample probability density $f = F'$, which is an even function. If $N(\rho)$ has a unique stationary point, which we assume to be the case in the examples treated, then the optimum $\{q_\alpha\}$ and $\{x_\alpha\}$ will clearly be symmetrically distributed around the origin. In applications we are usually interested in having an even number of quanta, $\nu = 2\mu$, so we renumber the positive endpoints and quanta according to

$$0 = x_0 < q_1 < x_1 < \cdots < q_{\mu-1} < x_{\mu-1} < q_\mu; \quad (23)$$

the endpoints and quanta for the negative half-axis are the negatives of these.

We normalize to unit signal power $S = 1$. The $\{q_\alpha\}$ and $\{x_\alpha\}$ for other values of S are to be obtained by multiplying the numbers in the tables by $\sqrt{S}$.

The simplest case is the uniform distribution:

$$f(x) = \frac{1}{2\sqrt{3}}, \qquad -\sqrt{3} \le x \le \sqrt{3}$$

$$= 0, \qquad \sqrt{3} < |x| < \infty.$$

Method II of the preceding section shows that $N(\rho)$ in this case has a unique stationary point, which is necessarily an absolute minimum. The optimum scheme is the usual one with ν equal intervals of width $1/(2\nu\sqrt{3})$ each; the quanta being the midpoints of these intervals. The minimum value of the noise is the familiar $N = 1/\nu^2$.

Another case of possible interest is the Gaussian:

$$f(x) = \frac{e^{-1/2x^2}}{\sqrt{2\pi}}, \qquad -\infty < x < \infty.$$

The optimum schemes for $\nu = 2^b$, $b = 1, 2, \cdots, 7$, are given in Tables I–VII,[3] respectively. The corresponding noise values appear in Table VIII together with the quantities $\nu^2 N$ and νx_1. The behavior of these latter with increasing ν hint at the existence of asymptotic properties; we examine this question in the next section.

[3]Since some of the tables were never completed, those tables although mentioned in text are not included in this paper.

TABLE I
GAUSSIAN, $\nu = 2$

α	q_α	x_α
1	0.7979	∞

TABLE II
GAUSSIAN, $\nu = 4$

α	q_α	x_α
1	0.4528	0.9816
2	1.5104	∞

TABLE III
GAUSSIAN, $\nu = 8$

α	q_α	x_α
1	0.2451	0.5006
2	0.7560	1.0500
3	1.3439	1.7480
4	2.1520	∞

TABLE IV
GAUSSIAN, $\nu = 16$

α	q_α	x_α
1	0.1284	0.2582
2	0.3880	0.5224
3	0.6568	0.7996
4	0.9423	1.0993
5	1.2562	1.4371
6	1.6181	1.8435
7	2.0690	2.4008
8	2.7326	∞

TABLE VIII
GAUSSIAN; OPTIMUM NOISE FOR VARIOUS VALUES OF ν

ν	N	$\nu^2 N$	νx_1
2	0.3634	1.452	
4	0.1175	1.880	3.93
8	3.455×10^{-2}	2.205	4.00
16	9.500×10^{-3}	2.430	4.13
32			
64			
128			
(∞)	(0)	(2.72)	(4.34)

For speech signals a distribution which has been found useful empirically is the Laplacian:[4]

$$f(x) = \frac{e^{-|x|\sqrt{2}}}{\sqrt{2}}, \qquad -\infty < x < \infty.$$

The optimum quantizing schemes for this distribution for $\nu = 2^b$, $b = 1, 2, \cdots, 7$, are given in Tables IX–XV, respectively. The corresponding N, $\nu^2 N$, and νx_1 values are given in Table XVI; again, we notice certain regularities.

VIII. Asymptotic Properties

Let us assume that the distribution F is absolutely continuous with density function $f = F'$, which is itself dif-

[4]The author is indebted to V. Vyssotsky of the Acoustics Research Group for this information (private communication).

ferentiable, and that for each ν there is a unique optimum quantization scheme. We revert to our original numbering (18).

Let the quantities $\{h_\alpha\}$ be defined by

$$h_\alpha = x_\alpha - q_\alpha = q_{\alpha+1} - x_\alpha, \qquad \alpha = 1,2,\cdots,\nu-1,$$

so that, for $\alpha = 2,3,\cdots,\nu-1$, Q_α consists of an interval of length h_α to the right of q_α together with an interval of length $h_{\alpha-1}$ to the left of q_α. We have already imposed the optimizing conditions (17) in the very definition of the $\{h_\alpha\}$. The center of mass conditions (14) (except for the first and last) may be written as

$$\int_{q_\alpha - h_{\alpha-1}}^{q_\alpha + h_\alpha} (x - q_\alpha) f(x)\, dx = 0, \qquad \alpha = 2,3,\cdots,\nu-1.$$

If we expand f here in Taylor's series around q_α, the integration gives

$$\tfrac{1}{2}\left(h_\alpha^2 - h_{\alpha-1}^2\right) f(q_\alpha) + \tfrac{1}{3}\left(h_\alpha^3 + h_{\alpha-1}^3\right) f'(q_\alpha) = o\left(h_\alpha^3\right) + o\left(h_{\alpha-1}^3\right), \qquad \alpha = 2,3,\cdots,\nu-1. \quad (24)$$

The numbers in Tables VIII and XVI suggest the existence of an asymptotic fractional density of quanta. Accordingly, we define the function $g_\nu(x)$, $-\infty < x < \infty$, by

$$\begin{aligned} g_\nu(x) &= 0, && -\infty < x \le q_1 \\ &= \frac{1}{2\nu h_\alpha}, && q_\alpha < x \le q_{\alpha+1}, \\ & && \alpha = 1,2,\cdots,\nu-1, \\ &= 0, && q_\nu < x < \infty. \end{aligned} \quad (25)$$

The definition is arranged so that (for given ν) the sets $Q_2, Q_3, \cdots, Q_{\nu-1}$ subtend equal areas of $1/\nu$ each under the graph of $g_\nu(x)$ versus x. We will proceed as if a limiting density,

$$g(x) = \lim_{\nu\to\infty} g_\nu(x), \qquad -\infty < x < \infty,$$

existed.

We wish to express g in terms of the given sample density function f. To do this we will use conditions (24), together with the following further assumptions. We assume that g has a derivative, and we assume that for given x and k the difference $\epsilon_\nu(x) = g_\nu(x) - g(x)$, $-\infty < x < \infty$, has the property[5]

$$\epsilon_\nu\left(x + \frac{k}{\nu}\right) - \epsilon_\nu(x) = o\left(\frac{1}{\nu}\right).$$

In (24), then, we may approximate $h_\alpha - h_{\alpha-1}$ by

$$\begin{aligned} h_\alpha - h_{\alpha-1} &= \frac{1}{2\nu}\left[\frac{1}{g_\nu(q_\alpha + h_\alpha)} - \frac{1}{g_\nu(q_\alpha - h_{\alpha-1})}\right] \\ &= -\frac{g'(q_\alpha)}{2\nu^2 g^3(q_\alpha)} + o\left(\frac{1}{\nu^2}\right), \qquad \alpha = 2,3,\cdots,\nu-1, \end{aligned}$$

and we find that the left-hand member of (24) is indeed $o(h^3) = o(1/\nu^3)$ provided that

$$\frac{g'(x)}{g(x)} = \frac{f'(x)}{3f(x)}, \qquad -\infty < x < \infty. \quad (26)$$

The normalized solution of (26) is

$$g(x) = \frac{f^{1/3}(x)}{\int_{-\infty}^{\infty} f^{1/3}(x')\, dx'}, \qquad -\infty < x < \infty, \quad (27)$$

provided that the integral in the denominator exists.

The noise power becomes

$$\begin{aligned} N &= \frac{1}{12\nu^2}\int_{-\infty}^{\infty} \frac{f(x)}{g^2(x)}\, dx + o\left(\frac{1}{\nu^2}\right) \\ &= \frac{1}{12\nu^2}\left[\int_{-\infty}^{\infty} f^{1/3}(x)\, dx\right]^3 + o\left(\frac{1}{\nu^2}\right), \end{aligned} \quad (28)$$

neglecting the contributions from the end quanta.[6] For the Gaussian example then, the numbers $\nu^2 N$ of Table VIII should have a limit easily evaluated from (28) as $\nu^2 N \to \pi\sqrt{3}/2 (\approx 2.72)$, and in the Laplacian case, Table XVI, we find $\nu^2 N \to 9/2$.

The quantities denoted by νx_1 in Tables VIII and XVI should have the limiting value

$$\lim_{\substack{\nu\to\infty \\ q_\alpha \to 0}} \nu(h_{\alpha-1} + h_\alpha) = \frac{1}{g(0)},$$

comparing with (25). In the Gaussian example we find $1/g(0) = \sqrt{6\pi}\,(\approx 4.34)$, and for the Laplacian: $1/g(0) = 3\sqrt{2}\,(\approx 4.24)$.

For large values of ν the sets $\{Q_\alpha\}$ should subtend approximately equal areas of $1/\nu$ each under the graph of $g(x)$ versus x, so that the number b_ν defined by

$$\frac{1}{\nu} = \int_{b_\nu}^{\infty} g(x)\, dx$$

might be expected to be near the rightmost division point. Comparing Table XVII with Tables I–VII and IX–XVI we see that the approximation is surprisingly good, at least in the examples considered.

TABLE XVII
APPROXIMATE LAST ENDPOINT FROM ASYMPTOTIC FORMULA

ν	Gaussian b_ν	Laplacian b_ν
2	0	
4	1.168	
8	1.992	
16	2.657	
32		
64		
128		

[5] The notation $u(\nu) = o(v(\nu))$ means in our case $\lim_{\nu\to\infty} u(\nu)/v(\nu) = 0$.

[6] Other derivations of (27)–(28) are given in [6] and [7].

Acknowledgment

The numerical results presented in the tables are due to Miss M. C. Gray and her assistants in the Numerical Analysis and Digital Processes Group; the programming of Method I for the IBM-650 electronic computer was done by Miss C. A. Conn.

After substantial progress had been made on the work described here there appeared in [11] a review of a paper by J. Lukaszewicz and H. Steinhaus on optimum go/no-go gauge sets. The present author has not been able to obtain a copy of this paper, but it seems likely that these authors have treated a problem similar or identical to the one discussed in Sections IV–VI. M. P. Schützenberger in [12] examines the quantization problem in the case where $\nu = 2$ and where F increases at 3 or 4 points.

Appendix A

Suppose $s(t)$, $-\infty < t < \infty$, is a continuous parameter stochastic process, real, separable, measurable, stationary, and of finite power:

$$S = E\{s^2(t)\} = \int_{-\infty}^{\infty} x^2\, dF(x) < \infty, \qquad -\infty < t < \infty,$$

(where F is the first-order distribution of the process, Section III). Then s has a spectral representation

$$s(t) = \int_{-\infty}^{\infty} e^{2\pi i\lambda t} d\xi(\lambda), \qquad -\infty < t < \infty, \tag{29}$$

where the spectral process $\xi(\lambda)$, $-\infty < \lambda < \infty$, has orthogonal increments [4, p. 527]. To say that s is band-limited to the frequency band $-W \le \lambda \le W$ is to say that the ξ process has vanishing increments outside of this band with probability one, and (29) becomes

$$s(t) = \int_{-W-0}^{W+0} e^{2\pi i\lambda t} d\xi(\lambda), \qquad -\infty < t < \infty. \tag{30}$$

Since we are particularly concerned with the behavior of ξ at the band edges, we rewrite (30) as

$$s(t) = \int_{-W+0}^{W-0} e^{2\pi i\lambda t} d\xi(\lambda) + 2\delta_1 \cos 2\pi Wt - 2\delta_2 \sin 2\pi Wt, \qquad -\infty < t < \infty,$$

where the real random variables δ_1 and δ_2 describe the jumps of ξ at the band edges:

$$\xi(\pm W + 0) - \xi(\pm W - 0) = \delta_1 \pm i\delta_2.$$

For fixed t, the function $e^{2\pi i\lambda t}$, $-W \le \lambda \le W$, has Fourier coefficients

$$\begin{aligned} c_j &= \frac{1}{2W}\int_{-W}^{W} e^{-2\pi ij\lambda/(2W)} e^{2\pi i\lambda t} d\lambda \\ &= \frac{\sin 2\pi W(t - j/(2W))}{2\pi W(t - j/(2W))} \\ &= K(t - t_j), \qquad -\infty < j < \infty, \end{aligned}$$

in the notation of Section I. This function is of bounded variation, so that the partial sums

$$S_l(\lambda) = \sum_{j=-l}^{l} e^{2\pi ij\lambda/(2W)} K(t - t_j), \qquad -W \le \lambda \le W,$$

converge boundedly to

$$\begin{aligned} S(\lambda) &= \lim_{l\to\infty} S_l(\lambda) \\ &= e^{2\pi i\lambda t}, \qquad -W < \lambda < W, \\ &= \cos^{2\pi Wt}, \qquad \lambda = \pm W, \end{aligned}$$

from [8]. Hence, using the representation (30) for the samples, the sampling series

$$\hat{s}(t) = \sum_{j=-\infty}^{\infty} s(t_j) K(t - t_j) \tag{31}$$

converges (in stochastic mean square) to

$$\begin{aligned} \hat{s}(t) &= \text{l.i.m.}_{l\to\infty} \int_{-W-0}^{W+0} S_l(\lambda)\, d\xi(\lambda) \\ &= \int_{-W+0}^{W-0} e^{2\pi i\lambda t} d\xi(\lambda) + 2\delta_1 \cos 2\pi Wt \\ &= s(t) + 2\delta_2 \sin 2\pi Wt, \qquad -\infty < t < \infty, \end{aligned} \tag{32}$$

from [4, p. 429]. (The corresponding result for deterministic functions is given in [9].) Since the orthogonal increments property of ξ requires $E\{\delta_1\delta_2\} = 0$ together with

$$E = \{\delta_1^2\} = E\{\delta_2^2\} = \tfrac{1}{2}E\{|\xi(\pm W + 0) - \xi(\pm W - 0)|^2\},$$

we see from (32) that the sampling series (31) represents s with probability 1, if and only if, the ξ process has no power concentrated at the band edges,

$$E\{|\xi(\pm W + 0) - \xi(\pm W - 0)|^2\} = 0.$$

(Other proofs of this result appear in [3] and in [10].)

Let s be as above and suppose $\varphi(x)$, $-\infty < x < \infty$, is a Baire function. Then the random variables

$$\cdots, \varphi(s(t_{-1})), \varphi(s(t_0)), \varphi(s(t_1)), \cdots \tag{33}$$

constitute a stationary discrete-parameter stochastic process. If the number

$$\Phi = E\{\varphi^2(s(t_j))\} = \int_{-\infty}^{\infty} \varphi^2(x)\, dF(x), \qquad -\infty < j < \infty,$$

is finite then the process (33) admits a spectral representation

$$\varphi(s(t_j)) = \int_{-W}^{W} e^{2\pi ij\lambda/(2W)} d\eta(\lambda), \qquad -\infty < j < \infty,$$

where the η process has orthogonal increments ([4, p. 481], with a change of scale).

A certain continuous parameter stochastic process $\theta(t)$, $-\infty < t < \infty$, may be defined in terms of the η process by

$$\theta(t) = \int_{-W}^{W} e^{2\pi i\lambda t} d\eta(\lambda), \qquad -\infty < t < \infty.$$

This process is stationary in the wide sense and has the given process (33) as its samples, clearly

$$\theta(t_j) = \varphi(s(t_j)), \qquad -\infty < j < \infty.$$

Moreover,

$$E\{\theta^2(t)\} = \int_{-W}^{W} E\{|d\eta(\lambda)|^2\} = \int_{-\infty}^{\infty} \varphi^2(x) dF(x), \qquad -\infty < t < \infty.$$

The θ process is represented by the sampling series

$$\theta(t) = \sum_{j=-\infty}^{\infty} \varphi(s(t_j)) K(t - t_j), \quad -\infty < t < \infty, \tag{34}$$

if and only if, the spectral process η has no power concentrated at

the band edges. The arguments are identical to those given above for the s process itself.

The r and n processes of Sections II and III are of the form just described, since the functions $y(x)$, (6), and $z(x)$, (9), will differ from certain Baire functions only on sets of measure zero $[dF]$ when we assume, as we do, that the sets $\{Q_\alpha\}$ are measureable $[dF]$.

The well-known mean-erogodic property,

$$\text{l.i.m.}_{m''-m'\to\infty} \sum_{j=m'}^{m''} \frac{(-1)^j \varphi(s(t_j))}{m''-m'+1}$$
$$= \eta(W+0) - \eta(W-0) + \eta(-W+0) - \eta(-W-0)$$
$$= 2\,\mathrm{Re}[\eta(\pm W+0) - \eta(\pm W-0)],$$

([4, p. 491]) shows that the requirement that η have no power at the band edges is equivalent to the condition

$$\lim_{m''-m'\to\infty} E\left\{\left|\sum_{j=m'}^{m''} \frac{(-1)^j \varphi(s(t_j))}{m''-m'+1}\right|\right\}^2 = 0.$$

Finally we note that if the ξ process has a discrete component at frequencies $\pm\lambda_0$, then depending on the form of φ, the derived η process is likely to have discrete components at all of the harmonic frequencies $\pm m\lambda_0$ (modulo $2W$) of λ_0, $m = 1, 2, \cdots$. In particular, if λ_0 is rational then the η process may have a discrete component at the band edges, a possibility which must be excluded if (34) is to hold.

Appendix B

A simple example shows that the conditions (14) and (17) are not sufficient for an absolute minimum of N. Suppose F is absolutely continuous, with a density $f = F'$ as shown in Fig. 1, where $c_1(b_2 - b_1) + c_2(b_4 - b_3) = 1$. If $\nu > 1$ quanta are desired, let $\nu_1, \nu_2 > 0$ be any integers such that $\nu_1 + \nu_2 = \nu$, and divide the interval (b_1, b_2) into ν_1 equal intervals and (b_3, b_4) into ν_2 equal intervals; let the quanta be the midpoints of these ν intervals. If we suppose that $b_2 < \frac{1}{2}(b_1 + b_3)$ and $b_3 > \frac{1}{2}(b_2 + b_4)$ then the division point which separates the right-hand $\{Q_\alpha\}$ in (b_1, b_2) and the left-hand $\{Q_\alpha\}$ in (b_3, b_4) will lie in the interval (b_2, b_3), so that the conditions (14) and (17) will be satisfied. Thus we have $\nu - 1$ distinct local minima of N. (If c_1, respectively c_2, is small enough there may even be another minimum, corresponding to $\nu_1 = 0$, respectively $\nu_2 = 0$.) Which of these is the true minimum depends on the values of the parameters. (Explicitly, the noise has the value

$$N = \frac{c_1(b_2 - b_1)^3}{12\nu_1^2} + \frac{c_2(b_4 - b_3)^3}{12\nu_2^2},$$

and the ν_2/ν_1 for which this is a minimum is given by

$$\frac{\nu_2/(b_4 - b_3)}{\nu_1/(b_2 - b_1)} = \left(\frac{c_2}{c_1}\right)^{1/3},$$

agreeing with (27).)

The following interesting example is due to J. L. Kelly, Jr., of the Visual Research Group. Let the density f be as in Fig. 2, with $c_2 > c_1$, $c_1 + c_2 = 1$. The signal power is $S = 1/3$, independently of c_1 and c_2. Suppose $\nu = 2$. One configuration for which conditions (14) and (17) are satisfied is $q_1 = -\frac{1}{2}$, $x_1 = 0$, $q_2 = \frac{1}{2}$, clearly, and the resulting noise in $N = 1/12$. When $c_2 > 3c_1$; however, there is another solution—it is the one which in the limit $c_1 = 0$, $c_2 = 1$ goes into the scheme where $(0, 1)$ is divided

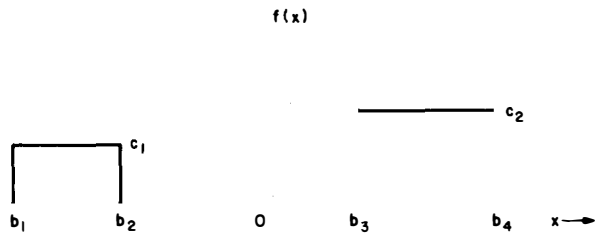


Fig. 1. The density $f(x)$ vanishes outside of the intervals (b_1, b_2), (b_3, b_4).

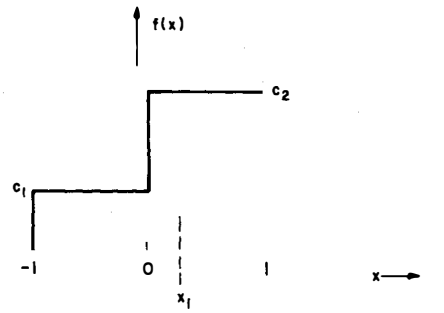


Fig. 2. The density $f(x)$ vanishes outside of the interval $(-1, 1)$.

into two equal parts and $(-1, 0)$ is ignored. The parameters for this configuration work out to be

$$q_1 = \frac{c_2 - 9c_1}{4c_2},$$
$$x_1 = \frac{c_2 - 3c_1}{2c_2},$$
$$q_2 = \frac{3(c_2 - c_1)}{4c_2},$$
$$N = \frac{1}{12} - \frac{(c_2 - 3c_1)^3}{16c_2^2}.$$

Hence if this configuration exists then it is better than the one first mentioned. We note, by the way, that method I of Section VI will converge to the $(-1/2, 0, 1/2)$ configuration, if the starting value $x_1^{(1)}$ is negative; if $c_2 > 3c_1$, however, this configuration is only a saddle point of N.

Author's Note 1981:

This is nearly a verbatim reproduction of a draft manuscript, which was circulated for comments at Bell Laboratories; the Mathematical Research Department log date is July 31, 1957. I wish to thank the editors for their invitation to publish this antique *samizdat* in the present issue.

The main reason the paper was not submitted for publication previously was that the numerical calculations were never completed. The Gaussian $\nu = 32$ case was done on the IBM 650 card programmable calculator; the Laplacian cases were done only for $\nu = 2$. Some time later the 650 was replaced by an IBM 701 electronic computer, but no quantizing program was written for it.

I was not satisfied with not having conditions for a unique minimum but would have published the paper

without this. Later, P. E. Fleischer of Bell Laboratories gave a neat sufficient condition in his paper [13].

In the examples of Appendix B, the direct current can be removed by changing the origin; the noise is not affected. The results of the paper are valid for the uncentered processes used.

I was aware when I wrote the paper that the methods for quantizing a real random variable extend to other loss functions. In the least squares case the process quantizing noise is just the noise per sample; the generalization is more complicated and was omitted.

References

[1] B. M. Oliver, J. R. Pierce, and C. E. Shannon, "The philosophy of PMC," *Proc. I.R.E.*, vol. 36, pp. 1324–1331, 1948.

[2] H. S. Black, *Modulation Theory*. Princeton, NJ: Van Nostrand, 1953.

[3] S. P. Lloyd and B. McMillan, "Linear least squares filtering and prediction of sampled signals," in *Proc. Symp. on Modern Network Synthesis*, vol. 5. Brooklyn, NY: Polytechnic Institute of Brooklyn, 1956, pp. 221–247.

[4] J. L. Doob, *Stochastic Processes*. New York: Wiley, 1953.

[5] H. Cramér, *Mathematical Methods of Statistics*. Princeton, NJ: Princeton University, 1951.

[6] P. F. Panter and W. Dite, "Quantization distortion in pulse-count modulation with nonuniform spacing of levels," *Proc. I.R.E.*, vol. 39, pp. 44–48, 1951.

[7] B. Smith, "Instantaneous companding of quantized signals," *Bell Syst. Tech. J.*, vol. 36, pp. 653–709, 1957.

[8] A. Zygmund, *Trigonometrical Series*. New York: Dover, 1955, p. 47.

[9] H. P. Kramer, "A generalized sampling theorem," *Bull. Am. Math. Soc.* vol. 63, p. 117, 1957.

[10] E. Parzen, "A simple proof and some extensions of the sampling theorem," Department of Statistics, Stanford Univ., CA, Tech. Report 7, Dec. 1956.

[11] J. Kukaszewicz and H. Steinhaus, "On measuring by comparison," *Zastos. Mat.*, vol. 2, pp. 225–231, 1955; *Math. Reviews*, vol. 17, p. 757, 1956.

[12] M. P. Schützenberger, "Contribution aux applications statistiques de la théorie de l'information," *Pub. de l'Inst. de Statis.* de l'Université de Paris, vol. 3, Fasc. 1-2 pp. 56–69, 1954.

[13] P. E. Fleischer, "Sufficient conditions for achieving minimum distortion in quantizer," *IEEE Int. Convention Record*, part I, vol. 12, pp. 104–111, 1964.

SOME METHODS FOR CLASSIFICATION AND ANALYSIS OF MULTIVARIATE OBSERVATIONS

J. MacQUEEN
UNIVERSITY OF CALIFORNIA, LOS ANGELES

1. Introduction

The main purpose of this paper is to describe a process for partitioning an N-dimensional population into k sets on the basis of a sample. The process, which is called 'k-means,' appears to give partitions which are reasonably efficient in the sense of within-class variance. That is, if p is the probability mass function for the population, $S = \{S_1, S_2, \cdots, S_k\}$ is a partition of E_N, and u_i, $i = 1, 2, \cdots, k$, is the conditional mean of p over the set S_i, then $w^2(S) = \sum_{i=1}^{k} \int_{S_i} |z - u_i|^2 \, dp(z)$ tends to be low for the partitions S generated by the method. We say 'tends to be low,' primarily because of intuitive considerations, corroborated to some extent by mathematical analysis and practical computational experience. Also, the k-means procedure is easily programmed and is computationally economical, so that it is feasible to process very large samples on a digital computer. Possible applications include methods for similarity grouping, nonlinear prediction, approximating multivariate distributions, and nonparametric tests for independence among several variables.

In addition to suggesting practical classification methods, the study of k-means has proved to be theoretically interesting. The k-means concept represents a generalization of the ordinary sample mean, and one is naturally led to study the pertinent asymptotic behavior, the object being to establish some sort of law of large numbers for the k-means. This problem is sufficiently interesting, in fact, for us to devote a good portion of this paper to it. The k-means are defined in section 2.1, and the main results which have been obtained on the asymptotic behavior are given there. The rest of section 2 is devoted to the proofs of these results. Section 3 describes several specific possible applications, and reports some preliminary results from computer experiments conducted to explore the possibilities inherent in the k-means idea. The extension to general metric spaces is indicated briefly in section 4.

The original point of departure for the work described here was a series of problems in optimal classification (MacQueen [9]) which represented special

This work was supported by the Western Management Science Institute under a grant from the Ford Foundation, and by the Office of Naval Research under Contract No. 233(75), Task No. 047-041.

cases of the problem of optimal information structures as formulated by Marschak [11], [12]. (For an interesting treatment of a closely related problem, see Blackwell [1].) In one instance the problem of finding optimal information structures reduces to finding a partition $S = \{S_1, S_2, \cdots, S_k\}$ of E_N which will minimize $w^2(S)$ as defined above. In this special model, individual A observes a random point $z \in E_N$, which has a known distribution p, and communicates to individual B what he has seen by transmitting one of k messages. Individual B interprets the message by acting as if the observed point z is equal to a certain point $\hat{z}$ to be chosen according to the message received. There is a loss proportional to the squared error $|z - \hat{z}|^2$ resulting from this choice. The object is to minimize expected loss. The expected loss becomes $w^2(S)$, where the i-th message is transmitted if $z \in S_i$, since the best way for B to interpret the information is to choose the conditional mean of p on the set associated with the message received. The mean, of course, minimizes the squared error. Thus the problem is to locate a partition minimizing $w^2(S)$. This problem was also studied by Fisher [5], who gives references to earlier related works.

The k-means process was originally devised in an attempt to find a feasible method of computing such an optimal partition. In general, the k-means procedure will not converge to an optimal partition, although there are special cases where it will. Examples of both situations are given in section 2.3. So far as the author knows, there is no feasible, general method which always yields an optimal partition. Cox [2] has solved the problem explicitly for the normal distribution in one dimension, with $k = 2, 3, \cdots, 6$, and a computational method for finite samples in one dimension has been proposed by Fisher [5]. A closely related method for obtaining reasonably efficient 'similarity groups' has been described by Ward [15]. Also, a simple and elegant method which would appear to yield partitions with low within-class variance, was noticed by Edward Forgy [7] and Robert Jennrich, independently of one another, and communicated to the writer sometime in 1963. This procedure does not appear to be known to workers in taxonomy and grouping, and is therefore described in section 3. For a thorough consideration of the biological taxonomy problem and a discussion of a variety of related classification methods, the reader is referred to the interesting book by Sokal and Sneath [14]. (See *Note added in proof* of this paper.)

Sebestyen [13] has described a procedure called "adaptive sample set construction," which involves the use of what amounts to the k-means process. This is the earliest explicit use of the process with which the author is familiar. Although arrived at in ignorance of Sebestyen's work, the suggestions we make in sections 3.1, 3.2, and 3.3, are anticipated in Sebestyen's monograph.

2. K-means; asymptotic behavior

2.1. *Preliminaries.* Let $z_1, z_2, \cdots$ be a random sequence of points (vectors) in E_N, each point being selected independently of the preceding ones using a fixed probability measure p. Thus $P[z_1 \in A] = p(A)$ and $P[z_{n+1} \in A | z_1, z_2, \cdots, z_n] =$

$p(A)$, $n = 1, 2, \cdots$, for A any measurable set in E_N. Relative to a given k-tuple $x = (x_1, x_2, \cdots, x_k)$, $x_i \in E_N$, $i = 1, 2, \cdots, k$, we define a *minimum distance partition* $S(x) = \{S_1(x), S_2(x), \cdots, S_k(x)\}$ of E_N, by

$$S_1(x) = T_1(x), S_2(x) = T_2(x)S_1'(x), \cdots,$$
$$S_k(x) = T_k(x)S_1'(x)S_2'(x) \cdots S_{k-1}'(x), \tag{2.1}$$

where

$$T_i(x) = \{\xi: \xi \in E_N, |\xi - x_i| \leq |\xi - x_j|, j = 1, 2, \cdots, k\}. \tag{2.2}$$

The set $S_i(x)$ contains the points in E_N nearest to x_i, with tied points being assigned arbitrarily to the set of lower index. Note that with this convention concerning tied points, if $x_i = x_j$ and $i < j$ then $S_j(x) = \varnothing$. Sample k-means $x^n = (x_1^n, x_2^n, \cdots, x_k^n)$, $x_i^n \in E_N$, $i = 1, \cdots, k$, with associated integer weights $(w_1^n, w_2^n, \cdots, w_k^n)$, are now defined as follows: $x_i^1 = z_i$, $w_i^1 = 1$, $i = 1, 2, \cdots, k$, and for $n = 1, 2, \cdots$, if $z_{k+n} \in S_i^n$, $x_i^{n+1} = (x_i^n w_i^n + z_{n+k})/(w_i^n + 1)$, $w_i^{n+1} = w_i^n + 1$, and $x_j^{n+1} = x_j^n$, $w_j^{n+1} = w_j^n$ for $j \neq i$, where $S^n = \{S_1^n, S_2^n, \cdots, S_k^n\}$ is the minimum distance partition relative to x^n.

Stated informally, the k-means procedure consists of simply starting with k groups each of which consists of a single random point, and thereafter adding each new point to the group whose mean the new point is nearest. After a point is added to a group, the mean of that group is adjusted in order to take account of the new point. Thus at each stage the k-means are, in fact, the means of the groups they represent (hence the term k-means).

In studying the asymptotic behavior of the k-means, we make the convenient assumptions, (i) p is absolutely continuous with respect to Lebesgue measure on E_N, and (ii) $p(R) = 1$ for a closed and bounded convex set $R \subset E_N$, and $p(A) > 0$ for every open set $A \subset R$. For a given k-tuple $x = (x_1, x_2, \cdots, x_k)$—such an entity being referred to hereafter as a k-point—let

$$W(x) = \sum_{i=1}^{k} \int_{S_i} |z - x_i|^2 \, dp(z),$$
$$V(x) = \sum_{i=1}^{k} \int_{S_i} |z - u_i(x)|^2 \, dp(z), \tag{2.3}$$

where $S = \{S_1, S_2, \cdots, S_k\}$ is the minimum distance partition relative to x, and $u_i(x) = \int_{S_i} z \, dp(z)/p(S_i)$ or $u_i(x) = x_i$, according to whether $p(S_i) > 0$ or $p(S_i) = 0$. If $x_i = u_i(x)$, $i = 1, 2, \cdots, k$ we say the k-point x is *unbiased*.

The principal result is as follows.

THEOREM 1. *The sequence of random variables $W(x^1)$, $W(x^2)$, $\cdots$ converges a.s. and $W_\infty = \lim_{n\to\infty} W(x^n)$ is a.s. equal to $V(x)$ for some x in the class of k-points $x = (x_1, x_2, \cdots, x_k)$ which are unbiased, and have the property that $x_i \neq x_j$ if $i \neq j$.*

In lieu of a satisfactory strong law of large numbers for k-means, we obtain the following theorem.

THEOREM 2. *Let $u_i^n = u_i(x^n)$ and $p_i^n = p(S_i(x^n))$; then*

$$\sum_{n=1}^{m} \left(\sum_{i=1}^{k} p_i^n |x_i^n - u_i^n| \right) \Big/ m \underset{\text{a.s.}}{\to} 0 \quad \text{as} \quad m \to \infty. \tag{2.4}$$

2.2. *Proofs.* The system of k-points forms a complete metric space if the distance $\rho(x, y)$ between the k-points $x = (x_1, x_2, \cdots, x_k)$ and $y = (y_1, y_2, \cdots, y_k)$, is defined by $\rho(x, y) = \sum_{i=1}^{k} |x_i - y_i|$. We designate this space by M and interpret continuity, limits, convergence, neighborhoods, and so on, in the usual way with respect to the metric topology of M. Of course, every bounded sequence of k-points contains a convergent subsequence.

Certain difficulties encountered in the proof of theorem 1 are caused by the possibility of the limit of a convergent sequence of k-points having some of its constituent points equal to each other. With the end in view of circumventing these difficulties, suppose that for a given k-point $x = (x_1, x_2, \cdots, x_k)$, $x_i \in R$, $i = 1, 2, \cdots, k$, we have $x_i = x_j$ for a certain pair i, j, $i < j$, and $x_i = x_j \neq x_m$ for $m \neq i, j$. The points x_i and x_j being distinct in this way, and considering assumption (ii), we necessarily have $p(S_i(x)) > 0$, for $S_i(x)$ certainly contains an open subset of R. The convention concerning tied points means $p(S_j(x)) = 0$. Now if $\{y^n\} = \{(y_1^n, y_2^n, \cdots, y_k^n)\}$ is a sequence of k-points satisfying $y_i^n \in R$, and $y_i^n \neq y_j^n$ if $i \neq j$, $n = 1, 2, \cdots$, and the sequence y^n approached x, then y_i^n and y_j^n approach $x_i = x_j$, and hence each other; they also approach the boundaries of $S_i(y^n)$ and $S_j(y^n)$ in the vicinity of x_i. The conditional means $u_i(y^n)$ and $u_j(y^n)$, however, must remain in the interior of the sets $S_i(y^n)$ and $S_j(y^n)$ respectively, and thus tend to become separated from the corresponding points y_i^n and y_j^n. In fact, for each sufficiently large n, the distance of $u_i(y^n)$ from the boundary of $S_i(y^n)$ *or* the distance of $u_j(y^n)$ from the boundary of $S_j(y^n)$, will exceed a certain positive number. For as n tends to infinity, $p(S_i(y^n)) + p(S_j(y^n))$ will approach $p(S_i(x)) > 0$—a simple continuity argument based on the absolute continuity of p will establish this—and for each sufficiently large n, at least one of the probabilities $p(S_i(y^n))$ or $p(S_j(y^n))$ will be positive by a definite amount, say δ. But in view of the boundedness of R, a convex set of p measure at least $\delta > 0$ cannot have its conditional mean arbitrarily near its boundary. This line of reasoning, which extends immediately to the case where some three or more members of $(x_1, x_2, \cdots, x_k)$ are equal, gives us the following lemma.

LEMMA 1. *Let $x = (x_1, x_2, \cdots, x_k)$ be the limit of a convergent sequence of k-points $\{y^n\} = \{(y_1^n, y_2^n, \cdots, y_k^n)\}$ satisfying $y_i^n \in R$, $y_i^n \neq y_j^n$ if $i \neq j$, $n = 1, 2, \cdots$. If $x_i = x_j$ for some $i \neq j$, then $\liminf_n \sum_{i=1}^{k} p(S_i(y^n))|y_i^n - u_i(y^n)| > 0$. Hence, if $\lim_{n\to\infty} \sum_{i=1}^{k} p(S_i(y^n))|y_i^n - u_i(y^n)| = 0$, each member of the k-tuple $(x_1, x_2, \cdots, x_k)$ is distinct from the others.*

We remark that if each member of the k-tuple $x = (x_1, x_2, \cdots, x_k)$ is distinct from the others, then $\pi(y) = (p(S_1(y)), p(S_2(y)), \cdots, p(S_k(y)))$, regarded as a mapping of M onto E_k, is continuous at x—this follows directly from the absolute continuity of p. Similarly, $u(y) = (u_1(y), u_2(y), \cdots, u_k(y))$ regarded as a mapping from M onto M is continuous at x—because of the absolute continuity of p and the boundness of R (finiteness of $\int z\, dp(z)$ would do). Putting this remark together with lemma 1, we get lemma 2.

LEMMA 2. *Let $x = (x_1, x_2, \cdots, x_k)$ be the limit of a convergent sequence of k-points $\{y^n\} = \{(y_1^n, y_2^n, \cdots, y_k^n)\}$ satisfying $y_i^n \in R$, $y_i^n \neq y_j^n$ if $i \neq j$, $n = 1, 2,$*

$\cdots$. *If* $\lim_{n\to\infty} \sum_{i=1}^{k} p(S_i(y^n))|y_i^n - u_i(y^n)| = 0$, *then* $\sum_{i=1}^{k} p(S_i(x))|x_i - u_i(x^n)| = 0$ *and each point* x_i *in the* k*-tuple* $(x_1, x_2, \cdots, x_k)$ *is distinct from the others.*

Lemmas 1 and 2 above are primarily technical in nature. The heart of the proofs of theorems 1 and 2 is the following application of martingale theory.

LEMMA 3. *Let* $t_1, t_2, \cdots$, *and* $\xi_1, \xi_2, \cdots$, *be given sequences of random variables, and for each* $n = 1, 2, \cdots$, *let* t_n *and* ξ_n *be measurable with respect to* β_n *where* $\beta_1 \subset \beta_2 \subset \cdots$ *is a monotone increasing sequence of* σ*-fields (belonging to the underlying probability space). Suppose each of the following conditions holds a.s.:* (i) $|t_n| \lesssim K < \infty$, (ii) $\xi_n \geq 0$, $\sum \xi_n < \infty$, (iii) $E(t_{n+1}|\beta_n) \leq t_n + \xi_n$. *Then the sequences of random variables* $t_1, t_2, \cdots$ *and* $s_0, s_1, s_2, \cdots$, *where* $s_0 = 0$ *and* $s_n = \sum_{i=1}^{n} (t_i - E(t_{i+1}|\beta_i))$, $n = 1, 2, \cdots$, *both converge a.s.*

PROOF. Let $y_n = t_n + s_{n-1}$ so that the y_n form a martingale sequence. Let c be a positive number and consider the sequence $\{\tilde{y}_n\}$ obtained by stopping y_n (see Doob [3], p. 300) at the first n for which $y_n \leq -c$. From (iii) we see that $y_n \geq -\sum_{i=1}^{n-1} \xi_i - K$, and since $y_n - y_{n-1} \geq 2K$, we have $\tilde{y}_n \geq \max(-\sum_{i=1}^{n-1} \xi_i - K, -(c + 2K))$. The sequence $\{\tilde{y}\}$ is a martingale, so that $E\tilde{y}_n = E\tilde{y}_1$, $n = 1, 2, \cdots$, and being bounded from below with $E|\tilde{y}_1| \leq K$, certainly $\sup_n E|\tilde{y}_n| < \infty$. The martingale theorem ([3], p. 319) shows $\tilde{y}_n$ converges a.s. But $y_n = \tilde{y}_n$ on the set A_c where $-\sum_{i=1}^{\infty} \xi_i > -c - K$, $i = 1, 2, \cdots$, and (ii) implies $P[A_c] \to 1$ as $c \to \infty$. Thus $\{y_n\}$ converge a.s. This means $s_n = y_{n+1} - t_{n+1}$ is a.s. bounded. Using (iii) we can write $-s_n = \sum_{i=1}^{n} \xi_i - \sum_{i=1}^{n} \Delta_i$ where $\Delta_i \geq 0$. But since s_n and $\sum_1^n \xi_i$ are a.s. bounded, $\sum \Delta_i$ converges a.s., s_n converges a.s., and finally, so does t_n. This completes the proof.

Turning now to the proof of theorem 1, let ω_n stand for the sequence $z_1, z_2, \cdots, z_{n-1+k}$, and let A_1^n be the event $[z_{n+k} \in S_1^n]$. Since S^{n+1} is the minimum distance partition relative to x^{n+1}, we have

$$
\begin{aligned}
(2.5) \qquad E[W(x^{n+1})|\omega_n] &= E\left[\sum_{i=1}^{k} \int_{S_i^{n+1}} |z - x_i^{n+1}|^2 \, dp(z)|\omega_n\right] \\
&\leq E\left[\sum_{i=1}^{k} \int_{S_i^n} |z - x_i^{n+1}|^2 \, dp(z)|\omega_n\right] \\
&= \sum_{j=1}^{k} E\left[\sum_{i=1}^{k} \int_{S_i^n} |z - x_i^{n+1}|^2 \, dp(z)|A_j^n, \omega_n\right] p_j^n.
\end{aligned}
$$

If $z_{n+k} \in S_j^n$, $x_i^{n+1} = x_i^n$ for $i \neq j$. Thus we obtain

$$
\begin{aligned}
(2.6) \qquad E[W(x^{n+1})|\omega_n] &\leq W(x^n) - \sum_{j=1}^{k} \left(\int_{S_j^n} |z - x_j^n|^2 \, dp(z)\right) p_j^n \\
&\quad + \sum_{j=1}^{k} E\left[\int_{S_j^n} |z - x_j^{n+1}|^2 \, dp(z)|A_j^n, \omega_n\right] p_j^n.
\end{aligned}
$$

Several applications of the relation $\int_A |z - x|^2 \, dp(z) = \int_A |z - u|^2 \, dp(z) + p(A)|x - u|^2$, where $\int_A (u - z) \, dp(z) = 0$, enables us to write the last term in (2.6) as

$$(2.7)\qquad \sum_{j=1}^{k}\Big[\int_{S_j^n}|z-x_j^n|^2\,dp(z)\,p_j^n-(p_j^n)^2|x_j^n-u_j^n|^2 + (p_j^n)^2|x_j^u-u_j^n|^2(w_j^n/(w_j^n+1))^2+\int_{S_j^n}|z-u_j^n|^2\,dp(z)\,p_j^n/(w_j^n+1)_2\Big].$$

Combining this with (2.6), we get

$$(2.8)\qquad E(W(x^{n+1})|\omega_n]\le W(x^n)-\sum_{j=1}^{k}|x_j^n-u_j^n|^2(p_j^n)^2(2w_j^n+1)/(w_j^n+1)^2 + \sum_{j=1}^{k}\sigma_{n,j}^2(p_j^n)^2/(w_j^n+1)^2,$$

where $\sigma_{n,j}^2=\int_{S_j^n}|z-u_j^n|^2\,dp(z)/p_j^n$.

Since we are assuming $p(R)=1$, certainly $W(x^n)$ is a.s. bounded, as is $\sigma_{n,j}^2$. We now show that

$$(2.9)\qquad \sum_n (p_j^n)^2/(w_j^n+1)^2$$

converges a.s. for each $j=1,2,\cdots,k$, thereby showing that

$$(2.10)\qquad \sum_n\Big(\sum_{j=1}^{k}[\sigma_{n,j}^2(p_j^n)^2/(w_j^n+1)^2]\Big)$$

converges a.s. Then lemma 3 can be applied with $t_n=W(x^n)$ and $\xi_n=\sum_{j=1}^{k}\sigma_{n,j}^2(p_j^n)^2/(w_j^n+1)^2$.

It suffices to consider the convergence of

$$(2.11)\qquad \sum_{n\ge 2}(p_j^n)^2/[(\beta+1+w_j^n)(\beta+1+w_j^{n+1})]$$

with $\beta>0$, since this implies convergence of (2.9). Also, this is convenient, for $E(I_j^n|\omega_n)=p_j^n$ where I_j^n is the characteristic function of the event $[z_{n+k}\in S_j^n]$, and on noting that $w_j^{n+1}=1+\sum_{i=1}^{n}I_j^i$, an application of theorem 1 in [4], p. 274, says that for any positive numbers α and β,

$$(2.12)\qquad P\Big[\beta+1+w_j^{n+1}\ge 1+\sum_{i=1}^{n}p_j^i-\alpha\sum_{i=1}^{n}v_j^i \text{ for all } n=1,2,\cdots\Big] > 1-(1+\alpha\beta)^{-1},$$

where $v_j^i=p_j^i-(p_j^i)^2$ is the conditional variance of I_j^i given ω_i. We take $\alpha=1$, and thus with probability at least $1-(1+\beta)^{-1}$ the series (2.11) is dominated by

$$(2.13)\qquad \sum_{n\ge 2}(p_j^n)^2\Big/\Big[\Big(1+\sum_{i=1}^{n-1}(p_j^i)^2\Big)\Big(1+\sum_{i=1}^{n}(p_j^i)^2\Big)\Big] = \sum_{n\ge 2}\Big[1\Big/\Big(1+\sum_{i=1}^{n-1}(p_j^i)^2\Big)-1\Big/\Big(1+\sum_{i=1}^{n}(p_j^i)^2\Big)\Big],$$

which clearly converges.

The choice of β being arbitrary, we have shown that (2.9) converges a.s. Application of lemma 3 as indicated above proves $W(x^n)$ converges a.s.

To identify the limit W_∞, note that with t_n and ξ_n taken as above, lemma 3

entails a.s. convergence of $\sum_n [W(x^n) - E[W(x^{n+1})|\omega_n]]$, and hence (2.8) implies a.s. convergence of

$$\sum_n \left(\sum_{j=1}^k |x^n - u_j^n|^2 (p_j^n)^2 (2w_j^n + 1)/(w_j^n + 1)^2 \right). \tag{2.14}$$

Since (2.14) dominates $\sum_n (\sum_{j=1}^k p_j^n |x_j^n - u_j^n|)/kn$, the latter converges a.s., and a little consideration makes it clear that

$$\sum_{j=1}^k p_j^n |x_j^n - u_j^n| = \sum_{j=1}^k p(S_j(x^n)) |x_j^n - u_j(x^n)| \tag{2.15}$$

converges to zero on a subsequence $\{x^{n_t}\}$ and that this subsequence has itself a convergent subsequence, say $\{x^{n_t}\}$. Let $x = (x_1, x_2, \cdots, x_k) = \lim_{t\to\infty} x^{n_t}$. Since $W(x) = V(x) + \sum_{j=1}^k p(S_j(x))|x_j - u(x)|^2$ and in particular,

$$W(x^n) = V(x^n) + \sum_{j=1}^k p(S_j(x^n)) |x_j^n - u(x_j^n)|^2, \tag{2.16}$$

we have only to show

(a) $\lim_{t\to\infty} W(x^{n_t}) = W_\infty = W(x)$, and

(b) $\lim_{t\to\infty} \sum_{j=1}^k p(S_j(x^{n_t})) |x_j^{n_t} - u(x_j^{n_t})|^2 = 0 = \sum_{j=1}^k p(S_j(x)) |x_j - u_j(x)|^2$.

Then $W(x) = V(x)$ and x is a.s. unbiased. (Obviously, $\sum_{i=1}^k p_i |a_i| = 0$ if and only if $\sum_{i=1}^k p_i |a_i|^2 = 0$, where $p_i \geq 0$.)

We show that (a) is true by establishing the continuity of $W(x)$. We have

$$\begin{aligned} W(x) &\leq \sum_{j=1}^k \int_{S_j(y)} |z - x_j|^2 \, dp(z) \\ &\leq \sum_{j=1}^k \int_{S_j(y)} |z - y_j|^2 + \sum_{j=1}^k [p(S_j(y)) |x_j - y_j|^2 \\ &\quad + 2|x_j - y_j| \int_{S_j(y)} |z - x_j| \, dp(z)], \end{aligned} \tag{2.17}$$

with the last inequality following easily from the triangle inequality. Thus $W(x) \leq W(y) + o(\rho(x, y))$, and similarly, $W(y) \leq W(x) + o(\rho(x, y))$.

To establish (b), lemma 2 can be applied with $\{y^n\}$ and $\{x^{n_t}\}$ identified, for a.s. $x_i^n \neq x_j^n$ for $i \neq j$, $n = 1, 2, \cdots$. It remains to remark that lemma 2 also implies a.s. $x_i \neq x_j$ for $i \neq j$. The proof of theorem 1 is complete.

Theorem 2 follows from the a.s. convergence of $\sum_n (\sum_{i=1}^k p_i^n |x_i^n - u_i^n|)/nk$ upon applying an elementary result (c.f. Halmos [8], theorem C, p. 203), which says that if $\sum a_n/n$ converges, $\sum_{i=1}^n a_i/n \to 0$.

2.3. *Remarks.* In a number of cases covered by theorem 1, all the unbiased k-points have the same value of W. In this situation, theorem 1 implies $\sum_{i=1}^k p_i^n |x_i^n - u_i^n|$ converges a.s. to zero. An example is provided by the uniform distribution over a disk in E_2. If $k = 2$, the unbiased k-point (x_1, x_2) with $x_1 \neq x_2$ consist of the family of points x_1 and x_2 opposite one another on a diameter, and at a certain fixed distance from the center of the disk. (There is one unbiased k-point with $x_1 = x_2$, both x_1 and x_2 being at the center of the disk in this case.)

The k-means thus converge to some such relative position, but theorem 1 does not quite permit us to eliminate the interesting possibility that the two means oscillate slowly but indefinitely around the center.

Theorem 1 provides for a.s. convergence of $\sum_{i=1}^{k} p_i^n|x_i^n - u_i^n|$ to zero in a slightly broader class of situations. This is where the unbiased k-points $x = (x_1, x_2, \cdots, x_k)$ with $x_i \neq x_j$ for $i \neq j$, are all *stable* in the sense that for each such x, $W(y) \geq W(x)$ (and hence $V(y) \geq V(x)$) for all y in a neighborhood of x. In this case, each such x falls in one of finitely many equivalence classes such that W is constant on each class. This is illustrated by the above example, where there is only a single equivalence class. If each of the equivalence classes contains only a single point, theorem 1 implies a.s. convergence of x^n to one of those points.

There are unbiased k-points which are not stable. Take a distribution on E_2 which has sharp peaks of probability at each corner of a square, and is symmetric about both diagonals. With $k = 2$, the two constituent points can by symmetrically located on a diagonal so that the boundary of the associated minimum distance partition coincides with the other diagonal. With some adjustment, such a k-point can be made to be unbiased, and if the probability is sufficiently concentrated at the corners of the square, any small movement of the two points off the diagonal in opposite directions, results in a decrease in $W(x)$. It seems likely that the k-means *cannot* converge to such a configuration.

For an example where the k-means converge with positive probability to a point x for which $V(x)$ is not a minimum, take equal probabilities at the corner points of a rectangle which is just slightly longer on one side than the other. Number with 1 the corner points, and 2 at the end points of one of the short edges, and 3 and 4, at the end points of the other short edge, with 1 opposite 3 on the long edge. Take $k = 2$. If the first four points fall at the corner points 1, 2, 3, 4 in that order, the two means at this stage are directly opposite one another at the middle of the long edges. New points falling at 1 and 3 will always be nearer the first mean, and points falling at 2 and 4 will always be nearer the second mean, unless one of the means has an excursion too near one of the corner points. By the strong law of large numbers there is positive probability this will *not* happen, and hence with positive probability the two means will converge to the midpoints of the long edges. The corresponding partition clearly does not have minimum within-class variance.

3. Applications

3.1. *Similarity grouping: coarsening and refining.* Perhaps the most obvious application of the k-means process is to the problem of "similarity grouping" or "clustering." The point of view taken in this application is *not* to find some unique, definitive grouping, but rather to simply aid the investigator in obtaining qualitative and quantitative understanding of large amounts of N-dimensional data by providing him with reasonably good similarity groups. The method should be used in close interaction with theory and intuition. Consequently, the

computer program actually prepared for this purpose involved several modifications of the k-means process, modifications which appear to be helpful in this sense.

First, the program involves two parameters: C for 'coarsening,' and R for 'refinement.' The program starts with a user specified value of k, and takes the first k points in the sample as initial means. The k-means process is started, each subsequent sample point being assigned to the nearest mean, the new mean computed, and so on, except that after each new point is added, and for the initial means as well, the program determines the pair of means which are nearest to each other among all pairs. If the distance between the members of this pair is less than C, they are averaged together, using their respective weights, to form a single mean. The nearest pair is again determined, their separation compared with C, and so on, until all the means are separated by an amount of C or more. Thus k is reduced and the partition defined by the means is coarsened. In addition, as each new point is processed and its distance from the nearest of the current means determined, this distance is compared with R. If the new point is found to be further than R from the nearest mean, it is left by itself as the seed point for a new mean. Thus k is increased and the partition is refined. Ordinarily we take $C \leq R$. After the entire sample is processed in this way, the program goes back and reclassifies all the points on the basis of nearness to the final means. The points thus associated with each mean constitutes the final grouping. The program prints out the points in each group along with as many as 18 characters of identifying information which may be supplied with each point. The distance of each point from its nearest mean, the distances between the means, the average for each group, of the squared distance of the points in each group from their respective defining means, and the grand average of these quantities over groups, are all printed out. The latter quantity, which is not quite the within-group variance, is called the within-class variation for purposes of the discussion below. If requested, the program determines frequencies of occurrence within each group of the values of discrete variables associated with each point. Up to twelve variables, with ten values for each variable, can be supplied. This makes it convenient to determine whether or not the groups finally obtained are related to other attributes of interest. (Copies of this experimental program are available from the author on request.)

The program has been applied with some success to several samples of real data, including a sample of five dimensional observations on the students' environment in 70 U.S. colleges, a sample of twenty semantic differential measurements on each of 360 common words, a sample of fifteen dimensional observations on 760 documents, and a sample of fifteen physiological observations on each of 560 human subjects. While analysis of this data is still continuing, and will be reported in detail elsewhere, the meaningfulness of the groups obtained is suggested by their obvious pertinence to other identifiable properties of the objects classified. This was apparent on inspection. For example, one group of colleges contained Reed, Swarthmore, Antioch, Oberlin, and Bryn

Mawr. Another group contained the Universities of Michigan, Minnesota, Arkansas, and Illinois, Cornell, Georgia Tech, and Purdue. Selecting at random a half-dozen words from several groups obtained from the semantic differential data, we find in one group the words calm, dusky, lake, peace, sleep, and white; in another group the words beggar, deformed, frigid, lagging, low; and in another group the words statue, sunlight, time, trees, truthful, wise.

When the sample points are rearranged in a new random order, there is some variation in the grouping which is obtained. However, this has not appeared to be a serious concern. In fact, when there are well separated clusters, as determined by inspection of the between-mean distances in relation to the within-class variation, repeated runs give virtually identical groupings. Minor shifts are due to the unavoidable difficulty that some points are located between clusters.

A degree of stability with respect to the random order in which the points are processed is also indicated by a tendency for the within-class variation to be similar in repeated runs. Thus when a sample of 250 points in five dimensions with $k = 18$, was run three times, each time with the points in a different random order, the within-class variation (see above) changed over the three runs by at most 7%. A certain amount of stability is to be expected simply because the within-class variation is the mean of k dependent random variables having the property that when one goes up the others generally go down. We can reasonably expect the within-class stability to generally increase with k and the sample size. Actually, it will usually be desirable to make several runs, with different values of C and R, and possibly adding, deleting, or rescaling variables, and so on, in an effort to understand the basic structure of the data. Thus any instabilities due to random ordering of the sample will be quickly noted. Being able to make numerous classifications cheaply and thereby look at the data from a variety of different perspectives is an important advantage.

Another general feature of the k-means procedure which is to be expected on intuitive grounds, and has been noted in practice, is a tendency for the means and the associated partition to avoid having the extreme of only one or two points in a set. In fact, there is an appreciable tendency for the frequency to be evenly split over groups. If there are a few relatively large groups, these tend to have relatively low within-class variation, as would be expected from a tendency for the procedure to approximate minimum variance partitions.

Running times of the above program on the IBM 7094 vary with C, R, the number of dimensions, and the number of points. A conservative estimate for 20-dimensional data, with C and R set so that k stays in the vicinity of 20, is one minute for two hundred sample points. Most of this computation time results from the coarsening and refining procedure and the auxiliary features. A limited amount of experience indicates the undecorated k-means procedure with $k = 20$ will process five hundred points in 20 dimensions in something like 10 seconds.

3.2. *Relevant classifications.* Suppose it is desired to develop a classification scheme on the basis of a sample, so that knowing the classification of a new point, it will be possible to predict a given dependent variable. The values of the de-

pendent variable are known for the sample. One way to do this, closely related to a procedure proposed by Fix and Hodges [6], is illustrated by the following computer experiment. A sample of 250 four-dimensional random vectors was prepared, with the values on each dimension being independently and uniformly distributed on the integers 1 through 10. Two of the dimensions were then arbitrarily selected, and if with respect to these two dimensions a point was either 'high' (above 5) on both *or* 'low' (5 or less) on both, it was called an A; otherwise, it was called a B. This gave 121 A's and 129 B's which were related to the selected dimensions in a strongly interactive fashion. The k-means with $k = 8$ were then obtained for the A's and B's separately. Finally, using the resulting 16 (four-dimensional) means, a prediction, A or B, was made for each of a *new* sample of 250 points on the basis of whether or not each point was nearest to an A mean or a B mean. These predictions turned out to be 87% correct.

As this example shows, the method is potentially capable of taking advantage of a highly nonlinear relationship. Also, the method has something to recommend it from the point of view of simplicity, and can easily be applied in many dimensions and to more than two-valued dependent variables.

3.3. *Approximating a general distribution.* Suppose it is desired to approximate a distribution on the basis of a sample of points. First the sample points are processed using the k-means concept or some other method which gives a minimum distance partition of the sample points. The approximation, involving a familiar technique, consists of simply fitting a joint normal distribution to the points in each group, and taking as the approximation the probability combination of these distributions, with the probabilities proportional to the number of points in each group.

Having fitted a mixture of normals in this way, it is computationally easy (on a computer) to do two types of analysis. One is predicting unknown coordinates of a new point given the remaining coordinates. This may be done by using the regression function determined on the assumption that the fitted mixture is the true distribution. Another possible application is a kind of nonlinear discriminant analysis. A mixture of k normals is fitted in the above fashion to two samples representing two given different populations; one can then easily compute the appropriate likelihood ratios for deciding to which population a new point belongs. This method avoids certain difficulties encountered in ordinary discriminant analysis, such as when the two populations are each composed of several distinct subgroups, but with some of the subgroups from one population actually between the subgroups of the other. Typically in this situation, one or several of the k-means will be centered in each of the subgroups—provided k is large enough—and the fitted normals then provide a reasonable approximation to the mixture.

To illustrate the application of the regression technique, consider the artificial sample of four-dimensional A's and B's described in the preceding section. On a fifth dimension, the A's were arbitrarily given a value of 10, and the B's a value of 0. The k-means procedure with $k = 16$ was used to partition the combined

sample of 250 five-dimensional points. Then the mixture of 16 normal distributions was determined as described above for this sample. The second sample of 250 points was prepared similarly, and predictions were made for the fifth dimension on the basis of the original four. The standard error of estimate on the new sample was 2.8. If, in terms of the original A-B classification, we had called a point on A if the predicted value exceeded 5, and a B otherwise, 96% of the designations would have been correct on the new sample. The mean of the predictions for the A's was 10.3, and for B's, 1.3.

Considering the rather complex and highly nonlinear relationship involved in the above sample, it is doubtful that any conventional technique would do as well. In the few instances which were tested, the method performed nearly as well as linear regression on normally distributed samples, provided k was not too large. This is not surprising inasmuch as with $k = 1$ the method *is* linear regression. In determining the choice of k, one procedure is to increase k as long as the error of estimate drops. Since this will probably result in "over fitting" the sample, a cross validation group is essential.

3.4. *A scrambled dimension test for independence among several variables.* As a general test for relationship among variables in a sample of N-dimensional observations, we propose proceeding as follows. First, the sample points are grouped into a minimum distance partition using k-means, and the within-class variance is determined. Then the relation among the variables is destroyed by randomly associating the values in each dimension; that is, a sample is prepared in which the variables are unrelated, but which has exactly the same marginal distributions as the original sample. A minimum distance partition and the associated within-class variance is now determined for this sample. Intuition and inspection of a few obvious examples suggest that on the average this "scrambling" will tend to *increase* the within-class variance, more or less regardless of whatever type of relation might have existed among the variables, and thus comparison of the two variances would reveal whether or not any such relation existed.

To illustrate this method, a sample of 150 points was prepared in which points were distributed uniformly outside a square 60 units on a side, but inside a surrounding square 100 units on a side. This gave a sample which involves essentially a zero correlation coefficient, and yet a substantial degree of relationship which could not be detected by any conventional quantitative technique known to the author (although it could be detected immediately by visual inspection). The above procedure was carried out using k-means with $k = 12$. As was expected, the variance after scrambling was increased by a factor of 1.6. The within-class variances were not only larger in the scrambled data, but were apparently more variable. This procedure was also applied to the five-dimensional sample described in the preceding section. Using $k = 6$, 12, and 18, the within-class variance increased after scrambling by the factors 1.40, 1.55, and 1.39, respectively.

A statistical test for nonindependence can be constructed by simply repeating the scrambling and partitioning a number of times, thus obtaining empirically a

sample from the conditional distribution of the within-class variance under the hypothesis that the variables are unrelated *and* given the marginal values of the sample. Under the hypothesis of independence, the unscrambled variance should have the same (conditional) distribution as the scrambled variance. In fact, the rank of the unscrambled variance in this empirical distribution should be equally likely to take on any of the possible values 1, 2, $\cdots$, $n + 1$, where n is the number of scrambled samples taken, regardless of the marginal distributions in the underlying population. Thus the rank can be used in a nonparametric test of the hypothesis of independence. For example, if the unscrambled variance is the lowest in 19 values of the scrambled variance, we can reject the hypothesis of independence with a Type I error of .05.

A computer program was not available to do the scrambling, and its being inconvenient to set up large numbers of scrambled samples using punched cards, further testing of this method was not undertaken. It is estimated, however, that an efficient computer program would easily permit this test to be applied at, say, the .01 level, on large samples in many dimensions.

The power of this procedure remains to be seen. On the encouraging side is the related conjecture, that for fixed marginal distributions, the within-class variance for the optimal partition as defined in section 1 is maximal when the joint distribution is actually the product of the marginals. If this is true (and it seems likely that it is, at least for a large class of reasonable distributions), then we reason that since the k-means process tends to give a good partition, this difference will be preserved in the scrambled and unscrambled variances, particularly for large samples. Variation in the within-class variance due to the random order in which the points are processed, can be reduced by taking several random orders, and averaging their result. If this is done for the scrambled runs as well, the Type I error is preserved, while the power is increased somewhat.

3.5. *Distance-based classification trees.* The k-means concept provides a number of simple procedures for developing lexigraphic classification systems (filing systems, index systems, and so on) for a large sample of points. To illustrate, we describe briefly a procedure which results in the within-group variance of each of the groups at the most refined level of classification being no more than a specified number, say R. The sample k-means are first determined with a selected value of k, for example, $k = 2$. If the variance of any of the groups of points nearest to these means is less than R, these groups are not subclassified further. The remaining groups are each processed in the same way, that is, k-means are determined for each of them, and then for the points nearest each of these, and so on. This is continued until only groups with within-group variance less than R remain. Thus for each mean at the first level, there is associated several means at the second level, and so on. Once the means at each level are determined from the sample in this fashion, the classification of a new point is defined by the rule: first, see which one of the first level k-means the point is nearest; then see which one of the second-level k-means associated with that mean the point is nearest,

and so on; finally the point is assigned to a group which in the determining sample has variance no more than R.

This procedure has some promising features. First, the amount of computation required to determine the index is approximately linear in the sample size and the number of levels. The procedure can be implemented easily on the computer. At each stage during the construction of the classification tree, we are employing a powerful heuristic, which consists simply of putting points which are near to each other in the same group. Each of the means at each level is a fair representation of its group, and can be used for certain other purposes, for instance, to compare other properties of the points as a function of their classification.

3.6. *A two-step improvement procedure.* The method of obtaining partitions with low within-class variance which was suggested by Forgy and Jennrich (see section 1.1) works as follows. Starting with an arbitrary partition into k sets, the means of the points in each set are first computed. Then a new partition of the points is formed by the rule of putting the points into groups of the basis of nearness to the first set of means. The average squared distance of the points in the new partition from the first set of means (that is, from their nearest means) is obviously less than the within-class variance of the first partition. But the average within-class *variance* of the new partition is even lower, for the variance of the squared distance of the points in each group from their respective means, and the mean, of course, is that point which minimizes the average squared distance from itself. Thus the new partition has lower variance. Computationally, the two steps of the method are (1) compute the means of the points in each set in the initial partition and (2) reclassify the points on the basis of nearness to these means, thus forming a new partition. This can be iterated and the series of the partitions thus produced have decreasing within-class variances and will converge in a finite number of steps.

For a given sample, one cycle of this method requires about as much computation as the k-means. The final partition obtained will depend on the initial partition, much as the partition produced by k-means will depend on random variation in the order in which the points are processed. Nevertheless, the procedure has much to recommend it. By making repeated runs with different initial starting points, it would seem likely that one would actually obtain the sample partition with minimum within-class variance.

4. General metric spaces

It may be something more than a mere mathematical exercise to attempt to extend the idea of k-means to general metric spaces. Metric spaces other than Euclidian ones do occur in practice. One prominent example is the space of binary sequences of fixed length under Hamming distance.

An immediate difficulty in making such an extension is the notion of mean itself. The arithmetic operations defining the mean in Euclidian space may not be available. However, with the communication problem of section 1 in mind,

one thinks of the problem of representing a population by a point, the goal being to have low average error in some sense. Thus we are led to proceed rather naturally as follows.

Let M be a compact metric space with distance ρ, let $\mathfrak{F}$ be the σ-algebra of subsets of M, and let p be a probability measure on $\mathfrak{F}$. For the measure p, a centroid of order $r \geq 0$ is any point in the set $\mathcal{C}^r$ of points x^* such that $\int \rho^r(x^*, z)\, dp(z) = \inf_x \int \rho^r(x, z)\, dp(z)$. The quantity $\int \rho^r(x^*, z)\, dp(z)$ is the r-th moment of p. The compactness and the continuity of ρ guarantee that $\mathcal{C}^r$ is nonempty. For finite samples, sample centroids are defined analogously, each point in the sample being treated as having measure $1/n$ where n is the sample size; namely, for a sample of size n, the sample centroid is defined up to an equivalence class $\mathcal{C}_n^r$ which consists of all those points $\hat{x}_n$ such that $\sum_{i=1}^n \rho^r(\hat{x}_n, z_i) = \inf_x \sum_{i=1}^n \rho^r(x, z_i)$, where $z_1, z_2, \cdots, z_n$ is the sample.

Note that with M the real line, and ρ ordinary distance, $r = 2$ yields the ordinary mean, and $r = 1$ yields the family of medians. As r tends to ∞, the elements of $\mathcal{C}_r$ will tend to have (in a manner which can easily be made precise) the property that they are centers for a spherical covering of the space with minimal radius. In particular, on the line, the centroid will tend to the mid-range. As r tends to zero, one obtains what may with some justification be called a mode, for on a compact set, $\rho^r(x, y)$ is approximately 1 for small r, except where x and y are very near, so that minimizing $\int \rho^r(x, y)\, dp(y)$ with respect to x, involves attempting to locate x so that there is a large amount of probability in its immediate vicinity. (This relationship can also be made precise.)

We note that the optimum communication problem mentioned in section 1.1 now takes the following general form. Find a partition $S = \{S_1, S_2, \cdots, S_k\}$ which minimizes $w = \sum_{i=1}^k \int_{S_i} \rho^r(x_i^*, y)\, dp(y)$, where x_i^* is the centroid of order r with respect to the (conditional) distribution on S_i. If there is any mass in a set S_i nearer to x_j than to x_i, $j \neq i$, then w can be reduced by modifying S_i and S_j so as to reassign this mass to S_j. It follows that in minimizing w we can restrict attention to partitions which are minimum distance partitions, analogous to those defined in section 2, that is, partitions of the form $S(x) = \{S_1(x), S_2(x), \cdots, S_k(x)\}$ where $x = (x_1, x_2, \cdots, x_k)$ is a k-tuple of points in M, and $S_i(x)$ is a set of points at least as near x_i (in terms of ρ) as to x_j if $j \neq i$. In keeping with the terminology of section 2, we may say that a k-tuple, or "k-point," $x = (x_1, x_2, \cdots, x_k)$ is unbiased if x_i, $i = 1, 2, \cdots, k$, belongs to the class of points which are centroids within $S_i(x)$.

It is now clear how to extend the concept of k-means to metric spaces; the notion of centroid replaces the more special concept of mean. The first 'k-centroid' $(x_1^1, x_2^1, \cdots, x_k^1)$ consists of the first k points in the sample, and thereafter as each new point is considered, the nearest of the centroids is determined. The new point is assigned to the corresponding group and the centroid of that group modified accordingly, and so on.

It would seem reasonable to suppose that the obvious extension of theorem 1 would hold. That is, under independent sampling, $\sum_{i=1}^k \int_{S_i(x^n)} \rho^r(z, x_i^n)\, dp(z)$ will

converge a.s., and the convergent subsequences of the sequence of sample k-centroids will have their limits in the class of unbiased k-points. This is true, at any rate, for $k = 1$ and $r = 1$, for if $z_1, z_2, \cdots, z_n$ are independent, $\sum_{i=1}^{n} \rho(z_i, y)/n$ is the mean of independent, identically distributed random variables, which because M is compact, are uniformly bounded in y. It follows (cf. Parzen [13]) that $\sum_{i=1}^{n} \rho(z_i, y)/n$ converges a.s. to $\int \rho(z, y)\, dp(z)$ uniformly in y. By definition of the sample centroid, we have $\sum_{i=1}^{n} \rho(z_i, x^*)/n \geq \sum_{i=1}^{n} \rho(z_i, \hat{x}_n)/n$; hence, $\int \rho(z, x^*)\, dp(z) \geq \limsup \sum_{i=1}^{n} \rho(z_i, \hat{x}_n)/n$ with probability 1. On the other hand, from the triangle inequality, $\sum_{i=1}^{n} \rho(z_i, y)/n \leq \sum_{i=1}^{n} \rho(z_i, \hat{x}_n)/n + \rho(\hat{x}_n, y)$. Using this inequality on a convergent subsequence $\hat{x}_{n1}, \hat{x}_{n2}, \cdots$, chosen so that

$$\lim_{t\to\infty} \sum_{i=1}^{n_t} \rho(z_i, \hat{x}_{n_t})/n_t = \liminf \sum_{i=1}^{n} \rho(z_i, \hat{x}_n)/n, \tag{4.1}$$

we see that with probability 1,

$$\int \rho(z, x^*)\, dp(z) \leq \int \rho(z, y)\, dp(z) \leq \liminf \sum_{i=1}^{n} \rho(z_i, \hat{x}_n)/n, \tag{4.2}$$

where $y = \lim_{t\to\infty} \hat{x}_{n_t}$.

Provided the necessary computations can be accomplished, the methods suggested in sections 3.1, 3.2, 3.4, 3.5, and 3.6 can all be extended to general metric spaces in a quite straightforward fashion.

ACKNOWLEDGMENTS

The author is especially indebted to Tom Ferguson, Edward Forgy, and Robert Jennrich, for many valuable discussions of the problems to which the above results pertain. Richard Tenney and Sonya Baumstein provided the essential programming support, for which the author is very grateful. Computing facilities were provided by the Western Data Processing Center.

Note added in proof. The author recently learned that C. S. Wallace of the University of Sidney and G. H. Ball of the Stanford Research Institute have independently used this method as a part of a more complex procedure. Ball has described his method, and reviewed earlier literature, in the interesting paper "Data analysis in the social sciences: What about the details?", *Proceedings of the Fall Joint Computer Conference*, Washington, D.C., Spartan Books, 1965.

REFERENCES

[1] DAVID BLACKWELL, "Comparison of experiments," *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability*, Berkeley and Los Angeles, University of California Press, 1951, pp. 93–102.

[2] D. R. Cox, "Note on grouping," *J. Amer. Statist. Assoc.*, Vol. 52 (1957), pp. 543–547.
[3] J. L. Doob, *Stochastic Processes*, New York, Wiley, 1953.
[4] L. E. Dubins and L. J. Savage, "A Tchebycheff-like inequality for stochastic processes," *Proc. Nat. Acad. Sci. U.S.A.*, Vol. 53 (1965), pp. 274–275.
[5] W. D. Fisher, "On grouping for maximum homogeneity," *J. Amer. Statist. Assoc.*, Vol. 53 (1958), pp. 789–798.
[6] Evelyn Fix and J. L. Hodges, Jr., "Discriminatory Analysis," USAF Project Report, School of Aviation Medicine, Project Number 21-49-004, No. 4 (1951).
[7] Edward Forgy, "Cluster analysis of multivariate data: efficiency vs. interpretability of classifications," abstract, *Biometrics*, Vol. 21 (1965), p. 768.
[8] Paul R. Halmos, *Measure Theory*, New York, Van Nostrand, 1950.
[9] J. MacQueen, "The classification problem," Western Management Science Institute Working Paper No. 5, 1962.
[10] ———, "On convergence of k-means and partitions with minimum average variance," abstract, *Ann. Math. Statist.*, Vol. 36 (1965), p. 1084.
[11] Jacob Marschak, "Towards an economic theory of organization and information," *Decision Processes*, edited by R. M. Thrall, C. H. Coombs, and R. C. Davis, New York, Wiley, 1954.
[12] ———, "Remarks on the economics of information," Proceedings of the scientific program following the dedication of the Western Data Processing Center, University of California, Los Angeles, January 29–30, 1959.
[13] Emanuel Parzen, "On uniform convergence of families of sequences of random variables," *Univ. California Publ. Statist.*, Vol. 2, No. 2 (1954), pp. 23–54.
[14] George S. Sebestyen, *Decision Making Process in Pattern Recognition*, New York, Macmillan, 1962.
[15] Robert R. Sokal and Peter H. Sneath, *Principles of Numerical Taxonomy*, San Francisco, Freeman, 1963.
[16] Joe Ward, "Hierarchical grouping to optimize an objective function," *J. Amer. Statist. Assoc.*, Vol. 58 (1963), pp. 236–244.

Survey of Clustering Data Mining Techniques

Pavel Berkhin
Accrue Software, Inc.

Clustering is a division of data into groups of similar objects. Representing the data by fewer clusters necessarily loses certain fine details, but achieves simplification. It models data by its clusters. Data modeling puts clustering in a historical perspective rooted in mathematics, statistics, and numerical analysis. From a machine learning perspective clusters correspond to *hidden patterns*, the search for clusters is *unsupervised learning*, and the resulting system represents a *data concept*. From a practical perspective clustering plays an outstanding role in data mining applications such as scientific data exploration, information retrieval and text mining, spatial database applications, Web analysis, CRM, marketing, medical diagnostics, computational biology, and many others.

Clustering is the subject of active research in several fields such as statistics, pattern recognition, and machine learning. This survey focuses on clustering in data mining. Data mining adds to clustering the complications of very large datasets with very many attributes of different types. This imposes unique computational requirements on relevant clustering algorithms. A variety of algorithms have recently emerged that meet these requirements and were successfully applied to real-life data mining problems. They are subject of the survey.

Categories and Subject Descriptors: I.2.6. [**Artificial Intelligence**]: Learning – *Concept learning*; I.4.6 [**Image Processing**]: Segmentation; I.5.1 [**Pattern Recognition**]: Models; I.5.3 [**Pattern Recognition**]: Clustering.

General Terms: Algorithms, Design

Additional Key Words and Phrases: Clustering, partitioning, data mining, unsupervised learning, descriptive learning, exploratory data analysis, hierarchical clustering, probabilistic clustering, k-means

Content:

Author's address: Pavel Berkhin, Accrue Software, 1045 Forest Knoll Dr., San Jose, CA, 95129; e-mail: pavelb@accrue.com

1. Introduction

The goal of this survey is to provide a comprehensive review of different clustering techniques in data mining. *Clustering* is a division of data into groups of similar objects. Each group, called cluster, consists of objects that are similar between themselves and dissimilar to objects of other groups. Representing data by fewer clusters necessarily loses certain fine details (akin to lossy data compression), but achieves simplification. It represents many data objects by few clusters, and hence, it models data by its clusters. Data modeling puts clustering in a historical perspective rooted in mathematics, statistics, and numerical analysis. From a machine learning perspective clusters correspond to *hidden patterns*, the search for clusters is *unsupervised learning*, and the resulting system represents a *data concept*. Therefore, clustering is unsupervised learning of a hidden data

concept. Data mining deals with large databases that impose on clustering analysis additional severe computational requirements. These challenges led to the emergence of powerful broadly applicable data mining clustering methods surveyed below.

1.1. Notations

To fix the context and to clarify prolific terminology, we consider a dataset X consisting of data points (or synonymously, *objects, instances, cases*, *patterns*, *tuples*, *transactions*) $x_i = (x_{i1},...,x_{id}) \in A$ in attribute space A, where $i = 1:N$, and each component $x_{il} \in A_l$ is a numerical or nominal categorical *attribute* (or synonymously*, feature*, *variable*, *dimension*, *component*, *field*). For a discussion of attributes data types see [Han & Kamber 2001]. Such point-by-attribute data format conceptually corresponds to a $N \times d$ matrix and is used by the majority of algorithms reviewed below. However, data of other formats, such as variable length sequences and heterogeneous data, is becoming more and more popular. The simplest attribute space subset is a direct Cartesian product of sub-ranges $C = \prod C_l \subset A$, $C_l \subseteq A_l$, $l = 1:d$, called a *segment* (also *cube, cell, region*). A *unit* is an elementary segment whose sub-ranges consist of a single category value, or of a small numerical bin. Describing the numbers of data points per every *unit* represents an extreme case of clustering, a *histogram*, where no actual clustering takes place. This is a very expensive representation, and not a very revealing one. User driven *segmentation* is another commonly used practice in data exploration that utilizes expert knowledge regarding the importance of certain sub-domains. We distinguish clustering from segmentation to emphasize the importance of the automatic learning process.

The ultimate goal of clustering is to assign points to a finite system of k subsets, clusters. Usually subsets do not intersect (this assumption is sometimes violated), and their union is equal to a full dataset with possible exception of outliers

$$X = C_1 \cup ... \cup C_k \cup C_{outliers},\ C_{j_1} \cap C_{j_2} = \phi .$$

1.2. Clustering Bibliography at Glance

General references regarding clustering include [Hartigan 1975; Spath 1980; Jain & Dubes 1988; Kaufman & Rousseeuw 1990; Dubes 1993; Everitt 1993; Mirkin 1996; Jain et al. 1999; Fasulo 1999; Kolatch 2001; Han et al. 2001; Ghosh 2002]. A very good introduction to contemporary data mining clustering techniques can be found in the textbook [Han & Kamber 2001].

There is a close relationship between clustering techniques and many other disciplines. Clustering has always been used in statistics [Arabie & Hubert 1996] and science [Massart & Kaufman 1983]. The classic introduction into pattern recognition framework is given in [Duda & Hart 1973]. Typical applications include *speech* and *character recognition*. Machine learning clustering algorithms were applied to *image segmentation* and *computer vision* [Jain & Flynn 1996]. For statistical approaches to pattern recognition see [Dempster et al. 1977] and [Fukunaga 1990]. Clustering can be viewed as a density estimation problem. This is the subject of traditional multivariate statistical estimation [Scott 1992]. Clustering is also widely used for data compression in image processing, which is also known as *vector quantization* [Gersho & Gray 1992]. Data

fitting in numerical analysis provides still another venue in data modeling [Daniel & Wood 1980].

This survey's emphasis is on clustering in data mining. Such clustering is characterized by large datasets with many attributes of different types. Though we do not even try to review particular applications, many important ideas are related to the specific fields. Clustering in data mining was brought to life by intense developments in information retrieval and text mining [Cutting et al. 1992; Steinbach et al. 2000; Dhillon et al. 2001], spatial database applications, for example, GIS or astronomical data, [Xu et al. 1998; Sander et al. 1998; Ester et al. 2000], sequence and heterogeneous data analysis [Cadez et al. 2001], Web applications [Cooley et al. 1999; Heer & Chi 2001; Foss et al. 2001], DNA analysis in computational biology [Ben-Dor & Yakhini 1999], and many others. They resulted in a large amount of application-specific developments that are beyond our scope, but also in some general techniques. These techniques and classic clustering algorithms that relate to them surveyed below.

1.3. Classification of Clustering Algorithms

Categorization of clustering algorithms is neither straightforward, nor canonical. In reality, groups below overlap. For reader's convenience we provide a classification closely followed by this survey. Corresponding terms are explained below.

Clustering Algorithms

- Hierarchical Methods
  - Agglomerative Algorithms
  - Divisive Algorithms
- Partitioning Methods
  - Relocation Algorithms
  - Probabilistic Clustering
  - *K*-medoids Methods
  - *K*-means Methods
  - Density-Based Algorithms
    - Density-Based Connectivity Clustering
    - Density Functions Clustering
- Grid-Based Methods
- Methods Based on Co-Occurrence of Categorical Data
- Constraint-Based Clustering
- Clustering Algorithms Used in Machine Learning
  - Gradient Descent and Artificial Neural Networks
  - Evolutionary Methods
- Scalable Clustering Algorithms
- Algorithms For High Dimensional Data
  - Subspace Clustering
  - Projection Techniques
  - Co-Clustering Techniques

1.4. Plan of Further Presentation

Traditionally clustering techniques are broadly divided in *hierarchical* and *partitioning*. Hierarchical clustering is further subdivided into *agglomerative* and *divisive*. The basics of hierarchical clustering include Lance-Williams formula, idea of *conceptual clustering*, now classic algorithms SLINK, COBWEB, as well as newer algorithms CURE and CHAMELEON. We survey them in the section *Hierarchical Clustering*.

While hierarchical algorithms build clusters gradually (as crystals are grown), partitioning algorithms learn clusters directly. In doing so, they either try to discover clusters by iteratively relocating points between subsets, or try to identify clusters as areas highly populated with data. Algorithms of the first kind are surveyed in the section *Partitioning Relocation Methods*. They are further categorized into *probabilistic clustering* (EM framework, algorithms SNOB, AUTOCLASS, MCLUST), *k-medoids* methods (algorithms PAM, CLARA, CLARANS, and its extension), and *k-means* methods (different schemes, initialization, optimization, harmonic means, extensions). Such methods concentrate on how well points fit into their clusters and tend to build clusters of proper convex shapes.

Partitioning algorithms of the second type are surveyed in the section *Density-Based Partitioning*. They try to discover dense connected components of data, which are flexible in terms of their shape. Density-based connectivity is used in the algorithms DBSCAN, OPTICS, DBCLASD, while the algorithm DENCLUE exploits space density functions. These algorithms are less sensitive to outliers and can discover clusters of irregular shapes. They usually work with low-dimensional data of numerical attributes, known as *spatial* data. Spatial objects could include not only points, but also extended objects (algorithm GDBSCAN).

Some algorithms work with data indirectly by constructing summaries of data over the attribute space subsets. They perform space segmentation and then aggregate appropriate segments. We discuss them in the section *Grid-Based Methods*. They frequently use hierarchical agglomeration as one phase of processing. Algorithms BANG, STING, WaveCluster, and an idea of fractal dimension are discussed in this section. Grid-based methods are fast and handle outliers well. Grid-based methodology is also used as an intermadiate step in many other algorithms (for example, CLIQUE, MAFIA).

Categorical data is intimately connected with transactional databases. The concept of a similarity alone is not sufficient for clustering such data. The idea of categorical data co-occurrence comes to rescue. The algorithms ROCK, SNN, and CACTUS are surveyed in the section *Co-Occurrence of Categorical Data*. The situation gets even more aggravated with the growth of the number of items involved. To help with this problem an effort is shifted from data clustering to pre-clustering of items or categorical attribute values. Development based on *hyper-graph* partitioning and the algorithm STIRR exemplify this approach.

Many other clustering techniques are developed, primarily in machine learning, that either have theoretical significance, are used traditionally outside the data mining community, or do not fit in previously outlined categories. The boundary is blurred. In the section *Other Clustering Techniques* we discuss *relationship to supervised learning*, *gradient descent* and *ANN* (LKMA, SOM), *evolutionary methods* (simulated annealing**,**

genetic algorithms (GA)), and the algorithm AMOEBA. We start, however, with the emerging field of *constraint-based clustering* that is influenced by requirements of real-world data mining applications.

Data Mining primarily works with large databases. Clustering large datasets presents scalability problems reviewed in the section *Scalability and VLDB Extensions*. Here we talk about algorithms like DIGNET, about BIRCH and other data squashing techniques, and about Hoffding or Chernoff bounds.

Another trait of real-life data is its high dimensionality. Corresponding developments are surveyed in the section *Clustering High Dimensional Data*. The trouble comes from a decrease in metric separation when the dimension grows. One approach to *dimensionality reduction* uses attributes transformations (DFT, PCA, wavelets). Another way to address the problem is through *subspace clustering* (algorithms CLIQUE, MAFIA, ENCLUS, OPTIGRID, PROCLUS, ORCLUS). Still another approach clusters attributes in groups and uses their derived proxies to cluster objects. This double clustering is known as *co-clustering*.

Issues that are common to different clustering methods are overviewed in the section *General Algorithmic Issues*. We talk about *assessment of results*, determination of *appropriate number of clusters* to build, *data preprocessing* (attribute selection, data scaling, special data indices), *proximity measures*, and *handling outliers*.

1.5. Important Issues

What are the properties of clustering algorithms we are concerned with in data mining? These properties include:

- Type of attributes algorithm can handle
- Scalability to large datasets
- Ability to work with high dimensional data
- Ability to find clusters of irregular shape
- Handling outliers
- Time complexity (when there is no confusion, we use the term *complexity*)
- Data order dependency
- Labeling or assignment (hard or strict vs. soft of fuzzy)
- Reliance on a priori knowledge and user defined parameters
- Interpretability of results

While we try to keep these issues in mind, realistically, we mention only few with every algorithm we discuss. The above list is in no way exhaustive. For example, we also discuss such properties as ability to work in pre-defined memory buffer, ability to restart and ability to provide an intermediate solution.

2. Hierarchical Clustering

Hierarchical clustering builds a cluster hierarchy or, in other words, a tree of clusters, also known as a *dendrogram*. Every cluster node contains child clusters; sibling clusters partition the points covered by their common parent. Such an approach allows exploring

data on different levels of granularity. Hierarchical clustering methods are categorized into ***agglomerative*** (bottom-up) and ***divisive*** (top-down) [Jain & Dubes 1988; Kaufman & Rousseeuw 1990]. An *agglomerative* clustering starts with one-point (singleton) clusters and recursively merges two or more most appropriate clusters. A *divisive* clustering starts with one cluster of all data points and recursively splits the most appropriate cluster. The process continues until a stopping criterion (frequently, the requested number k of clusters) is achieved. Advantages of hierarchical clustering include:

- Embedded flexibility regarding the level of granularity
- Ease of handling of any forms of similarity or distance
- Consequently, applicability to any attribute types

Disadvantages of hierarchical clustering are related to:

- Vagueness of termination criteria
- The fact that most hierarchical algorithms do not revisit once constructed (intermediate) clusters with the purpose of their improvement

The classic approaches to hierarchical clustering are presented in the sub-section *Linkage Metrics*. Hierarchical clustering based on linkage metrics results in clusters of proper (convex) shapes. Active contemporary efforts to build cluster systems that incorporate our intuitive concept of clusters as connected components of arbitrary shape, including the algorithms CURE and CHAMELEON, are surveyed in the sub-section *Hierarchical Clusters of Arbitrary Shapes*. Divisive techniques based on binary taxonomies are presented in the sub-section *Binary Divisive Partitioning*. The sub-section *Other Developments* contains information related to incremental learning, model-based clustering, and cluster refinement.

In hierarchical clustering our regular point-by-attribute data representation is sometimes of secondary importance. Instead, hierarchical clustering frequently deals with the $N \times N$ matrix of distances (dissimilarities) or similarities between training points. It is sometimes called *connectivity* matrix. Linkage metrics are constructed (see below) from elements of this matrix. The requirement of keeping such a large matrix in memory is unrealistic. To relax this limitation different devices are used to introduce into the connectivity matrix some sparsity. This can be done by omitting entries smaller than a certain threshold, by using only a certain subset of data representatives, or by keeping with each point only a certain number of its nearest neighbors. For example, nearest neighbor chains have decisive impact on memory consumption [Olson 1995]. A sparse matrix can be further used to represent intuitive concepts of closeness and connectivity. Notice that the way we process original (dis)similarity matrix and construct a linkage metric reflects our a priori ideas about the data model.

With the (sparsified) connectivity matrix we can associate the connectivity graph $G = (X, E)$ whose vertices X are data points, and edges E and their weights are pairs of points and the corresponding positive matrix entries. This establishes a connection between hierarchical clustering and graph partitioning.

One of the most striking developments in hierarchical clustering is the algorithm BIRCH. Since scalability is the major achievement of this blend strategy, this algorithm is discussed in the section *Scalable VLDB Extensions*. However, data squashing used by

BIRCH to achieve scalability, has independent importance. Hierarchical clustering of large datasets can be very sub-optimal, even if data fits in memory. Compressing data may improve performance of hierarchical algorithms.

2.1. Linkage Metrics

Hierarchical clustering initializes a cluster system as a set of singleton clusters (agglomerative case) or a single cluster of all points (divisive case) and proceeds iteratively with merging or splitting of the most appropriate cluster(s) until the stopping criterion is achieved. The appropriateness of a cluster(s) for merging/splitting depends on the (dis)similarity of cluster(s) elements. This reflects a general presumption that clusters consist of similar points. An important example of dissimilarity between two points is the distance between them. Other proximity measures are discussed in the section *General Algorithm Issues*.

To merge or split subsets of points rather than individual points, the distance between individual points has to be generalized to the distance between subsets. Such derived proximity measure is called a ***linkage metric***. The type of the linkage metric used significantly affects hierarchical algorithms, since it reflects the particular concept of *closeness* and *connectivity*. Major inter-cluster linkage metrics [Murtagh 1985, Olson 1995] include *single link*, *average link*, and *complete link*. The underlying dissimilarity measure (usually, distance) is computed for every pair of points with one point in the first set and another point in the second set. A specific operation such as minimum (single link), average (average link), or maximum (complete link) is applied to pair-wise dissimilarity measures:

$$d(C_1, C_2) = operation\{d(x, y) \mid x \in C_1, y \in C_2\}.$$

Early examples include the algorithm SLINK [Sibson 1973], which implements single link, Voorhees' method [Voorhees 1986], which implements average link, and the algorithm CLINK [Defays 1977], which implements complete link. Of these SLINK is referenced the most. It is related to the problem of finding the Euclidean minimal spanning tree [Yao 1982] and has $O(N^2)$ complexity. The methods using inter-cluster distances defined in terms of pairs with points in two respective clusters (subsets) are called ***graph*** methods. They do not use any cluster representation other than a set of points. This name naturally relates to the connectivity graph $G = (X, E)$ introduced above, since every data partition corresponds to a graph partition. Such methods can be appended by so-called ***geometric*** methods in which a cluster is represented by its central point. It results in *centroid*, *median,* and *minimum variance* linkage metrics. Under the assumption of numerical attributes, the center point is defined as a centroid or an average of two cluster centroids subject to agglomeration.

All of the above linkage metrics can be derived as instances of the Lance-Williams updating formula [Lance & Williams 1967]

$$d(C_i \cup C_j, C_k) = a(i)d(C_i, C_k) + a(k)d(C_j, C_k) + bd(C_i, C_j) + c\left|d(C_i, C_k) - d(C_j, C_k)\right|.$$

Here *a,b,c* are coefficients corresponding to a particular linkage. This formula expresses a linkage metric between the union of the two clusters and the third cluster in terms of

underlying components. The Lance-Williams formula has an utmost importance since it makes manipulation with dis(similarity) computationally feasible. Survey of linkage metrics can be found in [Murtagh 1983; Day & Edelsbrunner 1984]. When the base measure is distance, these methods capture inter-cluster closeness. However, a similarity-based view that results in intra-cluster connectivity considerations is also possible. This is how original average link agglomeration (Group-Average Method) [Jain & Dubes 1988] was introduced.

Linkage metrics-based hierarchical clustering suffers from time complexity. Under reasonable assumptions, such as *reducibility condition* (graph methods satisfy this condition), linkage metrics methods have $O(N^2)$ complexity [Olson 1995]. Despite the unfavorable time complexity, these algorithms are widely used. An example is algorithm AGNES (AGlomerative NESting) [Kaufman & Rousseeuw 1990] used in S-Plus.

When the connectivity $N \times N$ matrix is sparsified, graph methods directly dealing with the connectivity graph *G* can be used. In particular, hierarchical divisive MST (Minimum Spanning Tree) algorithm is based on graph partitioning [Jain & Dubes 1988].

2.2. Hierarchical Clusters of Arbitrary Shapes

Linkage metrics based on Euclidean distance for hierarchical clustering of spatial data naturally predispose to clusters of proper convex shapes. Meanwhile, visual scanning of spatial images frequently attests clusters with curvy appearance.

Guha et al. [1998] introduced the hierarchical agglomerative clustering algorithm CURE (Clustering Using REpresentatives). This algorithm has a number of novel features of general significance. It takes special care with outliers and with label assignment stage. It also uses two devices to achieve scalability. The first one is data sampling (section *Scalability and VLDB Extensions*). The second device is data partitioning in *p* partitions, so that fine granularity clusters are constructed in partitions first. A major feature of CURE is that it represents a cluster by a fixed number *c* of points scattered around it. The distance between two clusters used in the agglomerative process is equal to the minimum of distances between two scattered representatives. Therefore, CURE takes a middle-ground approach between the graph (all-points) methods and the geometric (one centroid) methods. Single and average link closeness is replaced by representatives' aggregate closeness. Selecting representatives scattered around a cluster makes it possible to cover non-spherical shapes. As before, agglomeration continues until requested number *k* of clusters is achieved. CURE employs one additional device: originally selected scattered points are shrunk to the geometric centroid of the cluster by user-specified factor α. Shrinkage suppresses the affect of the outliers since outliers happen to be located further from the cluster centroid than the other scattered representatives. CURE is capable of finding clusters of different shapes and sizes, and it is insensitive to outliers. Since CURE uses sampling, estimation of its complexity is not straightforward. For low-dimensional data authors provide a complexity estimate of $O(N_{sample}^2)$ defined in terms of sample size.

More exact bounds depend on input parameters: shrink factor α, number of representative points *c*, number of partitions, and sample size. Figure 1 illustrates agglomeration in Cure. Three clusters, each with three representatives, are shown before and after the merge and shrinkage. Two closest representatives are connected by arrow.

While the algorithm CURE works with numerical attributes (particularly low dimensional spatial data), the algorithm ROCK developed by the same researchers [Guha et al. 1999] targets hierarchical agglomerative clustering for categorical attributes. It is surveyed in the section *Co-Occurrence of Categorical Data*.

The hierarchical agglomerative algorithm CHAMELEON [Karypis et al. 1999a] utilizes dynamic modeling in cluster aggregation. It uses the connectivity graph G corresponding to the K-nearest neighbor model sparsification of the connectivity matrix: the edges of K most similar points to any given point are preserved, the rest are pruned. CHAMELEON has two stages. In the first stage small tight clusters are built to ignite the second stage. This involves a graph partitioning [Karypis & Kumar 1999]. In the second stage agglomerative process is performed. It utilizes measures of relative inter-connectivity $RI(C_i,C_j)$ and relative closeness $RC(C_i,C_j)$; both are locally normalized by quantities related to clusters C_i,C_j. In this sense the modeling is *dynamic*. Normalization involves certain non-obvious graph operations [Karypis & Kumar 1999]. CHAMELEON strongly relies on graph partitioning implemented in the library HMETIS (see the section *Co-Occurrence of Categorical Data*). Agglomerative process depends on user provided thresholds. A decision to merge is made based on the combination

$$RI(C_i,C_j)\cdot RC(C_i,C_j)^{\alpha}$$

of local relative measures. The algorithm does not depend on assumptions about the data model. This algorithm is proven to find clusters of different shapes, densities, and sizes in 2D (two-dimensional) space. It has a complexity of $O(Nm+N\log(N)+m^2\log(m))$, where m is number of sub-clusters built during first initialization phase. Figure 2 (analogous to the one in [Karypis & Kumar 1999]) presents a choice of four clusters (a)-(d) for a merge. While Cure would merge clusters (a) and (b), CHAMELEON makes intuitively better choice of merging (c) and (d).

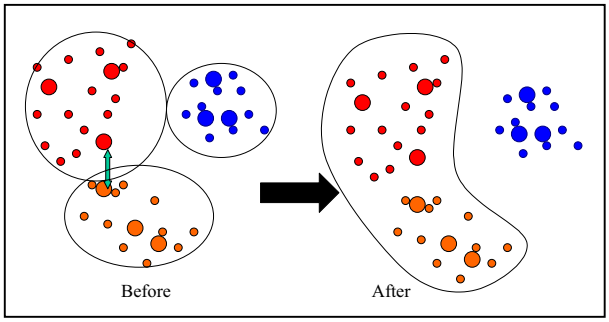


Figure 1: Agglomeration in Cure.

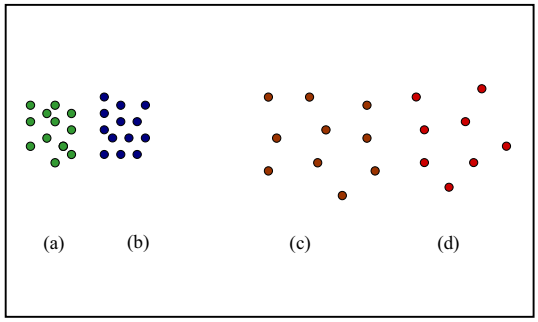


Figure 2: CHAMELEON merges (c) and (d).

2.3. Binary Divisive Partitioning

In linguistics, information retrieval, and document clustering applications binary taxonomies are very useful. Linear algebra methods, based on *singular value decomposition* (SVD) are used for this purpose in collaborative filtering and information retrieval [Berry & Browne 1999]. SVD application to hierarchical divisive clustering of document collections resulted in the PDDP (Principal Direction Divisive Partitioning)

algorithm [Boley 1998]. In our notations, object x is a document, l^{th} attribute corresponds to a word (*index term*), and matrix entry x_{il} is a measure (as TF-IDF) of l-term frequency in a document x. PDDP constructs SVD decomposition of the matrix

$$C = (X - e\bar{x}), \quad \bar{x} = \frac{1}{N} \sum_{i=1:N} x_i, \quad e = (1,...1)^T \in R^d .$$

This algorithm bisects data in Euclidean space by a hyperplane that passes through data centroid orthogonally to eigenvector with the largest singular value. The k-way splitting is also possible if the k largest singular values are considered. Bisecting is a good way to categorize documents and it results in a binary tree. When k-means (2-means) is used for bisecting, the dividing hyperplane is orthogonal to a line connecting two centroids. The comparative study of both approaches [Savaresi & Boley 2001] can be used for further references. Hierarchical divisive bisecting k-means was proven [Steinbach et al. 2000] to be preferable for document clustering.

While PDDP or 2-means are concerned with how to split a cluster, the problem of which cluster to split is also important. Casual strategies are: (1) split each node at a given level, (2) split the cluster with highest cardinality, and, (3) split the cluster with the largest intra-cluster variance. All three strategies have problems. For analysis regarding this subject and better alternatives, see [Savaresi et al. 2002].

2.4. Other Developments

Ward's method [Ward 1963] implements agglomerative clustering based not on linkage metric, but on an objective function used in k-means (sub-section *K-Means Methods*). The merger decision is viewed in terms of its effect on the objective function.

The popular hierarchical clustering algorithm for categorical data COBWEB [Fisher 1987] has two very important qualities. First, it utilizes ***incremental*** learning. Instead of following divisive or agglomerative approaches, it dynamically builds a dendrogram by processing one data point at a time. Second, COBWEB belongs to ***conceptual*** or ***model-based*** learning. This means that each cluster is considered as a model that can be described intrinsically, rather than as a collection of points assigned to it. COBWEB's dendrogram is called a *classification tree.* Each tree node C, a cluster, is associated with the conditional probabilities for categorical attribute-values pairs,

$$\Pr(x_l = v_{lp} \mid C), l = 1:d, p = 1:|A_l|.$$

This easily can be recognized as a C-specific Naïve Bayes classifier. During the classification tree construction, every new point is descended along the tree and the tree is potentially updated (by an insert/split/merge/create operation). Decisions are based on an analysis of a ***category utility*** [Corter & Gluck 1992]

$$CU\{C_1,...,C_k\} = \left(\sum_{j=1:k} CU(C_j)\right)/k,$$

$$CU(C_j) = \sum_{l,p} (\Pr(x_l = v_{lp} \mid C_j)^2 - (\Pr(x_l = v_{lp})^2)$$

similar to GINI index. It rewards clusters C_j for increases in predictability of the categorical attribute values v_{lp}. Being incremental, COBWEB is fast with a complexity

of $O(tN)$, though it depends non-linearly on tree characteristics packed into a constant *t*. There is the similar incremental hierarchical algorithm for all numerical attributes called CLASSIT [Gennari et al. 1989]. CLASSIT associates normal distributions with cluster nodes. Both algorithms can result in highly unbalanced trees.

Chiu et al. [2001] proposed another *conceptual* or *model-based* approach to hierarchical clustering. This development contains several different useful features, such as the extension of BIRCH-like preprocessing to categorical attributes, outliers handling, and a two-step strategy for monitoring the number of clusters including BIC (defined below). The model associated with a cluster covers both numerical and categorical attributes and constitutes a blend of Gaussian and multinomial models. Denote corresponding multivariate parameters by θ. With every cluster *C*, we associate a logarithm of its (classification) likelihood

$$l_C = \sum_{x_i \in C} \log p(x_i \mid \theta)$$

The algorithm uses *maximum likelihood estimates* for parameter θ. The distance between two clusters is defined (instead of linkage metric) as a decrease in log-likelihood

$$d(C_1, C_2) = l_{C_1} + l_{C_2} - l_{C_1 \cup C_2}$$

caused by merging of the two clusters under consideration. The agglomerative process continues until the stopping criterion is satisfied. As such, determination of the best *k* is automatic. This algorithm has the commercial implementation (in SPSS Clementine). The complexity of the algorithm is linear in *N* for the summarization phase.

Traditional hierarchical clustering is inflexible due to its greedy approach: after a merge or a split is selected it is not refined. Though COBWEB does reconsider its decisions, it is so inexpensive that the resulting classification tree can also have sub-par quality. Fisher [1996] studied iterative hierarchical cluster redistribution to improve once constructed dendrogram. Karypis et al. [1999b] also researched refinement for hierarchical clustering. In particular, they brought attention to a relation of such a refinement to a well-studied refinement of *k*-way graph partitioning [Kernighan & Lin 1970].

For references related to parallel implementation of hierarchical clustering see [Olson 1995].

3. Partitioning Relocation Clustering

In this section we survey data partitioning algorithms, which divide data into several subsets. Because checking all possible subset systems is computationally infeasible, certain greedy heuristics are used in the form of ***iterative optimization***. Specifically, this means different ***relocation*** schemes that iteratively reassign points between the *k* clusters. Unlike traditional hierarchical methods, in which clusters are not revisited after being constructed, relocation algorithms gradually improve clusters. With appropriate data, this results in high quality clusters.

One approach to data partitioning is to take a *conceptual* point of view that identifies the cluster with a certain model whose unknown parameters have to be found. More

specifically, ***probabilistic*** models assume that the data comes from a mixture of several populations whose distributions and priors we want to find. Corresponding algorithms are described in the sub-section *Probabilistic Clustering.* One clear advantage of probabilistic methods is the interpretability of the constructed clusters. Having concise cluster representation also allows inexpensive computation of intra-clusters measures of fit that give rise to a global *objective function* (see log-likelihood below).

Another approach starts with the definition of ***objective function*** depending on a partition. As we have seen (sub-section *Linkage Metrics*), pair-wise distances or similarities can be used to compute measures of iter- and intra-cluster relations. In iterative improvements such pair-wise computations would be too expensive. Using unique cluster representatives resolves the problem: now computation of objective function becomes linear in N (and in a number of clusters $k << N$). Depending on how representatives are constructed, iterative optimization partitioning algorithms are subdivided into ***k*-medoids** and ***k*-means** methods. *K*-medoid is the most appropriate data point within a cluster that represents it. Representation by *k*-medoids has two advantages. First, it presents no limitations on attributes types, and, second, the choice of medoids is dictated by the location of a predominant fraction of points inside a cluster and, therefore, it is lesser sensitive to the presence of outliers. In *k*-means case a cluster is represented by its centroid, which is a mean (usually weighted average) of points within a cluster. This works conveniently only with numerical attributes and can be negatively affected by a single outlier. On the other hand, centroids have the advantage of clear geometric and statistical meaning. The corresponding algorithms are reviewed in the sub-sections *K-Medoids Methods* and *K-Means Methods.*

3.1. Probabilistic Clustering

In the ***probabilistic*** approach, data is considered to be a sample independently drawn from a ***mixture model*** of several probability distributions [McLachlan & Basford 1988]. The main assumption is that data points are generated by, first, randomly picking a model j with probability τ_j, $j = 1:k$, and, second, by drawing a point x from a corresponding distribution. The area around the mean of each (supposedly unimodal) distribution constitutes a natural cluster. So we associate the cluster with the corresponding distribution's parameters such as mean, variance, etc. Each data point carries not only its (observable) attributes, but also a (hidden) cluster ID (*class* in pattern recognition). Each point x is assumed to belong to one and only one cluster, and we can *estimate* the probabilities of the assignment $\Pr(C_j \mid x)$ to j^{th} model. The overall ***likelihood*** of the training data is its probability to be drawn from a given mixture model

$$L(X \mid C) = \prod_{i=1:N} \sum_{j=1:k} \tau_j \Pr(x_i \mid C_j)$$

Log-likelihood $\log(L(X \mid C))$ serves as an *objective function*, which gives rise to the ***Expectation-Maximization*** (EM) method. For a quick introduction to EM, see [Mitchell 1997]. Detailed descriptions and numerous references regarding this topic can be found in [Dempster et al. 1977; McLachlan & Krishnan 1997]. EM is a two-step iterative optimization. Step (E) estimates probabilities $\Pr(x \mid C_j)$, which is equivalent to a soft

(fuzzy) reassignment. Step (M) finds an approximation to a mixture model, given current soft assignments. This boils down to finding mixture model parameters that maximize log-likelihood. The process continues until log-likelihood convergence is achieved.

Restarting and other tricks are used to facilitate finding better local optimum. Moore [1999] suggested acceleration of EM method based on a special data index, KD-tree. Data is divided at each node into two descendents by splitting the widest attribute at the center of its range. Each node stores sufficient statistics (including covariance matrix) similar to BIRCH. Approximate computing over a pruned tree accelerates EM iterations.

Probabilistic clustering has some important features:

- It can be modified to handle recodes of complex structure
- It can be stopped and resumed with consecutive batches of data, since clusters have representation totally different from sets of points
- At any stage of iterative process the intermediate mixture model can be used to assign cases (on-line property)
- It results in easily interpretable cluster system

Because the mixture model has clear probabilistic foundation, the determination of the most suitable number of clusters *k* becomes a more tractable task. From a data mining perspective, excessive parameter set causes overfitting, while from a probabilistic perspective, number of parameters can be addressed within the Bayesian framework. See the sub-section "*How Many Clusters?*" for more details including terms MML and BIC used in the next paragraph.

The algorithm SNOB [Wallace & Dowe 1994] uses a mixture model in conjunction with the MML principle. Algorithm AUTOCLASS [Cheeseman & Stutz 1996] utilizes a mixture model and covers a broad variety of distributions, including Bernoulli, Poisson, Gaussian, and log-normal distributions. Beyond fitting a particular fixed mixture model, AUTOCLASS extends the search to different models and different *k*. To do this AUTOCLASS heavily relies on Bayesian methodology, in which a model complexity is reflected through certain coefficients (priors) in the expression for the likelihood previously dependent only on parameters' values. This algorithm has a history of industrial usage. The algorithm MCLUST [Fraley & Raftery 1999] is a software package (commercially linked with S-PLUS) for hierarchical, mixture model clustering, and discriminant analysis using BIC for estimation of goodness of fit. MCLUST uses Gaussian models with ellipsoids of different volumes, shapes, and orientations.

An important property of probabilistic clustering is that mixture model can be naturally generalized to clustering *heterogeneous* data. This is important in practice, where an individual (data object) has multivariate static data (demographics) in combination with variable length dynamic data (customer profile) [Smyth 1999]. The dynamic data can consist of finite sequences subject to a first-order Markov model with a transition matrix dependent on a cluster. This framework also covers data objects consisting of *several* sequences, where number n_i of sequences per x_i is subject to geometric distribution [Cadez et al. 2000]. To emulate sessions of different lengths, finite-state Markov model (transitional probabilities between Web site pages) has to be augmented with a special "end" state. Cadez et al. [2001] used mixture model for customer profiling based on transactional information.

Model-based clustering is also used in a hierarchical framework: COBWEB, CLASSIT and development by Chiu et al. [2001] were already presented above. Another early example of conceptual clustering is algorithm CLUSTER/2 [Michalski & Stepp 1983].

3.2. *K*-Medoids Methods

In *k*-medoids methods a cluster is represented by one of its points. We have already mentioned that this is an easy solution since it covers any attribute types and that medoids have embedded resistance against outliers since peripheral cluster points do not affect them. When medoids are selected, clusters are defined as subsets of points close to respective medoids, and the objective function is defined as the averaged distance or another dissimilarity measure between a point and its medoid.

Two early versions of *k*-medoid methods are the algorithm PAM (Partitioning Around Medoids) and the algorithm CLARA (Clustering LARge Applications) [Kaufman & Rousseeuw 1990]. PAM is iterative optimization that combines relocation of points between perspective clusters with re-nominating the points as potential medoids. The guiding principle for the process is the effect on an objective function, which, obviously, is a costly strategy. CLARA uses several (five) samples, each with 40+2*k* points, which are each subjected to PAM. The whole dataset is assigned to resulting medoids, the objective function is computed, and the best system of medoids is retained.

Further progress is associated with Ng & Han [1994] who introduced the algorithm CLARANS (Clustering Large Applications based upon RANdomized Search) in the context of clustering in *spatial* databases. Authors considered a graph whose nodes are the sets of *k* medoids and an edge connects two nodes if they differ by exactly one medoid. While CLARA compares very few neighbors corresponding to a fixed small sample, CLARANS uses random search to generate neighbors by starting with an arbitrary node and randomly checking *maxneighbor* neighbors. If a neighbor represents a better partition, the process continues with this new node. Otherwise a local minimum is found, and the algorithm restarts until *numlocal* local minima are found (value *numlocal*=2 is recommended). The best node (set of medoids) is returned for the formation of a resulting partition. The complexity of CLARANS is $O(N^2)$ in terms of number of points. Ester et al. [1995] extended CLARANS to spatial VLDB. They used R*-trees [Beckmann 1990] to relax the original requirement that all the data resides in core memory, which allowed *focusing* exploration on the relevant part of the database that resides at a branch of the whole data tree.

3.3. *K*-Means Methods

The ***k-means*** algorithm [Hartigan 1975; Hartigan & Wong 1979] is by far the most popular clustering tool used in scientific and industrial applications. The name comes from representing each of *k* clusters C_j by the mean (or weighted average) c_j of its points, the so-called *centroid*. While this obviously does not work well with categorical attributes, it has the good geometric and statistical sense for numerical attributes. The sum of discrepancies between a point and its centroid expressed through appropriate distance is used as the objective function. For example, the L_2-norm based objective

function, the sum of the squares of errors between the points and the corresponding centroids, is equal to the total intra-cluster variance

$$E(C) = \sum_{j=1:k} \sum_{x_i \in C_j} \left\| x_i - c_j \right\|^2 .$$

The sum of the squares of errors can be rationalized as (a negative of) log-likelihood for normally distributed mixture model and is widely used in statistics (SSE). Therefore, *k*-means algorithm can be derived from general probabilistic framework (see sub-section *Probabilistic Clustering*) [Mitchell 1997]. Note that only means are estimated. A simple modification would normalize individual errors by cluster radii (cluster standard deviation), which makes a lot of sense when clusters have different dispersions. An objective function based on L_2 -norm has many unique algebraic properties. For example, it coincides with pair-wise errors

$$E'(C) = \frac{1}{2} \sum_{j=1:k} \sum_{x_i, y_i \in C_j} \left\| x_i - y_i \right\|^2 ,$$

and with the difference between the total data variance and the inter-cluster variance. Therefore, the cluster separation is achieved simultaneously with the cluster tightness.

Two versions of *k*-means iterative optimization are known. The first version is similar to EM algorithm and consists of two-step major iterations that (1) reassign all the points to their nearest centroids, and (2) recompute centroids of newly assembled groups. Iterations continue until a stopping criterion is achieved (for example, no reassignments happen). This version is known as Forgy's algorithm [Forgy 1965] and has many advantages:

- It easily works with any L_p -norm
- It allows straightforward parallelization [Dhillon & Modha 1999]
- It is insensitive with respect to data ordering.

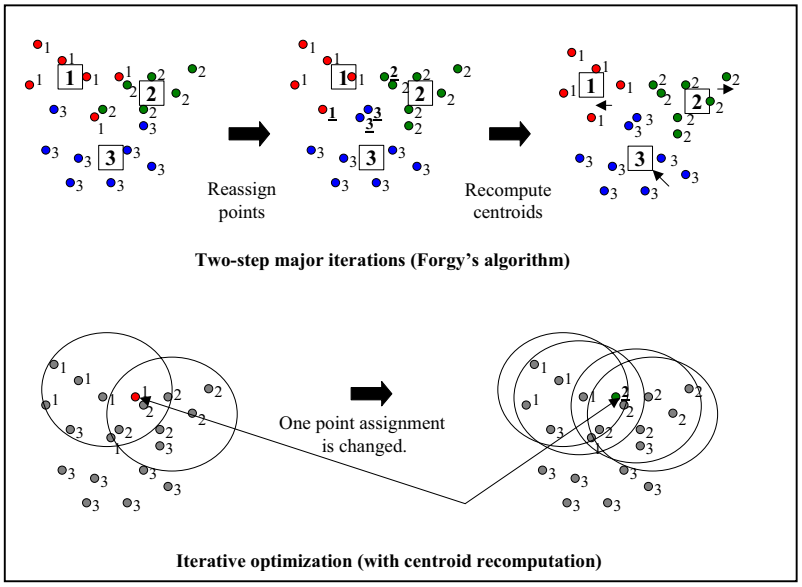
The second (classic in iterative optimization) version of *k*-means iterative optimization reassigns points based on more detailed analysis of effects on the objective function caused by moving a point from its current cluster to a potentially new one. If a move has a positive effect, the point is relocated and the two centroids are recomputed. It is not clear that this version is computationally feasible, because the outlined analysis requires an inner loop over all member points of involved clusters affected by centroids shifts. However, in L_2 case it is known [Duda & Hart 1973; Berkhin & Becher 2002] that all computations can be algebraically reduced to simply computing a single distance! Therefore, in this case both versions have the same computational complexity.

There is experimental evidence that compared with Forgy's algorithm, the second (classic) version frequently yields better results [Larsen & Aone 1999; Steinbach et al. 2000]. In particular, Dhillon et al. [2002] noticed that a Forgy's *spherical k*-means (using cosine similarity instead of Euclidean distance) has a tendency to get stuck when applied to document collections. They noticed that a version reassigning points and immediately recomputing centroids works much better. Figure 3 illustrates both implementations.

Besides these two versions, there have been other attempts to find minimum of *k*-means objective function. For example, the early algorithm ISODATA [Ball & Hall 1965] used merges and splits of intermediate clusters.

The wide popularity of *k*-means algorithm is well deserved. It is simple, straightforward, and is based on the firm foundation of analysis of variances. The *k*-means algorithm also suffers from all the usual suspects:

- The result strongly depends on the initial guess of centroids (or assignments)
- Computed local optimum is known to be a far cry from the global one
- It is not obvious what is a good *k* to use
- The process is sensitive with respect to outliers
- The algorithm lacks scalability
- Only numerical attributes are covered
- Resulting clusters can be unbalanced (in Forgy's version, even empty)



A simple way to mitigate the affects of clusters initialization was suggested by Bradley & Fayyad [1998]. First, *k*-means is performed on several small samples of data with a random initial guess. Each of these constructed systems is then used as a potential initialization for a union of all the samples. Centroids of the best system constructed this way are suggested as an intelligent initial guesses to ignite the *k*-means algorithm on the full data. Another interesting attempt [Babu & Murty 1993] is based on GA (see below). No initialization actually guarantees global minimum for *k*-means. As is common to any combinatorial optimization, a logical attempt to cure this problem is to use simulated annealing [Brown & Huntley 1991]. Zhang [2001] suggested another way to rectify optimization process by soft assignment of points to different clusters with appropriate weights (as EM does), rather than by moving them decisively from one cluster to another. The weights take into account how well a point fits into recipient clusters. This process involves so-called *harmonic means*.

We discuss scalability issues in the section *Scalability and VLDB Extensions*. For a comprehensive approach in relation to *k*-means see an excellent study [Bradley et al. 1998]. A generic method to achieve scalability is to preprocess or *squash* the data. Such preprocessing usually also takes care of outliers. Preprocessing has its drawbacks. It results in approximations that sometimes negatively affect final cluster quality. Pelleg & Moore [1999] suggested how to directly (without any squashing) accelerate *k*-means iterative process by utilizing KD-trees [Moore 1999]. The algorithm *X*-means [Pelleg &

Moore 2000] goes a step further: in addition to accelerating the iterative process it tries to incorporate a search for the best k in the process itself. While more comprehensive criteria discussed in the sub-section "*How Many Clusters?"* require running independent k-means and then comparing the results (costly experementation), X-means tries to split a part of already constructed cluster based on outcome of BIC criterion. This gives a much better initial guess for the next iteration and covers a user specified range of admissible k.

The tremendous popularity of k-means algorithm has brought to life many other extensions and modifications. Mahalanobis distance can be used to cover hyper-ellipsoidal clusters [Mao & Jain 1996]. Maximum of intra-cluster variances, instead of the sum, can serve as an objective function [Gonzales 1985]. Generalizations that incorporate categorical attributes are known. Sometimes the term *k-prototypes* is used in this context [Huang 1998]. Modifications which constructs clusters of balanced size are discussed in the sub-section *Constrained-Based Clustering*.

4. Density-Based Partitioning

An open set in the Euclidean space can be divided into a set of its connected components. The implementation of this idea for partitioning of a finite set of points requires concepts of density, connectivity and boundary. They are closely related to a point's nearest neighbors. A cluster, defined as a connected dense component, grows in any direction that density leads. Therefore, density-based algorithms are capable of discovering clusters of arbitrary shapes. Also this provides a natural protection against outliers. Figure 4 illustrates some cluster shapes that present a problem for partitioning relocation clustering (e.g., k-means), but are handled properly by density-based algorithms. They also have good scalability. These outstanding properties are tempered with certain inconveniencies.

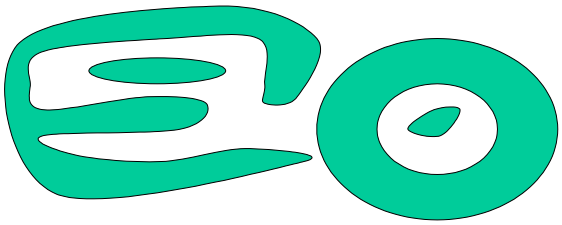


Figure 4. Irregular shapes difficult for k-means are

From a very general data description point of view, a single dense cluster consisting of two adjacent areas with significantly different densities (both higher than a threshold) is not very informative. Another drawback is a lack of interpretability. An excellent introduction to density-based methods is contained in the textbook [Han & Kamber 2001].

Since density-based algorithms require a metric space, the natural setting for them is ***spatial*** data clustering [Han et al. 2001; Kolatch 2001]. To make computations feasible, some index of data is constructed (such as R*-tree). This is a topic of active research. Classic indices were effective only with reasonably low-dimensional data. The algorithm DENCLUE that, in fact, is a blend of a density-based clustering and a grid-based preprocessing is lesser affected by data dimensionality.

There are two major approaches for density-based methods. The first approach pins density to a training data point and is reviewed in the sub-section *Density-Based Connectivity*. Representative algorithms include DBSCAN, GDBSCAN, OPTICS, and

DBCLASD. The second approach pins density to a point in the attribute space and is explained in the sub-section *Density Functions*. It includes the algorithm DENCLUE.

4.1. Density-Based Connectivity

Crucial concepts of this section are ***density*** and ***connectivity*** both measured in terms of local distribution of nearest neighbors.

The algorithm DBSCAN (Density Based Spatial Clustering of Applications with Noise) [Ester et al. 1996] targeting low-dimensional spatial data is the major representative in this category. Two input parameters ε and *MinPts* are used to define:

1) An ***ε-neighborhood*** $N_\varepsilon(x) = \{y \in X \mid d(x,y) \le \varepsilon\}$ of the point x,
2) A ***core object*** (a point with a neighborhood consisting of more than *MinPts* points)
3) A concept of a point y ***density-reachable*** from a core object x (a finite sequence of core objects between x and y exists such that each next belongs to an ε-neighborhood of its predecessor)
4) A ***density-connectivity*** of two points *x, y* (they should be density-reachable from a common core object).

So defined density-connectivity is a symmetric relation and all the points reachable from core objects can be factorized into maximal connected components serving as clusters. The points that are not connected to any core point are declared to be outliers (they are not covered by any cluster). The non-core points inside a cluster represent its *boundary*. Finally, core objects are *internal* points. Processing is independent of data ordering. So far, nothing requires any limitations on the dimension or attribute types. Obviously, an effective computing of ε-neighborhoods presents a problem. However, in the case of low-dimensional ***spatial*** data, different effective indexation schemes exist (meaning $O(\log(N))$ rather than $O(N)$ fetches per search). DBSCAN relies on R*-tree indexation [Kriegel et al. 1990]. Therefore, on low-dimensional spatial data theoretical complexity of DBSCAN is $O(N\log(N))$. Experiments confirm slight super-linear runtime.

Notice that DBSCAN relies on $\mathring{U}$-neighborhoods and on frequency count within such neighborhoods to define a concept of a core object. Many spatial databases contain extended objects such as polygons instead of points. Any reflexive and symmetric predicate (for example, two polygons have a non-empty intersection) suffice to define a "neighborhood". Additional measures (as intensity of a point) can be used instead of a simple count as well. These two generalizations lead to the algorithm GDBSCAN [Sander et al. 1998], which uses the same two parameters as algorithm DBSCAN.

With regard to these two parameters ε and *MinPts*, there is no straightforward way to fit them to data. Moreover, different parts of data could require different parameters – the problem discussed earlier in conjunction with CHAMELEON. The algorithm OPTICS (Ordering Points To Identify the Clustering Structure) [Ankerst et al. 1999] adjusts DBSCAN to this challenge. It builds an augmented ordering of data which is consistent with DBSCAN, but goes a step further: keeping the same two parameters ε, *MinPts*, OPTICS covers a spectrum of all different $\varepsilon' \le \varepsilon$. The constructed ordering can be used automatically or interactively. With each point, OPTICS stores only two additional fields,

the so-called core- and reachability-distances. For example, the core-distance is the distance to *MinPts*' nearest neighbor when it does not exceeds ε, or undefined otherwise. Experimentally, OPTICS exhibits runtime roughly equal to 1.6 of DBSCAN runtime.

While OPTICS can be considered as a DBSCAN extension in direction of different local densities, a more mathematically sound approach is to consider a random variable equal to the distance from a point to its nearest neighbor, and to learn its probability distribution. Instead of relying on user-defined parameters, a possible conjuncture is that each cluster has its own typical distance-to-nearest-neighbor scale. The goal is to discover such scales. Such *nonparametric* approach is implemented in the algorithm DBCLASD (Distribution Based Clustering of Large Spatial Databases) [Xu et al. 1998]. Assuming that points inside each cluster are uniformly distributed which may or may not be realistic, DBSCLAD defines a cluster as a non-empty arbitrary shape subset in X that has the expected distribution of distance to the nearest neighbor with a required confidence, and is the *maximal connected* set with this quality. This algorithm handles spatial data (minefield example is used). χ^2 -test is used to check distribution requirement (standard consequence is a requirement for each cluster to have at least 30 points). Regarding connectivity, DBCLASD relies on *grid-based* approach to generate cluster-approximating polygons. The algorithm contains devices for handling real databases with noise and implements *incremental* unsupervised learning. Two venues are used. First, assignments are not final: points can change cluster membership. Second, certain points (noise) are not assigned, but are tried later. Therefore, once incrementally fetched points can be revisited internally. DBCLASD is known to run faster than CLARANS by a factor of 60 on some examples. In comparison with much more efficient DBSCAN, it can be 2-3 times slower. However, DBCLASD requires no user input, while empirical search for appropriate parameter requires several DBSCAN runs. In addition, DBCLASD discovers clusters of different densities.

4.2. Density Functions

Hinneburg & Keim [1998] shifted the emphasis from computing densities pinned to data points to computing density functions defined over the underlying attribute space. They proposed the algorithm DENCLUE (DENsity-based CLUstEring). Along with DBCLASD, it has a firm mathematical foundation. DENCLUE uses a ***density function***

$$f^D(x) = \mathop{\text{—}}_{y \in D} f(x, y)$$

that is the superposition of several ***influence functions***. When the f-term depends on x - y, the formula can be recognized as a convolution with a kernel. Examples include a *square wave function* $f(x,y) = \theta(\|x-y\|/\sigma)$ equal to 1, if distance between x and y is less than or equal to σ, and a Gaussian influence function $f(x,y) = e^{-\|x-y\|^2/2\sigma^2}$. This provides a high level of generality: the first example leads to DBSCAN, the second one to k-means clusters! Both examples depend on parameter σ. Restricting the summation to $D = \{y : \|x-y\| < k\sigma\} \subset X$ enables a practical implementation. DENCLUE concentrates on local maxima of density functions called *density-attractors* and uses a flavor of gradient hill-climbing technique for finding them. In addition to *center-defined* clusters,

arbitrary-shape clusters are defined as continuations along sequences of points whose local densities are no less than prescribed threshold ξ. The algorithm is stable with respect to outliers and authors show how to choose parameters σ and ξ. DENCLUE scales well, since at its initial stage it builds a *map* of hyper-rectangle cubes with edge length 2σ. For this reason, the algorithm can be classified as a *grid-based* method. Applications include high dimensional multimedia and molecular biology data. While no clustering algorithm could have less than $O(N)$ complexity, the runtime of DENCLUE scales with N sub-linearly! The explanation is that though all the points are fetched, the bulk of analysis (in clustering stage) involves only points in highly populated areas.

5. Grid-Based Methods

In the previous section crucial concepts of density, connectivity, and boundary were used which required elaborate definitions. Another way of dealing with them is to inherit the topology from the underlying attribute space. To limit the search combinations, multi-rectangular segments are considered. Recall that a ***segment*** (also *cube, cell, region*). is a direct Cartesian product of individual attribute sub-ranges (contiguous in case of numerical attributes). Since some binning is usually adopted for numerical attributes, methods partitioning space are frequently called grid-based methods. The elementary segment corresponding to single-bin or single-value sub-ranges is called a ***unit***.

Overall, we shift our attention from data to space partitioning. Data partitioning is induced by points' membership in segments resulted from space partitioning, while space partitioning is based on grid-characteristics accumulated from input data. One advantage of this indirect handling (data grid-data space-partitioning data-partitioning) is that accumulation of grid-data makes grid-based clustering techniques independent of data ordering. In contrast, relocation methods and all incremental algorithms are very sensitive with respect to data ordering. While density-based partitioning methods work best with numerical attributes, grid-based methods work with attributes of different types.

To some extent, the grid-based methodology reflects a technical point of view. The category is eclectic: it contains both partitioning and hierarchical algorithms. The algorithm DENCLUE from the previous section uses grids at its initial stage. The very important grid-based algorithm CLIQUE and its descendent, algorithm MAFIA, are presented in the section *Clustering High Dimensional Data*. In this section we survey algorithms that use grid-based technique as their major principle instrument.

BANG-clustering [Schikuta & Erhart 1997] improves the similar hierarchical algorithm GRIDCLUST [Schikuta 1996]. Grid-based segments are used to summarize data. The segments are stored in a special BANG-structure that is a grid-directory incorporating different scales. Adjacent segments are neighbors. If a common face has maximum dimension they are called nearest neighbors. More generally, neighbors of degree between 0 and d-1 can be defined. The density of a segment is defined as a ratio between number of points in it and its volume. From the grid-directory, a dendrogram is directly calculated.

The algorithm STING (STatistical INformation Grid-based method) [Wang et al. 97] works with numerical attributes (spatial data) and is designed to facilitate "region

oriented" queries. In doing so, STING constructs data summaries in a way similar to BIRCH. It, however, assembles statistics in a hierarchical tree of nodes that are grid-cells. Figure 5 presents the proliferation of cells in 2-dimensional space and the construction of the corresponding tree. Each cell has four (default) children and stores a point count, and attribute-dependent measures: mean, standard deviation, minimum, maximum, and distribution type. Measures are accumulated starting from bottom level cells, and further propagate to higher-level cells (e.g., minimum is equal to a minimum among the children-minimums). Only distribution type presents a problem – χ^2-test is used after bottom cell distribution types are handpicked. When the cell-tree is constructed (in $O(N)$ time), certain cells are identified and connected in clusters similar to DBSCAN. If the number of leaves is K, the cluster construction phase depends on K and not on N. This algorithm has a simple structure suitable for parallelization and allows for multi-resolution, though defining appropriate granularity is not straightforward. STING has been further enhanced to algorithm STING+ [Wang et al. 1999] that targets dynamically evolving spatial databases, and uses similar hierarchical cell organization as its predecessor. In addition, STING+ enables *active* data mining.

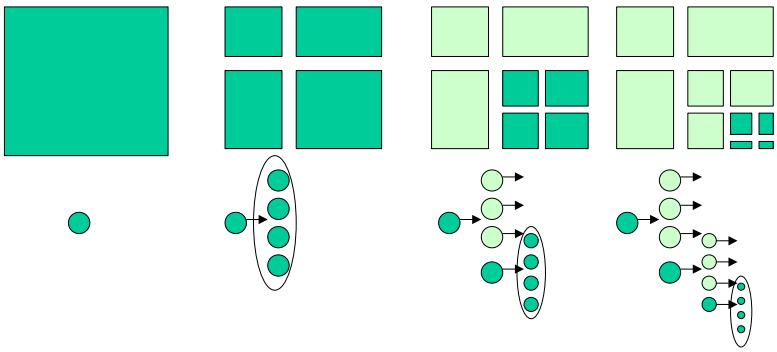


Figure 5. Cell generation and tree construction in STING.

To do so, it supports user defined *trigger conditions* (e.g., there is a region where at least 10 cellular phones are in use per square mile with total area of at least 10 square miles, or usage drops by 20% in a described region). The related measures, sub-triggers, are stored and updated over the hierarchical cell tree. They are suspended until the trigger *fires* with user-defined action. Four types of conditions are supported: absolute and relative conditions on regions (a set of adjacent cells), absolute and relative conditions on certain attributes.

The algorithm WaveCluster [Sheikholeslami et al. 1998] works with numerical attributes and has an advanced multi-resolution. It is also known for other outstanding properties:

- High quality of clusters
- Ability to work well in relatively high dimensional spatial data
- Successful handling of outliers
- $O(N)$ complexity

WaveCluster is based on ideas of signal processing. It applies wavelet transforms to filter the data. Notice that high-frequency parts of a signal correspond to boundaries, while low

frequency high amplitude parts of a signal correspond to clusters' interiors. Wavelet transform provides us with useful filters. For example, hat-shape filter forces dense areas to serve as attractors and simultaneously suppresses lesser dense boundary areas. After getting back from signal to attribute space this makes clusters more sharp and eliminates outliers. WaveCluster goes in stages. It:

1) Bins every dimension and assigns points to corresponding units
2) Applies discrete Wavelet transform to so accumulated units
3) Finds connected components (clusters) in a transformed attribute space (corresponding to a certain level of resolution)
4) Assigns points

The algorithm's complexity is $O(N)$ for low dimensions, but exponentially grows with the dimension.

The hierarchy of grids allows definition of the *Hausdorff Fractal Dimension* (HFD) [Schalkoff 1991]. HFD of a set is the negative slope of a log-log plot of the number of cells $Cell(r)$ (occupied by a set) as a function of a grid size r. A fast algorithm (*box counting*) to compute HFD was introduced in [Liebovitch & Toth 1989]. The concept of HFD is fundamental to the FC (Fractal Clustering) algorithm [Barbara & Chen 2000] for numeric attributes, which works with several layers of grids (cardinality of each dimension is increased 4 times with each next layer). Although only occupied cells are kept to save memory, memory usage is still a significant problem. FC starts with initializing of k clusters. Initialization threshold and a data sample are used at this stage to come up with the appropriate k. Then FC scans full data incrementally. It tries to add an incoming point to each cluster that results in certain increase of HFD. If the smallest increase exceeds a threshold τ, a point is declared an outlier; otherwise a point is assigned so that HFD would be minimally impacted. The FC algorithm has few appealing properties:

- Incremental structure (batches of data are fetched into core memory)
- Suspendable nature always ready for on-line assignments
- Ability to discover clusters of irregular shapes
- $O(N)$ complexity

It also has a few problems:

- Data order dependency
- Strong dependency on clusters initialization
- Dependency on parameters (threshold used in initialization, and τ)

6. Co-Occurrence of Categorical Data

In this section we talk about categorical data, which frequently relates to the concept of a variable size ***transaction*** that is a finite set of elements called ***items*** from a common item universe. For example, market basket data has this form. Every transaction can be presented in a point-by-attribute format, by enumerating all items j, and by associating with a transaction the binary attributes that indicate whether j-items belong to a transaction or not. Such representation is sparse and two random transactions have very few items in common. This is why similarity (sub-section *Proximity Measures*) between

them is usually measured by Jaccard coefficient $sim(T_1, T_2) = |T_1 \cap T_2| / |T_1 \cup T_2|$. Common to this and others examples of point-by-attribute format for categorical data, is high dimensionality, significant amount of zero values, and small number of common values between two objects. Conventional clustering methods, based on similarity measures, do not work well. Since categorical/transactional data is important in customer profiling, assortment planning, Web analysis, and other applications, different clustering methods founded on the idea of ***co-occurrence*** of categorical data have been developed.

The algorithm ROCK (Robust Clustering algorithm for Categorical Data) [Guha et al. 1999] deals with categorical data and has many common features with the algorithm CURE (section *Hierarchical Clustering*): (1) it is a hierarchical clustering, (2) agglomeration continues until specified number k of clusters is constructed, and (3) it uses data sampling in the same way as CURE does. ROCK defines a neighbor of a point x as a point y such that $sim(x, y) \geq \theta$ for some threshold θ, and proceeds to a definition of links $link(x,y)$ between two points x, y equal to number of their common neighbors. Clusters consist of points with a high degree of connectivity – pair-wise points inside a cluster have on average a high number of links. ROCK utilizes the objective function

$$E = \sum_{j=1:k} |C_j| \cdot \sum_{x,y \in C_j} link(x,y) / |C_j|^{1+2f(\theta)},$$

where $f(\theta)$ is a data dependent function. E represents specifically normalized intra-connectivity measure.

To put this formula into perspective, notice that linkage metrics normalize the aggregate measures by the number of edges. For example, the average link metric is the sum of distances between each point C_i and each point in C_j divided by the factor $L = |C_i| \cdot |C_j|$. The value L can be rationalized on a more general level. If the expected number of edges per cluster is $|C|^{\beta}, \beta \in [1,2]$, then the aggregate inter-cluster similarity has to be normalized by the factor $\left(|C_i| + |C_j|\right)^{\beta} - |C_i|^{\beta} - |C_j|^{\beta}$ representing the number of inter-cluster edges. The average link normalization factor L corresponds to $\beta = 2$, the highest expected connectivity indeed. The ROCK objective function uses the same idea, but fits it with parameters. Whether a model fits particular data is an open question. Frequently, different regions of data have different properties, and therefore, global fit is impossible. ROCK relies on an input parameter θ and on a function $f(\theta)$ that have to fit data. It has a complexity of $O(c_m N_{sample} + N_{sample}^2 \log(N_{sample}))$, where coefficient c_m is a product of average and maximum number of neighbors.

The algorithm SNN (Shared Nearest Neighbors) [Ertoz et al. 2002] blends a density-based approach with the idea of ROCK. SNN sparsifies similarity matrix (therefore, unfortunately resulting in $O(N^2)$ complexity) by only keeping K-nearest neighbors, and thus derives the total strength of links for each x.

For this matter, the idea to use shared nearest neighbors in clustering was suggested by Jarvis & Patrick [1973] long ago. See also [Gowda & Krishna 1978].

The algorithm CACTUS (Clustering Categorical Data Using Summaries) [Ganti et al. 1999a] looks for hyper-rectangular clusters (called *interval regions*) in point-by-attribute data with categorical attributes. In our terminology such clusters are segments. CACTUS is based on the idea of co-occurrence for attribute-value pairs. (Implicitly uniform distribution within the range of values for each attribute is assumed). Two values *a, b* of two different attributes are *strongly connected* if the number of data points having both *a* and *b* is larger than the frequency expected under independency assumption by a user-defined margin α. This definition is extended to subsets *A*, *B* of two different attributes (each value pair $a \in A, b \in B$ has to be strongly connected), to segments (each 2D projection is strongly connected), and to the similarity of pair of values of a single attribute via connectivity to other attributes. The cluster is defined as the maximal strongly connected segment having at least α times more elements than expected from the segment under attributes independency assumption. CACTUS uses data summaries to generate all the strongly connected and similar attribute value pairs. As a second step, a heuristic is used to generate maximum segments. The complexity of the summarization phase is $O(cN)$, where the constant *c* depends on whether all the attribute-value summaries fit in memory (one data scan), or not (multiple data scans).

The situation with clustering transactional data becomes more aggravated when size of item universe grows. Here we have a classic case of low separation in high-dimensional space (section *Clustering High Dimensional Data*). With categorical data, the idea of auxiliary clustering of items, or more generally of categorical attribute values, gained popularity. It is very similar to the idea of co-clustering (sub-section *Co-Clustering*). This, formally speaking, preprocessing step becomes the major concern, while the following data clustering remains a lesser issue.

We start with the development of Han et al. [1997] that exemplifies this approach. After items are clustered (major step), a very simple method to cluster transactions themselves is used: each transaction *T* is assigned to a cluster C_j of items having most in common with *T*, as defined by a function $\left|T \mathbin{1} C_j\right| / \left|C_j\right|$. Other choices come to mind, but again the primary objective is to find item groups. To achieve this *association rules* and *hyper-graph* machineries are used. First, frequent item-sets are generated from transactional data. A hyper-graph $H = (V, E)$ can be associated with item universe, so that vertices *V* are items. In a common graph, pairs of vertices are connected by edges, but in a hyper-graph several vertices are connected by hyper-edges. Hyper-edge $e \in E$ in *H* corresponds to a frequent item-set $\{v_1, ..., v_s\}$ and has a *weight* equal to an average of confidences among all association rules involving this item-set. A solution to the problem of *k*-way partitioning of a hyper-graph *H* is provided by algorithm HMETIS [Karypis et al. 1997].

The algorithm STIRR (Sieving Through Iterated Reinfircement) [Gibson et al. 1998] deals with co-occurrence for *d*-dimensional categorical objects, *tuples*. Extension to transactional data is obvious. It uses beautiful technique from functional analysis. Define *configurations* as weights $w = \{w_v\}$ over all different values *v* for all *d* attributes. Consider, for example, a value *v* of the first attribute. The tuples $x = (v, u_1, ..., u_{d-1})$

containing v result in a weight update $w_v' = \sum_{x, v \in x} z_x$, where terms $z_x = \Phi(w_{u_1}, ..., w_{u_{d-1}})$ depend on a *combining operator* Φ. An example of a combining operator is $\Phi(w_1, ..., w_{d-1}) = w_1 + ... + w_{d-1}$. So the weight is redistributed among different values. The major iteration scans the data X and results in the propagation of weights between different nodes $w_{new} = f(w)$ equal to a described update followed by the normalization of weights among the values of each attribute. Function f can be considered as a ***dynamic system*** (non-linear, if Φ is non-linear). STIRR relies on a deep analogy with the *spectral graph partitioning*. For linear dynamic system defined over the graph, a re-orthogonalization Gram-Schmidt process can be engaged to compute its eigenvectors that introduces negative weights. The few first non-principal eigenvectors (non-principle *basins*) define graph partitioning corresponding to positive/negative weights. The process works like this: few weights (configurations) $w^q = \{w_v^q\}$ are initialized. A major iteration updates them, $w_{new}^q = f(w^q)$, and new weights are re-orthogonalized. The process continues until *fixed point* of a dynamic system is achieved. Non-principle basins are analyzed. In STIRR a dynamic system instead of association rules formalizes co-occurrence. Additional references related to spectral graph partitioning can be found in [Gibson et al. 1998]. As the *convergence* of the process can cause a problem, the further progress is related to the modification of the dynamic system that guarantees it [Zhang et al. 2000].

7. Other Clustering Techniques

A number of other clustering algorithms have been developed. Some deal with the specific application requirements. *Constraint-based clustering* belongs to this category. Others have theoretical significance or are mostly used in other than data mining applications. We briefly discuss these developments in the sub-sections *Relation to Supervised Learning*, *Gradient Descent and ANN*, and *Evolutionary Methods*. Finally, in the sub-section *Other Developments* we very briefly mention developments that simply do not fit well in our classification.

7.1. Constraint-Based Clustering

In real-world applications customers are rarely interested in unconstrained solutions. Clusters are frequently subjected to some problem-specific limitations that make them suitable for particular business actions. Building of so conditioned cluster partitions is the subject of active research; for example, see survey [Han et al. 2001].

The framework for the constrained-based clustering is introduced in [Tung et al. 2001]. The taxonomy of clustering constraints includes constraints on individual objects (e.g., customer who recently purchased) and parameter constraints (e.g., number of clusters) that can be addressed through preprocessing or external cluster parameters. The taxonomy also includes constraints on individual clusters that can be described in terms of bounds on aggregate functions (min, avg, etc.) over each cluster. These constrains are essential, since they require a new methodology. In particular, an *existential* constraint is

a bound from below on a count of objects of a certain subset (i.e. frequent customers) in each cluster. Iterative optimization used in partitioning clustering relies on moving objects to their nearest cluster representatives. This may violate such constraint. A methodology of how to resolve this conflict is developed in [Tung et al. 2001].

The most frequent requirement is to bound number of cluster points from below. Unfortunately, *k*-means algorithm, which is used most frequently, sometimes provides a number of very small (in certain implementations empty) clusters. The modification of the *k*-means objective function and of *k*-means updates that incorporate lower limits on cluster volumes is suggested in [Bradley et al. 2000]. This includes soft assignments of data points with coefficients subject to linear program requirements. Banerjee & Ghosh [2002] presented another modification to *k*-means algorithm. Their objective function corresponds to an isotropic Gaussian mixture with widths inversely proportional to numbers of points in the clusters. The result is the *frequency sensitive k*-means. Still another approach to building balanced clusters is to convert a task into a graph-partitioning problem [Strehl & Ghosh 2000].

Important constraint-based clustering application is to cluster 2D spatial data in the presence of *obstacles*. Instead of regular Euclidean distance, a length of the shortest path between two points can be used as an *obstacle distance*. The COD (Clustering with Obstructed Distance) algorithm [Tung et al. 2001] deals with this problem. It is best illustrated by the figure 6, showing the difference in constructing three clusters in absence of obstacle (left) and in presence of a river with a bridge (right).

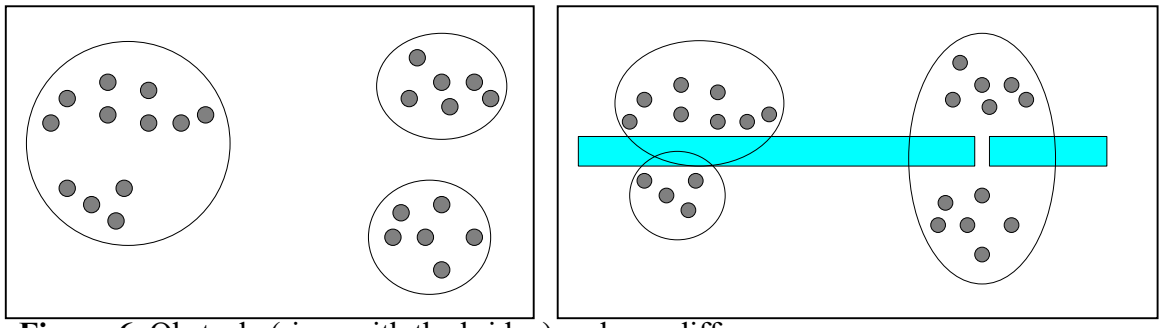


Figure 6. Obstacle (river with the bridge) makes a difference.

7.2. Relation to Supervised Learning

Both Forgy's *k*-means implementation and EM algorithms are *iterative optimizations*. Both initialize *k* models and then engage in a series of two-step iterations that: (1) reassign (*hard* or *soft*) data points, (2) update a combined model. This process can be generalized to a *framework* relating clustering with predictive mining [Kalton et al. 2001]. The model update is considered as the training of a predictive classifier based on current assignments serving as the target attribute values supervising the learning. Points' reassignments correspond to the forecasting using the recently trained classifier.

Liu et al. [2000] suggested another elegant connection to supervised learning. They considered binary target attribute defined as *Yes* on points subject to clustering, and defined as *No* on non-existent artificial points uniformly distributed in a whole attribute space. A decision tree classifier is applied to the full synthetic data. *Yes*–labeled leaves correspond to clusters of input data. The new technique CLTree (CLustering based on

decision Trees) resolves the challenges of populating the input data with artificial *No*–points such as: (1) adding points gradually following the tree construction; (2) making this process virtual (without physical additions to input data); (3) problems with uniform distribution in higher dimensions.

7.3. Gradient Descent and Artificial Neural Networks

Soft reassignments make a lot of sense, if *k*-means objective function is slightly modified to incorporate (similar to EM) "fuzzy errors", that is if it accounts for distances not only to the closest, but also to the less fit centroids:

$$E'(C) = \sum_{i=1:N} \sum_{j=1:k} \left\| x_i - c_j \right\|^2 \omega_{ij}^2$$

Exponential probabilities γ are defined based on Gaussian models. This makes the objective function differentiable with respect to means and allows application of general ***gradient decent*** method. Marroquin & Girosi [1993] presented a detailed introduction to this subject in the context of *vector quantization*. Gradient decent method in *k*-means is known as LKMA (Local K-Means Algorithm). At each iteration *t*, it modifies means c_j^t

$$c_j^{t+1} = c_j^t + a_t \sum_{i=1:N} (x_i - c_j^t) w_{ij}^2 \quad \text{or} \quad c_j^{t+1} = c_j^t + a_t (x_i - c_j^t) w_{ij}^2$$

in the direction of gradient decent. In the second case one *x* is selected randomly. Scalars a_t satisfy certain monotone asymptotic behavior and converge to zero, coefficients *w* are defined trough γ [Bottou & Bengio 1995]. Such updates are also used in a different context of ***artificial neural network*** (ANN) clustering, namely SOM (Self-Organized Map) [Kohonen 1990]. SOM is popular in vector quantization. Bibliography related to this dynamic field can be found in the monograph [Kohonen 2001]. We will not elaborate here about SOM except for two important features: (1) SOM uses incremental approach – points (patterns) are processed one-by-one; (2) SOM allows to map centroids into 2D plane that provides for a straightforward visualization. In addition to SOM, other ANN developments, such as *adaptive resonance theory* [Carpenter et al. 1991], have relation to clustering. For further discussion see [Jain & Mao 1996].

7.4. Evolutionary Methods

Substantial information on ***simulated annealing*** in the context of partitioning (main focus) or hierarchical clustering is accumulated, including the algorithm SINICC (SImulation of Near-optima for Internal Clustering Criteria) [Brown & Huntley 1991]. The perturbation operator used in general annealing has a simple meaning in clustering: it amounts to a relocation of a point from its current to a new randomly chosen cluster (very similar to *k*-means scheme). SINICC also tries to address the interesting problem of choosing the most appropriate objective function. It has a real application – surveillance monitoring of ground-based entities by airborne and ground-based sensors. Similar to simulating annealing is the so-called ***tabu search*** [Al-Sultan 1995].

Genetic Algorithms (GA) [Goldberg 1989] are also used in cluster analysis. An example is the GGA (Genetically Guided Algorithm) for fuzzy and hard *k*-means [Hall et al.

1999]. This article can be used for further references. Sarafis et al. [2002] applied GA in the context of *k*-means objective function. A *population* is a set of "*k*-means" systems represented by grid segments instead of centroids. Every segment is described by *d* rules (genes), one per attribute range. The population is improved through mutation and crossover specifically devised for these rules. Unlike in normal *k*-means, clusters can have different size and elongation; however, shapes are restricted to segments, a far cry from density-based methods. GA were also applied to clustering of categorical data using so-called generalized entropy to define the dissimilarity [Cristofor and Simovici 2002].

Evolutionary techniques rely on certain parameters to empirically fit data and have high computational costs that limit their application in data mining. However, usage of combined strategies (e.g., generation of initial guess for *k*-means) has been attempted [Babu & Murty 1993; Babu & Murty 1994]. Usage of GA with variable length genome to simultaneously improve *k*-means centroids and *k* itself [Lee & Antonsson 2000] also has a merit in comparison with running multiple *k*-means to determine a *k*, since changes in *k* happen before full convergence is achieved.

7.5. Other Developments

There are other developments that in terms of their performance qualify for data mining.

For 2D spatial data (for example, GIS database) the algorithm AMOEBA [Estivill-Castro & Lee 2000] uses Delaunay diagram (the dual of Voronoi diagram) to represent data proximity and has $O(N\log(N))$ complexity.

Harel & Koren [2001] suggested an approach related to agglomerative hierarchical graph methodology that they showed to successfully find local clusters in 2D. As above, consider a connectivity graph $G=(X,E)$. Using Delaunay diagram or keeping with any point only its *K*-nearest neighbors sparsifies the graph. The method relies on *random walk* to find separating edges F so that clusters become connected components of $G=(V,E-F)$.

8. Scalability and VLDB Extensions

Clustering algorithms face problems of scalability both in terms of computing time and memory requirements. In data mining reasonable runtime and ability to use certain limited core memory become especially important. There have been many interesting attempts to extend clustering to very large databases (VLDB), which can be divided into:

- Incremental mining,
- Data squashing,
- Reliable sampling.

The algorithm DIGNET [Thomopoulos et al. 1995; Wann & Thomopoulos 1997] (compare with "*the leader*" clustering algorithm in [Hartigan 1975]) is an example of incremental unsupervised learning. This means that it handles one data point at a time, and then discards it. DIGNET uses *k*-means cluster representation without iterative optimization. Centroids are instead *pushed* or *pulled* depending on whether they loose or win each next coming point. Such on-line clustering needs only one data pass, but

strongly depends on data ordering, and it can result in sub-quality clusters. However, it handles outliers, clusters can be dynamically born or discarded, and the training process is resumable. This makes it very appealing for dynamic VLDB. Some further tools can be used to improve obtained clusters.

Data squashing techniques scan data to compute certain data summaries (*sufficient statistics*) [DuMouchel et al. 1999]. The obtained summaries are then used instead of the original data for further clustering. The pivotal role here belongs to the algorithm BIRCH (Balanced Iterative Reduction and Clustering using Hierarchies) [Zhang et al. 1996; Zhang et al. 1997]. This work had a significant impact on overall direction of scalability research in clustering. BIRCH creates a height-balanced tree of nodes that summarize data by accumulating its zero, first, and second moments. A node, called ***Cluster Feature*** (CF), is a tight small cluster of numerical data. The construction of a tree residing in core memory is controlled by some parameters. A new data point descends along the tree to the closest CF leaf. If it fits the leaf well and if the leaf is not overcrowded, CF statistics are incremented for all nodes from the leaf to the root. Otherwise a new CF is constructed. Since the maximum number of children per node (*branching factor*) is limited, one or several splits can happen. When the tree reaches the assigned memory size, it is rebuilt and a threshold controlling whether a new point is assigned to a leaf or starts a new leaf is updated to a coarser one. The outliers are sent to disk, and refitted gradually during tree rebuilds. The final leaves constitute input to any algorithm of choice. The fact that a CF-tree is balanced allows the log-efficient search. BIRCH depends on parameters that control CF tree construction (branching factor, maximum of points per leaf, leaf threshold), and it also depends on data ordering. When the tree is constructed (one data pass), it can be additionally condensed in the optional 2^{nd} phase to further fit desired input cardinality of post-processing clustering algorithm. Next, in the 3^{rd} phase a global clustering of CF (considered as individual points) happens. Finally, certain irregularities (for example, identical points getting to different CFs) can be resolved in an optional 4^{th} phase. It makes one or more passes through data reassigning points to best possible clusters, as *k*-means does. The overall complexity is $O(N)$. Summarization phase of BIRCH was extended to mixed numerical and categorical attributes [Chiu et al. 2001].

A full interface between VLDB and relocation clustering (as *k*-means) includes following requirements [Bradley et al. 1998]. Algorithm has to:

- Take one (or less – early termination) data scan
- Provide on-line solution: some solution in-progress should always be available
- Be suspendable, stoppable, resumable
- Be able to incorporate additional data incrementally
- Be able to work in prescribed memory buffer
- Utilize different scanning modes (sequential, index, sample)
- Be able to operate in forward-only cursor over a view of database

The article suggests data compression that accumulates sufficient statistics like BIRCH does, but makes it in phases. Points that are compressed over the primary stage are discarded. They can be attributed to their clusters with very high confidence even if other points would shift. The rest is taken care of in the secondary phase, which tries to find

dense subsets by *k*-means method with higher than requested *k*. Violators of this stage are still kept in retained set (RT) of singletons to be analyzed later.

BIRCH-like preprocessing substantially relies on vector-space operations. Meanwhile, in many applications, objects (for example, strings) belong to a metric space. In other words, all we can do with data points is to compute distances between them. Ganti et al. [1999b] proposed BIRCH-type data squashing BUBBLE for VLDB in metric spaces. Each leaf of the BUBBLE-tree is characterized by:

1) Number of its points
2) Medoid (called *clustroid*) that delivers a minimum to an error – a squared distance between it and all other points belonging to the leaf
3) Radius equal to the square root of an average error per a point

The problem to overcome is how to insert new points in the absence of a vector structure. BUBBLE uses a heuristic that relates to a distance preserving embedding of leaf points into a low-dimensional Euclidean vector space. Such embedding is known as isometric map in geometry and as multidimensional scaling in statistics. Certain analogy can also be made with embeddings used in support vector machines. While Euclidean distance (used in BIRCH) is cheap, the computation of a distance in a metric space (for example, edit distance for strings) can be expensive. Meanwhile, every insertion requires to compute distances to all the nodes descending to a leaf. The similar algorithm BUBBLE-FM handles this difficulty. It relaxes the computations by using *approximate* isometric embedding. This is possible due to the algorithm FastMap [Faloutsos & Lin 1995].

In the context of hierarchical density-based clustering in VLDB, Breunig et al. [2001] analyzed data reduction techniques such as sampling and BIRCH summarization, and noticed that they result in deterioration of cluster quality. To cure this, they approached data reduction through accumulation of *data bubbles* that are summaries of local information about distances and nearest neighbors. A data bubble contains an *extent*, the distance from a bubble's representative to most points in *X*, and the array of distances to each of *MinPts* nearest neighbors. Data bubbles are then used in conjunction with the algorithm OPTICS (see sub-section *Density-Based Connectivity*).

Grid-methods also generate data summaries, though their summarization phase relates to units and segments and not to CFs. Therefore, they are scalable.

Many algorithms use old-fashioned sampling with or without rigorous statistical reasoning. It is especially handy for different initializations as in CLARANS (sub-section *K-Medoids Methods*), Fractal Clustering (section *Grid-Based Methods*), or *k*-means [Bradley & Fayyad 98]. Notice that when clusters are constructed using whatever sample, assigning the whole data to the most appropriate clusters minimally adds the term $O(N)$ to the overall complexity.

Sampling has got a new life with the adoption by the data mining community of a special uniform check to control its adequacy. This check is based on *Hoffding* or *Chernoff* bounds [Motwani & Raghavan 1995] and says that, independent of the distribution of a real-valued random variable $Y,\ 0 \le Y \le R,$ the average of *n* independent observations lies within $\mathring{U}$ of the actual mean

$$\left| \bar{Y} - 1/n \sum_{j=1:n} Y_j \right| \leq \varepsilon$$

with probability 1-ŭ as soon as

$$\varepsilon = \sqrt{R^2 \ln(1/\delta)/2n} \, .$$

These bounds were used in the clustering algorithm CURE [Guha et al. 1998] and in the development of scalable decision trees in predictive mining [Hulten et al. 2001]. In the context of balanced clustering, a statistical estimation of a sample size is provided in [Banerjee & Ghosh 2002]. Due to their nonparametric nature, the bounds have a ubiquitous significance.

9. Clustering High Dimensional Data

The objects in data mining could have hundreds of attributes. Clustering in such high dimensional spaces presents tremendous difficulty, much more so than in predictive learning. In decision trees, for example, irrelevant attributes simply will not be picked for node splitting, and it is known that they do not affect Naïve Bayes as well. In clustering, however, high dimensionality presents a dual problem. First, under whatever definition of similarity, the presence of irrelevant attributes eliminates any hope on *clustering tendency*. After all, searching for clusters where there are no clusters is a hopeless enterprise. While this could also happen with low dimensional data, the likelihood of presence and number of irrelevant attributes grows with dimension.

The second problem is the ***dimensionality curse*** that is a loose way of speaking about a lack of data separation in high dimensional space. Mathematically, nearest neighbor query becomes *unstable*: the distance to the nearest neighbor becomes indistinguishable from the distance to the majority of points [Beyer et al. 1999]. This effect starts to be severe for dimensions greater than 15. Therefore, construction of clusters founded on the concept of proximity is doubtful in such situations. For interesting insights into complications of high dimensional data, see [Aggarwal et al. 2000].

Basic exploratory data analysis (attribute selection) preceding the clustering step is the best way to address the first problem of irrelevant attributes. We consider this topic in the section *General Algorithmic Issues*. Below we present some techniques dealing with a situation when the number of already pre-selected attributes *d* is still high.

In the sub-section *Dimensionality Reduction* we talk briefly about traditional methods of dimensionality reduction. In the sub-section *Subspace Clustering* we review algorithms that try to circumvent high dimensionality by building clusters in appropriate subspaces of original attribute space. Such approach has a perfect sense in applications, since it is only better if we can describe data by fewer attributes. Still another approach that divides attributes into similar groups and comes up with good new derived attributes representing each group is discussed in the sub-section *Co-Clustering*.

Important source of high dimensional categorical data comes from transactional (market basket) analysis. Idea to group items very similar to co-clustering has already been discussed in the section *Co-Occurrence of Categorical Data*.

9.1. Dimensionality Reduction

When talking about high dimensionality, *how high is high*?

Many spatial clustering algorithms depend on indices in spatial datasets (sub-section *Data Preparation*) to facilitate quick search of the nearest neighbors. Therefore, indices can serve as good proxies with respect to *dimensionality curse* performance impact. Indices used in clustering algorithms are known to work effectively for dimensions below 16. For a dimension $d > 20$ their performance degrades to the level of sequential search (though newer indices achieve significantly higher limits). Therefore, we can arguably claim that data with more than 16 attributes is high dimensional.

How large is the gap? If we are dealing with a retail application, 52-weeks sales volumes represent a typical set of features, which is a special example of more general class of time series data. In customer profiling dozens of generalized item categories plus basic demographics result in at the least 50-100 attributes. Web clustering based on site contents results in 200-1000 attributes (pages/contents) for modest Web sites. Biology and genomic data can have dimensions that easily surpass 2000-5000 attributes. Finally, text mining and information retrieval also deal with many thousands of attributes (words or *index terms*). So, the gap *is significant*.

Two general purpose techniques are used to fight high dimensionality: (1) ***attributes transformations*** and (2) ***domain decomposition***.

Attribute transformations are simple functions of existent attributes. For sales profiles and OLAP-type data, roll-ups as sums or averages over time intervals (e.g., monthly volumes) can be used. Due to a fine seasonality of sales such brute force approaches rarely work. In multivariate statistics *principal components analysis* (PCA) is popular [Mardia et al. 1980; Joliffe 1986], but this approach is problematic since it leads to clusters with poor interpretability. Singular value decomposition (SVD) technique is used to reduce dimensionality in information retrieval [Berry et al. 1995; Berry & Browne 1999] and statistics [Fukunaga 1990]. Low-frequency Fourier harmonics in conjunction with Parseval's theorem are successfully used in analysis of time series [Agrawal et al. 1993], as well as wavelets and other transformations [Keogh et al. 2001].

Domain decomposition divides the data into subsets, *canopies*, [McCallum et al. 2000] using some inexpensive similarity measure, so that the high dimensional computation happens over smaller datasets. Dimension stays the same, but the costs are reduced. This approach targets the situation of high dimension, large data, and many clusters.

9.2. Subspace Clustering

Some algorithms better adjust to high dimensions. For example, the algorithm CACTUS (section *Co-Occurrence of Categorical Data*) adjusts well since it defines a cluster only in terms of a cluster's 2D projections. In this section we cover techniques that are specifically designed to work with high dimensional data.

The algorithm CLIQUE (Clustering In QUEst) [Agrawal et al. 1998] for numerical attributes is fundamental in subspace clustering. It marries the ideas of:

- Density-based clustering

- Grid-based clustering
- Induction through dimensions similar to *Apriori* algorithm in association rules
- MDL principle to select appropriate subspaces
- Interpretability of clusters in terms of DNF representation

CLIQUE starts with the definition of a unit – elementary rectangular cell in a subspace. Only units whose densities exceed a threshold τ are retained. A bottom-up approach of finding such units is applied. First, 1-dimensional units are found by dividing intervals in ξ equal-width bins (a grid). Both parameters τ and ξ are the algorithm's inputs. The recursive step from q-1-dimensional units to q-dimensional units involves self-join of q-1 units having *first* common q-2 dimensions (Apriori-reasoning). All the subspaces are sorted by their coverage and lesser-covered subspaces are pruned. A cut point is selected based on MDL principle. A cluster is defined as a maximal set of connected dense units. It is represented by a DNF expression that is associated with a finite set of maximal segments (called *regions*) whose union is equal to a cluster. Effectively, CLIQUE results in attribute selection (it selects several subspaces) and produces a view of data from different perspectives! The result is a series of cluster systems in different subspaces. This versatility goes more in vein with data description rather than with data partitioning: different clusters overlap. If q is a highest subspace dimension selected, the complexity of dense units generations is $O(const^q + qN)$. Identification of clusters is a quadratic task in terms of units.

The algorithm ENCLUS (ENtropy-based CLUStering) [Cheng et al. 1999] follows in the footsteps of CLIQUE, but uses a different criterion for subspace selection. The criterion is derived from entropy related considerations: the subspace spanned by attributes $A_1,...,A_q$ with entropy $H(A_1,...,A_q)$ smaller than a threshold ω is considered good for clustering. Any subspace of a good subspace is also good, since

$$H(A_1,...,A_{q-1}) = H(A_1,...,A_q) - H(A_q \mid A_1,...,A_{q-1}) \le H(A_1,...,A_q) < \omega .$$

Low entropy subspace corresponds to a skewed distribution of unit densities. The computational costs of ENCLUS are high.

The algorithm MAFIA (Merging of Adaptive Finite Intervals) [Goil et al. 1999; Nagesh et al. 2001] significantly modifies CLIQUE. It starts with one data pass to construct *adaptive grids* in each dimension. Many (1000) bins are used to compute histograms by reading blocks of data in core memory, which are then merged together to come up with a smaller number of variable-size bins than CLIQUE does. The algorithm uses a parameter $\breve{U}$, called cluster dominance factor, to select bins that are $\breve{U}$-times more densely populated relative to their volume than on average. These are q=1 candidate dense units (CDUs). Then MAFIA proceeds recursively to higher dimensions (every time a data scan is involved). The difference between MAFIA and CLIQUE is that to construct a new q-CDU, MAFIA tries two q-1-CDUs as soon as they share *any* (not only *first* dimensions) q-2-face. This creates an order of magnitude more candidates. Adjacent CDUs are merged into clusters and clusters that are proper subsets of the higher dimension clusters are eliminated. The parameter $\breve{U}$ (default value 1.5 works fine) presents no problem in comparison with global density threshold used in CLIQUE. Reporting for a range of $\breve{U}$ in

a single run is supported. If q is a highest dimensionality of CDU, the complexity is $O(const^q + qN)$.

The algorithm OPTIGRID [Hinneburg & Keim 1999] uses data partitioning based on divisive recursion by multi-dimensional grids. Authors present a very good introduction into the effects of high-dimension geometry. Familiar concepts, as for example, uniform distribution, become blurred for large *d*. OPTIGRID uses density estimations in the same way the algorithm DENCLUE (by the same authors) does. It primarily focuses on separation of clusters by (hyper) planes that are not necessarily axes parallel. To find such planes consider a set of *contracting* linear projectors (functionals) $P_1,...,P_k, \|P_j\| \le 1$ of the attribute space A at a 1D line. For a density kernel of the form $K(x-y)$ (a tool of trade in DENCLUE) and a contracting projection, density induced after projection is more concentrated. A cutting plane is chosen so that it goes through the point of minimal density and discriminates two significantly dense half-spaces. Several cutting planes are chosen, and recursion continues with each subset of data.

The algorithm PROCLUS (PROjected CLUstering) [Aggarwal et al. 1999a] associates with a subset C a low-dimensional subspace such that the projection of C into the subspace is a tight cluster. The subset – subspace pair when exists constitutes a *projected cluster*. The number k of clusters and the average subspace dimension l are user inputs. The iterative phase of the algorithm deals with finding k good medoids, each associated with its subspace. A sample of data is used in a greedy hill-climbing technique. Manhattan distance divided by the subspace dimension is a useful normalized metric for searching among different dimensions. An additional data pass follows after iterative stage is finished to refine clusters including subspaces associated with the medoids.

The algorithm ORCLUS (ORiented projected CLUSter generation) [Aggarwal & Yu 2000] uses a similar approach of projected clustering, but employs non-axes parallel subspaces of high dimensional space. In fact, both developments address a more generic issue: even in a low dimensional space, different portions of data could exhibit clustering tendency in different subspaces (consider several non-parallel non-intersecting cylinders in 3D space). If this is the case, any attribute selection is doomed. ORCLUS has a *k*-means-like transparent model that defines clusters as sets of points (partition) that have low sum-of-squares of errors (*energy*) in a certain subspace. More specifically, for $x \in C$, and directions $E = \{e_1,...,e_l\}$ (specific to C), the projection is defined as $\{x \cdot e_1,...,x \cdot e_l\}$. The projection only *decreases* energy. SVD diagonalization can be used to find directions (eigenvectors) corresponding to the lowest l eigenvalues of the covariance matrix. In reality, the algorithm results in X partitioning (the outliers excluded) into k clusters C_j together with their subspace directions E_j. The algorithm builds more than k clusters, with larger than l-dimensional E gradually fitting the optimal subspace and requested k. Though suggestion of picking a good parameter l is provided, uniform l is a certain liability.

Any other comparison aside, projected clusters provide data partitioning, while cluster systems resulted from CLIQUE overlap.

9.3. Co-Clustering

In OLAP attribute roll-ups can be viewed as representatives of the attribute groups. An interesting general idea of producing attribute groups in conjunction with clustering of points themselves leads to the concept of ***co-clustering***. Co-clustering is a simultaneous clustering of both points and their attributes. This approach reverses the struggle: to improve clustering of points based on their attributes, it tries to cluster attributes based on the points. So far we were concerned with grouping only rows of a matrix *X*. Now we are talking about grouping its columns as well. This utilizes a canonical ***duality*** contained in the *point-by-attribute* data representation.

The idea of co-clustering of data points and attributes is old [Anderberg 1973; Hartigan 1975] and is known under the names *simultaneous clustering*, *bi-dimensional clustering*, *block clustering*, *conjugate clustering*, *distributional clustering*, and *information bottleneck method*. The use of duality for analysis of categorical data (dual or multidimensional scaling) also has a long history in statistics [Nishisato 1980]. The

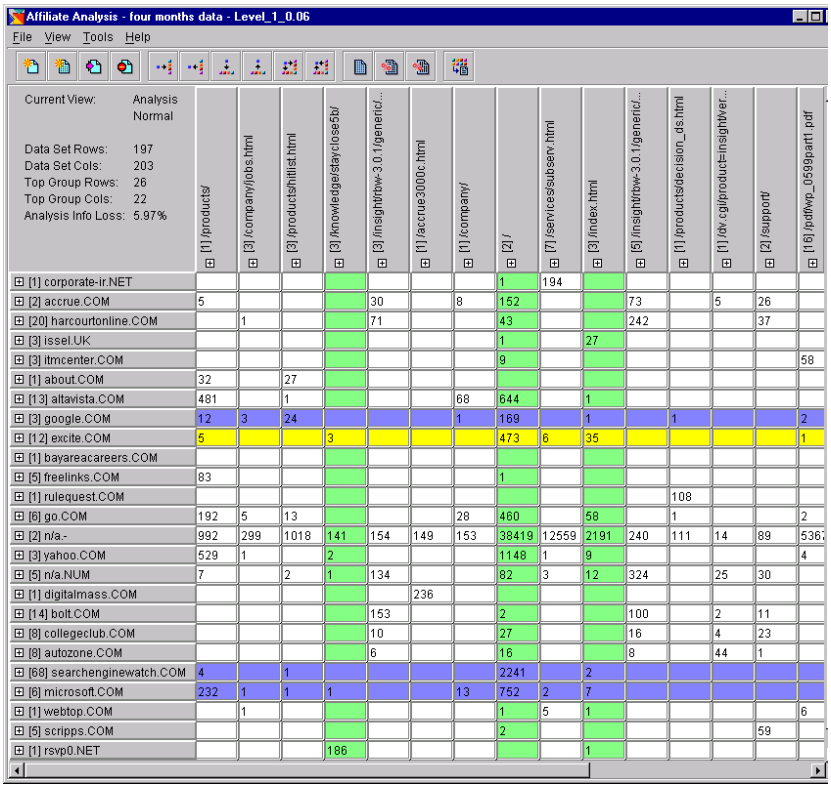


Figure 7. Learning referring traffic on a Web site.

similar approach of building groups of item was presented in the section *Co-Occurrence of Categorical Data*. In this section we turn to numerical attributes. Assume that the matrix *X* has non-negative elements. In this context it is known as *incidence*, *relational*, *frequency*, *contingency* matrix. In applications it can reflect intensity of a gene response in a tissue sample, frequency of visitation activity of a page, or the amount of a sale in a store per item category.

Govaert [1995] researched simultaneous block clustering of the rows and columns of contingency tables. He also reviewed an earlier work on the subject. An advanced algebraic approach to co-clustering based on bi-partite graphs and their minimal cuts in conjunction with text mining was proposed in [Dhillon 2001]. This paper contains an excellent introduction in relations between simultaneous clustering and graph

partitioning, as well as in connection with SVD. A simple algorithm Ping-Pong [Oyanagi et al. 2001] was suggested to find populated areas in a sparse binary matrix. It redistributes influence of columns on rows and vice versa (compare with algorithm STIRR above) by transversal connection through matrix elements and provides an example of other than co-clustering, but a related development.

A series of publications deal with distributional clustering of attributes based on the informational measures of attribute similarity. Two attributes (two columns in matrix X) with exactly the same probability distributions are identical for the purpose of data mining, and so, one can be deleted. Attributes that have probability distributions that are close in terms of their Kullback-Leibler (KL) distance [Kullback & Leibler 1951] can still be grouped together without much of an impact. In addition, a natural derived attribute, the mixed distribution (a normalized sum of two columns) is now available to represent the group. This process can be generalized. The grouping simplifies the original matrix X to the compressed form $\overline{X}$. Attribute clustering is productive when it minimally impacts information reduction $R = I(X) - I(\overline{X})$, where $I(X)$ is mutual information contained in X [Cover & Thomas 1990]. Such attribute grouping is intimately related to Naïve Bayes classification in predictive mining [Baker & McCallum 1998].

The outlined technique is very much relevant to grouping words in text mining. In this context the technique was explored under the name *information bottleneck method* [Tishby et al. 1999]. It was used to facilitate agglomerative co-clustering of words in document clustering [Slonim & Tishby 2000] and classification [Slonim & Tishby 2001].

Berkhin & Becher [2002] showed deep algebraic connection of distributional clustering to *k*-means. They used *k*-means adaptation to KL-distance as a major iterative step in the algorithm SIMPLIFYRELATION that gradually co-clusters points and attributes. This development has industrial application in Web analysis. Figure 7 shows how an original incidence matrix of Web site traffic between 197 referrers (rows) and 203 Web site pages (columns) is clustered into 26x22 matrix with 6% information loss. While KL-distance is not actually a distance, since it is not symmetric, it can be symmetrized to the so-called Jensen-Shanon divergence. Dhillon et al. [2002] used Jensen-Shanon divergence to cluster words in *k*-means fashion in text classification. Besides text and Web mining, the idea of co-clustering finds its way into other applications, as for example, clustering of gene microarrays [Busygin et al. 2002].

10. General Algorithmic Issues

We have presented many different clustering techniques. However, there are common issues that must be addressed to make any clustering algorithm successful. Some are so ubiquitous that they are not even specific to unsupervised learning and can be considered as a part of overall data mining framework. Others are resolved in certain algorithms we presented. In fact, many algorithms were specifically designed for this reason. Now we overview common issues, and necessarily our coverage will be very fragmented.

Scalability for VLDB and high dimensional clustering were already surveyed above, but several others significant issues are discussed below:

- Assessment of results

- Choice of appropriate number of clusters
- Data preparation
- Proximity measures
- Handling outliers

10.1. Assessment of Results

The data mining clustering process starts with the assessment of whether any *cluster tendency* has a place at all, and correspondingly includes, appropriate attribute selection, and in many cases feature construction. It finishes with the *validation* and *evaluation* of the resulting clustering system. The clustering system can be assessed by an expert, or by a particular automated procedure. Traditionally, the first type of assessment relates to two issues: (1) cluster interpretability, (2) cluster visualization. Interpretability depends on the technique used. Model-based probabilistic and conceptual algorithms, as COBWEB, have better scores in this regard. *K*-means and *k*-medoid methods generate clusters that are interpreted as dense areas around centroids or medoids and, therefore, also score well. The review [Jain et al. 1999] extensively covers cluster validation, while a discussion of cluster visualization and related references can be found in [Kandogan 2001].

Regarding automatic procedures, when two partitions are constructed (with the same or different number of subsets *k*), the first order of business is to compare them. Sometimes the actual class label *s* of one partition is known. Still clustering is performed generating another label *j*. The situation is similar to testing a classifier in predictive mining when the actual target is known. Comparison of *s* and *j* labels is similar to computing an error, confusion matrix, etc., in predictive mining. Simple criterion ***Rand*** serves this purpose [Rand 1971]. Computation of a Rand index (defined below) involves pairs of points that were assigned to the same and to the different clusters in each of two partitions. Hence it has $O(N^2)$ complexity and is not always feasible. ***Conditional entropy*** of a known label *s* given clustering partitioning [Cover & Thomas 1990]

$$H(S \mid J) = -\sum_j p_j \sum_s p_{s|j} \log(p_{s|j})$$

is another measure used. Here $p_j, p_{s|j}$ are probabilities of *j* cluster, and conditional probabilities of *s* given *j*. It has $O(N)$ complexity. Other measures are also used, for example, the ***F-measure*** [Larsen & Aone 1999].

10.2. How Many Clusters?

In many methods number *k* of clusters to construct is an input user parameter. Running an algorithm several times leads to a sequence of clustering systems. Each system consists of more granular and less-separated clusters. In the case of *k*-means, the objective function is monotone decreasing. Therefore, the answer to the question of which system is preferable is not trivial.

Many criteria have been introduced to find an optimal *k*. Some industrial applications (SAS, NeoVista) report pseudo ***F-statistic***. This only makes sense for *k*-means clustering in context of ANOVA. Earlier publications on the subject analyzed cluster separation for different *k* [Engleman & Hartigan 1969; Milligan & Cooper 1985]. For instance, a

distance between any two centroids (medoids) normalized by corresponding cluster's radii (standard deviations) and averaged (with cluster weights) is a reasonable choice of *coefficient of separation*. This coefficient has a very low $O(k^2)$ complexity. Another popular choice for separation measure is a ***Silhouette coefficient*** [Kaufman & Rousseeuw 1990]. For example, Silhouette coefficient is used in conjunction with CLARANS in [Ng & Han 1994]. It has $O(N^2)$ complexity. Consider average distance between the point x of cluster C and other points within C and compare it with averaged distance to the best fitting cluster G other than C

$$a(x) = \frac{1}{|C|-1} \sum_{y \in C, y \neq x} d(x,y), \; b(x) = \min_{G \neq C} \frac{1}{|G|} \sum_{y \in G} d(x,y)$$

The Silhouette coefficient of x is $s(x) = (b(x) - a(x)) / \max\{a(x), b(x)\}$, values close to +1 corresponding to a perfect and values below 0 to a bad clustering choice. The overall average of individual $s(x)$ gives a good indication of cluster system appropriateness.

Still another approach to separation is to employ possible soft (or fuzzy) assignments. It has an intermediate $O(kN)$ complexity. Remember that assignment of a point to a particular cluster may frequently involve certain arbitrariness. Depending on how well a point fits a particular cluster C, different probabilities or weights $w(x,C)$ can be introduced so that a hard (strict) assignment is defined as

$$C(x) = \min \arg_C w(x,C).$$

A ***Partition coefficient*** [Bezdek 1981] is equal to the sum of squares of the weights

$$W = \frac{1}{N} \sum_{x \in X} w(x, C(x))^2$$

(compare with GINI index). Each of the discussed measures can be plotted as a function of k and the graph can be used to choose the best k.

The strong probabilistic foundation of the *mixture model*, discussed in sub-section *Probabilistic Clustering*, allows viewing a choice of optimal k as a problem of fitting the data by the best model. The question is whether adding new parameters results in a better model. In Bayesian learning (for example, AUTOCLASS [Cheeseman & Stutz 1995]) the likelihood of the model is directly affected (through priors) by the model complexity (number of parameters is proportional to k). Several criteria were suggested including:

- *Minimum Description Length* (MDL) criterion [Rissanen 1978; Schwarz 1978; Rissanen 1989]
- *Minimum Message Length* (MML) criterion [Wallace & Freeman 87; Wallace & Dowe 94]
- *Bayesian Information Criterion* (BIC) [Schwarz 1978; Fraley & Raftery 1998]
- *Akaike's Information Criterion* (AIC) [Bozdogan 1983]
- *Non-coding Information Theoretic Criterion* (ICOMP) [Bozdogan 1994]

- *Approximate Weight of Evidence* (AWE) criterion [Banfield & Raftery 1993]
- *Bayes Factors* [Kass & Raftery 1995]

All these criteria are expressed through combinations of log-likelihood L, number of clusters k, number of parameters per cluster, total number of estimated parameters p, and different flavors of Fisher information matrix. For example,

$$MDL(k) = -L + p/2 - \log(p),\ k_{best} = \min\arg MDL(k),$$

$$BIC(k) = L - {}^{p}\!/_{2}\log(n),\ k_{best} = \max\arg BIC(k).$$

Further details and discussion can be found in [Bock 1996; Oliver et al. 1996; Fraley & Raftery 1998]. Few examples: MCLUST and *X*-means use BIC criterion, SNOB uses MML criterion, CLIQUE and evolutionary approach to k determination [Lee & Antonsson 2000] use MDL. Significant expertise is developed in estimation of goodness of fit based on the criteria above. For example, different ranges of BIC are suggested for weak, positive, and very strong evidence in favor of one clustering system versus another [Fraley & Raftery 1999]. Smyth [1998] suggested a *likelihood cross-validation* technique for determination the best k.

10.3. Data Preparation

Irrelevant attributes make chances of a successful clustering futile, because they negatively affect proximity measures and eliminate *clustering tendency*. Therefore, sound exploratory data analysis (EDA) is essential. An overall framework for EDA can be found in [Becher et al. 2000]. As its first order of business, EDA eliminates inappropriate attributes and reduces the cardinality of the retained categorical attributes. Next it provides attribute selection. Different attribute selection methods exist. Inconsistency rates are utilized in [Liu & Setiono 1996]. The idea of a Markov blanket is used in [Koller & Sahami 1996]. While there are others methods (for example, [Jebara & Jaakkola 2000]), most are used primarily for predictive and not descriptive mining and thus do not address general-purpose attribute selection for clustering. We conclude that cluster-specific attribute selection yet to be invented.

Attributes transformation and clustering have already been discussed in the context of dimensionality reduction. The practice of assigning different weights to attributes and/or scaling of their values (especially, standardization) is widespread and allows constructing clusters of better shapes. To some extent *attribute scaling* can be viewed as the continuation of attribute selection.

In real-life applications it is crucial to handle attributes of different nature. For example, images are characterized by color, texture, shape, and location, resulting in four attribute subsets. Modha & Spangler [2002] suggested a very interesting approach for attribute scaling that pursues the objective of clustering in each attribute subset by computing weights (a simplex) that minimize the product of intra-cluster to inter-cluster ratios for the attribute subset projections (called *generalized Fisher ratio*).

In many applications data points have different significance. For example, in assortment planning, stores can be characterized by the profiles of sales of particular items in

percentage. However, the overall sale volume gives additional weight to larger stores. Partitioning relocation algorithms easily handle weighted data, because centroids become centers of weights instead of means. The described practice is called *case scaling*.

Some algorithms depend on the effectiveness of data access. To facilitate this process data indices are constructed. Examples include the extension of the algorithm CLARANS [Ester et al. 1995] and the algorithm DBSCAN [Ester et al. 1996]. Index structures used for spatial data, include KD-trees [Friedman et al. 1977], R-trees [Guttman 1984], R*-trees [Kriegel et al. 1990]. A blend of attribute transformations (DFT, Polynomials) and indexing technique is presented in [Keogh et al. 2001a]. Other indices and numerous generalizations exist [Beckmann 1990; Faloutsos et al. 1994; Berchtold et al. 98; Wang et al. 1998; Karypis & Han 2000; Keogh et al. 2001b]. The major application of such data structures is in nearest neighbors search.

Preprocessing of multimedia data that is based on its embedding in Euclidean space (the algorithm FastMap) [Faloutsos & Lin 1995].

A fairly diverse range of preprocessing is used for variable length sequences. Instead of handling them directly (as in the sub-section *Probabilistic Clustering*), a fixed set of features to represent variable length sequences can be derived [Guralnik & Karypis 2001; Manilla & Rusakov 2001].

10.4. Proximity Measures

Both hierarchical and partitioning methods use different distances and similarity measures [Jain & Dubes 1988]. The usual L_p distance

$$d(x,y) = \|x-y\|_p,\ \|z\|_p = \left(\sum_{j=1:d} |z_j|^p\right)^{1/p},\ \|z\| = \|z\|_2$$

is used for numerical data, $1 \le p < \infty$, in which lower p corresponds to a more *robust* estimation (therefore, less affected by outliers). Euclidean (p=2) distance is by far the most popular choice used in k-means objective function (sum of squares of distances between points and centroids) that has a clear statistical meaning of total inter-clusters variance. Manhattan distance corresponds to p=1. The distance that returns the maximum of absolute difference in coordinates is also used and corresponds to $p = \infty$. In many applications (profile analyses) points are scaled to have a unit norm, so that the proximity measure is an angle between the points,

$$d(x,y) = \arccos\left(x^T y / \|x\| \cdot \|y\|\right).$$

It is used, in specific tools, as DIGNET (section Scalability and VLDB Extensions), and in particular applications, as text mining. All above distances assume attributes independence (diagonal covariance matrix S). Mahanalabonis distance

$$d(x,y) = \sqrt{(x-y)^T S^{-1} (x-y)}$$

[Mardia et al. 1980] is used in algorithms, as ORCLUS [Aggarwal & Yu 2000], that do not make this assumption.

Formula $s(x,y) = 1/(1+d(x,y))$ defines similarity for numerical attributes. Other choices are *cosine, Dice coefficients* and distance *exponent*

$$s_{\cos}(x,y) = x^T y / \|x\|\|y\|, \;\; s_{Dice} = 2x^T y / \left(\|x\|^2 + \|y\|^2\right), \; s_{\exp} = \exp\left(-\|x-y\|^{\alpha}\right).$$

Now we shift our attention to categorical data. A number of similarity measures exist for categorical attributes [Dubes 1993; Everitt 1993]. Assuming binary attributes with values $\alpha, \beta = \pm$, let $d_{\alpha\beta}$ be a number of attributes having outcomes α in *x* and β in *y*. Then the *Rand* and *Jaccard* (also known as *Tanimoto*) indices *R*, *J* are equal to

$$R(x,y) = (d_{++} + d_{--})/(d_{++} + d_{+-} + d_{-+} + d_{--}), \; J(x,y) = (d_{++})/(d_{++} + d_{+-} + d_{-+})$$

Notice that Jaccard index treats positive and negative values asymmetrically, which makes it the measure of choice for transactional data, + meaning that an item is present. It is simply the fraction of common items of two transactions relative to the number of items in both transactions. It is also used in collaborative filtering, sequence analysis, text mining, and pattern recognition. *Extended Jaccard* coefficient is advocated in [Ghosh 2002]. For construction of similarity measures for market basket analysis see [Aggarwal et al. 1999b; Baeza-Yates 1992]. Similarity can also be constructed axiomatically based on information-theoretical considerations [Lin 1998]. The last two references contain material related to strings similarity (biology is one application). For strings over the same alphabet, *edit distance* is a frequent choice [Arslan & Egecioglu 2000]. It is based on the length of a sequence of transformations (such as insertion, deletion, transposition, etc.) that are necessary to transform one string into another. A classic Hamming distance [Cover & Thomas 1990] is also used. Further references can be found in the review [Jain et al. 1999]. Historically textual mining was a source of major inspirations for a concept of similarity [Resnik 1995].

Proximity measures between two clusters that can be derived from proximities between pairs of their points were discussed in the sub-section *Linkage Metrics*.

10.5. Handling Outliers

Applications that derive their data from measurements have an associated amount of noise, which can be viewed as outliers. Alternately, outliers can be viewed as legitimate records having abnormal behavior. In general, clustering techniques do not distinguish between the two: neither noise nor abnormalities fit into clusters. Correspondingly, the preferable way to deal with outliers in partitioning the data is to keep one extra set of outliers, so as not to pollute factual clusters.

There are multiple ways of how descriptive learning handles outliers. If a summarization or data preprocessing phase is present, it usually takes care of outliers. For example, this is the case with grid-based methods. They simply rely on input thresholds to eliminate low-populated cells. Algorithms in the section *Scalability and VLDB Extensions* provide further examples. The algorithm BIRCH [Zhang et al. 1996; Zhang et al. 1997] revisits outliers during the major CF tree rebuilds, but in general handles them separately. This approach is shared by other similar systems [Chiu et al. 2001]. The framework of [Bradley et al. 1998] utilizes a multiphase approach to outliers.

Certain algorithms have specific features for outliers handling. The algorithm CURE [Guha et al. 1998] uses shrinkage of cluster's representives to suppress the effects of outliers. *K*-medoids methods are generally more robust than *k*-means methods with respect to outliers: medoids do not "feel" outliers. The algorithm DBCSAN [Ester et al. 1996] uses concepts of internal (core), boundary (reachable), and outliers (non-reachable) points. The algorithm CLIQUE [Agrawal et al. 1998] goes a step further: it eliminates subspaces with low coverage. The algorithm WaveCluster [Sheikholeslami et al. 1998] is known to handle outliers very well through its filtering DSP foundation. The algorithm ORCLUS [Aggarwal & Yu 2000] produces a partition plus a set of outliers.

What is an outlier? Statistics defines an outlier as a point that does not fit a *probability distribution*. This approach has the problem with *discordance testing* for unknown multivariate distribution. Classic data analysis utilizes a concept of *depth* [Tukey 1977] and defines an outlier as a point of low depth. This concept becomes computationally infeasible for $d > 3$. Data mining is gradually develops its own definitions.

Consider two positive parameters $\mathring{U}$, $\breve{u}$. A point can be declared an outlier if its $\mathring{U}$-neighborhood contains less than 1-$\breve{u}$ fraction of a whole dataset X [Knorr & Ng 1998]. Ramaswamy et al. [2000] noticed that this definition can be improved by eliminating parameter $\mathring{U}$. Rank all the points by their distance to the K-nearest neighbor and define the $\breve{u}$ fraction of points with highest ranks as outliers. Both definitions are uniformly global. How to describe local outliers? In essence, different subsets of data have different densities and may be governed by different distributions. A point close to a tight cluster can be a more probable outlier than a point that is further away from a more dispersed cluster. The concept of *local outlier factor* (LOF) that specifies a degree of outlier-ness comes to rescue [Breunig et al. 2000]. The definition is based on the distance to the *k*-nearest neighbor. Knorr et al. [2001] addressed a related problem of how to eliminate outliers in order to compute an appropriate covariance matrix that describes a given locality. To do so, they utilized *Donoho-Stahel estimator* in two-dimensional space.

Crude handling of outliers works surprisingly well in many applications, because the simple truth is that many applications are concerned with systematic patterns. An example is customer segmentation with an objective of a direct mail campaign. On the other hand, philosophically outlier is a non-typical leftover after a regular clustering and, as such, it can easily attain a prominent significance. Therefore, in addition to eliminating negative effects of outliers on clusters construction, there is a separate reason driving interest in outlier detection. The reason is that in some applications, the outlier is the commodity of trade. This is so in medical diagnostics, fraud detection, network security, anomaly detection, and computer immunology. Some connections and further references can be found in [Forrest et al. 1997; Lee & Stolfo 1998; Ghosh et al. 1999]. In CRM, E-commerce, Web-site analytics outliers relate to a concept of *interesting* and *unexpected* [Piatetsky-Shapiro & Matheus 1994; Silberschatz & Tuzhilin 1996; Padmanabhan & Tuzhilin 1999; Padmanabhan & Tuzhilin 2000]. Most of the research in these applications is not directly related to clustering (but to pruning association rules).

Acknowledgements

Cooperation with Jonathan Becher was essential for the appearance of this text. It resulted in numerous discussions and various improvements. I am grateful to Jiawei Han for reading the text and his thoughtful remarks concerning the presentation of the material.

I am very much thankful to Sue Krouscup for her help with text preparation.

References

AGGARWAL, C.C., HINNEBURG, A., and KEIM, D.A. 2000. On the surprising behavior of distance metrics in high dimensional space. *IBM Research report*, RC 21739.

AGGARWAL, C.C., PROCOPIUC, C., WOLF, J.L., YU, P.S., and PARK, J.S. 1999a. Fast algorithms for projected clustering. In *Proceedings of the ACM SIGMOD Conference*, 61-72, Philadelphia, PA.

AGGARWAL, C.C., WOLF, J.L., and YU, P.S. 1999b. A new method for similarity indexing of market basket data. In *Proceedings of the ACM SIGMOD Conference*, 407-418, Philadelphia, PA.

AGGARWAL, C.C. and YU, P.S. 2000. Finding generalized projected clusters in high dimensional spaces. *Sigmod Record*, 29, 2, 70-92.

AGRAWAL, R., FALOUTSOS, C., and SWAMI, A. 1993. Efficient similarity search in sequence databases. In *Proceedings of the 4th International Conference on Foundations of Data Organization and Algorithms*, Chicago, IL.

AGRAWAL, R., GEHRKE, J., GUNOPULOS, D., and RAGHAVAN, P. 1998. Automatic subspace clustering of high dimensional data for data mining applications. In *Proceedings of the ACM SIGMOD Conference*, 94-105, Seattle, WA.

AL-SULTAN, K. 1995. A Tabu search approach to the clustering problem. *Pattern Recognition*, 28, 9, 1443-1451.

ANDERBERG, M. 1973. *Cluster Analysis and Applications*. Academic Press, New York.

ANKERST, M., BREUNIG, M., KRIEGEL, H.-P., and SANDER, J. 1999. OPTICS: Ordering points to identify clustering structure. In *Proceedings of the ACM SIGMOD Conference*, 49-60, Philadelphia, PA.

ARABIE, P. and HUBERT, L.J. 1996. An overview of combinatorial data analysis, in: Arabie, P., Hubert, L.J., and Soete, G.D. (Eds.) *Clustering and Classification*, 5-63, World Scientific Publishing Co., NJ.

ARSLAN, A.N. and EGECIOGLU, O. 2000. Efficient algorithms for normalized edit distance. *Journal of Discrete Algorithms*, 1, 1.

BABU, G.P. and MURTY, M.N. 1993. A near-optimal initial seed value selection in K-means algorithm using a genetic algorithm. *Pattern Recogn. Lett.* 14, 10, 763-169.

BABU, G.P. and MARTY, M.N. 1994. Clustering with evolution strategies. *Pattern Recognition*, 27, 2, 321-329.

BAEZA-YATES, R. 1992. Introduction to data structures and algorithms related to information retrieval. In Frakes, W.B. and Baeza-Yates, R. (Eds.) *Information Retrieval, Data Structures and Algorithms*, 13-27, Prentice-Hall.

BAKER, L.D. and MCCALLUM, A. K. 1998. Distributional clustering of words for text classification. In *Proceedings of the 21st ACM SIGIR Conference*, Melbourne, Australia.

BALL, G. and HALL, D. 1965. ISODATA, a novel method of data analysis and classification. *Technical Report AD-699616*, SRI, Stanford, CA.

BANERJEE, A. and GHOSH, J. 2002. On scaling up balanced clustering algorithms. In *Proceedings of the 2nd SIAM ICDM*, 333-349, Arlington, VA.

BANFIELD, J. and RAFTERY, A. 1993. Model-based Gaussian and non-Gaussian clustering. *Biometrics*, 49, 803-821.

BARBARA, D. and CHEN, P. 2000. Using the fractal dimension to cluster datasets. In *Proceedings of the 6th ACM SIGKDD*, 260-264, Boston, MA.

BECHER, J., BERKHIN, P., and FREEMAN, E. 2000. Automating exploratory data analysis for efficient data mining. In *Proceedings of the 6th ACM SIGKDD*, 424-429, Boston, MA.

BECKMANN, N., KRIEGEL, H-P., SCHNEIDER, R., and SEEGER, B. 1990. The R*-tree: An efficient access method for points and rectangles. In *Proceedings of International Conference on Geographic Information Systems*, Ottawa, Canada.

BERCHTOLD, S., BÖHM, C., and KRIEGEL, H-P. 1998. The Pyramid-technique: towards breaking the curse of dimensionality. In *Proceedings of the ACM SIGMOD Conference*, 142-153, Seattle, WA.

BERKHIN, P. and BECHER, J. 2002. Learning Simple Relations: Theory and Applications. In *Proceedings of the 2nd SIAM ICDM*, 420-436, Arlington, VA.

BEN-DOR, A. and YAKHINI, Z. 1999. Clustering gene expression patterns. In *Proceedings of the 3rd Annual International Conference on Computational Molecular Biology (RECOMB 99)*, 11-14, Lyon, France.

BERRY, M.W. and BROWNE, M. 1999. *Understanding Search Engines: Mathematical Modeling and Text Retrieval*. SIAM.

BERRY, M., DUMAIS, S., LANDAUER, T., and O'BRIEN, G. 1995. Using linear algebra for intelligent information retrieval. *SIAM Review*, 37, 4, 573-595.

BOTTOU, L. and BENGIO, Y. 1995. Convergence properties of the K-means algorithms. In Tesauro, G. and Touretzky, D. (Eds.) *Advances in Neural Information Processing Systems 7*, 585-592, The MIT Press, Cambridge, MA.

BEYER, K., GOLDSTEIN, J., RAMAKRISHNAN, R., and SHAFT, U. 1999. When is nearest neighbor meaningful? In *Proceedings of the* 7th ICDT, Jerusalem, Israel.

BEZDEK, D. 1981. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Plenum Press, New York, NY.

BOCK, H.H. 1996. Probability models in partitional cluster analysis. In Ferligoj, A. and Kramberger, A. (Eds.) *Developments in Data Analysis*, 3-25, Slovenia.

BOLEY, D.L. 1998. Principal direction divisive partitioning. *Data Mining and Knowledge Discovery*, 2, 4, 325-344.

BOZDOGAN, H. 1983. Determining the number of component clusters in the standard multivariate normal mixture model using model-selection criteria. *TR UIC/DQM/A83-1*, Quantitative Methods Department, University of Illinois, Chicago, IL.

BOZDOGAN, H. 1994. Mixture-model cluster analysis using model selection criteria and a new information measure of complexity. In *Proceedings of the 1st US/Japan Conference on the Frontiers of Statistical Modeling: An Informational Approach*, 69-113, Dordrecht, Netherlands.

BRADLEY, P. S., BENNETT, K. P., and DEMIRIZ, A. 2000. Constrained k-means clustering. *Technical Report MSR-TR-2000-65*. Microsoft Research, Redmond, WA.

BRADLEY, P. and FAYYAD, U. 1998. Refining initial points for k-means clustering. In *Proceedings of the 15th ICML*, 91-99, Madison, WI.

BRADLEY, P., FAYYAD, U., and REINA, C. 1998. Scaling clustering algorithms to large databases. In *Proceedings of the 4th ACM SIGKDD*, 9-15, New York, NY.

BREUNIG, M., KRIEGEL, H-P., KROGER, P., and SANDER, J. 2001. Data Bubbles: quality preserving performance boosting for hierarchical clustering. In *Proceedings of the ACM SIGMOD Conference*, Santa Barbara, CA.

BREUNIG, M.M., KRIEGEL, H.-P., NG, R.T., and SANDER, J. 2000. LOF: identifying density-based local outliers. In *Proceedings of the ACM SIGMOD Conference*, 93-104, Dallas, TX.

BROWN, D. and HUNTLEY, C. 1991. A practical application of simulated annealing to clustering. *Technical report IPC-TR-91-003*, University of Virginia.

BUSYGIN, S., JACOBSEN, G., and KRÄMER, E. 2002. Double conjugated clustering applied to leukemia microarray data, *2nd SIAM ICDM, Workshop on clustering high dimensional data*, Arlington, VA.

CADEZ, I., GAFFNEY, S., and SMYTH, P. 2000. A general probabilistic framework for clustering individuals. *Technical Report UCI-ICS 00-09*.

CADEZ, I., SMYTH, P., and MANNILA, H. 2001. Probabilistic modeling of transactional data with applications to profiling, Visualization, and Prediction, In *Proceedings of the 7th ACM SIGKDD*, 37-46, San Francisco, CA.

Carpenter, G.A., Grossberg, S., and Rosen, D.B. 1991. Fuzzy art: Fast stable learning and categorization of analog patterns by an adaptive resonance system. Neural Networks, 4, 759-771.

CHEESEMAN, P. and STUTZ, J. 1996. Bayesian Classification (AutoClass): Theory and Results. In Fayyad, U.M., Piatetsky-Shapiro, G., Smyth, P., and Uthurusamy , R. (Eds.) *Advances in Knowledge Discovery and Data Mining*, AAAI Press/MIT Press.

CHENG, C., FU, A., and ZHANG, Y. 1999. Entropy-based subspace clustering for mining numerical data. In *Proceedings of the 5th ACM SIGKDD*, 84-93, San Diego, CA.

CHIU, T., FANG, D., CHEN, J., and Wang, Y. 2001. A Robust and scalable clustering algorithm for mixed type attributes in large database environments. In *Proceedings of the 7th ACM SIGKDD*, 263-268, San Francisco, CA.

COOLEY, R., MOBASHER, B., and SRIVASTAVA, J. 1999. Data preparation for mining world wide web browsing. *Journal of Knowledge Information Systems*, 1, 1, 5-32.

CORTER, J. and GLUCK, M. 1992. Explaining basic categories: feature predictability and information. *Psychological Bulletin*, 111, 291-303.

COVER, T.M. and THOMAS, J.A. 1990. *Elements of Information Theory*. John Wiley & Sons, New York, NY.

CRISTOFOR, D. and SIMOVICI, D.A. 2002. An information-theoretical approach to clustering categorical databases using genetic algorithms. *2nd SIAM ICDM, Workshop on clustering high dimensional data*, Arlington, VA.

CUTTING, D., KARGER, D., PEDERSEN, J., and TUKEY, J. 1992. Scatter/gather: a cluster-based approach to browsing large document collection. In *Proceedings of the 15th ACM SIGIR Conference*, 318-329, Copenhagen, Denmark.

DANIEL, C. and WOOD, F.C. 1980. *Fitting Equations To Data: Computer Analysis of Multifactor Data*. John Wiley & Sons, New York, NY.

DAY, W. and EDELSBRUNNER, H. 1984. Efficient algorithms for agglomerative hierarchical clustering methods. *Journal of Classification*, 1, 7, 7-24.

DEFAYS, D. 1977. An efficient algorithm for a complete link method. *The Computer Journal*, 20, 364-366.

DEMPSTER, A., LAIRD, N., and RUBIN, D. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, Series B, 39, 1, 1-38.

DHILLON, I. 2001. Co-clustering documents and words using bipartite spectral graph partitioning. In *Proceedings of the 7th ACM SIGKDD*, 269-274, San Francisco, CA.

DHILLON, I., FAN, J., and GUAN, Y. 2001. Efficient clustering of very large document collections. In Grossman, R.L., Kamath, C., Kegelmeyer, P., Kumar, V., and Namburu, R.R. (Eds.) *Data Mining for Scientific and Engineering Applications*, Kluwer Academic Publishers.

DHILLON, I., GUAN, Y., and KOGAN, J. 2002. Refining clusters in high dimensional data. *2nd SIAM ICDM, Workshop on clustering high dimensional data*, Arlington, VA.

DHILLON, I., MALLELA, S., and KUMAR, R. 2002. Enhanced Word Clustering for Hierarchical Text Classification, In *Proceedings of the 8th ACM SIGKDD*, 191-200, Edmonton, Canada.

DHILLON, I. and MODHA, D. 1999. A data clustering algorithm on distributed memory multiprocessors. *5th ACM SIGKDD*, *Large-scale Parallel KDD Systems Workshop*, 245-260, San Diego, CA.

DUBES, R.C. 1993. Cluster Analysis and Related Issues. In Chen, C.H., Pau, L.F., and Wang, P.S. (Eds.) *Handbook of Pattern Recognition and Computer Vision*, 3-32, World Scientific Publishing Co., River Edge, NJ.

DUDA, R. and HART, P. 1973. *Pattern Classification and Scene Analysis*. John Wiley & Sons, New York, NY.

DUMOUCHEL, W., VOLINSKY, C., JOHNSON, T., CORTES, C., and PREGIBON, D. 1999. Squashing flat files flatter. In *Proceedings of the 5th ACM SIGKDD*, 6-15, San Diego, CA.

ENGLEMAN, L. and HARTIGAN, J. 1969. Percentage points of a test for clusters. *Journal of the American Statistical Association*, 64, 1647-1648.

ERTOZ, L., STEINBACH, M., and KUMAR, V. 2002. Finding clusters of different sizes, shapes, and densities in noisy, high dimensional data, *Technical Report*.

ESTER M., FROMMELT A., KRIEGEL H.-P., and SANDER J. 2000. Spatial data mining: database primitives, algorithms and efficient DBMS support. *Data Mining and Knowledge Discovery*, Kluwer Academic Publishers, 4, 2-3, 193-216.

ESTER, M., KRIEGEL, H-P., SANDER, J. and XU, X. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the 2nd ACM SIGKDD*, 226-231, Portland, Oregon.

ESTER, M., KRIEGEL, H-P., and XU, X. 1995. A database interface for clustering in large spatial databases. In *Proceedings of the 1st ACM SIGKDD*, 94-99, Montreal, Canada.

ESTIVILL-CASTRO, V. and LEE, I. 2000. AMOEBA: Hierarchical Clustering Based on Spatial Proximity Using Delaunay Diagram. In *Proceedings of the 9th International Symposium on Spatial Data Handling*. Beijing, China.

EVERITT, B. 1993. *Cluster Analysis* (3rd ed.). Edward Arnold, London, UK.

FALOUTSOS, C. and LIN, K. 1995. Fastmap: A fast algorithm for indexing, data mining and visualization of traditional and multimedia. In *Proceedings of the ACM SIGMOD Conference*, 163-174, San Jose, CA.

FALOUTSOS, C., RANGANATHAN, M., and MANOLOPOULOS, Y. 1994. Fast subsequence matching in time-series databases. In *Proceedings of the ACM SIGMOD Conference*, 419-429, Minneapolis, MN.

FASULO, D. 1999. An analysis of recent work on clustering algorithms. *Technical Report UW-CSE01 -03-02, University of Washington*.

FISHER, D. 1987. Knowledge acquisition via incremental conceptual clustering. *Machine Learning*, 2, 139-172.

FISHER, D. 1996. Iterative optimization and simplification of hierarchical clustering. *Journal of Artificial Intelligence Research*, 4, 147-179.

FORGY, E. 1965. Cluster analysis of multivariate data: Efficiency versus interpretability of classification. *Biometrics*, 21, 768-780.

FORREST, S., HOFMEYR, S.A., and SOMAYAJI, A. 1997. Computer immunology. *Communications of the ACM*, 40, 88-96.

FOSS, A., WANG, W., and ZAANE, O. 2001. A non-parametric approach to Web log analysis. *1st SIAM ICDM*, *Workshop on Web Mining,* 41-50, Chicago, IL.

FRALEY, C. and RAFTERY, A. 1999. MCLUST: Software for model-based cluster and discriminant analysis, *Tech Report 342*, Dept. Statistics, Univ. of Washington.

FRALEY, C. and RAFTERY, A. How many clusters? 1998. Which clustering method? Answers via model-based cluster analysis. *The Computer Journal*, 41, 8, 578-588.

FRIEDMAN, J.H., BENTLEY, J.L., and FINKEL, R.A. 1977. An algorithm for finding best matches in logarithmic expected time. *ACM Trans. Math. Software*, 3, 3, 209-226.

FUKUNAGA, K. 1990. *Introduction to Statistical Pattern Recognition*. Academic Press, San Diego, CA.

GANTI, V., GEHRKE, J. and RAMAKRISHNAN, R. 1999a. CACTUS-Clustering Categorical Data Using Summaries. In *Proceedings of the 5th ACM SIGKDD*, 73-83, San Diego, CA.

GANTI, V., RAMAKRISHNAN, R., GEHRKE, J., POWELL, A., and FRENCH, J. 1999b. Clustering large datasets in arbitrary metric spaces. In *Proceedings of the 15th ICDE*, 502-511, Sydney, Australia.

GENNARI, J., LANGLEY, P., and FISHER, D. 1989. Models of incremental concept formation. *Artificial Intelligence*, 40, 11-61.

GERSHO, A. and GRAY, R. M. 1992. *Vector Quantization and Signal Compression. Communications and Information Theory*. Kluwer Academic Publishers, Norwell, MA.

GHOSH, J., 2002. Scalable Clustering Methods for Data Mining. In Nong Ye (Ed.) *Handbook of Data Mining*, Lawrence Erlbaum, to appear.

GHOSH, A.K., SCHWARTZBARD, A., and SCHATZ. M. 1999. Learning program behavior profiles for intrusion detection. In *Proceedings of the SANS Conference and Workshop on Intrusion Detection and Response*, San Francisco, CA.

GIBSON, D., KLEINBERG, J., and RAGHAVAN, P. 1998. Clustering categorical data: An approach based on dynamic systems. In *Proceedings of the 24th International Conference on Very Large Databases*, 311-323, New York, NY.

GOIL, S., NAGESH, H., and CHOUDHARY, A. 1999. MAFIA: Efficient and scalable subspace clustering for very large data sets. *Technical Report CPDC-TR-9906-010*, Northwestern University.

GOLDBERG, D. 1989. Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley.

GONZALEZ, T.F. 1985. Clustering to minimize the maximum intercluster distance. *Theoretical Computer Science*, 38, 293-306.

GOVAERT, G. 1995. Simultaneous clustering of rows and columns. *Control and Cybernetics*, 24, 437-458.

GOWDA, K.C. and KRISHNA, G. 1978. Agglomerative clustering using the concept of mutual nearest neighborhood. *Pattern Recognition*, 10, 105-112.

GUHA, S., RASTOGI, R., and SHIM, K. 1998. CURE: An efficient clustering algorithm for large databases. In *Proceedings of the ACM SIGMOD Conference*, 73-84, Seattle, WA.

GUHA, S., RASTOGI, R., and SHIM, K. 1999. ROCK: A robust clustering algorithm for categorical attributes. In *Proceedings of the 15th ICDE*, 512-521, Sydney, Australia.

GURALNIK, V. and KARYPIS, G. 2001. A Scalable algorithm for clustering sequential data. *IEEE ICDM 2001*, Silicon Valley, CA.

GUTTMAN, A. 1984. R-trees: a dynamic index structure for spatial searching. In *Proceedings of the ACM SIGMOD Conference*, 47-57, Boston, MA.

HALL, L.O., OZYURT, B., and BEZDEK, J.C. 1999. Clustering with a genetically optimized approach. *IEEE Trans. on Evolutionary Computation*, 3, 2, 103-112.

HAN, E-H., KARYPIS, G., KUMAR, V., and MOBASHER, B. 1997. Clustering based on association rule hypergraphs. *ACM SIGMOD Conference, Data Mining Workshop* (DMKD'97), Tucson, Arizona.

HAN, J. and KAMBER, M. 2001. *Data Mining*. Morgan Kaufmann Publishers.

HAN, J., KAMBER, M., and TUNG, A. K. H. 2001. Spatial clustering methods in data mining: A survey. In Miller, H. and Han, J. (Eds.) *Geographic Data Mining and Knowledge Discovery*, Taylor and Francis.

HAREL, D. and KOREN, Y. 2001. Clustering spatial data using random walks, In *Proceedings of the 7th ACM SIGKDD*, 281-286. San Francisco, CA.

HARTIGAN, J. 1975. *Clustering Algorithms*. John Wiley & Sons, New York, NY.

HARTIGAN, J. and WONG, M. 1979. Algorithm AS136: A k-means clustering algorithm. *Applied Statistics*, 28, 100-108.

HEER, J. and CHI, E. 2001. Identification of Web user traffic composition using multi-modal clustering and information scent. *1st SIAM ICDM*, *Workshop on Web Mining*, 51-58, Chicago, IL.

HINNEBURG, A. and KEIM, D. 1998. An efficient approach to clustering large multimedia databases with noise. In *Proceedings of the 4th ACM SIGKDD*, 58-65, New York, NY.

HINNEBURG, A. and KEIM, D. 1999. Optimal grid-clustering: Towards breaking the curse of dimensionality in high-dimensional clustering. In *Proceedings of the 25th Conference on VLDB*, 506-517, Edinburgh, Scotland.

HUANG, Z. 1998. Extensions to the k-means algorithm for clustering large data sets with categorical values. *Data Mining and Knowledge Discovery*, 2, 3, 283-304.

HULTEN, G., SPENCER, L., and DOMINGOS, P. 2001. Mining time-changing data streams. In *Proceedings of the 7th ACM SIGKDD*, 97-106, San Francisco, CA.

JAIN, A. and DUBES, R. 1988. *Algorithms for Clustering Data*. Prentice-Hall, Englewood Cliffs, NJ.

JAIN, A.K. and FLYNN, P.J. 1966. Image segmentation using clustering. In *Advances in Image Understanding: A Festschrift for Azriel Rosenfeld*, IEEE Press, 65-83.

JAIN, A.K. and MAO, J. 1996. Artificial neural networks: A tutorial. *IEEE Computer*, 29, 3, 31-44.

JAIN, A.K, MURTY, M.N., and FLYNN P.J. 1999. Data clustering: a review. *ACM Computing Surveys*, 31, 3, 264-323.

JARVIS, R.A. and PATRICK, E.A. 1973. Clustering using a similarity measure based on shared nearest neighbors. *IEEE Transactions on Computers*, C-22, 11.

JEBARA, T. and JAAKKOLA, T. 2000. Feature selection and dualities in maximum entropy discrimination. In *Proceedings of the 16th UIA Conference*, Stanford, CA.

JOLIFFE, I. 1986. *Principal Component Analysis*. Springer-Verlag, New York, NY.

KALTON, A., LANGLEY, P., WAGSTAFF, K., and YOO, J. 2001. Generalized clustering, supervised learning, and data assignment. In *Proceedings of the 7th ACM SIGKDD*, 299-304, San Francisco, CA.

KANDOGAN, E. 2001. Visualizing multi-dimensional clusters, trends, and outliers using star coordinates. In *Proceedings of the 7th ACM SIGKDD*, 107-116, San Francisco, CA.

KARYPIS, G., AGGARWAL, R., KUMAR, V., and SHEKHAR, S. 1997. Multilevel hypergraph partitioning: application in VLSI domain, In *Proceedings ACM/IEEE Design Automation Conference.*

KARYPIS, G., HAN, E.-H., and KUMAR, V. 1999a. CHAMELEON: A hierarchical clustering algorithm using dynamic modeling, *COMPUTER*, 32, 68-75.

KARYPIS, G., HAN, E.-H., and KUMAR, V. 1999b. Multilevel refinement for hierarchical clustering. *Technical Report # 99-020*.

KARYPIS, G. and HAN, E.-H. 2000. Concept indexing: A fast dimensionality reduction algorithm with applications to document retrieval & categorization. *Technical Report TR-00-016*, Department of Computer Science, University of Minnesota, Minneapolis.

KARYPIS, G. and KUMAR, V., 1999. A fast and highly quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on Scientific Computing*, 20, 1.

KASS, R. and RAFTERY, A. 1995. Bayes factors. *Journal of Amer. Statistical Association*, 90, 773-795.

KAUFMAN, L. and ROUSSEEUW, P. 1990. *Finding Groups in Data: An Introduction to Cluster Analysis*. John Wiley and Sons, New York, NY.

KEOGH, E., CHAKRABARTI, K., MEHROTRA, S., and PAZZANI, M. 2001. Locally adaptive dimensionality reduction for indexing large time series databases. In *Proceedings of the ACM SIGMOD Conference*, Santa Barbara, CA.

KEOGH, E., CHAKRABARTI, K., PAZZANI M., and MEHROTRA, S. 2001a. Dimensionality reduction for fast similarity search in large time series databases. *Journal of Knowledge and Information Systems*, 3, 3.

KEOGH, E, CHU, S., and PAZZANI, M. 2001b. Ensemble-index: A new approach to indexing large databases. In *Proceedings of the 7th ACM SIGKDD*, 117-125, San Francisco, CA.

KERNIGHAN, B.W. and LIN, S. 1970. An efficient heuristic procedure for partitioning graphs. *The Bell System Technical Journal*, 49, 2, 291-307.

KOHONEN, T. 1990. The self-organizing map. *Proceedings of the IEEE*, 9, 1464-1479.

KOHONEN, T. 2001. *Self-Organizing Maps*. Springer Series in Information Sciences, 30, Springer.

KOLATCH, E. 2001. Clustering Algorithms for Spatial Databases: A Survey. PDF is available on the Web.

KOLLER, D. and SAHAMI, M. 1996. Toward optimal feature selection. In *Proceedings of the 13th ICML*, 284-292, Bari, Italy.

KNORR E. and NG R. 1998. Algorithms for mining distance-based outliers in large datasets. In *Proceedings of the 24h Conference on VLDB*, 392-403, New York, NY.

KNORR, E.M., NG, R.T., and ZAMAR, R.H. 2001. Robust Space Transformations for distance-based operations. In *Proceedings of the 7th ACM SIGKDD*, 126-135, San Francisco, CA.

KRIEGEL H.-P., SEEGER B., SCHNEIDER R., and BECKMANN N. 1990. The R*-tree: an efficient access method for geographic information systems. In *Proceedings International Conference on Geographic Information Systems*, Ottawa, Canada.

KULLBACK, S. and LEIBLER, R.A. 1951. On information and sufficiency. *Annals of Mathematical Statistics*, 22, 76-86.

LANCE, G. and WILLIAMS W. 1967. A general theory of classification sorting strategies. *Computer Journal*, 9, 373-386.

LARSEN, B. and AONE, C. 1999. Fast and effective text mining using linear-time document clustering. In *Proceedings of the 5th ACM SIGKDD*, 16-22, San Diego, CA.

LEE, C-Y. and ANTONSSON, E.K. 2000. Dynamic partitional clustering using evolution strategies. In *Proceedings of the 3rd Asia-Pacific Conference on Simulated Evolution and Learning*, Nagoya, Japan.

LEE, W. and STOLFO, S. 1998. Data mining approaches for intrusion detection. In *Proceedings of the 7th USENIX Security Symposium*, San Antonio, TX.

LIEBOVITCH, L. and TOTH, T. 1989. A fast algorithm to determine fractal dimensions by box counting. *Physics Letters*, 141A, 8.

LIN, D. 1998. An information-theoretic definition of similarity. In *Proceedings of the 15th ICML*, 296-304, Madison, WI.

LIU, B., XIA, Y., and YU, P.S. 2000. Clustering through decision tree construction. In *SIGMOD 2000.*

LIU, H. and SETIONO, R. 1996. A probabilistic approach to feature selection - a filter solution. In *Proceedings of the 13th ICML*, 319-327, Bari, Italy.

MANILLA, H. and RUSAKOV, D. 2001. Decomposition of event sequences into independent components. In *Proceedings of the 1st SIAM ICDM*, Chicago, IL.

MAO, J. and JAIN, A.K. 1996. A Self-organizing network for hyperellipsoidal clustering (HEC). *IEEE Transactions on Neural Networks*, 7, 1, 16-29.

MARDIA, K., KENT, J. and BIBBY, J. 1980. *Multivariate Analysis*. Academic Press.

MARROQUIN, J.L. and GIROSI, F. 1993. Some extensions of the k-means algorithm for image segmentation and pattern classification. *A.I. Memo 1390*. MIT, Cambridge, MA.

MASSART, D. and KAUFMAN, L. 1983. The Interpretation of Analytical Chemical Data by the Use of Cluster Analysis. John Wiley & Sons, New York, NY.

MCCALLUM, A., NIGAM, K., and UNGAR, L.H. 2000. Efficient clustering of high-dimensional data sets with application to reference matching. In *Proceedings of the 6th ACM SIGKDD*, 169-178, Boston, MA.

MCLACHLAN, G. and BASFORD, K. 1988. *Mixture Models: Inference and Applications to Clustering*. Marcel Dekker, New York, NY.

MCLACHLAN, G. and KRISHNAN, T. 1997. *The EM Algorithm and Extensions*. John Wiley & Sons, New York, NY.

MICHALSKI, R.S. and STEPP, R. 1983. Learning from observations: conceptual clustering. In *Machine Learning: An Artificial Intelligence Approach*. San Mateo, CA, Morgan Kaufmann.

MILLIGAN, G. and COOPER, M. 1985. An examination of procedures for determining the number of clusters in a data set. *Psychometrika*, 50, 159-179.

MIRKIN, B. 1996. *Mathematic Classification and Clustering*. Kluwer Academic Publishers.

MITCHELL, T. 1997. *Machine Learning*. McGraw-Hill, New York, NY.

MODHA, D. and SPANGLER, W. 2002. Feature weighting in k-means clustering. *Machine Learning*, 47.

MOORE, A. 1999. Very fast EM-based mixture model clustering using multiresolution kd-trees. *Advances in Neural Information Processing Systems*, 11.

MOTWANI, R. and RAGHAVAN, P. 1995. *Randomized Algorithms*. Cambridge University Press.

MURTAGH, F. 1983. A survey of recent advances in hierarchical clustering algorithms. *Computer Journal*, 26, 4, 354-359.

MURTAGH, F. 1985. *Multidimensional Clustering Algorithms*. Physica-Verlag, Vienna.

NAGESH, H., GOIL, S., and CHOUDHARY, A. 2001. Adaptive grids for clustering massive data sets, In *Proceedings of the 1st SIAM ICDM*, Chicago, IL.

NG, R. and HAN, J. 1994. Efficient and effective clustering methods for spatial data mining. In *Proceedings of the 20th Conference on VLDB*, 144-155, Santiago, Chile.

NISHISATO, S. 1980. *Analysis of Categorical Data: Dual Scaling and Its Applications*. University of Toronto.

OLIVER, J., BAXTER, R. and WALLACE, C. 1996. Unsupervised learning using MML. In *Proceedings of the 13th ICML*, Bari, Italy.

OLSON, C. 1995. Parallel algorithms for hierarchical clustering. *Parallel Computing*, 21, 1313-1325.

OYANAGI, S., KUBOTA, K., and NAKASE, A. 2001. Application of matrix clustering to Web log analysis and access prediction. *7th ACM SIGKDD, WEBKDD Workshop*, San Francisco, CA.

PADMANABHAN, B. and TUZHILIN, A. 1999. Unexpectedness as a measure of interestingness in knowledge discovery. *Decision Support Systems Journal*, 27, 3, 303-318.

PADMANABHAN, B. and TUZHILIN, A. 2000. Small is beautiful: discovering the minimal set of unexpected patterns. In *Proceedings of the 6th ACM SIGKDD*, 54-63, Boston, MA.

PELLEG, D. and MOORE, A. 1999. Accelerating exact k-means algorithms with geometric reasoning. In *Proceedings of the 5th ACM SIGKDD*, 277-281, San Diego, CA.

PELLEG, D. and MOORE, A. 2000. X-means: Extending K-means with Efficient Estimation of the Number of Clusters. In *Proceedings 17th ICML*, Stanford University.

PIATETSKY-SHAPIRO, G. and MATHEUS, C.J. 1994. The interestingness of deviations. In *Proceedings of the AAAI-94 Workshop on Knowledge Discovery in Databases*.

RAMASWAMY, S., RASTOGI, R., and SHIM, K. 2000. Efficient algorithms for mining outliers from large data sets, *Sigmoid Record*, 29, 2, 427-438.

RAND, W.M. 1971. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Assoc*, 66, 846-850.

RESNIK, P. 1995. Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of IJCAI*-95, 448-453, Montreal, Canada.

RISSANEN, J. 1978. Modeling by shortest data description. *Automatica*, 14, 465-471.

RISSANEN J. 1989. *Stochastic Complexity in Statistical Inquiry*. World Scientific Publishing Co., Singapore.

SANDER, J., ESTER, M., KRIEGEL, H.-P., and XU, X. 1998. Density-based clustering in spatial databases: the algorithm GDBSCAN and its applications. In *Data Mining and Knowledge Discovery*, 2, 2, 169-194.

SARAFIS, I., ZALZALA, A.M.S., and TRINDER, P.W. 2002. A genetic rule-based data clustering toolkit. To be published in *Congress on Evolutionary Computation (CEC)*, Honolulu, USA.

SAVARESI, S. and BOLEY, D. 2001. On performance of bisecting k-means and PDDP. In *Proceedings of the 1st SIAM ICDM*, Chicago, IL.

SAVARESI, S.M., BOLEY, D.L., BITTANTI, S., and GAZZANIGA, G. 2002. Cluster Selection in divisive clustering algorithms. In *Proceedings of the 2nd SIAM ICDM*, 299-314, Arlington, VA.

SCHALKOFF, R. 1991. *Pattern Recognition. Statistical, Structural and Neural Approaches*. John Wiley & Sons, New York, NY.

SCHIKUTA, E. 1996. Grid-clustering: a fast hierarchical clustering method for very large data sets. In *Proceedings 13th International Conference on Pattern Recognition,* 2, 101-105.

SCHIKUTA, E., ERHART, M. 1997. The BANG-clustering system: grid-based data analysis. In *Proceeding of Advances in Intelligent Data Analysis, Reasoning about Data, 2nd International Symposium,* 513-524, London, UK.

SCHWARZ, G. 1978. Estimating the dimension of a model. *The Annals of Statistics*, 6, 461-464.

SCOTT, D.W. 1992. *Multivariate Density Estimation*. Wiley, New York, NY.

SHEIKHOLESLAMI, G., CHATTERJEE, S., and ZHANG, A. 1998. WaveCluster: A multi-resolution clustering approach for very large spatial databases. In *Proceedings of the 24th Conference on VLDB*, 428-439, New York, NY.

SIBSON, R. 1973. SLINK: An optimally efficient algorithm for the single link cluster method. *Computer Journal*, 16, 30-34.

SILBERSCHATZ, A. and TUZHILIN, A. 1996. What makes patterns interesting in knowledge discovery systems. *IEEE Trans. on Knowledge and Data Eng.*, 8, 6.

SLONIM, N. and TISHBY, N. 2000. Document clustering using word clusters via the Information Bottleneck Method. In *Proceedings SIGIR*, 208-215.

SLONIM, N. and TISHBY, N. 2001. The power of word clusters for text classification. In *23rd European Colloquium on Information Retrieval Research*.

SMYTH, P. 1998. Model selection for probabilistic clustering using cross-validated likelihood. *ICS Tech Report 98-09*, Statistics and Computing.

SMYTH, P. 1999. Probabilistic model-based clustering of multivariate and sequential data. In *Proceedings of the 7th International Workshop on AI and Statistics*, 299-304.

SPATH H. 1980. *Cluster Analysis Algorithms*. Ellis Horwood, Chichester, England.

STEINBACH, M., KARYPIS, G., and KUMAR, V. 2000. A comparison of document clustering techniques. *6th ACM SIGKDD, World Text Mining Conference*, Boston, MA.

STREHL, A. and GHOSH, J. 2000. A scalable approach to balanced, high-dimensional clustering of market baskets, In *Proceedings of 17th International Conference on High Performance Computing,* Springer LNCS, 525-536, Bangalore, India.

THOMOPOULOS, S., BOUGOULIAS, D., and WANN, C-D. 1995. Dignet: An unsupervised-learning clustering algorithm for clustering and data fusion. *IEEE Trans. on Aerospace and Electr. Systems*, 31, 1, 2,1-38.

TISHBY, N., PEREIRA, F.C., and BIALEK, W. 1999. The information bottleneck method. In *Proceedings of the 37-th Annual Allerton Conference on Communication, Control and Computing*, 368-377.

TUKEY, J.W. 1977. *Exploratory Data Analysis*. Addison-Wesley.

TUNG, A.K.H., HOU, J., and HAN, J. 2001. Spatial clustering in the presence of obstacles. In *Proceedings of the 17th ICDE*, 359-367, Heidelberg, Germany.

TUNG, A.K.H., NG, R.T., LAKSHMANAN, L.V.S., and HAN, J. 2001. Constraint-Based Clustering in Large Databases, In *Proceedings of the* 8th ICDT, London, UK.

VOORHEES, E.M. 1986. Implementing agglomerative hierarchical clustering algorithms for use in document retrieval. *Information Processing and Management*, 22, 6, 465-476.

WALLACE, C. and DOWE, D. 1994. Intrinsic classification by MML – the Snob program. In the *Proceedings of the 7th Australian Joint Conference on Artificial Intelligence*, 37-44, UNE, World Scientific Publishing Co., Armidale, Australia.

WALLACE, C. and FREEMAN, P. 1987. Estimation and inference by compact coding. *Journal of the Royal Statistical Society,* Series B, 49, 3, 240-265.

XU, X., ESTER, M., KRIEGEL, H.-P., and SANDER, J. 1998. A distribution-based clustering algorithm for mining large spatial datasets. In *Proceedings of the 14th ICDE*, 324-331, Orlando, FL.

WANN C.-D. and THOMOPOULOS, S.A. 1997. A Comparative study of self-organizing clustering algorithms Dignet and ART2. *Neural Networks*, 10, 4, 737-743.

WARD, J.H. 1963. Hierarchical grouping to optimize an objective function. *Journal Amer. Stat, Assoc.*, 58, 301, 235-244.

WANG, W., YANG, J., and MUNTZ, R. 1997. STING: a statistical information grid approach to spatialdata mining. In *Proceedings of the 23rd Conference on VLDB*, 186-195, Athens, Greece.

WANG, W., YANG, J., and MUNTZ, R.R. 1998. PK-tree: a spatial index structure for high dimensional point data. In *Proceedings of the 5th International Conference of Foundations of Data Organization*.

WANG, W., YANG, J., and MUNTZ, R.R. 1999. STING+: An approach to active spatial data mining. In *Proceedings 15th ICDE*, 116-125, Sydney, Australia.

XU, X., ESTER, M., KRIEGEL, H.-P., and SANDER, J. 1998. A distribution-based clustering algorithm for mining in large spatial databases. In *Proceedings of the 14th ICDE*, 324-331, Orlando, FL.

YAO, A. 1982. On constructing minimum spanning trees in k-dimensional space and related problems. *SIAM Journal on Computing*, 11, 4, 721-736.

ZHANG, B. 2001. Generalized k-harmonic means – dynamic weighting of data in unsupervised learning. In *Proceedings of the 1st SIAM ICDM*, Chicago, IL.

ZHANG, T., RAMAKRISHNAN, R. and LIVNY, M. 1996. BIRCH: an efficient data clustering method for very large databases. In *Proceedings of the ACM SIGMOD Conference*, 103-114, Montreal, Canada.

ZHANG, T., Ramakrishnan, R., and LIVNY, M. 1997. BIRCH: A new data clustering algorithm and its applications. *Journal of Data Mining and Knowledge Discovery*, 1, 2, 141-182.

ZHANG, Y., FU, A.W., CAI, C.H., and Heng. P.-A. 2000. Clustering categorical data. In *Proceedings of the 16th ICDE*, 305, San Diego, CA.

Journal of Artificial Intelligence Research 16 (2002) 321–357 Submitted 09/01; published 06/02

SMOTE: Synthetic Minority Over-sampling Technique

Nitesh V. Chawla CHAWLA@CSEE.USF.EDU
Department of Computer Science and Engineering, ENB 118
University of South Florida
4202 E. Fowler Ave.
Tampa, FL 33620-5399, USA

Kevin W. Bowyer KWB@CSE.ND.EDU
Department of Computer Science and Engineering
384 Fitzpatrick Hall
University of Notre Dame
Notre Dame, IN 46556, USA

Lawrence O. Hall HALL@CSEE.USF.EDU
Department of Computer Science and Engineering, ENB 118
University of South Florida
4202 E. Fowler Ave.
Tampa, FL 33620-5399, USA

W. Philip Kegelmeyer WPK@CALIFORNIA.SANDIA.GOV
Sandia National Laboratories
Biosystems Research Department, P.O. Box 969, MS 9951
Livermore, CA, 94551-0969, USA

Abstract

An approach to the construction of classifiers from imbalanced datasets is described. A dataset is imbalanced if the classification categories are not approximately equally represented. Often real-world data sets are predominately composed of "normal" examples with only a small percentage of "abnormal" or "interesting" examples. It is also the case that the cost of misclassifying an abnormal (interesting) example as a normal example is often much higher than the cost of the reverse error. Under-sampling of the majority (normal) class has been proposed as a good means of increasing the sensitivity of a classifier to the minority class. This paper shows that a combination of our method of over-sampling the minority (abnormal) class and under-sampling the majority (normal) class can achieve better classifier performance (in ROC space) than only under-sampling the majority class. This paper also shows that a combination of our method of over-sampling the minority class and under-sampling the majority class can achieve better classifier performance (in ROC space) than varying the loss ratios in Ripper or class priors in Naive Bayes. Our method of over-sampling the minority class involves creating synthetic minority class examples. Experiments are performed using C4.5, Ripper and a Naive Bayes classifier. The method is evaluated using the area under the Receiver Operating Characteristic curve (AUC) and the ROC convex hull strategy.

1. Introduction

A dataset is imbalanced if the classes are not approximately equally represented. Imbalance on the order of 100 to 1 is prevalent in fraud detection and imbalance of up to 100,000 to

©2002 AI Access Foundation and Morgan Kaufmann Publishers. All rights reserved.

1 has been reported in other applications (Provost & Fawcett, 2001). There have been attempts to deal with imbalanced datasets in domains such as fraudulent telephone calls (Fawcett & Provost, 1996), telecommunications management (Ezawa, Singh, & Norton, 1996), text classification (Lewis & Catlett, 1994; Dumais, Platt, Heckerman, & Sahami, 1998; Mladenić & Grobelnik, 1999; Lewis & Ringuette, 1994; Cohen, 1995a) and detection of oil spills in satellite images (Kubat, Holte, & Matwin, 1998).

The performance of machine learning algorithms is typically evaluated using predictive accuracy. However, this is not appropriate when the data is imbalanced and/or the costs of different errors vary markedly. As an example, consider the classification of pixels in mammogram images as possibly cancerous (Woods, Doss, Bowyer, Solka, Priebe, & Kegelmeyer, 1993). A typical mammography dataset might contain 98% normal pixels and 2% abnormal pixels. A simple default strategy of guessing the majority class would give a predictive accuracy of 98%. However, the nature of the application requires a fairly high rate of correct detection in the minority class and allows for a small error rate in the majority class in order to achieve this. Simple predictive accuracy is clearly not appropriate in such situations. The Receiver Operating Characteristic (ROC) curve is a standard technique for summarizing classifier performance over a range of tradeoffs between true positive and false positive error rates (Swets, 1988). The Area Under the Curve (AUC) is an accepted traditional performance metric for a ROC curve (Duda, Hart, & Stork, 2001; Bradley, 1997; Lee, 2000). The ROC convex hull can also be used as a robust method of identifying potentially optimal classifiers (Provost & Fawcett, 2001). If a line passes through a point on the convex hull, then there is no other line with the same slope passing through another point with a larger true positive (TP) intercept. Thus, the classifier at that point is optimal under any distribution assumptions in tandem with that slope.

The machine learning community has addressed the issue of class imbalance in two ways. One is to assign distinct costs to training examples (Pazzani, Merz, Murphy, Ali, Hume, & Brunk, 1994; Domingos, 1999). The other is to re-sample the original dataset, either by over-sampling the minority class and/or under-sampling the majority class (Kubat & Matwin, 1997; Japkowicz, 2000; Lewis & Catlett, 1994; Ling & Li, 1998). Our approach (Chawla, Bowyer, Hall, & Kegelmeyer, 2000) blends under-sampling of the majority class with a special form of over-sampling the minority class. Experiments with various datasets and the C4.5 decision tree classifier (Quinlan, 1992), Ripper (Cohen, 1995b), and a Naive Bayes Classifier show that our approach improves over other previous re-sampling, modifying loss ratio, and class priors approaches, using either the AUC or ROC convex hull.

Section 2 gives an overview of performance measures. Section 3 reviews the most closely related work dealing with imbalanced datasets. Section 4 presents the details of our approach. Section 5 presents experimental results comparing our approach to other re-sampling approaches. Section 6 discusses the results and suggests directions for future work.

2. Performance Measures

The performance of machine learning algorithms is typically evaluated by a confusion matrix as illustrated in Figure 1 (for a 2 class problem). The columns are the Predicted class and the rows are the Actual class. In the confusion matrix, TN is the number of negative examples

	Predicted Negative	Predicted Positive
Actual Negative	TN	FP
Actual Positive	FN	TP

Figure 1: Confusion Matrix

correctly classified (True Negatives), FP is the number of negative examples incorrectly classified as positive (False Positives), FN is the number of positive examples incorrectly classified as negative (False Negatives) and TP is the number of positive examples correctly classified (True Positives).

Predictive accuracy is the performance measure generally associated with machine learning algorithms and is defined as $Accuracy = (TP + TN)/(TP + FP + TN + FN)$. In the context of balanced datasets and equal error costs, it is reasonable to use error rate as a performance metric. Error rate is $1 - Accuracy$. In the presence of imbalanced datasets with unequal error costs, it is more appropriate to use the ROC curve or other similar techniques (Ling & Li, 1998; Drummond & Holte, 2000; Provost & Fawcett, 2001; Bradley, 1997; Turney, 1996).

ROC curves can be thought of as representing the family of best decision boundaries for relative costs of TP and FP. On an ROC curve the X-axis represents $\%FP = FP/(TN+FP)$ and the Y-axis represents $\%TP = TP/(TP+FN)$. The ideal point on the ROC curve would be (0,100), that is all positive examples are classified correctly and no negative examples are misclassified as positive. One way an ROC curve can be swept out is by manipulating the balance of training samples for each class in the training set. Figure 2 shows an illustration. The line y = x represents the scenario of randomly guessing the class. Area Under the ROC Curve (AUC) is a useful metric for classifier performance as it is independent of the decision criterion selected and prior probabilities. The AUC comparison can establish a dominance relationship between classifiers. If the ROC curves are intersecting, the total AUC is an average comparison between models (Lee, 2000). However, for some specific cost and class distributions, the classifier having maximum AUC may in fact be suboptimal. Hence, we also compute the ROC convex hulls, since the points lying on the ROC convex hull are potentially optimal (Provost, Fawcett, & Kohavi, 1998; Provost & Fawcett, 2001).

3. Previous Work: Imbalanced datasets

Kubat and Matwin (1997) selectively under-sampled the majority class while keeping the original population of the minority class. They have used the geometric mean as a performance measure for the classifier, which can be related to a single point on the ROC curve. The minority examples were divided into four categories: some noise overlapping the positive class decision region, borderline samples, redundant samples and safe samples. The borderline examples were detected using the Tomek links concept (Tomek, 1976). Another

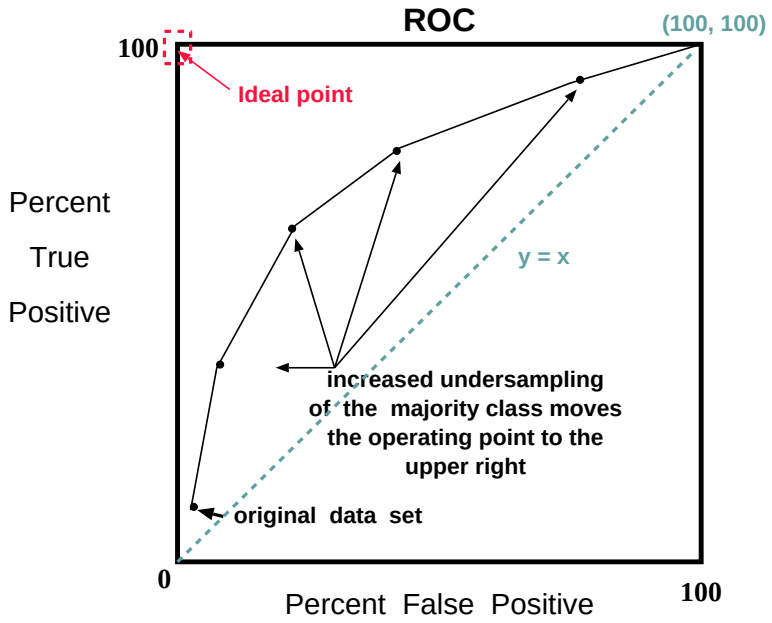


Figure 2: Illustration of sweeping out a ROC curve through under-sampling. Increased under-sampling of the majority (negative) class will move the performance from the lower left point to the upper right.

related work proposed the SHRINK system that classifies an overlapping region of minority (positive) and majority (negative) classes as positive; it searches for the "best positive region" (Kubat et al., 1998).

Japkowicz (2000) discussed the effect of imbalance in a dataset. She evaluated three strategies: under-sampling, resampling and a recognition-based induction scheme. We focus on her sampling approaches. She experimented on artificial 1D data in order to easily measure and construct concept complexity. Two resampling methods were considered. Random resampling consisted of resampling the smaller class at random until it consisted of as many samples as the majority class and "focused resampling" consisted of resampling only those minority examples that occurred on the boundary between the minority and majority classes. Random under-sampling was considered, which involved under-sampling the majority class samples at random until their numbers matched the number of minority class samples; focused under-sampling involved under-sampling the majority class samples lying further away. She noted that both the sampling approaches were effective, and she also observed that using the sophisticated sampling techniques did not give any clear advantage in the domain considered (Japkowicz, 2000).

One approach that is particularly relevant to our work is that of Ling and Li (1998). They combined over-sampling of the minority class with under-sampling of the majority class. They used lift analysis instead of accuracy to measure a classifier's performance. They proposed that the test examples be ranked by a confidence measure and then lift be used as the evaluation criteria. A lift curve is similar to an ROC curve, but is more tailored for the

marketing analysis problem (Ling & Li, 1998). In one experiment, they under-sampled the majority class and noted that the best lift index is obtained when the classes are equally represented (Ling & Li, 1998). In another experiment, they over-sampled the positive (minority) examples with replacement to match the number of negative (majority) examples to the number of positive examples. The over-sampling and under-sampling combination did not provide significant improvement in the lift index. However, our approach to over-sampling differs from theirs.

Solberg and Solberg (1996) considered the problem of imbalanced data sets in oil slick classification from SAR imagery. They used over-sampling and under-sampling techniques to improve the classification of oil slicks. Their training data had a distribution of 42 oil slicks and 2,471 look-alikes, giving a prior probability of 0.98 for look-alikes. This imbalance would lead the learner (without any appropriate loss functions or a methodology to modify priors) to classify almost all look-alikes correctly at the expense of misclassifying many of the oil slick samples (Solberg & Solberg, 1996). To overcome this imbalance problem, they over-sampled (with replacement) 100 samples from the oil slick, and they randomly sampled 100 samples from the non oil slick class to create a new dataset with equal probabilities. They learned a classifier tree on this balanced data set and achieved a 14% error rate on the oil slicks in a leave-one-out method for error estimation; on the look alikes they achieved an error rate of 4% (Solberg & Solberg, 1996).

Another approach that is similar to our work is that of Domingos (1999). He compares the "metacost" approach to each of majority under-sampling and minority over-sampling. He finds that metacost improves over either, and that under-sampling is preferable to minority over-sampling. Error-based classifiers are made cost-sensitive. The probability of each class for each example is estimated, and the examples are relabeled optimally with respect to the misclassification costs. The relabeling of the examples expands the decision space as it creates new samples from which the classifier may learn (Domingos, 1999).

A feed-forward neural network trained on an imbalanced dataset may not learn to discriminate enough between classes (DeRouin, Brown, Fausett, & Schneider, 1991). The authors proposed that the learning rate of the neural network be adapted to the statistics of class representation in the data. They calculated an attention factor from the proportion of samples presented to the neural network for training. The learning rate of the network elements was adjusted based on the attention factor. They experimented on an artificially generated training set and on a real-world training set, both with multiple (more than two) classes. They compared this to the approach of replicating the minority class samples to balance the data set used for training. The classification accuracy on the minority class was improved.

Lewis and Catlett (1994) examined heterogeneous uncertainty sampling for supervised learning. This method is useful for training samples with uncertain classes. The training samples are labeled incrementally in two phases and the uncertain instances are passed on to the next phase. They modified C4.5 to include a loss ratio for determining the class values at the leaves. The class values were determined by comparison with a probability threshold of $LR/(LR+1)$, where LR is the loss ratio (Lewis & Catlett, 1994).

The information retrieval (IR) domain (Dumais et al., 1998; Mladenić & Grobelnik, 1999; Lewis & Ringuette, 1994; Cohen, 1995a) also faces the problem of class imbalance in the dataset. A document or web page is converted into a bag-of-words representation;

that is, a feature vector reflecting occurrences of words in the page is constructed. Usually, there are very few instances of the interesting category in text categorization. This over-representation of the negative class in information retrieval problems can cause problems in evaluating classifiers' performances. Since error rate is not a good metric for skewed datasets, the classification performance of algorithms in information retrieval is usually measured by *precision* and *recall*:

$$recall = \frac{TP}{TP + FN}$$

$$precision = \frac{TP}{TP + FP}$$

Mladenić and Grobelnik (1999) proposed a feature subset selection approach to deal with imbalanced class distribution in the IR domain. They experimented with various feature selection methods, and found that the *odds ratio* (van Rijsbergen, Harper, & Porter, 1981) when combined with a Naive Bayes classifier performs best in their domain. *Odds ratio* is a probabilistic measure used to rank documents according to their relevance to the positive class (minority class). *Information gain* for a word, on the other hand, does not pay attention to a particular target class; it is computed per word for each class. In an imbalanced text dataset (assuming 98 to 99% is the negative class), most of the features will be associated with the negative class. *Odds ratio* incorporates the target class information in its metric giving better results when compared to *information gain* for text categorization.

Provost and Fawcett (1997) introduced the ROC convex hull method to estimate the classifier performance for imbalanced datasets. They note that the problems of unequal class distribution and unequal error costs are related and that little work has been done to address either problem (Provost & Fawcett, 2001). In the ROC convex hull method, the ROC space is used to separate classification performance from the class and cost distribution information.

To summarize the literature, under-sampling the majority class enables better classifiers to be built than over-sampling the minority class. A combination of the two as done in previous work does not lead to classifiers that outperform those built utilizing only under-sampling. However, the over-sampling of the minority class has been done by sampling with replacement from the original data. Our approach uses a different method of over-sampling.

4. SMOTE: Synthetic Minority Over-sampling TEchnique

4.1 Minority over-sampling with replacement

Previous research (Ling & Li, 1998; Japkowicz, 2000) has discussed over-sampling with replacement and has noted that it doesn't significantly improve minority class recognition. We interpret the underlying effect in terms of decision regions in feature space. Essentially, as the minority class is over-sampled by increasing amounts, the effect is to identify similar but more specific regions in the feature space as the decision region for the minority class. This effect for decision trees can be understood from the plots in Figure 3.

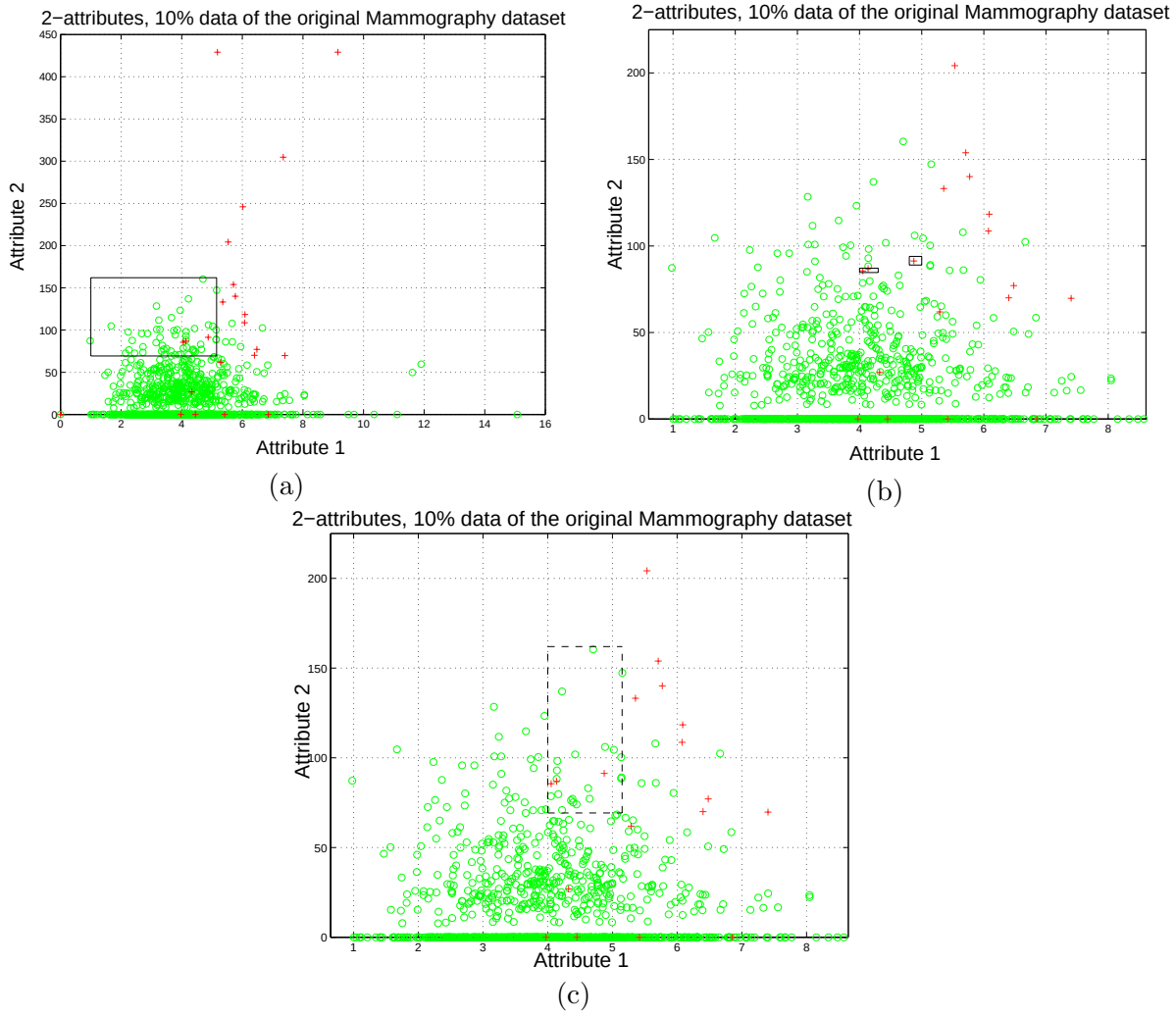


Figure 3: a) Decision region in which the three minority class samples (shown by '+') reside after building a decision tree. This decision region is indicated by the solid-line rectangle. b) A zoomed-in view of the chosen minority class samples for the same dataset. Small solid-line rectangles show the decision regions as a result of over-sampling the minority class with replication. c) A zoomed-in view of the chosen minority class samples for the same dataset. Dashed lines show the decision region after over-sampling the minority class with synthetic generation.

The data for the plot in Figure 3 was extracted from a Mammography dataset[1] (Woods et al., 1993). The minority class samples are shown by + and the majority class samples are shown by *o* in the plot. In Figure 3(a), the region indicated by the solid-line rectangle is a majority class decision region. Nevertheless, it contains three minority class samples shown by '+' as false negatives. If we replicate the minority class, the decision region for the minority class becomes very specific and will cause new splits in the decision tree. This will lead to more terminal nodes (leaves) as the learning algorithm tries to learn more and more specific regions of the minority class; in essence, overfitting. Replication of the minority class does not cause its decision boundary to spread into the majority class region. Thus, in Figure 3(b), the three samples previously in the majority class decision region now have very specific decision regions.

4.2 SMOTE

We propose an over-sampling approach in which the minority class is over-sampled by creating "synthetic" examples rather than by over-sampling with replacement. This approach is inspired by a technique that proved successful in handwritten character recognition (Ha & Bunke, 1997). They created extra training data by performing certain operations on real data. In their case, operations like rotation and skew were natural ways to perturb the training data. We generate synthetic examples in a less application-specific manner, by operating in "feature space" rather than "data space". The minority class is over-sampled by taking each minority class sample and introducing synthetic examples along the line segments joining any/all of the k minority class nearest neighbors. Depending upon the amount of over-sampling required, neighbors from the k nearest neighbors are randomly chosen. Our implementation currently uses five nearest neighbors. For instance, if the amount of over-sampling needed is 200%, only two neighbors from the five nearest neighbors are chosen and one sample is generated in the direction of each. Synthetic samples are generated in the following way: Take the difference between the feature vector (sample) under consideration and its nearest neighbor. Multiply this difference by a random number between 0 and 1, and add it to the feature vector under consideration. This causes the selection of a random point along the line segment between two specific features. This approach effectively forces the decision region of the minority class to become more general.

Algorithm *SMOTE*, on the next page, is the pseudo-code for SMOTE. Table 4.2 shows an example of calculation of random synthetic samples. The amount of over-sampling is a parameter of the system, and a series of ROC curves can be generated for different populations and ROC analysis performed.

The synthetic examples cause the classifier to create larger and less specific decision regions as shown by the dashed lines in Figure 3(c), rather than smaller and more specific regions. More general regions are now learned for the minority class samples rather than those being subsumed by the majority class samples around them. The effect is that decision trees generalize better. Figures 4 and 5 compare the minority over-sampling with replacement and SMOTE. The experiments were conducted on the mammography dataset. There were 10923 examples in the majority class and 260 examples in the minority class originally. We have approximately 9831 examples in the majority class and 233 examples

1. The data is available from the USF Intelligent Systems Lab, http://morden.csee.usf.edu/~chawla.

in the minority class for the training set used in 10-fold cross-validation. The minority class was over-sampled at 100%, 200%, 300%, 400% and 500% of its original size. The graphs show that the tree sizes for minority over-sampling with replacement at higher degrees of replication are much greater than those for SMOTE, and the minority class recognition of the minority over-sampling with replacement technique at higher degrees of replication isn't as good as SMOTE.

Algorithm *SMOTE*(T, N, k)
Input: Number of minority class samples T; Amount of SMOTE N%; Number of nearest neighbors k
Output: $(N/100)$ * T synthetic minority class samples
1. (∗ *If N is less than 100%, randomize the minority class samples as only a random percent of them will be SMOTEd.* ∗)
2. **if** $N < 100$
3. **then** Randomize the T minority class samples
4. $T = (N/100) * T$
5. $N = 100$
6. **endif**
7. $N = (int)(N/100)$ (∗ *The amount of SMOTE is assumed to be in integral multiples of 100.* ∗)
8. k = Number of nearest neighbors
9. $numattrs$ = Number of attributes
10. $Sample[\,][\,]$: array for original minority class samples
11. $newindex$: keeps a count of number of synthetic samples generated, initialized to 0
12. $Synthetic[\,][\,]$: array for synthetic samples
(∗ *Compute k nearest neighbors for each minority class sample only.* ∗)
13. **for** $i \leftarrow 1$ **to** T
14. Compute k nearest neighbors for i, and save the indices in the $nnarray$
15. Populate(N, i, $nnarray$)
16. **endfor**

Populate(N, i, $nnarray$) (∗ *Function to generate the synthetic samples.* ∗)
17. **while** $N \neq 0$
18. Choose a random number between 1 and k, call it nn. This step chooses one of the k nearest neighbors of i.
19. **for** $attr \leftarrow 1$ **to** $numattrs$
20. Compute: $dif = Sample[nnarray[nn]][attr] - Sample[i][attr]$
21. Compute: gap = random number between 0 and 1
22. $Synthetic[newindex][attr] = Sample[i][attr] + gap * dif$
23. **endfor**
24. $newindex$++
25. $N = N - 1$
26. **endwhile**
27. **return** (∗ *End of Populate.* ∗)
End of Pseudo-Code.

Consider a sample (6,4) and let (4,3) be its nearest neighbor.
(6,4) is the sample for which k-nearest neighbors are being identified.
(4,3) is one of its k-nearest neighbors.
Let:
f1_1 = 6 f2_1 = 4 f2_1 - f1_1 = -2
f1_2 = 4 f2_2 = 3 f2_2 - f1_2 = -1
The new samples will be generated as
(f1',f2') = (6,4) + rand(0-1) * (-2,-1)
rand(0-1) generates a random number between 0 and 1.

Table 1: Example of generation of synthetic examples (SMOTE).

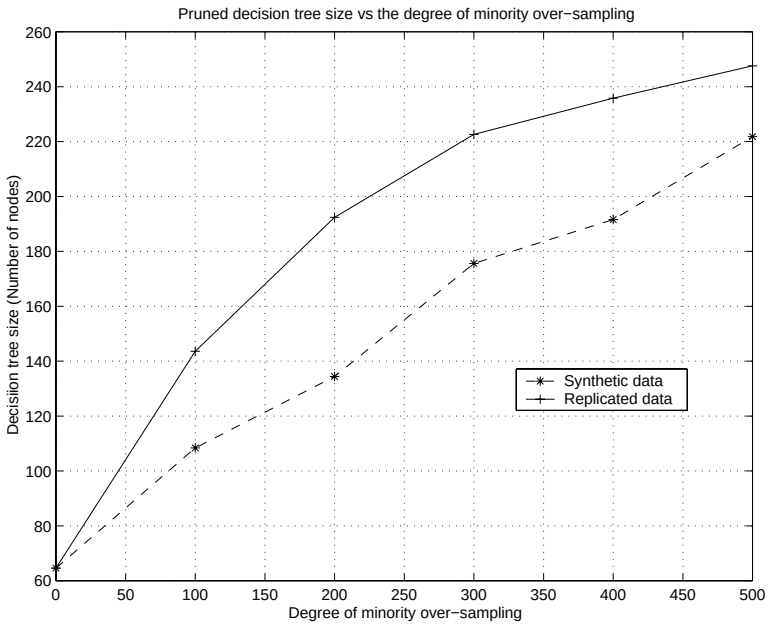


Figure 4: Comparison of decision tree sizes for replicated over-sampling and SMOTE for the Mammography dataset

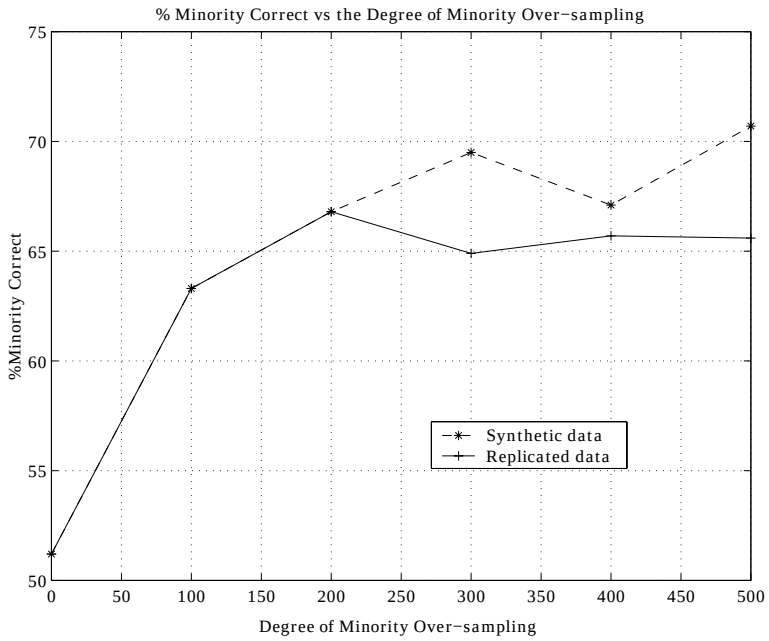


Figure 5: Comparison of % Minority correct for replicated over-sampling and SMOTE for the Mammography dataset

4.3 Under-sampling and SMOTE Combination

The majority class is under-sampled by randomly removing samples from the majority class population until the minority class becomes some specified percentage of the majority class. This forces the learner to experience varying degrees of under-sampling and at higher degrees of under-sampling the minority class has a larger presence in the training set. In describing our experiments, our terminology will be such that if we *under-sample the majority class at 200%*, it would mean that the modified dataset will contain *twice as many elements from the minority class as from the majority class*; that is, if the minority class had 50 samples and the majority class had 200 samples and we under-sample majority at 200%, the majority class would end up having 25 samples. By applying a combination of under-sampling and over-sampling, the initial bias of the learner towards the negative (majority) class is reversed in the favor of the positive (minority) class. Classifiers are learned on the dataset perturbed by "SMOTING" the minority class and under-sampling the majority class.

5. Experiments

We used three different machine learning algorithms for our experiments. Figure 6 provides an overview of our experiments.

1. **C4.5:** We compared various combinations of SMOTE and under-sampling with plain under-sampling using C4.5 release 8 (Quinlan, 1992) as the base classifier.

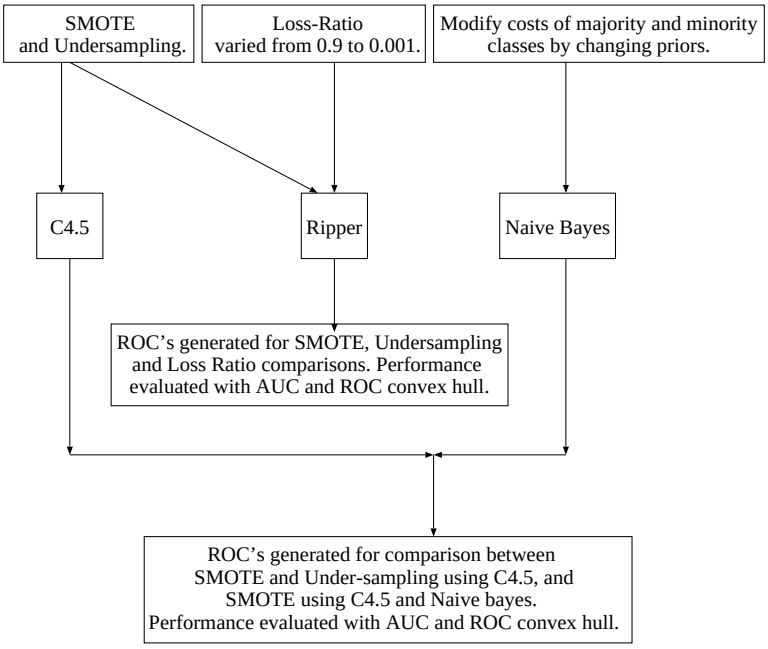


Figure 6: Experiments Overview

2. **Ripper:** We compared various combinations of SMOTE and under-sampling with plain under-sampling using Ripper (Cohen, 1995b) as the base classifier. We also varied Ripper's loss ratio (Cohen & Singer, 1996; Lewis & Catlett, 1994) from 0.9 to 0.001 (as a means of varying misclassification cost) and compared the effect of this variation with the combination of SMOTE and under-sampling. By reducing the loss ratio from 0.9 to 0.001 we were able to build a set of rules for the minority class.

3. **Naive Bayes Classifier:** The Naive Bayes Classifier[2] can be made cost-sensitive by varying the priors of the minority class. We varied the priors of the minority class from 1 to 50 times the majority class and compared with C4.5's SMOTE and under-sampling combination.

These different learning algorithms allowed SMOTE to be compared to some methods that can handle misclassification costs directly. %FP and %TP were averaged over 10-fold cross-validation runs for each of the data combinations. The minority class examples were over-sampled by calculating the five nearest neighbors and generating synthetic examples. The AUC was calculated using the trapezoidal rule. We extrapolated an extra point of TP = 100% and FP = 100% for each ROC curve. We also computed the ROC convex hull to identify the optimal classifiers, as the points lying on the hull are potentially optimal classifiers (Provost & Fawcett, 2001).

2. The source code was downloaded from http://fuzzy.cs.uni-magdeburg.de/~borgelt/software.html.

5.1 Datasets

We experimented on nine different datasets. These datasets are summarized in Table 5.2. These datasets vary extensively in their size and class proportions, thus offering different domains for SMOTE. In order of increasing imbalance they are:

1. The Pima Indian Diabetes (Blake & Merz, 1998) has 2 classes and 768 samples. The data is used to identify the positive diabetes cases in a population near Phoenix, Arizona. The number of positive class samples is only 268. Good sensitivity to detection of diabetes cases will be a desirable attribute of the classifier.

2. The Phoneme dataset is from the ELENA project[3]. The aim of the dataset is to distinguish between nasal (class 0) and oral sounds (class 1). There are 5 features. The class distribution is 3,818 samples in class 0 and 1,586 samples in class 1.

3. The Adult dataset (Blake & Merz, 1998) has 48,842 samples with 11,687 samples belonging to the minority class. This dataset has 6 continuous features and 8 nominal features. SMOTE and SMOTE-NC (see Section 6.1) algorithms were evaluated on this dataset. For SMOTE, we extracted the continuous features and generated a new dataset with only continuous features.

4. The E-state data[4] (Hall, Mohney, & Kier, 1991) consists of electrotopological state descriptors for a series of compounds from the National Cancer Institute's Yeast Anti-Cancer drug screen. E-state descriptors from the NCI Yeast AntiCancer Drug Screen were generated by Tripos, Inc. Briefly, a series of about 60,000 compounds were tested against a series of 6 yeast strains at a given concentration. The test was a high-throughput screen at only one concentration so the results are subject to contamination, etc. The growth inhibition of the yeast strain when exposed to the given compound (with respect to growth of the yeast in a neutral solvent) was measured. The activity classes are either active — at least one single yeast strain was inhibited more than 70%, or inactive — no yeast strain was inhibited more than 70%. The dataset has 53,220 samples with 6,351 samples of active compounds.

5. The Satimage dataset (Blake & Merz, 1998) has 6 classes originally. We chose the smallest class as the minority class and collapsed the rest of the classes into one as was done in (Provost et al., 1998). This gave us a skewed 2-class dataset, with 5809 majority class samples and 626 minority class samples.

6. The Forest Cover dataset is from the UCI repository (Blake & Merz, 1998). This dataset has 7 classes and 581,012 samples. This dataset is for the prediction of forest cover type based on cartographic variables. Since our system currently works for binary classes we extracted data for two classes from this dataset and ignored the rest. Most other approaches only work for only two classes (Ling & Li, 1998; Japkowicz, 2000; Kubat & Matwin, 1997; Provost & Fawcett, 2001). The two classes we considered are Ponderosa Pine with 35,754 samples and Cottonwood/Willow with 2,747

3. ftp.dice.ucl.ac.be in the directory pub/neural-nets/ELENA/databases.
4. We would like to thank Steven Eschrich for providing the dataset and description to us.

Dataset	Majority Class	Minority Class
Pima	500	268
Phoneme	3818	1586
Adult	37155	11687
E-state	46869	6351
Satimage	5809	626
Forest Cover	35754	2747
Oil	896	41
Mammography	10923	260
Can	435512	8360

Table 2: Dataset distribution

samples. Nevertheless, the SMOTE technique can be applied to a multiple class problem as well by specifying what class to SMOTE for. However, in this paper, we have focused on 2-classes problems, to explicitly represent positive and negative classes.

7. The Oil dataset was provided by Robert Holte and is used in their paper (Kubat et al., 1998). This dataset has 41 oil slick samples and 896 non-oil slick samples.

8. The Mammography dataset (Woods et al., 1993) has 11,183 samples with 260 calcifications. If we look at predictive accuracy as a measure of goodness of the classifier for this case, the default accuracy would be 97.68% when every sample is labeled non-calcification. But, it is desirable for the classifier to predict most of the calcifications correctly.

9. The Can dataset was generated from the Can ExodusII data using the AVATAR (Chawla & Hall, 1999) version of the Mustafa Visualization tool[5]. The portion of the can being crushed was marked as "very interesting" and the rest of the can was marked as "unknown." A dataset of size 443,872 samples with 8,360 samples marked as "very interesting" was generated.

5.2 ROC Creation

A ROC curve for SMOTE is produced by using C4.5 or Ripper to create a classifier for each one of a series of modified training datasets. A given ROC curve is produced by first over-sampling the minority class to a specified degree and then under-sampling the majority class at increasing degrees to generate the successive points on the curve. The amount of under-sampling is identical to plain under-sampling. So, each corresponding point on each ROC curve for a dataset represents the same number of majority class samples. Different ROC curves are produced by starting with different levels of minority over-sampling. ROC curves were also generated by varying the loss ratio in Ripper from 0.9 to 0.001 and by varying the priors of the minority class from the original distribution to up to 50 times the majority class for a Naive Bayes Classifier.

5. The Mustafa visualization tool was developed by Mike Glass of Sandia National Labs.

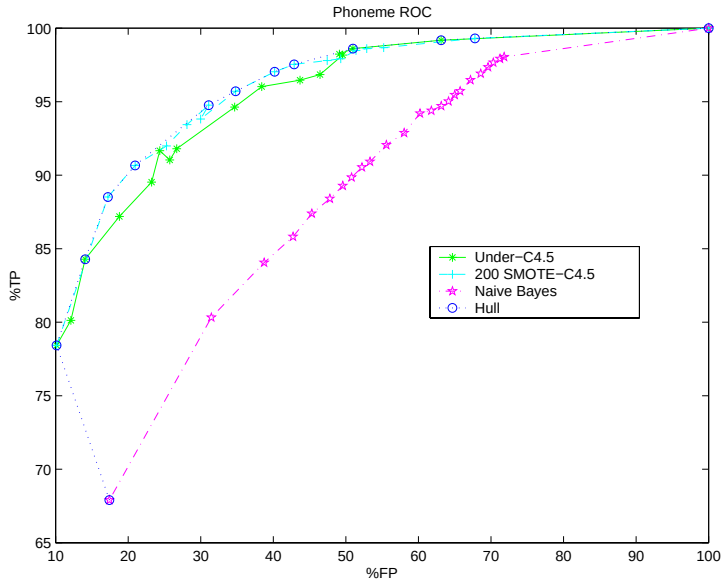


Figure 7: Phoneme. Comparison of SMOTE-C4.5, Under-C4.5, and Naive Bayes. SMOTE-C4.5 dominates over Naive Bayes and Under-C4.5 in the ROC space. SMOTE-C4.5 classifiers are potentially optimal classifiers.

Figures 9 through 23 show the experimental ROC curves obtained for the nine datasets with the three classifiers. The ROC curve for plain under-sampling of the majority class (Ling & Li, 1998; Japkowicz, 2000; Kubat & Matwin, 1997; Provost & Fawcett, 2001) is compared with our approach of combining synthetic minority class over-sampling (SMOTE) with majority class under-sampling. The plain under-sampling curve is labeled "Under", and the SMOTE and under-sampling combination ROC curve is labeled "SMOTE". Depending on the size and relative imbalance of the dataset, one to five SMOTE and under-sampling curves are created. We only show the best results from SMOTE combined with under-sampling and the plain under-sampling curve in the graphs. The SMOTE ROC curve from C4.5 is also compared with the ROC curve obtained from varying the priors of minority class using a Naive Bayes classifier — labeled as "Naive Bayes". "SMOTE", "Under", and "Loss Ratio" ROC curves, generated using Ripper are also compared. For a given family of ROC curves, an ROC convex hull (Provost & Fawcett, 2001) is generated. The ROC convex hull is generated using the Graham's algorithm (O'Rourke, 1998). For reference, we show the ROC curve that would be obtained using minority over-sampling by replication in Figure 19.

Each point on the ROC curve is the result of either a classifier (C4.5 or Ripper) learned for a particular combination of under-sampling and SMOTE, a classifier (C4.5 or Ripper) learned with plain under-sampling, or a classifier (Ripper) learned using some loss ratio or a classifier (Naive Bayes) learned for a different prior for the minority class. Each point represents the average (%TP and %FP) 10-fold cross-validation result. The lower leftmost point for a given ROC curve is from the raw dataset, without any majority class under-

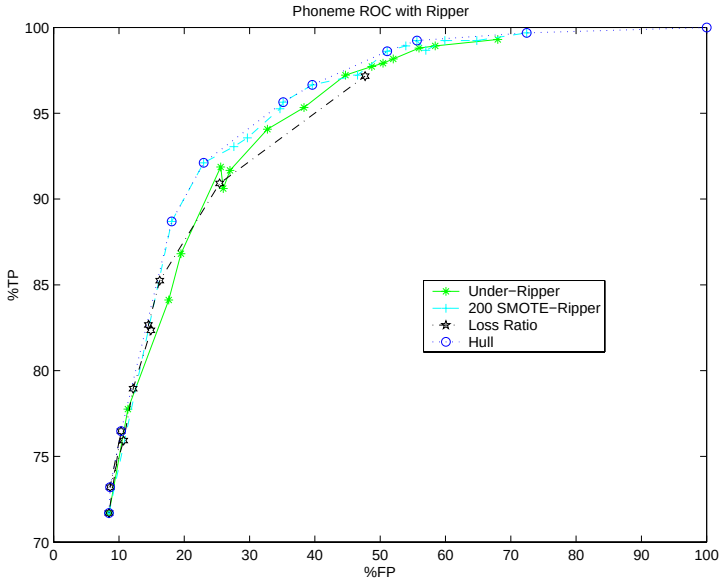


Figure 8: Phoneme. Comparison of SMOTE-Ripper, Under-Ripper, and modifying Loss Ratio in Ripper. SMOTE-Ripper dominates over Under-Ripper and Loss Ratio in the ROC space. More SMOTE-Ripper classifiers lie on the ROC convex hull.

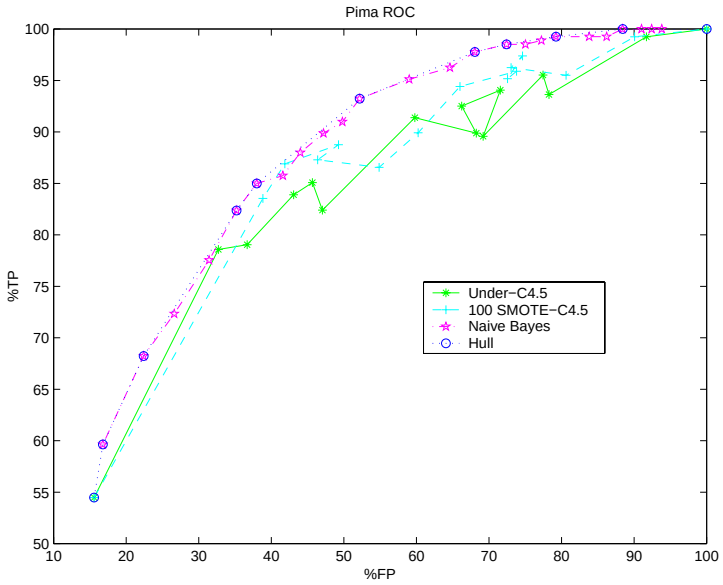


Figure 9: Pima Indians Diabetes. Comparison of SMOTE-C4.5, Under-C4.5, and Naive Bayes. Naive Bayes dominates over SMOTE-C4.5 in the ROC space.

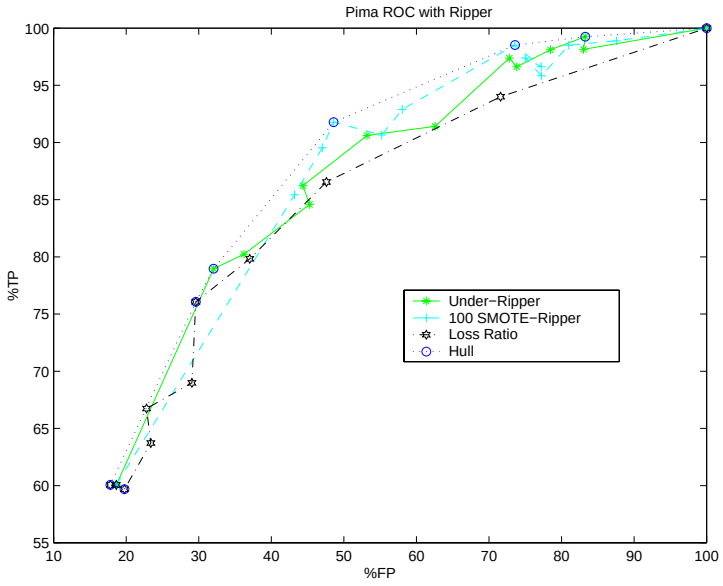


Figure 10: Pima Indians Diabetes. Comparison of SMOTE-Ripper, Under-Ripper, and modifying Loss Ratio in Ripper. SMOTE-Ripper dominates over Under-Ripper and Loss Ratio in the ROC space.

sampling or minority class over-sampling. The minority class was over-sampled at 50%, 100%, 200%, 300%, 400%, 500%. The majority class was under-sampled at 10%, 15%, 25%, 50%, 75%, 100%, 125%, 150%, 175%, 200%, 300%, 400%, 500%, 600%, 700%, 800%, 1000%, and 2000%. The amount of majority class under-sampling and minority class over-sampling depended on the dataset size and class proportions. For instance, consider the ROC curves in Figure 17 for the mammography dataset. There are three curves — one for plain majority class under-sampling in which the range of under-sampling is varied between 5% and 2000% at different intervals, one for a combination of SMOTE and majority class under-sampling, and one for Naive Bayes — and one ROC convex hull curve. The ROC curve shown in Figure 17 is for the minority class over-sampled at 400%. Each point on the SMOTE ROC curves represents a combination of (synthetic) over-sampling and under-sampling, the amount of under-sampling follows the same range as for plain under-sampling. For a better understanding of the ROC graphs, we have shown different sets of ROC curves for one of our datasets in Appendix A.

For the Can dataset, we had to SMOTE to a lesser degree than for the other datasets due to the structural nature of the dataset. For the Can dataset there is a structural neighborhood already established in the mesh geometry, so SMOTE can lead to creating neighbors which are under the surface (and hence not interesting), since we are looking at the feature space of physics variables and not the structural information.

The ROC curves show a trend that as we increase the amount of under-sampling coupled with over-sampling, our minority classification accuracy increases, of course at the expense of more majority class errors. For almost all the ROC curves, the SMOTE approach dom-

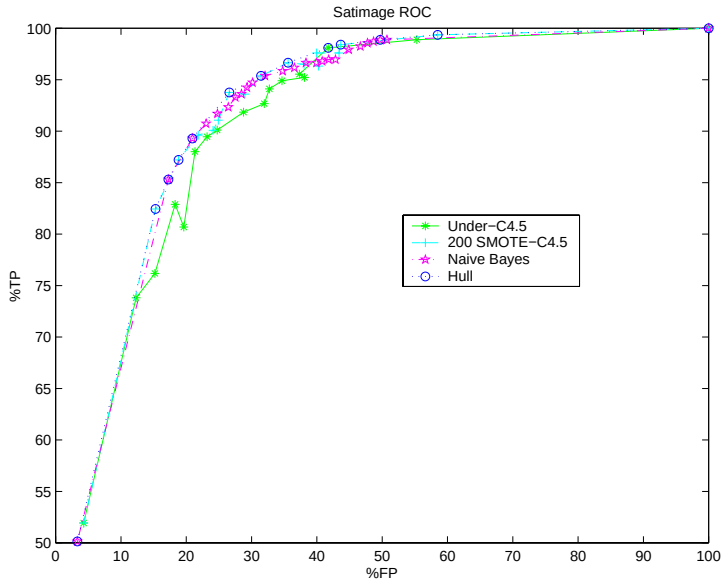


Figure 11: Satimage. Comparison of SMOTE-C4.5, Under-C4.5, and Naive Bayes. The ROC curves of Naive Bayes and SMOTE-C4.5 show an overlap; however, at higher TP's more points from SMOTE-C4.5 lie on the ROC convex hull.

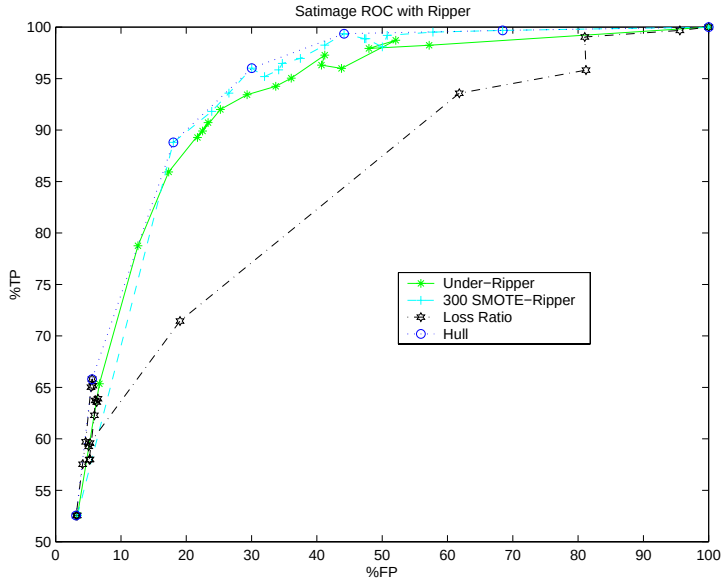


Figure 12: Satimage. Comparison of SMOTE-Ripper, Under-Ripper, and modifying Loss Ratio in Ripper. SMOTE-Ripper dominates the ROC space. The ROC convex hull is mostly constructed with points from SMOTE-Ripper.

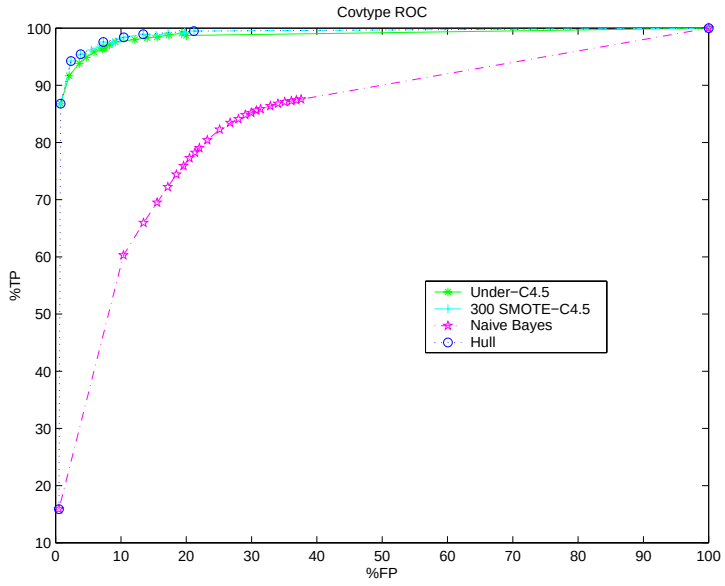


Figure 13: Forest Cover. Comparison of SMOTE-C4.5, Under-C4.5, and Naive Bayes. SMOTE-C4.5 and Under-C4.5 ROC curves are very close to each other. However, more points from the SMOTE-C4.5 ROC curve lie on the ROC convex hull, thus establishing a dominance.

inates. Adhering to the definition of ROC convex hull, most of the potentially optimal classifiers are the ones generated with SMOTE.

5.3 AUC Calculation

The Area Under the ROC curve (AUC) is calculated using a form of the trapezoid rule. The lower leftmost point for a given ROC curve is a classifier's performance on the raw data. The upper rightmost point is always (100%, 100%). If the curve does not naturally end at this point, the point is added. This is necessary in order for the AUC's to be compared over the same range of %FP.

The AUCs listed in Table 5.3 show that for all datasets the combined synthetic minority over-sampling and majority over-sampling is able to improve over plain majority under-sampling with C4.5 as the base classifier. Thus, our SMOTE approach provides an improvement in correct classification of data in the underrepresented class. The same conclusion holds from an examination of the ROC convex hulls. Some of the entries are missing in the table, as SMOTE was not applied at the same amounts to all datasets. The amount of SMOTE was less for less skewed datasets. Also, we have not included AUC's for Ripper/Naive Bayes. The ROC convex hull identifies SMOTE classifiers to be potentially optimal as compared to plain under-sampling or other treatments of misclassification costs, generally. Exceptions are as follows: for the Pima dataset, Naive Bayes dominates over SMOTE-C4.5; for the Oil dataset, Under-Ripper dominates over SMOTE-Ripper. For the Can dataset, SMOTE-*classifier* (*classifier* = C4.5 or Ripper) and Under-*classifier* ROC

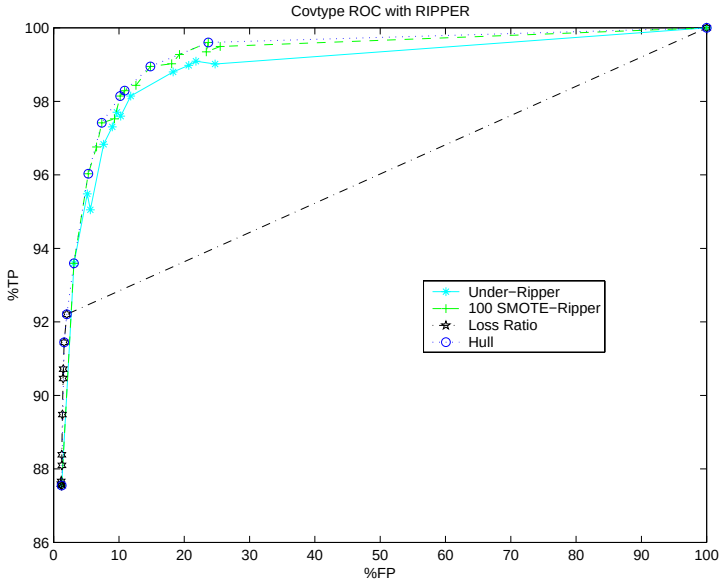


Figure 14: Forest Cover. Comparison of SMOTE-Ripper, Under-Ripper, and modifying Loss Ratio in Ripper. SMOTE-Ripper shows a domination in the ROC space. More points from SMOTE-Ripper curve lie on the ROC convex hull.

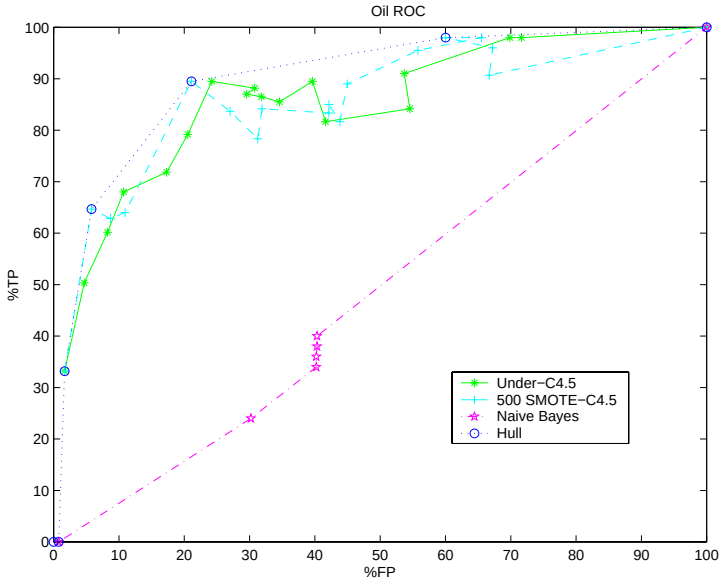


Figure 15: Oil. Comparison of SMOTE-C4.5, Under-C4.5, and Naive Bayes. Although, SMOTE-C4.5 and Under-C4.5 ROC curves intersect at points, more points from SMOTE-C4.5 curve lie on the ROC convex hull.

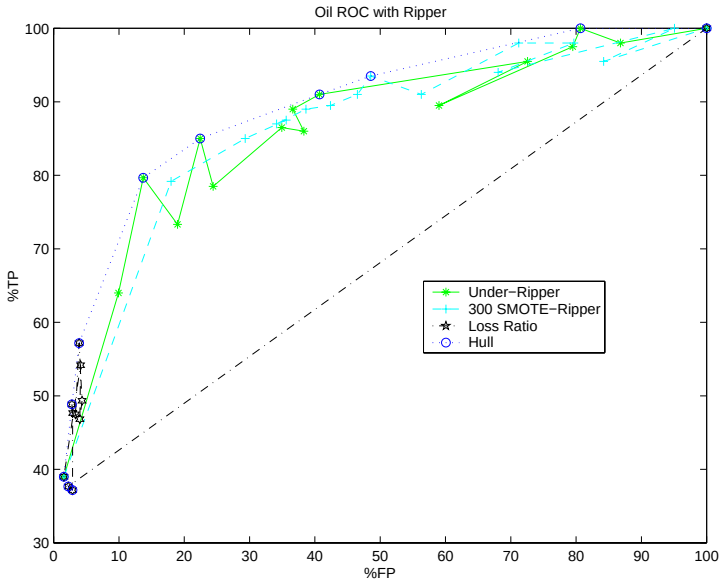


Figure 16: Oil. Comparison of SMOTE-Ripper, Under-Ripper, and modifying Loss Ratio in Ripper. Under-Ripper and SMOTE-Ripper curves intersect, and more points from the Under-Ripper curve lie on the ROC convex hull.

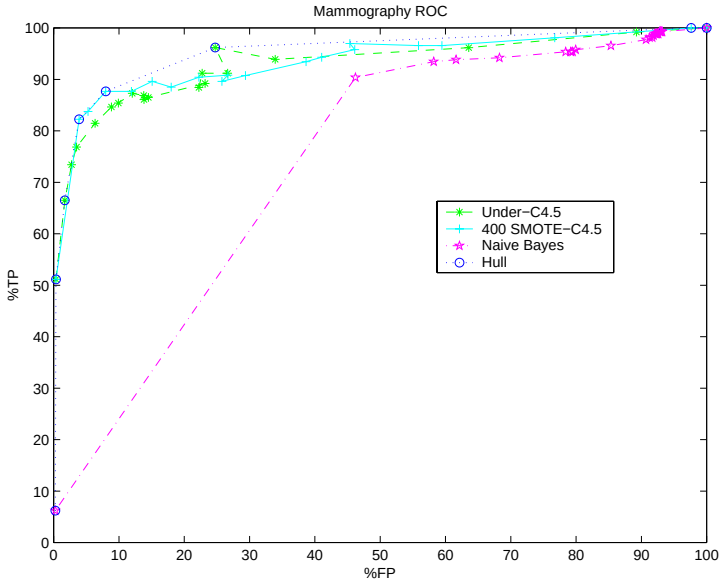


Figure 17: Mammography. Comparison of SMOTE-C4.5, Under-C4.5, and Naive Bayes. SMOTE-C4.5 and Under-C4.5 curves intersect in the ROC space; however, by virtue of number of points on the ROC convex hull, SMOTE-C4.5 has more potentially optimal classifiers.

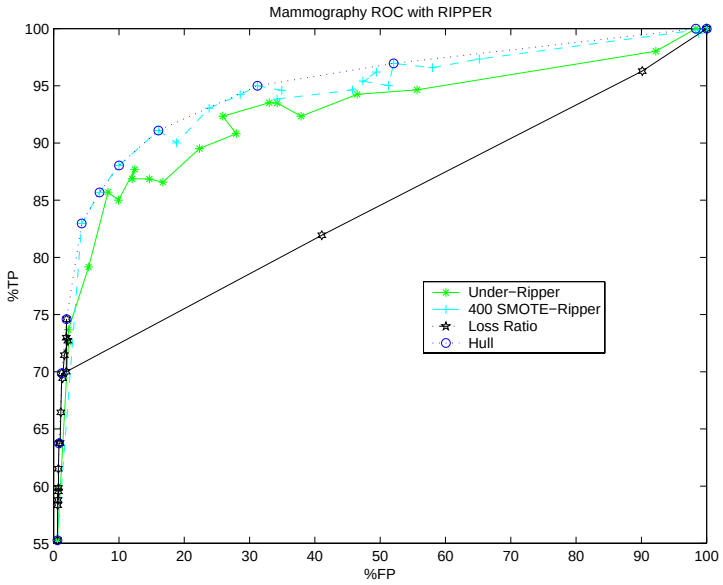


Figure 18: Mammography. Comparison of SMOTE-Ripper, Under-Ripper, and modifying Loss Ratio in Ripper. SMOTE-Ripper dominates the ROC space for TP $> 75\%$.

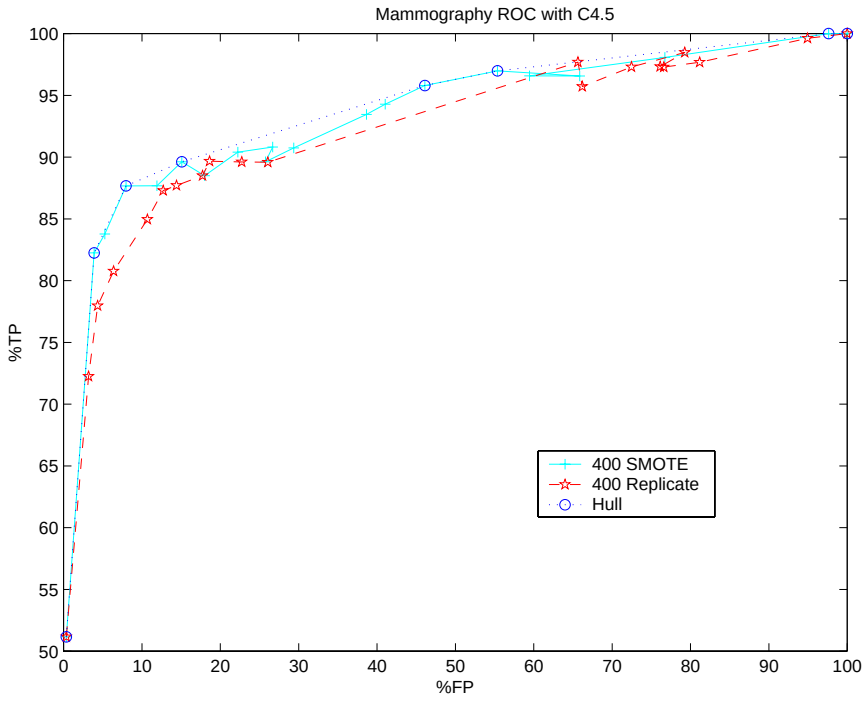


Figure 19: A comparison of over-sampling minority class examples by SMOTE and over-sampling the minority class examples by replication for the Mammography dataset.

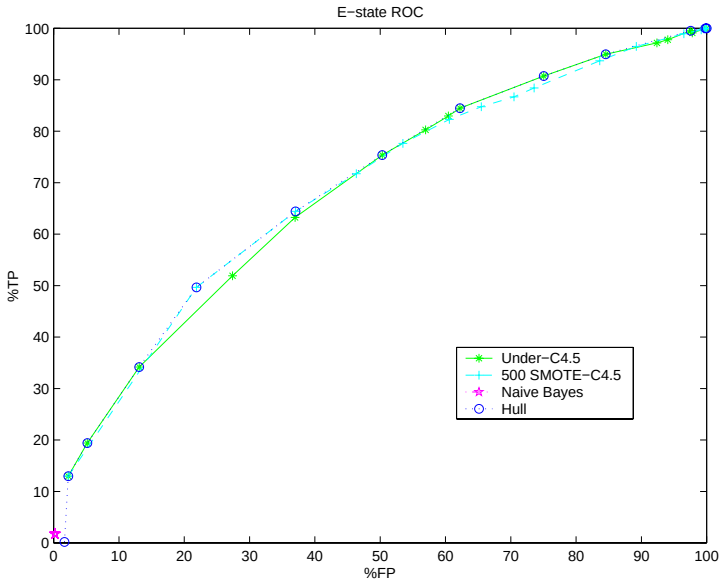


Figure 20: E-state. (a) Comparison of SMOTE-C4.5, Under-C4.5, and Naive Bayes. SMOTE-C4.5 and Under-C4.5 curves intersect in the ROC space; however, SMOTE-C4.5 has more potentially optimal classifiers, based on the number of points on the ROC convex hull.

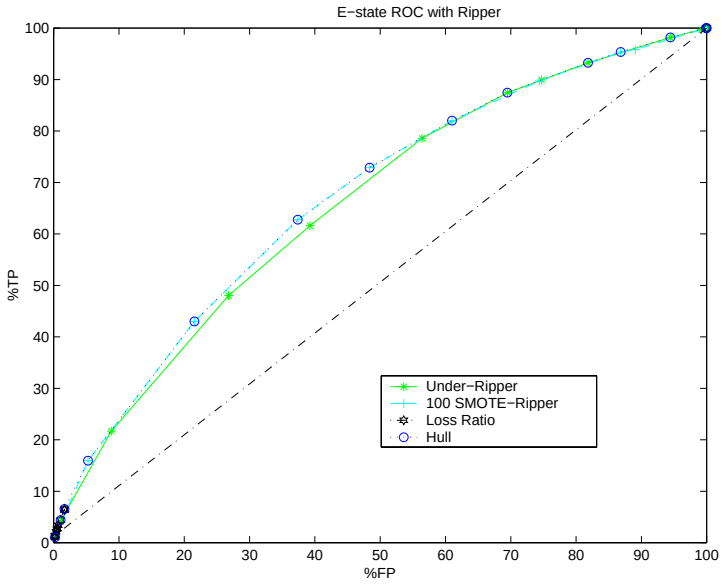


Figure 21: E-state. Comparison of SMOTE-Ripper, Under-Ripper, and modifying Loss Ratio in Ripper. SMOTE-Ripper has more potentially optimal classifiers, based on the number of points on the ROC convex hull.

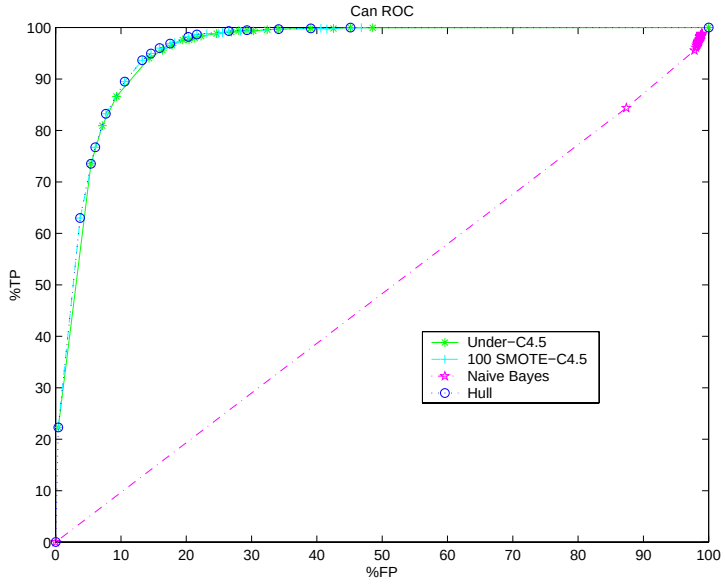


Figure 22: Can. Comparison of SMOTE-C4.5, Under-C4.5, and Naive Bayes. SMOTE-C4.5 and Under-C4.5 ROC curves overlap for most of the ROC space.

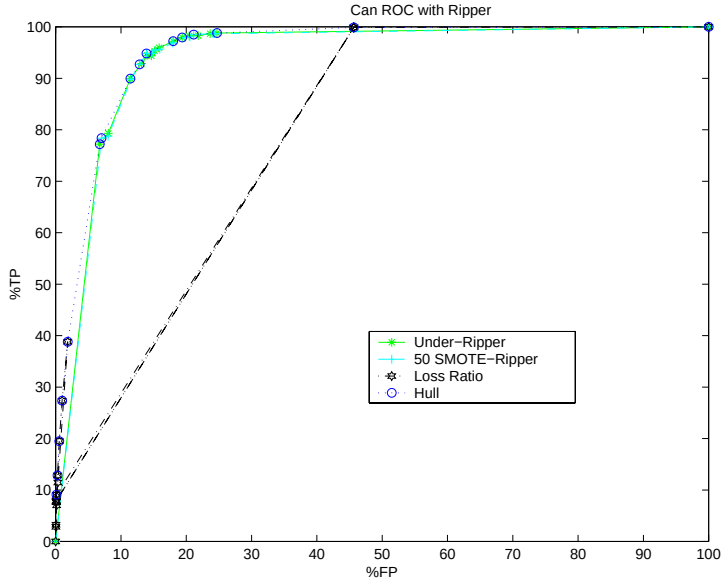


Figure 23: Can. Comparison of SMOTE-Ripper, Under-Ripper, and modifying Loss Ratio in Ripper. SMOTE-Ripper and Under-Ripper ROC curves overlap for most of the ROC space.

Dataset	Under	50 SMOTE	100 SMOTE	200 SMOTE	300 SMOTE	400 SMOTE	500 SMOTE
Pima	7242		**7307**				
Phoneme	8622		8644	**8661**			
Satimage	8900		8957	**8979**	8963	8975	8960
Forest Cover	9807		9832	9834	**9849**	9841	9842
Oil	8524		8523	8368	8161	8339	**8537**
Mammography	9260		9250	9265	9311	**9330**	9304
E-state	6811		6792	**6828**	6784	6788	6779
Can	9535	**9560**	9505	9505	9494	9472	9470

Table 3: AUC's [C4.5 as the base classifier] with the best highlighted in bold.

curves overlap in the ROC space. For all the other datasets, SMOTE-*classifier* has more potentially optimal classifiers than any other approach.

5.4 Additional comparison to changing the decision thresholds

Provost (2000) suggested that simply changing the decision threshold should always be considered as an alternative to more sophisticated approaches. In the case of C4.5, this would mean changing the decision threshold at the leaves of the decision trees. For example, a leaf could classify examples as the minority class even if more than 50% of the training examples at the leaf represent the majority class. We experimented by setting the decision thresholds at the leaves for the C4.5 decision tree learner at 0.5, 0.45, 0.42, 0.4, 0.35, 0.32, 0.3, 0.27, 0.25, 0.22, 0.2, 0.17, 0.15, 0.12, 0.1, 0.05, 0.0. We experimented on the Phoneme dataset. Figure 24 shows the comparison of the SMOTE and under-sampling combination against C4.5 learning by tuning the bias towards the minority class. The graph shows that the SMOTE and under-sampling combination ROC curve is dominating over the entire range of values.

5.5 Additional comparison to one-sided selection and SHRINK

For the oil dataset, we also followed a slightly different line of experiments to obtain results comparable to (Kubat et al., 1998). To alleviate the problem of imbalanced datasets the authors have proposed (a) one-sided selection for under-sampling the majority class (Kubat & Matwin, 1997) and (b) the SHRINK system (Kubat et al., 1998). Table 5.5 contains the results from (Kubat et al., 1998). Acc+ is the accuracy on positive (minority) examples and Acc− is the accuracy on the negative (majority) examples. Figure 25 shows the trend for Acc+ and Acc− for one combination of the SMOTE strategy and varying degrees of under-sampling of the majority class. The Y-axis represents the accuracy and the X-axis represents the percentage majority class under-sampled. The graphs indicate that in the band of under-sampling between 50% and 125% the results are comparable to those achieved by SHRINK and better than SHRINK in some cases. Table 5.5 summarizes the results for the SMOTE at 500% and under-sampling combination. We also tried combinations of SMOTE at 100-400% and varying degrees of under-sampling and achieved comparable results. The

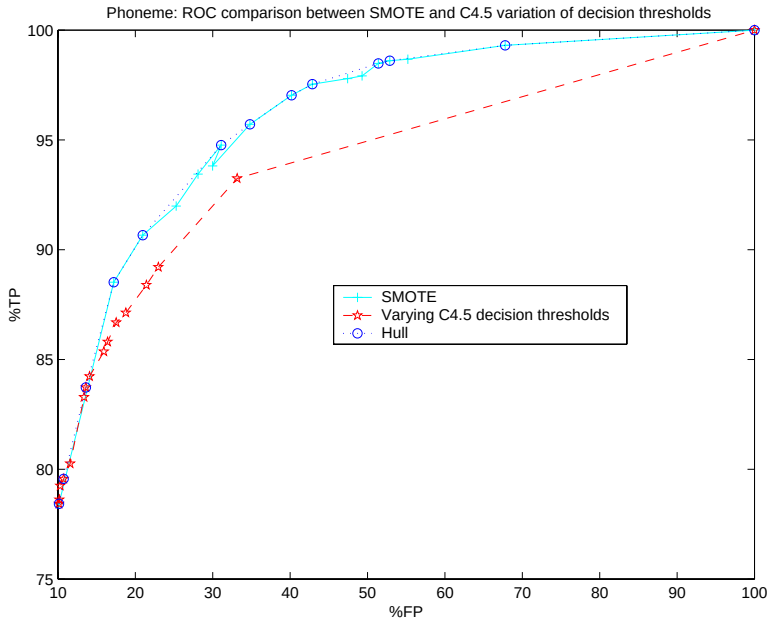


Figure 24: SMOTE and Under-sampling combination against C4.5 learning by tuning the bias towards the minority class

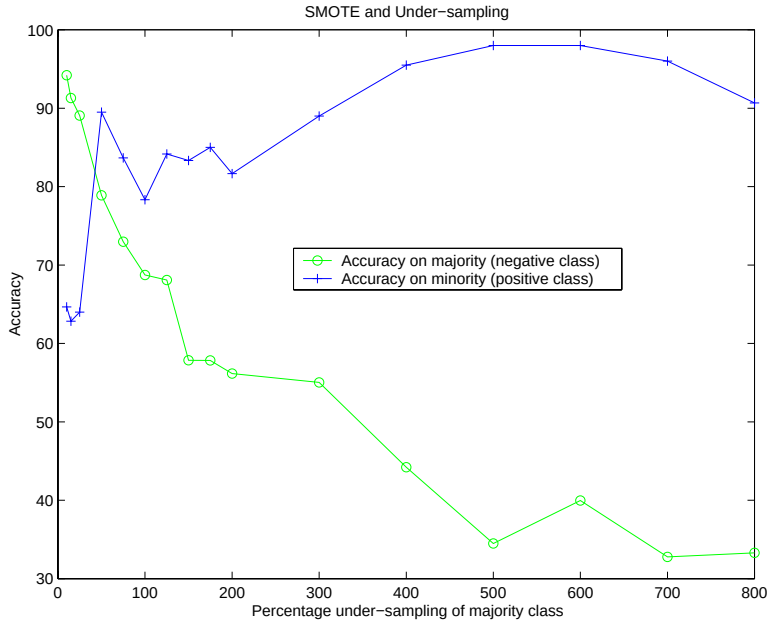


Figure 25: SMOTE (500 OU) and Under-sampling combination performance

SHRINK approach and our SMOTE approach are not directly comparable, though, as they see different data points. SMOTE offers no clear improvement over one-sided selection.

Method	Acc+	Acc−
SHRINK	82.5%	60.9%
One-sided selection	76.0%	86.6%

Table 4: Cross-validation results (Kubat et al., 1998)

Under-sampling %	Acc+	Acc−
10%	64.7%	94.2%
15%	62.8%	91.3%
25%	64.0%	89.1%
50%	89.5%	78.9%
75%	83.7%	73.0%
100%	78.3%	68.7%
125%	84.2%	68.1%
150%	83.3%	57.8%
175%	85.0%	57.8%
200%	81.7%	56.7%
300%	89.0%	55.0%
400%	95.5%	44.2%
500%	98.0%	35.5%
600%	98.0%	40.0%
700%	96.0%	32.8%
800%	90.7%	33.3%

Table 5: Cross-validation results for SMOTE at 500% SMOTE on the Oil data set.

6. Future Work

There are several topics to be considered further in this line of research. Automated adaptive selection of the number of nearest neighbors would be valuable. Different strategies for creating the synthetic neighbors may be able to improve the performance. Also, selecting nearest neighbors with a focus on examples that are incorrectly classified may improve performance. A minority class sample could possibly have a majority class sample as its nearest neighbor rather than a minority class sample. This crowding will likely contribute to the redrawing of the decision surfaces in favor of the minority class. In addition to these topics, the following subsections discuss two possible extensions of SMOTE, and an application of SMOTE to information retrieval.

6.1 SMOTE-NC

While our SMOTE approach currently does not handle data sets with all nominal features, it was generalized to handle mixed datasets of continuous and nominal features. We call this approach Synthetic Minority Over-sampling TEchnique-Nominal Continuous [SMOTE-NC]. We tested this approach on the Adult dataset from the UCI repository. The SMOTE-NC algorithm is described below.

1. Median computation: Compute the median of standard deviations of all continuous features for the minority class. If the nominal features differ between a sample and its potential nearest neighbors, then this median is included in the Euclidean distance computation. We use median to penalize the difference of nominal features by an amount that is related to the typical difference in continuous feature values.

2. Nearest neighbor computation: Compute the Euclidean distance between the feature vector for which k-nearest neighbors are being identified (minority class sample) and the other feature vectors (minority class samples) using the continuous feature space. For every differing nominal feature between the considered feature vector and its potential nearest-neighbor, include the median of the standard deviations previously computed, in the Euclidean distance computation. Table 2 demonstrates an example.

F1 = 1 2 3 A B C [Let this be the sample for which we are computing nearest neighbors]
F2 = 4 6 5 A D E
F3 = 3 5 6 A B K
So, Euclidean Distance between F2 and F1 would be:
Eucl = sqrt[$(4\text{-}1)^2$ + $(6\text{-}2)^2$ + $(5\text{-}3)^2$ + Med^2 + Med^2]
Med is the median of the standard deviations of continuous features of the minority class.
The median term is included twice for feature numbers 5: B→D and 6: C→E, which differ for the two feature vectors: F1 and F2.

Table 6: Example of nearest neighbor computation for SMOTE-NC.

3. Populate the synthetic sample: The continuous features of the new synthetic minority class sample are created using the same approach of SMOTE as described earlier. The nominal feature is given the value occuring in the majority of the k-nearest neighbors.

The SMOTE-NC experiments reported here are set up the same as those with SMOTE, except for the fact that we examine one dataset only. SMOTE-NC with the Adult dataset differs from our typical result: it performs worse than plain under-sampling based on AUC, as shown in Figures 26 and 27. We extracted only continuous features to separate the effect of SMOTE and SMOTE-NC on this dataset, and to determine whether this oddity was due to our handling of nominal features. As shown in Figure 28, even SMOTE with only continuous features applied to the Adult dataset, does not achieve any better performance than plain under-sampling. Some of the minority class continuous features have a very high variance, so, the synthetic generation of minority class samples could be overlapping with the majority class space, thus leading to more false positives than plain under-sampling. This hypothesis is also supported by the decreased AUC measure as we SMOTE at degrees greater than 50%. The higher degrees of SMOTE lead to more minority class samples in the dataset, and thus a greater overlap with the majority class decision space.

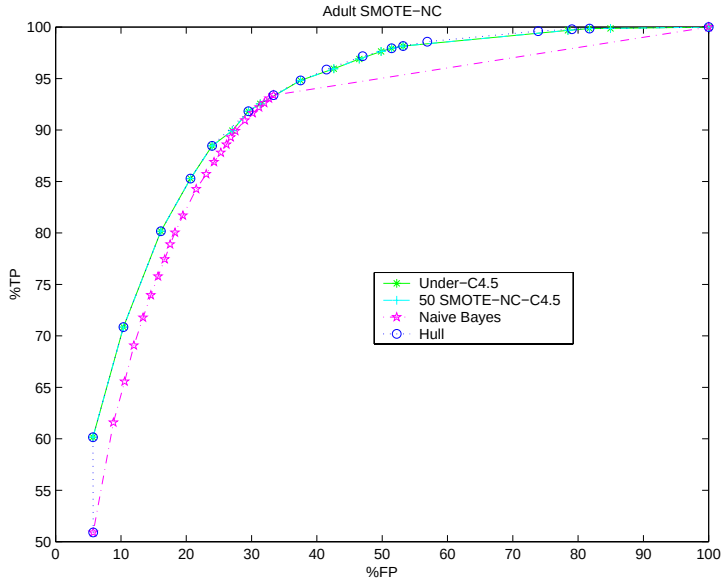


Figure 26: Adult. Comparison of SMOTE-C4.5, Under-C4.5, and Naive Bayes. SMOTE-C4.5 and Under-C4.5 ROC curves overlap for most of the ROC space.

6.2 SMOTE-N

Potentially, SMOTE can also be extended for nominal features — SMOTE-N — with the nearest neighbors computed using the modified version of Value Difference Metric (Stanfill & Waltz, 1986) proposed by Cost and Salzberg (1993). The Value Difference Metric (VDM) looks at the overlap of feature values over all feature vectors. A matrix defining the distance

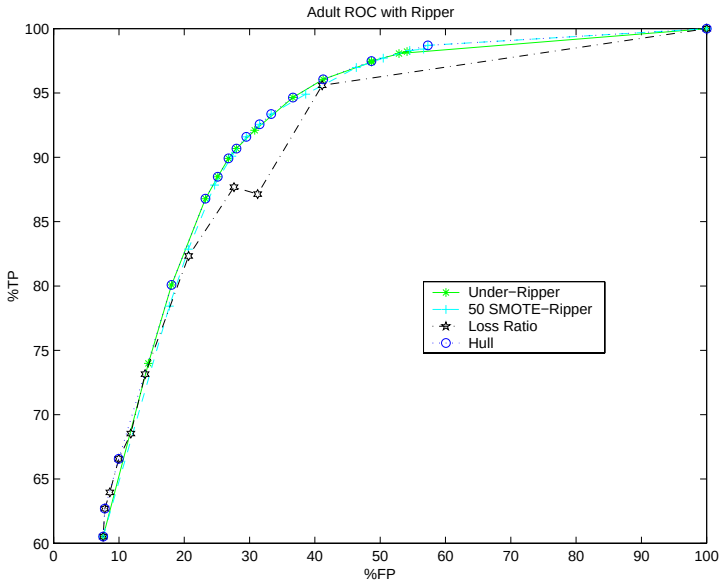


Figure 27: Adult. Comparison of SMOTE-Ripper, Under-Ripper, and modifying Loss Ratio in Ripper. SMOTE-Ripper and Under-Ripper ROC curves overlap for most of the ROC space.

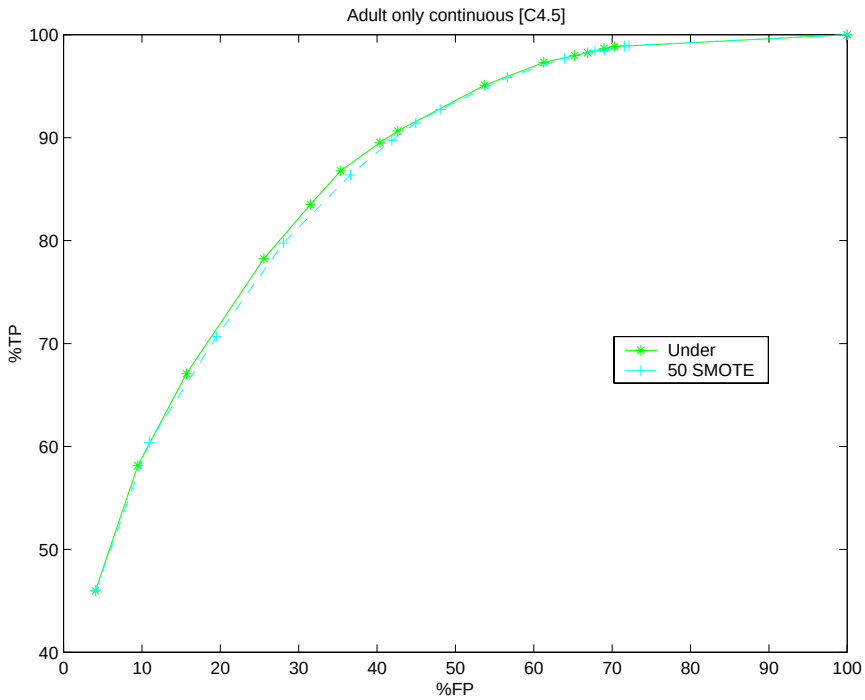


Figure 28: Adult with only continuous features. The overlap of SMOTE-C4.5 and Under-C4.5 is observed under this scenario as well.

between corresponding feature values for all feature vectors is created. The distance δ between two corresponding feature values is defined as follows.

$$\delta(V_1, V_2) = \sum_{i=1}^{n} |\frac{C_{1i}}{C_1} - \frac{C_{2i}}{C_2}|^k \quad (1)$$

In the above equation, V_1 and V_2 are the two corresponding feature values. C_1 is the total number of occurrences of feature value V_1, and C_{1i} is the number of occurrences of feature value V_1 for class i. A similar convention can also be applied to C_{2i} and C_2. k is a constant, usually set to 1. This equation is used to compute the matrix of value differences for each nominal feature in the given set of feature vectors. Equation 1 gives a geometric distance on a fixed, finite set of values (Cost & Salzberg, 1993). Cost and Salzberg's modified VDM omits the weight term w_f^a included in the δ computation by Stanfill and Waltz, which has an effect of making δ symmetric. The distance Δ between two feature vectors is given by:

$$\Delta(X, Y) = w_x w_y \sum_{i=1}^{N} \delta(x_i, y_i)^r \quad (2)$$

$r = 1$ yields the Manhattan distance, and $r = 2$ yields the Euclidean distance (Cost & Salzberg, 1993). w_x and w_y are the exemplar weights in the modified VDM. $w_y = 1$ for a new example (feature vector), and w_x is the bias towards more reliable examples (feature vectors) and is computed as the ratio of the number of uses of a feature vector to the number of correct uses of the feature vector; thus, more accurate feature vectors will have $w_x \approx 1$. For SMOTE-N we can ignore these weights in equation 2, as SMOTE-N is not used for classification purposes directly. However, we can redefine these weights to give more weight to the minority class feature vectors falling closer to the majority class feature vectors; thus, making those minority class features appear further away from the feature vector under consideration. Since, we are more interested in forming broader but accurate regions of the minority class, the weights might be used to avoid populating along neighbors which fall closer to the majority class. To generate new minority class feature vectors, we can create new set feature values by taking the majority vote of the feature vector in consideration and its k nearest neighbors. Table 6.2 shows an example of creating a synthetic feature vector.

Let F1 = A B C D E be the feature vector under consideration and let its 2 nearest neighbors be F2 = A F C G N F3 = H B C D N The application of SMOTE-N would create the following feature vector: FS = A B C D N

Table 7: Example of SMOTE-N

6.3 Application of SMOTE to Information Retrieval

We are investigating the application of SMOTE to information retrieval (IR). The IR problems come with a plethora of features and potentially many categories. SMOTE would have to be applied in conjunction with a feature selection algorithm, after transforming the given document or web page in a bag-of-words format.

An interesting comparison to SMOTE would be the combination of Naive Bayes and *Odds ratio*. *Odds ratio* focuses on a target class, and ranks documents according to their relevance to the target or positive class. SMOTE also focuses on a target class by creating more examples of that class.

7. Summary

The results show that the SMOTE approach can improve the accuracy of classifiers for a minority class. SMOTE provides a new approach to over-sampling. The combination of SMOTE and under-sampling performs better than plain under-sampling. SMOTE was tested on a variety of datasets, with varying degrees of imbalance and varying amounts of data in the training set, thus providing a diverse testbed. The combination of SMOTE and under-sampling also performs better, based on domination in the ROC space, than varying loss ratios in Ripper or by varying the class priors in Naive Bayes Classifier: the methods that could directly handle the skewed class distribution. SMOTE forces focused learning and introduces a bias towards the minority class. Only for Pima — the least skewed dataset — does the Naive Bayes Classifier perform better than SMOTE-C4.5. Also, only for the Oil dataset does the Under-Ripper perform better than SMOTE-Ripper. For the Can dataset, SMOTE-*classifier* and Under-*classifier* ROC curves overlap in the ROC space. For all the rest of the datasets SMOTE-*classifier* performs better than Under-*classifier*, Loss Ratio, and Naive Bayes. Out of a total of 48 experiments performed, SMOTE-*classifier* does not perform the best only for 4 experiments.

The interpretation of why synthetic minority over-sampling improves performance where as minority over-sampling with replacement does not is fairly straightforward. Consider the effect on the decision regions in feature space when minority over-sampling is done by replication (sampling with replacement) versus the introduction of synthetic examples. With replication, the decision region that results in a classification decision for the minority class can actually become smaller and more specific as the minority samples in the region are replicated. This is the opposite of the desired effect. Our method of synthetic over-sampling works to cause the classifier to build larger decision regions that contain nearby minority class points. The same reasons may be applicable to why SMOTE performs better than Ripper's loss ratio and Naive Bayes; these methods, nonetheless, are still learning from the information provided in the dataset, albeit with different cost information. SMOTE provides more related minority class samples to learn from, thus allowing a learner to carve broader decision regions, leading to more coverage of the minority class.

Acknowledgments

This research was partially supported by the United States Department of Energy through the Sandia National Laboratories ASCI VIEWS Data Discovery Program, contract number

DE-AC04-76DO00789. We thank Robert Holte for providing the oil spill dataset used in their paper. We also thank Foster Provost for clarifying his method of using the Satimage dataset. We would also like to thank the anonymous reviewers for their various insightful comments and suggestions.

Appendix A. ROC graphs for Oil Dataset

The following figures show different sets of ROC curves for the oil dataset. Figure 29 (a) shows the ROC curves for the Oil dataset, as included in the main text; Figure 29(b) shows the ROC curves without the ROC convex hull; Figure 29(c) shows the two convex hulls, obtained with and without SMOTE. The ROC convex hull shown by dashed lines and stars in Figure 29(c), was computed by including Under-C4.5 and Naive Bayes in the family of ROC curves. The ROC convex hull shown by solid line and small circles in Figure 29(c) was computed by including 500 SMOTE-C4.5, Under-C4.5, and Naive Bayes in the family of ROC curves. The ROC convex hull with SMOTE dominates the ROC convex hull without SMOTE, hence SMOTE-C4.5 contributes more optimal classifiers.

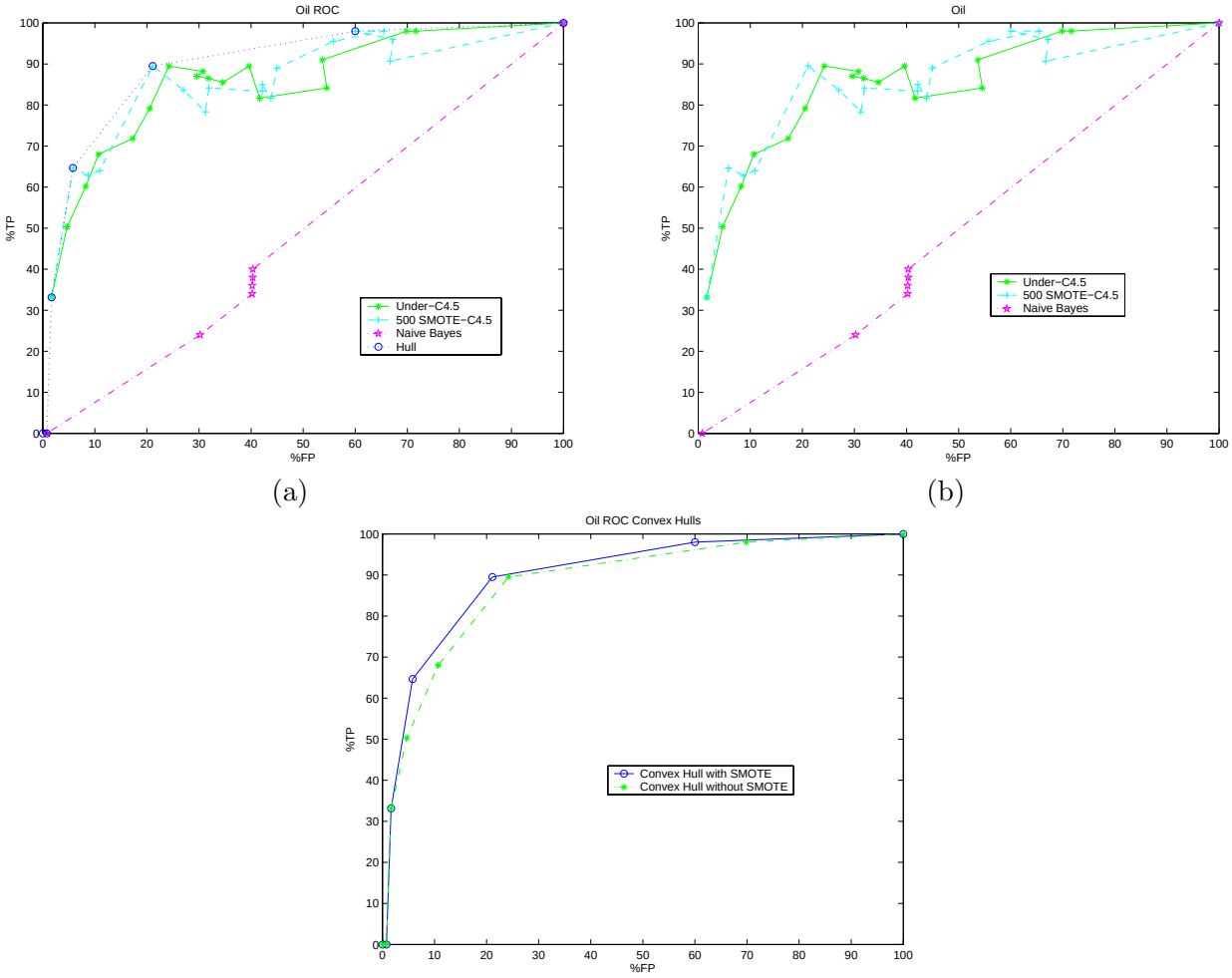


Figure 29: ROC curves for the Oil Dataset. (a) ROC curves for SMOTE-C4.5, Under-C4.5, Naive Bayes, and their ROC convex hull. (b) ROC curves for SMOTE-C4.5, Under-C4.5, and Naive Bayes. (c) ROC convex hulls with and without SMOTE.

References

Blake, C., & Merz, C. (1998). UCI Repository of Machine Learning Databases http://www.ics.uci.edu/∼mlearn/∼MLRepository.html. Department of Information and Computer Sciences, University of California, Irvine.

Bradley, A. P. (1997). The Use of the Area Under the ROC Curve in the Evaluation of Machine Learning Algorithms. *Pattern Recognition*, *30(6)*, 1145–1159.

Chawla, N., Bowyer, K., Hall, L., & Kegelmeyer, P. (2000). SMOTE: Synthetic Minority Over-sampling TEchnique. In *International Conference of Knowledge Based Computer Systems*, pp. 46–57. National Center for Software Technology, Mumbai, India, Allied Press.

Chawla, N., & Hall, L. (1999). Modifying MUSTAFA to capture salient data. Tech. rep. ISL-99-01, University of South Florida, Computer Science and Eng. Dept.

Cohen, W. (1995a). Learning to Classify English Text with ILP Methods. In *Proceedings of the 5th International Workshop on Inductive Logic Programming*, pp. 3–24. Department of Computer Science, Katholieke Universiteit Leuven.

Cohen, W. W. (1995b). Fast Effective Rule Induction. In *Proc. 12th International Conference on Machine Learning*, pp. 115–123 Lake Tahoe, CA. Morgan Kaufmann.

Cohen, W. W., & Singer, Y. (1996). Context-sensitive Learning Methods for Text Categorization. In Frei, H.-P., Harman, D., Schäuble, P., & Wilkinson, R. (Eds.), *Proceedings of SIGIR-96, 19th ACM International Conference on Research and Development in Information Retrieval*, pp. 307–315 Zürich, CH. ACM Press, New York, US.

Cost, S., & Salzberg, S. (1993). A Weighted Nearest Neighbor Algorithm for Learning with Symbolic Features. *Machine Learning*, *10*(1), 57–78.

DeRouin, E., Brown, J., Fausett, L., & Schneider, M. (1991). Neural Network Training on Unequally Represented Classes. In *Intellligent Engineering Systems Through Artificial Neural Networks*, pp. 135–141 New York. ASME Press.

Domingos, P. (1999). Metacost: A General Method for Making Classifiers Cost-sensitive. In *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 155–164 San Diego, CA. ACM Press.

Drummond, C., & Holte, R. (2000). Explicitly Representing Expected Cost: An Alternative to ROC Representation. In *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 198–207 Boston. ACM.

Duda, R., Hart, P., & Stork, D. (2001). *Pattern Classification*. Wiley-Interscience.

Dumais, S., Platt, J., Heckerman, D., & Sahami, M. (1998). Inductive Learning Algorithms and Representations for Text Categorization. In *Proceedings of the Seventh International Conference on Information and Knowledge Management.*, pp. 148–155.

Ezawa, K., J., Singh, M., & Norton, S., W. (1996). Learning Goal Oriented Bayesian Networks for Telecommunications Risk Management. In *Proceedings of the International Conference on Machine Learning, ICML-96*, pp. 139–147 Bari, Italy. Morgan Kauffman.

Fawcett, T., & Provost, F. (1996). Combining Data Mining and Machine Learning for Effective User Profile. In *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining*, pp. 8–13 Portland, OR. AAAI.

Ha, T. M., & Bunke, H. (1997). Off-line, Handwritten Numeral Recognition by Perturbation Method. *Pattern Analysis and Machine Intelligence*, *19/5*, 535–539.

Hall, L., Mohney, B., & Kier, L. (1991). The Electrotopological State: Structure Information at the Atomic Level for Molecular Graphs. *Journal of Chemical Information and Computer Science*, *31*(76).

Japkowicz, N. (2000). The Class Imbalance Problem: Significance and Strategies. In *Proceedings of the 2000 International Conference on Artificial Intelligence (IC-AI'2000): Special Track on Inductive Learning* Las Vegas, Nevada.

Kubat, M., Holte, R., & Matwin, S. (1998). Machine Learning for the Detection of Oil Spills in Satellite Radar Images. *Machine Learning*, *30*, 195–215.

Kubat, M., & Matwin, S. (1997). Addressing the Curse of Imbalanced Training Sets: One Sided Selection. In *Proceedings of the Fourteenth International Conference on Machine Learning*, pp. 179–186 Nashville, Tennesse. Morgan Kaufmann.

Lee, S. (2000). Noisy Replication in Skewed Binary Classification. *Computational Statistics and Data Analysis*, *34*.

Lewis, D., & Catlett, J. (1994). Heterogeneous Uncertainity Sampling for Supervised Learning. In *Proceedings of the Eleventh International Conference of Machine Learning*, pp. 148–156 San Francisco, CA. Morgan Kaufmann.

Lewis, D., & Ringuette, M. (1994). A Comparison of Two Learning Algorithms for Text Categorization. In *Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval*, pp. 81–93.

Ling, C., & Li, C. (1998). Data Mining for Direct Marketing Problems and Solutions. In *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining (KDD-98)* New York, NY. AAAI Press.

Mladenić, D., & Grobelnik, M. (1999). Feature Selection for Unbalanced Class Distribution and Naive Bayes. In *Proceedings of the 16th International Conference on Machine Learning.*, pp. 258–267. Morgan Kaufmann.

O'Rourke, J. (1998). *Computational Geometry in C.* Cambridge University Press, UK.

Pazzani, M., Merz, C., Murphy, P., Ali, K., Hume, T., & Brunk, C. (1994). Reducing Misclassification Costs. In *Proceedings of the Eleventh International Conference on Machine Learning* San Francisco, CA. Morgan Kauffmann.

Provost, F., & Fawcett, T. (2001). Robust Classification for Imprecise Environments. *Machine Learning*, *42/3*, 203–231.

Provost, F., Fawcett, T., & Kohavi, R. (1998). The Case Against Accuracy Estimation for Comparing Induction Algorithms. In *Proceedings of the Fifteenth International Conference on Machine Learning*, pp. 445–453 Madison, WI. Morgan Kauffmann.

Quinlan, J. (1992). *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Mateo, CA.

Solberg, A., & Solberg, R. (1996). A Large-Scale Evaluation of Features for Automatic Detection of Oil Spills in ERS SAR Images. In *International Geoscience and Remote Sensing Symposium*, pp. 1484–1486 Lincoln, NE.

Stanfill, C., & Waltz, D. (1986). Toward Memory-based Reasoning. *Communications of the ACM*, *29*(12), 1213–1228.

Swets, J. (1988). Measuring the Accuracy of Diagnostic Systems. *Science*, *240*, 1285–1293.

Tomek, I. (1976). Two Modifications of CNN. *IEEE Transactions on Systems, Man and Cybernetics*, *6*, 769–772.

Turney, P. (1996). Cost Sensitive Bibliography. http://ai.iit.nrc.ca/bibiliographies/cost-sensitive.html.

van Rijsbergen, C., Harper, D., & Porter, M. (1981). The Selection of Good Search Terms. *Information Processing and Management*, *17*, 77–91.

Woods, K., Doss, C., Bowyer, K., Solka, J., Priebe, C., & Kegelmeyer, P. (1993). Comparative Evaluation of Pattern Recognition Techniques for Detection of Microcalcifications in Mammography. *International Journal of Pattern Recognition and Artificial Intelligence*, *7(6)*, 1417–1436.

EXTENDED PDF FORMAT
SPONSORED BY
usb
www.usbweb.com

Reducing the Dimensionality of Data with Neural Networks

G. E. Hinton, *et al.*

Science **313**, 504 (2006);
DOI: 10.1126/science.1127647

The following resources related to this article are available online at www.sciencemag.org (this information is current as of September 14, 2008):

Updated information and services, including high-resolution figures, can be found in the online version of this article at:
http://www.sciencemag.org/cgi/content/full/313/5786/504

Supporting Online Material can be found at:
http://www.sciencemag.org/cgi/content/full/313/5786/504/DC1

A list of selected additional articles on the Science Web sites **related to this article** can be found at:
http://www.sciencemag.org/cgi/content/full/313/5786/504#related-content

This article **cites 8 articles**, 6 of which can be accessed for free:
http://www.sciencemag.org/cgi/content/full/313/5786/504#otherarticles

This article has been **cited by** 15 article(s) on the ISI Web of Science.

This article has been **cited by** 3 articles hosted by HighWire Press; see:
http://www.sciencemag.org/cgi/content/full/313/5786/504#otherarticles

Information about obtaining **reprints** of this article or about obtaining **permission to reproduce this article** in whole or in part can be found at:
http://www.sciencemag.org/about/permissions.dtl

Downloaded from www.sciencemag.org on September 14, 2008

Science (print ISSN 0036-8075; online ISSN 1095-9203) is published weekly, except the last week in December, by the American Association for the Advancement of Science, 1200 New York Avenue NW, Washington, DC 20005. Copyright 2006 by the American Association for the Advancement of Science; all rights reserved. The title *Science* is a registered trademark of AAAS.

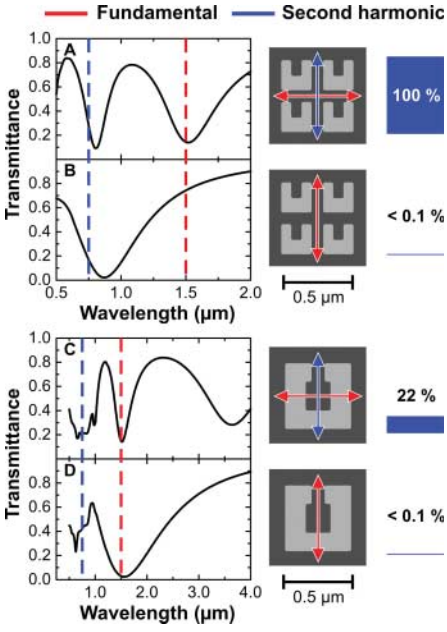


Fig. 3. Theory, presented as the experiment (see Fig. 1). The SHG source is the magnetic component of the Lorentz force on metal electrons in the SRRs.

The setup for measuring the SHG is described in the supporting online material (*22*). We expect that the SHG strongly depends on the resonance that is excited. Obviously, the incident polarization and the detuning of the laser wavelength from the resonance are of particular interest. One possibility for controlling the detuning is to change the laser wavelength for a given sample, which is difficult because of the extremely broad tuning range required. Thus, we follow an alternative route, lithographic tuning (in which the incident laser wavelength of 1.5 μm, as well as the detection system, remains fixed), and tune the resonance positions by changing the SRR size. In this manner, we can also guarantee that the optical properties of the SRR constituent materials are identical for all configurations. The blue bars in Fig. 1 summarize the measured SHG signals. For excitation of the *LC* resonance in Fig. 1A (horizontal incident polarization), we find an SHG signal that is 500 times above the noise level. As expected for SHG, this signal closely scales with the square of the incident power (Fig. 2A). The polarization of the SHG emission is nearly vertical (Fig. 2B). The small angle with respect to the vertical is due to deviations from perfect mirror symmetry of the SRRs (see electron micrographs in Fig. 1). Small detuning of the *LC* resonance toward smaller wavelength (i.e., to 1.3-μm wavelength) reduces the SHG signal strength from 100% to 20%. For excitation of the Mie resonance with vertical incident polarization in Fig. 1D, we find a small signal just above the noise level. For excitation of the Mie resonance with horizontal incident polarization in Fig. 1C, a small but significant SHG emission is found, which is again polarized nearly vertically. For completeness, Fig. 1B shows the off-resonant case for the smaller SRRs for vertical incident polarization.

Although these results are compatible with the known selection rules of surface SHG from usual nonlinear optics (*23*), these selection rules do not explain the mechanism of SHG. Following our above argumentation on the magnetic component of the Lorentz force, we numerically calculate first the linear electric and magnetic field distributions (*22*); from these fields, we compute the electron velocities and the Lorentz-force field (fig. S1). In the spirit of a metamaterial, the transverse component of the Lorentz-force field can be spatially averaged over the volume of the unit cell of size a by a by t. This procedure delivers the driving force for the transverse SHG polarization. As usual, the SHG intensity is proportional to the square modulus of the nonlinear electron displacement. Thus, the SHG strength is expected to be proportional to the square modulus of the driving force, and the SHG polarization is directed along the driving-force vector. Corresponding results are summarized in Fig. 3 in the same arrangement as Fig. 1 to allow for a direct comparison between experiment and theory. The agreement is generally good, both for linear optics and for SHG. In particular, we find a much larger SHG signal for excitation of those two resonances (Fig. 3, A and C), which are related to a finite magnetic-dipole moment (perpendicular to the SRR plane) as compared with the purely electric Mie resonance (Figs. 1D and 3D), despite the fact that its oscillator strength in the linear spectrum is comparable. The SHG polarization in the theory is strictly vertical for all resonances. Quantitative deviations between the SHG signal strengths of experiment and theory, respectively, are probably due to the simplified SRR shape assumed in our calculations and/or due to the simplicity of our modeling. A systematic microscopic theory of the nonlinear optical properties of metallic metamaterials would be highly desirable but is currently not available.

References and Notes

1. J. B. Pendry, A. J. Holden, D. J. Robbins, W. J. Stewart, *IEEE Trans. Microw. Theory Tech.* **47**, 2075 (1999).
2. J. B. Pendry, *Phys. Rev. Lett.* **85**, 3966 (2000).
3. R. A. Shelby, D. R. Smith, S. Schultz, *Science* **292**, 77 (2001).
4. T. J. Yen *et al.*, *Science* **303**, 1494 (2004).
5. S. Linden *et al.*, *Science* **306**, 1351 (2004).
6. C. Enkrich *et al.*, *Phys. Rev. Lett.* **95**, 203901 (2005).
7. A. N. Grigorenko *et al.*, *Nature* **438**, 335 (2005).
8. G. Dolling, M. Wegener, S. Linden, C. Hormann, *Opt. Express* **14**, 1842 (2006).
9. G. Dolling, C. Enkrich, M. Wegener, C. M. Soukoulis, S. Linden, *Science* **312**, 892 (2006).
10. J. B. Pendry, D. Schurig, D. R. Smith, *Science* **312**, 1780; published online 25 May 2006.
11. U. Leonhardt, *Science* **312**, 1777 (2006); published online 25 May 2006.
12. M. W. Klein, C. Enkrich, M. Wegener, C. M. Soukoulis, S. Linden, *Opt. Lett.* **31**, 1259 (2006).
13. W. J. Padilla, A. J. Taylor, C. Highstrete, M. Lee, R. D. Averitt, *Phys. Rev. Lett.* **96**, 107401 (2006).
14. D. R. Smith, S. Schultz, P. Markos, C. M. Soukoulis, *Phys. Rev. B* **65**, 195104 (2002).
15. S. O'Brien, D. McPeake, S. A. Ramakrishna, J. B. Pendry, *Phys. Rev. B* **69**, 241101 (2004).
16. J. Zhou *et al.*, *Phys. Rev. Lett.* **95**, 223902 (2005).
17. A. K. Popov, V. M. Shalaev, available at http://arxiv.org/abs/physics/0601055 (2006).
18. V. G. Veselago, *Sov. Phys. Usp.* **10**, 509 (1968).
19. M. Wegener, *Extreme Nonlinear Optics* (Springer, Berlin, 2004).
20. H. M. Barlow, *Nature* **173**, 41 (1954).
21. S.-Y. Chen, M. Maksimchuk, D. Umstadter, *Nature* **396**, 653 (1998).
22. Materials and Methods are available as supporting material on *Science* Online.
23. P. Guyot-Sionnest, W. Chen, Y. R. Shen, *Phys. Rev. B* **33**, 8254 (1986).

24. We thank the groups of S. W. Koch, J. V. Moloney, and C. M. Soukoulis for discussions. The research of M.W. is supported by the Leibniz award 2000 of the Deutsche Forschungsgemeinschaft (DFG), that of S.L. through a Helmholtz-Hochschul-Nachwuchsgruppe (VH-NG-232).

Supporting Online Material

www.sciencemag.org/cgi/content/full/313/5786/502/DC1
Materials and Methods
Figs. S1 and S2
References

26 April 2006; accepted 22 June 2006
10.1126/science.1129198

Reducing the Dimensionality of Data with Neural Networks

G. E. Hinton* and R. R. Salakhutdinov

High-dimensional data can be converted to low-dimensional codes by training a multilayer neural network with a small central layer to reconstruct high-dimensional input vectors. Gradient descent can be used for fine-tuning the weights in such "autoencoder" networks, but this works well only if the initial weights are close to a good solution. We describe an effective way of initializing the weights that allows deep autoencoder networks to learn low-dimensional codes that work much better than principal components analysis as a tool to reduce the dimensionality of data.

Dimensionality reduction facilitates the classification, visualization, communication, and storage of high-dimensional data. A simple and widely used method is principal components analysis (PCA), which finds the directions of greatest variance in the data set and represents each data point by its coordinates along each of these directions. We describe a nonlinear generalization of PCA that uses an adaptive, multilayer "encoder" network

Downloaded from www.sciencemag.org on September 14, 2008

to transform the high-dimensional data into a low-dimensional code and a similar "decoder" network to recover the data from the code. Starting with random weights in the two networks, they can be trained together by minimizing the discrepancy between the original data and its reconstruction. The required gradients are easily obtained by using the chain rule to backpropagate error derivatives first through the decoder network and then through the encoder network (*1*). The whole system is called an "autoencoder" and is depicted in Fig. 1.

Department of Computer Science, University of Toronto, 6 King's College Road, Toronto, Ontario M5S 3G4, Canada.

*To whom correspondence should be addressed; E-mail: hinton@cs.toronto.edu

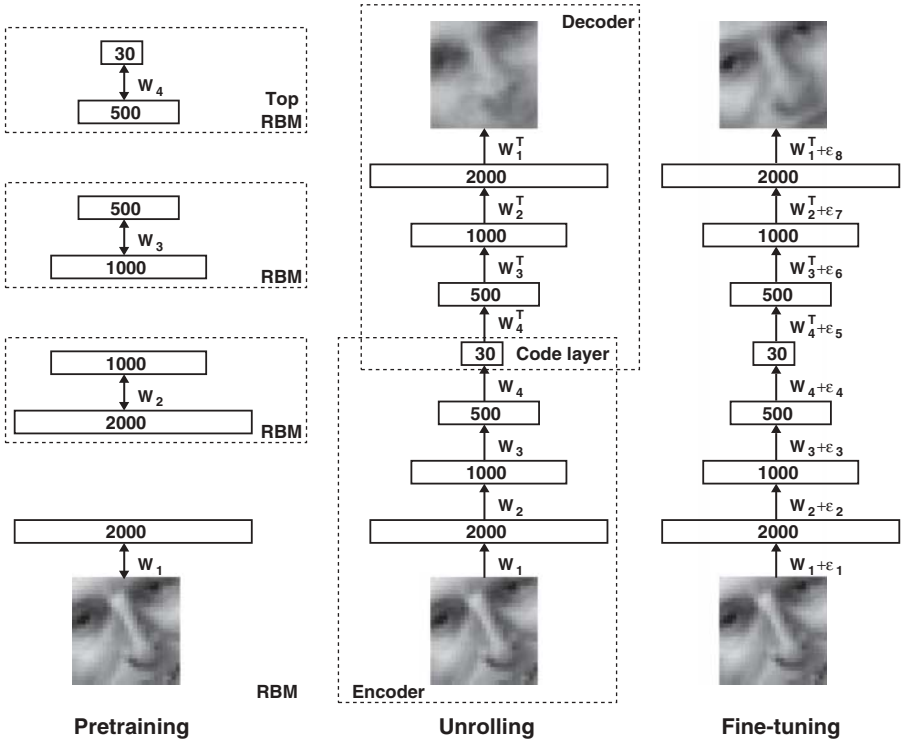


Fig. 1. Pretraining consists of learning a stack of restricted Boltzmann machines (RBMs), each having only one layer of feature detectors. The learned feature activations of one RBM are used as the "data" for training the next RBM in the stack. After the pretraining, the RBMs are "unrolled" to create a deep autoencoder, which is then fine-tuned using backpropagation of error derivatives.

It is difficult to optimize the weights in nonlinear autoencoders that have multiple hidden layers (*2*–*4*). With large initial weights, autoencoders typically find poor local minima; with small initial weights, the gradients in the early layers are tiny, making it infeasible to train autoencoders with many hidden layers. If the initial weights are close to a good solution, gradient descent works well, but finding such initial weights requires a very different type of algorithm that learns one layer of features at a time. We introduce this "pretraining" procedure for binary data, generalize it to real-valued data, and show that it works well for a variety of data sets.

An ensemble of binary vectors (e.g., images) can be modeled using a two-layer network called a "restricted Boltzmann machine" (RBM) (*5*, *6*) in which stochastic, binary pixels are connected to stochastic, binary feature detectors using symmetrically weighted connections. The pixels correspond to "visible" units of the RBM because their states are observed; the feature detectors correspond to "hidden" units. A joint configuration ($\mathbf{v}$, $\mathbf{h}$) of the visible and hidden units has an energy (*7*) given by

$$E(\mathbf{v}, \mathbf{h}) = -\sum_{i \in \text{pixels}} b_i v_i - \sum_{j \in \text{features}} b_j h_j - \sum_{i,j} v_i h_j w_{ij} \quad (1)$$

where v_i and h_j are the binary states of pixel i and feature j, b_i and b_j are their biases, and w_{ij} is the weight between them. The network assigns a probability to every possible image via this energy function, as explained in (*8*). The probability of a training image can be raised by

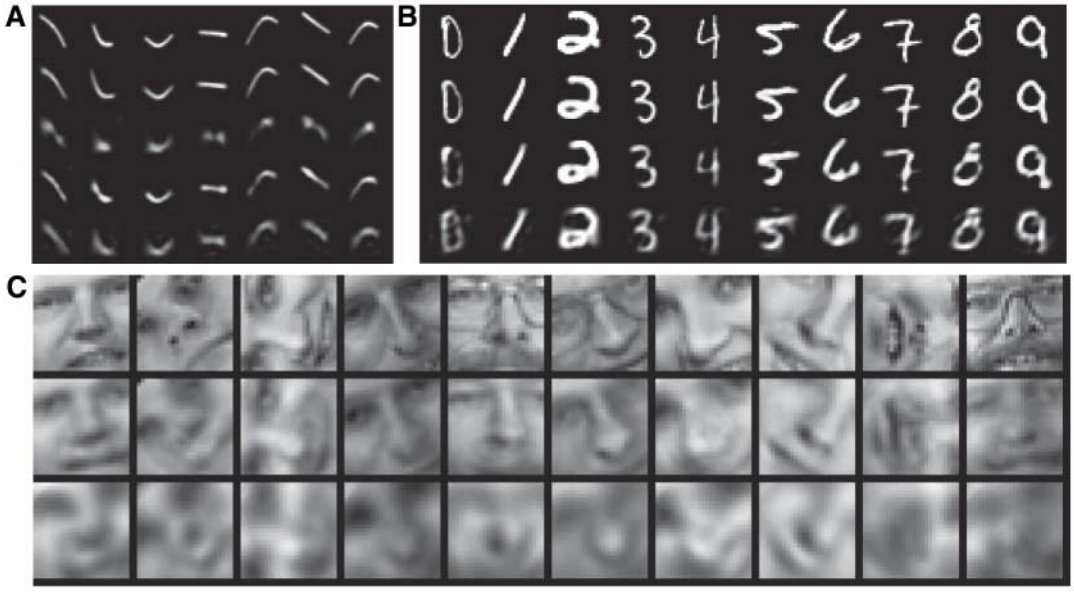


Fig. 2. (**A**) Top to bottom: Random samples of curves from the test data set; reconstructions produced by the six-dimensional deep autoencoder; reconstructions by "logistic PCA" (*8*) using six components; reconstructions by logistic PCA and standard PCA using 18 components. The average squared error per image for the last four rows is 1.44, 7.64, 2.45, 5.90. (**B**) Top to bottom: A random test image from each class; reconstructions by the 30-dimensional autoencoder; reconstructions by 30-dimensional logistic PCA and standard PCA. The average squared errors for the last three rows are 3.00, 8.01, and 13.87. (**C**) Top to bottom: Random samples from the test data set; reconstructions by the 30-dimensional autoencoder; reconstructions by 30-dimensional PCA. The average squared errors are 126 and 135.

Downloaded from www.sciencemag.org on September 14, 2008

adjusting the weights and biases to lower the energy of that image and to raise the energy of similar, "confabulated" images that the network would prefer to the real data. Given a training image, the binary state h_j of each feature detector j is set to 1 with probability $\sigma(b_j + \sum_i v_i w_{ij})$, where $\sigma(x)$ is the logistic function $1/[1 + \exp(-x)]$, b_j is the bias of j, v_i is the state of pixel i, and w_{ij} is the weight between i and j. Once binary states have been chosen for the hidden units, a "confabulation" is produced by setting each v_i to 1 with probability $\sigma(b_i + \sum_j h_j w_{ij})$, where b_i is the bias of i. The states of the hidden units are then updated once more so that they represent features of the confabulation. The change in a weight is given by

$$\Delta w_{ij} = \varepsilon\left(\langle v_i h_j\rangle_{\text{data}} - \langle v_i h_j\rangle_{\text{recon}}\right) \quad (2)$$

where ε is a learning rate, $\langle v_i h_j\rangle_{\text{data}}$ is the fraction of times that the pixel i and feature detector j are on together when the feature detectors are being driven by data, and $\langle v_i h_j\rangle_{\text{recon}}$ is the corresponding fraction for confabulations. A simplified version of the same learning rule is used for the biases. The learning works well even though it is not exactly following the gradient of the log probability of the training data (*6*).

A single layer of binary features is not the best way to model the structure in a set of images. After learning one layer of feature detectors, we can treat their activities—when they are being driven by the data—as data for learning a second layer of features. The first layer of feature detectors then become the visible units for learning the next RBM. This layer-by-layer learning can be repeated as many

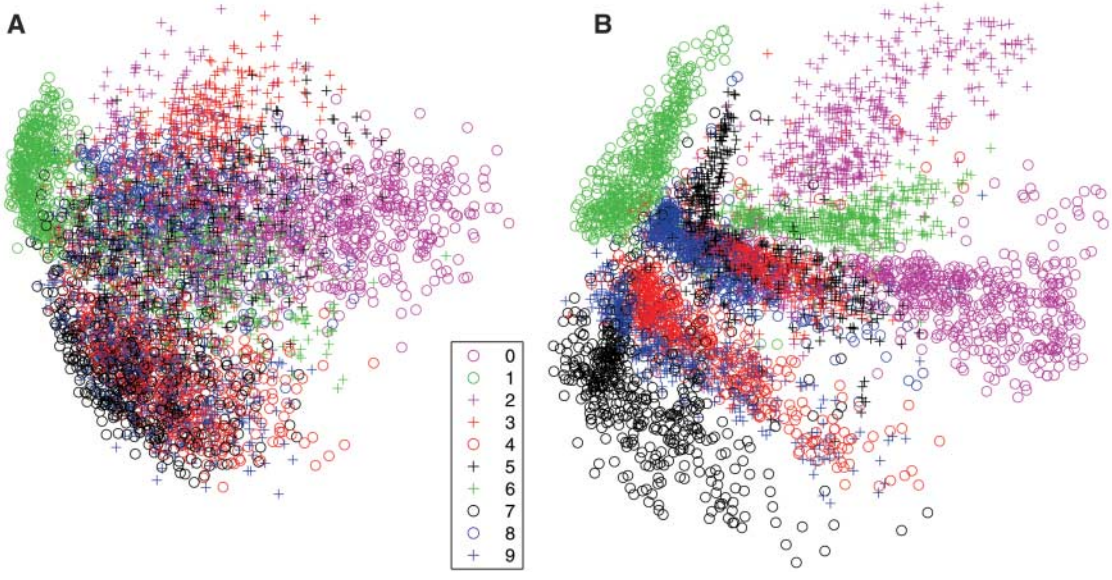


Fig. 3. (**A**) The two-dimensional codes for 500 digits of each class produced by taking the first two principal components of all 60,000 training images. (**B**) The two-dimensional codes found by a 784-1000-500-250-2 autoencoder. For an alternative visualization, see (*8*).

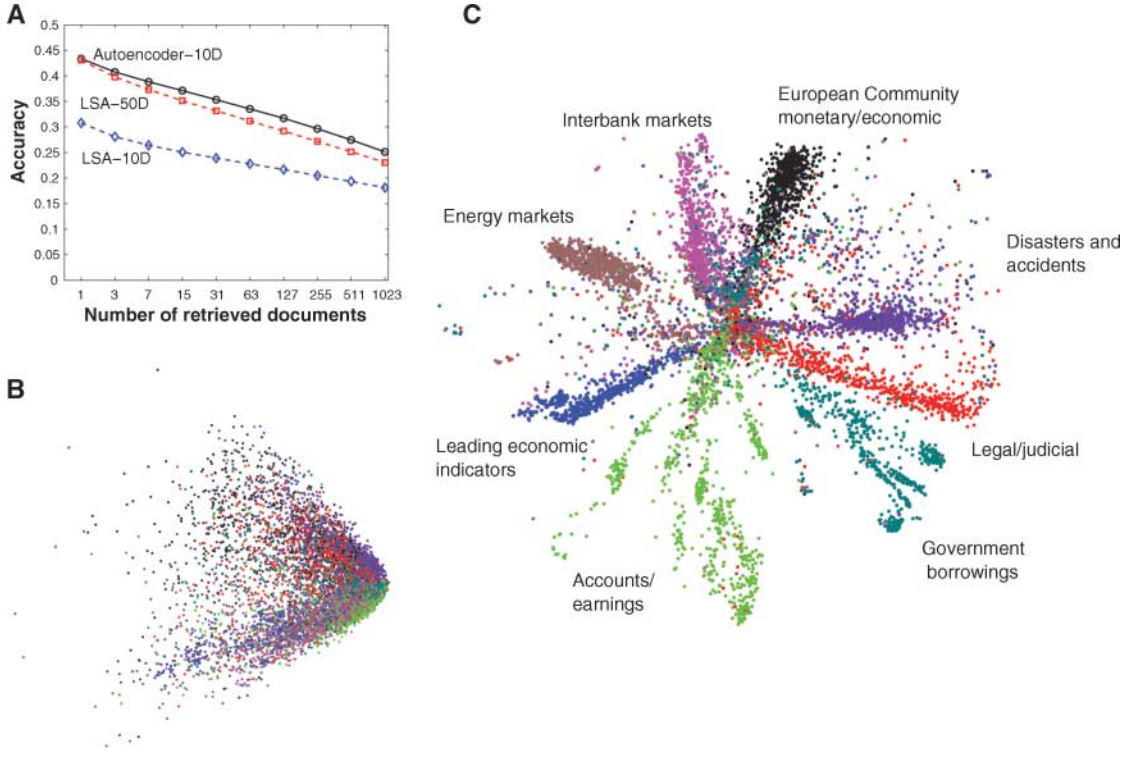


Fig. 4. (**A**) The fraction of retrieved documents in the same class as the query when a query document from the test set is used to retrieve other test set documents, averaged over all 402,207 possible queries. (**B**) The codes produced by two-dimensional LSA. (**C**) The codes produced by a 2000-500-250-125-2 autoencoder.

Downloaded from www.sciencemag.org on September 14, 2008

times as desired. It can be shown that adding an extra layer always improves a lower bound on the log probability that the model assigns to the training data, provided the number of feature detectors per layer does not decrease and their weights are initialized correctly (*9*). This bound does not apply when the higher layers have fewer feature detectors, but the layer-by-layer learning algorithm is nonetheless a very effective way to pretrain the weights of a deep autoencoder. Each layer of features captures strong, high-order correlations between the activities of units in the layer below. For a wide variety of data sets, this is an efficient way to progressively reveal low-dimensional, nonlinear structure.

After pretraining multiple layers of feature detectors, the model is "unfolded" (Fig. 1) to produce encoder and decoder networks that initially use the same weights. The global fine-tuning stage then replaces stochastic activities by deterministic, real-valued probabilities and uses backpropagation through the whole autoencoder to fine-tune the weights for optimal reconstruction.

For continuous data, the hidden units of the first-level RBM remain binary, but the visible units are replaced by linear units with Gaussian noise (*10*). If this noise has unit variance, the stochastic update rule for the hidden units remains the same and the update rule for visible unit i is to sample from a Gaussian with unit variance and mean $b_i + \sum_j h_j w_{ij}$.

In all our experiments, the visible units of every RBM had real-valued activities, which were in the range [0, 1] for logistic units. While training higher level RBMs, the visible units were set to the activation probabilities of the hidden units in the previous RBM, but the hidden units of every RBM except the top one had stochastic binary values. The hidden units of the top RBM had stochastic real-valued states drawn from a unit variance Gaussian whose mean was determined by the input from that RBM's logistic visible units. This allowed the low-dimensional codes to make good use of continuous variables and facilitated comparisons with PCA. Details of the pretraining and fine-tuning can be found in (*8*).

To demonstrate that our pretraining algorithm allows us to fine-tune deep networks efficiently, we trained a very deep autoencoder on a synthetic data set containing images of "curves" that were generated from three randomly chosen points in two dimensions (*8*). For this data set, the true intrinsic dimensionality is known, and the relationship between the pixel intensities and the six numbers used to generate them is highly nonlinear. The pixel intensities lie between 0 and 1 and are very non-Gaussian, so we used logistic output units in the autoencoder, and the fine-tuning stage of the learning minimized the cross-entropy error $[-\sum_i p_i \log \hat{p}_i - \sum_i (1 - p_i) \log(1 - \hat{p}_i)]$, where p_i is the intensity of pixel i and $\hat{p}_i$ is the intensity of its reconstruction.

The autoencoder consisted of an encoder with layers of size (28×28)-400-200-100-50-25-6 and a symmetric decoder. The six units in the code layer were linear and all the other units were logistic. The network was trained on 20,000 images and tested on 10,000 new images. The autoencoder discovered how to convert each 784-pixel image into six real numbers that allow almost perfect reconstruction (Fig. 2A). PCA gave much worse reconstructions. Without pretraining, the very deep autoencoder always reconstructs the average of the training data, even after prolonged fine-tuning (*8*). Shallower autoencoders with a single hidden layer between the data and the code can learn without pretraining, but pretraining greatly reduces their total training time (*8*). When the number of parameters is the same, deep autoencoders can produce lower reconstruction errors on test data than shallow ones, but this advantage disappears as the number of parameters increases (*8*).

Next, we used a 784-1000-500-250-30 autoencoder to extract codes for all the handwritten digits in the MNIST training set (*11*). The Matlab code that we used for the pretraining and fine-tuning is available in (*8*). Again, all units were logistic except for the 30 linear units in the code layer. After fine-tuning on all 60,000 training images, the autoencoder was tested on 10,000 new images and produced much better reconstructions than did PCA (Fig. 2B). A two-dimensional autoencoder produced a better visualization of the data than did the first two principal components (Fig. 3).

We also used a 625-2000-1000-500-30 autoencoder with linear input units to discover 30-dimensional codes for grayscale image patches that were derived from the Olivetti face data set (*12*). The autoencoder clearly outperformed PCA (Fig. 2C).

When trained on documents, autoencoders produce codes that allow fast retrieval. We represented each of 804,414 newswire stories (*13*) as a vector of document-specific probabilities of the 2000 commonest word stems, and we trained a 2000-500-250-125-10 autoencoder on half of the stories with the use of the multiclass cross-entropy error function $[-\sum_i p_i \log \hat{p}_i]$ for the fine-tuning. The 10 code units were linear and the remaining hidden units were logistic. When the cosine of the angle between two codes was used to measure similarity, the autoencoder clearly outperformed latent semantic analysis (LSA) (*14*), a well-known document retrieval method based on PCA (Fig. 4). Autoencoders (*8*) also outperform local linear embedding, a recent nonlinear dimensionality reduction algorithm (*15*).

Layer-by-layer pretraining can also be used for classification and regression. On a widely used version of the MNIST handwritten digit recognition task, the best reported error rates are 1.6% for randomly initialized backpropagation and 1.4% for support vector machines. After layer-by-layer pretraining in a 784-500-500-2000-10 network, backpropagation using steepest descent and a small learning rate achieves 1.2% (*8*). Pretraining helps generalization because it ensures that most of the information in the weights comes from modeling the images. The very limited information in the labels is used only to slightly adjust the weights found by pretraining.

It has been obvious since the 1980s that backpropagation through deep autoencoders would be very effective for nonlinear dimensionality reduction, provided that computers were fast enough, data sets were big enough, and the initial weights were close enough to a good solution. All three conditions are now satisfied. Unlike nonparametric methods (*15*, *16*), autoencoders give mappings in both directions between the data and code spaces, and they can be applied to very large data sets because both the pretraining and the fine-tuning scale linearly in time and space with the number of training cases.

Downloaded from www.sciencemag.org on September 14, 2008

References and Notes

1. D. C. Plaut, G. E. Hinton, *Comput. Speech Lang.* **2**, 35 (1987).
2. D. DeMers, G. Cottrell, *Advances in Neural Information Processing Systems 5* (Morgan Kaufmann, San Mateo, CA, 1993), pp. 580–587.
3. R. Hecht-Nielsen, *Science* **269**, 1860 (1995).
4. N. Kambhatla, T. Leen, *Neural Comput.* **9**, 1493 (1997).
5. P. Smolensky, *Parallel Distributed Processing: Volume 1: Foundations*, D. E. Rumelhart, J. L. McClelland, Eds. (MIT Press, Cambridge, 1986), pp. 194–281.
6. G. E. Hinton, *Neural Comput.* **14**, 1711 (2002).
7. J. J. Hopfield, *Proc. Natl. Acad. Sci. U.S.A.* **79**, 2554 (1982).
8. See supporting material on *Science* Online.
9. G. E. Hinton, S. Osindero, Y. W. Teh, *Neural Comput.* **18**, 1527 (2006).
10. M. Welling, M. Rosen-Zvi, G. Hinton, *Advances in Neural Information Processing Systems 17* (MIT Press, Cambridge, MA, 2005), pp. 1481–1488.
11. The MNIST data set is available at http://yann.lecun.com/exdb/mnist/index.html.
12. The Olivetti face data set is available at www.cs.toronto.edu/ roweis/data.html.
13. The Reuter Corpus Volume 2 is available at http://trec.nist.gov/data/reuters/reuters.html.
14. S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, R. A. Harshman, *J. Am. Soc. Inf. Sci.* **41**, 391 (1990).
15. S. T. Roweis, L. K. Saul, *Science* **290**, 2323 (2000).
16. J. A. Tenenbaum, V. J. de Silva, J. C. Langford, *Science* **290**, 2319 (2000).

17. We thank D. Rumelhart, M. Welling, S. Osindero, and S. Roweis for helpful discussions, and the Natural Sciences and Engineering Research Council of Canada for funding. G.E.H. is a fellow of the Canadian Institute for Advanced Research.

Supporting Online Material

www.sciencemag.org/cgi/content/full/313/5786/504/DC1
Materials and Methods
Figs. S1 to S5
Matlab Code

20 March 2006; accepted 1 June 2006
10.1126/science.1127647

An Analysis of the t-SNE Algorithm for Data Visualization

Sanjeev Arora ARORA@CS.PRINCETON.EDU
Princeton University

Wei Hu HUWEI@CS.PRINCETON.EDU
Princeton University

Pravesh K. Kothari KOTHARI@CS.PRINCETON.EDU
Princeton University & Institute for Advanced Study

Editors: Sébastien Bubeck, Vianney Perchet and Philippe Rigollet

Abstract

A first line of attack in exploratory data analysis is *data visualization*, i.e., generating a 2-dimensional representation of data that makes *clusters* of similar points visually identifiable. Standard Johnson-Lindenstrauss dimensionality reduction does not produce data visualizations. The *t-SNE* heuristic of van der Maaten and Hinton, which is based on non-convex optimization, has become the *de facto* standard for visualization in a wide range of applications.

This work gives a formal framework for the problem of data visualization – finding a 2-dimensional embedding of clusterable data that correctly separates individual clusters to make them visually identifiable. We then give a rigorous analysis of the performance of t-SNE under a natural, deterministic condition on the "ground-truth" clusters (similar to conditions assumed in earlier analyses of clustering) in the underlying data. These are the first provable guarantees on t-SNE for constructing good data visualizations.

We show that our deterministic condition is satisfied by considerably general probabilistic generative models for clusterable data such as mixtures of well-separated log-concave distributions. Finally, we give theoretical evidence that t-SNE provably succeeds in *partially* recovering cluster structure even when the above deterministic condition is not met.

Keywords: Clustering, t-SNE, Visualization

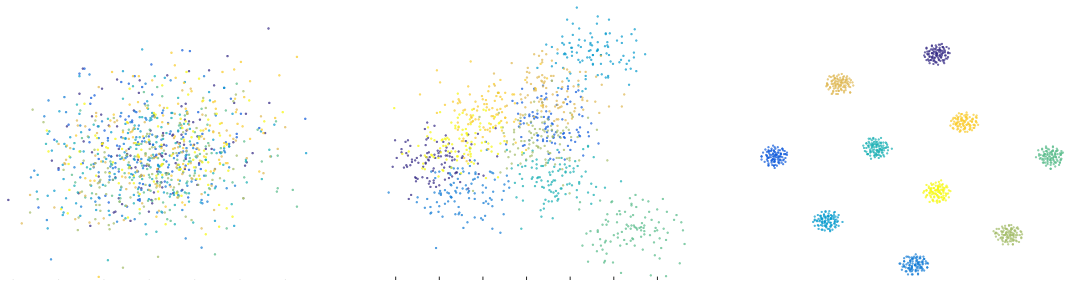
1. Introduction

Many scientific applications, especially those involving exploratory data analysis, rely on visually identifying high-level qualitative structures in the data, such as clusters or groups of similar points. This is not easy since the data of interest is usually high-dimensional and it is unclear how to capture the qualitative cluster structure in a 2-dimensional visualization. For example, linear dimensionality reduction techniques (e.g., data oblivious Johnson-Lindenstrauss (JL) embedding or data-dependent embedding using PCA) are incapable of reducing dimension down to 2 in any meaningful way (see Figure 1) - they merge distinct clusters into a uniform-looking sea of points.

In 2008, van der Maaten and Hinton (2008) introduced a nonlinear algorithm, *t-Distributed Stochastic Neighbor Embedding or t-SNE* (an improvement over the earlier SNE algorithm of Hinton and Roweis (2002)) for this task, which has become the de facto standard (see Figure 1(*c*)) for

. Extended abstract. Full version appears as arXiv:1803.01768.

© 2018 S. Arora, W. Hu & P.K. Kothari.



(*a*) (*b*) (*c*)

Figure 1: 2D embeddings of a mixture of 10 Gaussians with pairwise center separation $0.5\times$radius via: (a) random projection (JL), (b) projection to the subspace of top 2 singular vectors (PCA), (c) t-SNE.

visualizing high-dimensional datasets with diverse applications such as computer security (Gashi et al., 2009), music analysis (Hamel and Eck, 2010), cancer biology (Abdelmoula et al., 2016) and bioinformatics (Wallach and Lilien, 2009).

At a high level, t-SNE (like SNE) chooses two similarity measures between pairs of points - one for the high dimensional data and one for the 2-dimensional embedding. It then attempts to construct a 2-dimensional embedding that minimizes the KL divergence between the vector of similarities between pairs of points in the original dataset and the similarities between pairs of points in the embedding. This is a non-convex optimization problem and t-SNE employs gradient descent with random initialization (along with other tricks such as *early exaggeration*) to compute a reasonable solution to it. See the full version of this paper for details.

Of course, non-convex optimization drives much of today's progress in machine learning and data science, and thus poses a rich set of theoretical questions. Researchers have managed to rigorously analyze non-convex optimization algorithms in a host of settings (Dasgupta, 1999; Arora et al., 2012, 2014; Bhojanapalli et al., 2016; Ge et al., 2015a; Sun et al., 2017; Ge et al., 2017, 2016; Park et al., 2017). These analyses usually involve making clean assumptions about the structure of data, usually with a generative model. The goal of the current paper is to rigorously analyze t-SNE in a similar vein.

At the outset such a project runs into definitional issues about what a good *visualization* of clustering is. Many such issues are inherited from well-known issues in formalizing the goals of clustering (Kleinberg, 2002). In theoretical studies of clustering, such issues were sidestepped by going with a standard clustering formalization and assuming that data come with an (unknown) ground-truth clustering (for instance, mixtures of Gaussians, k-means, etc.). We make similar assumptions and assume that our goal is to produce a 2-dimensional embedding such that the points in the same clusters are noticeably closer together compared with points in different clusters. Under some of these standard models we show that t-SNE provably succeeds in computing a good visualization.

We emphasize that the focus of this paper is on formalizing the notion of visualization and providing a theoretical analysis of t-SNE. We do not advocate for t-SNE over other visualization methods.

We now begin by describing our formalization of the visualization problem followed by describing our results that give the first provable guarantees on t-SNE for computing visualization of clusterable data.

Formalizing Visualization. We assume that we are given a collection of points $\mathcal{X} = \{x_1, x_2, \ldots, x_n\} \subset \mathbb{R}^d$ and that there exists a "ground-truth" clustering described by a partition $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_k$ of $[n]$ into k clusters.

A visualization is described by a 2-dimensional embedding $\mathcal{Y} = \{y_1, y_2, \ldots, y_n\} \subseteq \mathbb{R}^2$ of $\mathcal{X}$, where each $x_i \in \mathcal{X}$ is mapped to the corresponding $y_i \in \mathcal{Y}$. Intuitively, a cluster $\mathcal{C}_\ell$ in the original data is *visualized* if the corresponding points in the 2-dimensional embedding $\mathcal{Y}$ are well-separated from all the rest. The following definition formalizes this idea.

Definition 1.1 (Visible cluster) *Let $\mathcal{Y}$ be a 2-dimensional embedding of a dataset $\mathcal{X}$ with ground-truth clustering $\mathcal{C}_1, \ldots, \mathcal{C}_k$. Given $\epsilon \geq 0$, a cluster $\mathcal{C}_\ell$ in $\mathcal{X}$ is said to be* $(1-\epsilon)$-visible *in $\mathcal{Y}$ if there exist $\mathcal{P}, \mathcal{P}_{\text{err}} \subseteq [n]$ such that:*

1. $|(\mathcal{P} \setminus \mathcal{C}_\ell) \cup (\mathcal{C}_\ell \setminus \mathcal{P})| \leq \epsilon \cdot |\mathcal{C}_\ell|$, $|\mathcal{P}_{\text{err}}| \leq \epsilon n$, *and*
2. *for every $i, i' \in \mathcal{P}$ and $j \in [n] \setminus (\mathcal{P} \cup \mathcal{P}_{\text{err}})$, $\|y_i - y_{i'}\| \leq \frac{1}{2}\|y_i - y_j\|$.*

In such a case, we say that $\mathcal{P}$ $(1-\epsilon)$-visualizes *$\mathcal{C}_i$ in $\mathcal{Y}$.*

It is now easy to define when $\mathcal{Y}$ is a good *visualization* - we ask that every cluster $\mathcal{C}_\ell$ in the dataset $\mathcal{X}$ is visualized in $\mathcal{Y}$.

Definition 1.2 (Visualization) *Let $\mathcal{Y}$ be a 2-dimensional embedding of a dataset $\mathcal{X}$ with ground-truth clustering $\mathcal{C}_1, \ldots, \mathcal{C}_k$. Given $\epsilon \geq 0$, we say that $\mathcal{Y}$ is a* $(1-\epsilon)$-visualization *of $\mathcal{X}$ if there exists a partition $\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_k, \mathcal{P}_{\text{err}}$ of $[n]$ such that:*

(i) For each $i \in [k]$, $\mathcal{P}_i$ $(1-\epsilon)$-visualizes $\mathcal{C}_i$ in $\mathcal{Y}$, and

(ii) $|\mathcal{P}_{\text{err}}| \leq \epsilon n$.

In particular, when $\epsilon = 0$, we say that $\mathcal{Y}$ is a full visualization *of $\mathcal{X}$.*

Remark *Note that this formalization of visualization should be considered a first cut, since ultimately human psychology must come into play. For instance, humans may reasonably visualize two parallel lines as two clusters, but these violate our definition.*

A natural question is whether clustering inferred from a visualization is unique. Our definition above does not guarantee this. Indeed, this is inherently impossible and relates to the ambiguity in the definition of clustering: for example, it can be impossible to determine whether a given set of points should be viewed as one cluster or two different smaller clusters. See Figure 2 for an example.

It is, however, not hard to establish that under an additional assumption that the size (number of points) of any cluster is smaller than twice the size of any other, full visualization as defined in Definition 1.2 uniquely determines a clustering.

In order to study fine-grained behaviors of t-SNE, we also define a weaker variant of visualization where at least one cluster is visualized.

Definition 1.3 (Partial visualization) *Given $\epsilon \geq 0$, we say that $\mathcal{Y}$ is a* $(1-\epsilon)$-partial visualization *of $\mathcal{X}$ if there exists a subset $\mathcal{P} \subseteq [n]$ such that $\mathcal{P}$ $(1-\epsilon)$-visualizes $\mathcal{C}_\ell$ for some $\ell \in [k]$.*

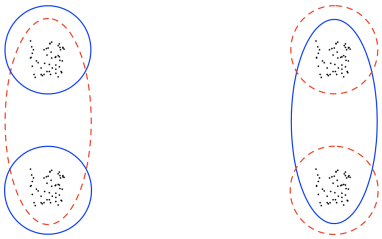


Figure 2: If we knew that there are 3 clusters in the original data, the blue and red outlines denote equally valid guesses for the underlying clustering based on the above visualization.

1.1. Our Results

Our main result identifies a simple deterministic condition on the clusterable data under which t-SNE provably succeeds in computing a full visualization.

Definition 1.4 (Well-separated, spherical data) *Let $\mathcal{X} = \{x_1, x_2, \ldots, x_n\} \subset \mathbb{R}^d$ be clusterable data with $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_k$ defining the individual clusters such that for each $\ell \in [k]$, $|\mathcal{C}_\ell| \geq 0.1(n/k)$. We say that $\mathcal{X}$ is γ-spherical and γ-well-separated if for some $b_1, b_2, \ldots, b_k > 0$, we have:*

1. *γ-**Spherical:** For any $\ell \in [k]$ and $i, j \in \mathcal{C}_\ell$ $(i \neq j)$, we have $\|x_i - x_j\|^2 \geq \frac{b_\ell}{1+\gamma}$, and for any $i \in \mathcal{C}_\ell$ we have $\left|\left\{j \in \mathcal{C}_\ell \setminus \{i\} : \|x_i - x_j\|^2 \leq b_\ell\right\}\right| \geq 0.51|C_\ell|$.*
2. *γ-**Well-separated:** For any $\ell, \ell' \in [k]$ $(\ell \neq \ell')$, $i \in \mathcal{C}_\ell$ and $j \in \mathcal{C}_{\ell'}$ we have $\|x_i - x_j\|^2 \geq (1 + \gamma \log n) \max\{b_\ell, b_{\ell'}\}$.*

The first condition asks for the distances between points in the same cluster ("intra-cluster distances") to be concentrated around a single value (with γ controlling the "amount" of concentration). The second condition requires that the distances between two points from different clusters should be somewhat larger than the intra-cluster distances for each of the two clusters involved. In addition, we require that none of the clusters has too few points. Such assumptions are satisfied by well-studied probabilistic generative models for clusterable data such as mixture of Gaussians and more generally, mixture of log-concave distributions, and have been used in previous work (Dasgupta, 1999; Arora and Kannan, 2005) studying "distance-based" clustering algorithms.

For spherical and well-separated data, our main theorem below shows that t-SNE with early exaggeration succeeds in finding a full visualization.

Theorem 1.5 (Informal) *Let $\mathcal{X} = \{x_1, x_2, \ldots, x_n\} \subset \mathbb{R}^d$ be γ-spherical and γ-well-separated clusterable data with $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_k$ defining the individual clusters. Then, t-SNE with early exaggeration on input $\mathcal{X}$ outputs a full visualization of $\mathcal{X}$ with high probability.*

Proof Technique. At a high level, t-SNE starts with a randomly initialized embedding and makes iterative gradient updates to it. The analysis thus demands understanding the effect of this update rule to the embedding of the high-dimensional points as a function of whether they lie in the same cluster or not. In a recent work, Linderman and Steinerberger (2017) established a "shrinkage" result for this update rule - they showed that points in the same cluster move towards each other under some mild conditions, that is, the embedding of any cluster "shrinks" as the iterations proceed.

This result, however, is insufficient to establish that t-SNE succeeds in finding a full visualization as it does not rule out multiple clusters merging into each other.

We resort to a more fine-grained analysis built on the one by Linderman and Steinerberger (2017) and obtain an update rule for the *centroids* of the embeddings of all underlying clusters. This allows us to track the changes to the positions of the centroids and show that the distance between distinct centroids remains *lower-bounded* whenever the data is γ-spherical and γ-well-separated. Combined with the shrinkage result for points in the same cluster, this implies that t-SNE outputs a full visualization of the data.

Our analysis implicitly relies on the update rule in t-SNE closely mimicking those appearing in the well-studied *noisy power method* (with non-random noise). We make this connection explicit and show that the behavior of t-SNE (with early exaggeration) on γ-spherical and well-separated data can in fact be closely approximated by power method run on a natural matrix of pairwise similarities.

Application to Visualizing Mixture Models. Mixture of Gaussians and more generally, mixture of log-concave distributions, are well-studied probabilistic generative models for clusterable data. As an immediate application of our main theorem above, we show that t-SNE produces a full visualization for data generated according to such models. Before describing the result, we quickly recall the definition of mixture of log-concave distributions.

A distribution $\mathcal{D}$ with density function f on $\mathbb{R}^d$ is said to be *log-concave* if $\log(f)$ is a concave function. $\mathcal{D}$ is said to be *isotropic* if its covariance is I. Many natural distributions including Gaussian distributions and the uniform distribution on any convex set are log-concave.

A mixture of k log-concave distributions is described by k positive mixing weights $w_1, w_2, \ldots, w_k$ ($\sum_{\ell=1}^{k} w_\ell = 1$) and k log-concave distributions $\mathcal{D}_1, \ldots \mathcal{D}_k$ in $\mathbb{R}^d$. To sample a point from this model, we pick cluster ℓ with probability w_ℓ and draw x from $\mathcal{D}_\ell$.

Theorem 1.5 immediately implies that t-SNE constructs a full visualization for data generated from a mixture of isotropic Gaussians or log-concave distributions with well-separated means. For isotropic Gaussians, the required pairwise separation between means is $\tilde{\Omega}(d^{1/4})$. For more general isotropic log-concave distributions, we require that the means be separated by $\tilde{\Omega}(d^{5/12})$.

Observe that the radius of samples from an isotropic log-concave distribution is $\approx d^{1/2}$ - thus, t-SNE succeeds in constructing 2D visualizations for clustering models far below the separation at which the clusters are non-overlapping. This is in stark contrast to standard linear dimensionality reduction techniques such as the Johnson-Lindenstrauss embedding that require mean separation of $\Omega(d^{1/2})$ to construct 2D visualizations that correctly separate 99% of points.

Corollary 1.6 (Informal) *Let $\mathcal{X} = \{x_1, \ldots, x_n\} \subset \mathbb{R}^d$ be i.i.d. samples from an equal-weighted mixture of k isotropic Gaussians in $\mathbb{R}^d$ with every pair of distinct means separated by $\tilde{\Omega}(d^{1/4})$. Then, with high probability t-SNE with early exaggeration on input $\mathcal{X}$ outputs a full visualization of $\mathcal{X}$. Moreover, the same result holds for mixture of isotropic log-concave distributions with every pair of distinct means separated by $\tilde{\Omega}(d^{5/12})$.*

Remark *Our result actually holds for a larger subclass of mixtures of* non-isotropic *log-concave distributions that may not be equal-weighted. See the full version of this paper for details. Mixture of log-concave distributions is among the weakest assumptions under which clustering algorithms with provable guarantees have been designed (Arora and Kannan, 2005; Vempala and Wang, 2004).*

We show that the t-SNE heuristic can visualize clusters under assumptions similar to the more sophisticated methods in previous theoretical work.

Finally, we show that even when the conditions in Definition 1.4 are not met, t-SNE can still provably visualize at least one cluster in the original data in some cases. As an example, using a more fine-grained analysis, we show that t-SNE computes a partial visualization for data obtained from a mixture of two *concentric* (thus, no mean separation at all!) spherical Gaussians with variances differing by a constant factor.

Theorem 1.7 (Informal) *Let $\mathcal{X}$ be generated from an equal-weighted mixture of two Gaussians $\mathcal{N}(0, \sigma_1^2)$ and $\mathcal{N}(0, \sigma_2^2)$ such that $1.5 \leq \sigma_2/\sigma_1 \leq 10$. Then t-SNE with early exaggeration on input $\mathcal{X}$ outputs a $(1 - d^{-\Omega(1)})$-partial visualization of $\mathcal{X}$ where $\mathcal{C}_1$ is $(1 - d^{-\Omega(1)})$-visible.*

1.2. Related Work

This paper continues the line of work focused on analyzing gradient descent and related heuristics for non-convex optimization problems, examples of which we have discussed before. Theoretically analyzing t-SNE, in particular, was recently considered in a work of Linderman and Steinerberger (2017) who showed that running t-SNE with early exaggeration causes points from the same cluster to move towards each other (i.e., embedding of any cluster shrinks). As discussed before, however, this does not imply that t-SNE ends up with a visualization as all the clusters could potentially collapse into each other. Another work by Shaham and Steinerberger (2017) derived a theoretical property of SNE, but their result is only nontrivial when the number of clusters is significantly larger than the number of points per cluster, which is an unrealistic assumption.

Mixture models are natural average-case generative models for clusterable data which have been studied as benchmarks for analyzing various clustering algorithms and have a long history of theoretical work. By now, a sequence of results (Dasgupta et al., 2007, 2006; Arora and Kannan, 2005; Vempala and Wang, 2004; Achlioptas and McSherry, 2005; Kannan et al., 2005; Vempala, 2007; Hsu and Kakade, 2013; Ge et al., 2015b; Kalai et al., 2012; Belkin and Sinha, 2010; Kalai et al., 2010; Kothari and Steinhardt, 2017; Hopkins and Li, 2017; Diakonikolas et al., 2017) have identified efficient algorithms for clustering data from such models under various natural assumptions.

Acknowledgments

This research was done with support from NSF, ONR, Simons Foundation, Mozilla Research, and Schmidt Foundation.

References

Walid M. Abdelmoula, Benjamin Balluff, Sonja Englert, Jouke Dijkstra, Marcel J. T. Reinders, Axel Walch, Liam A. McDonnell, and Boudewijn P. F. Lelieveldt. Data-driven identification of prognostic tumor subpopulations using spatially mapped t-sne of mass spectrometry imaging data. *Proceedings of the National Academy of Sciences*, 113(43):12244–12249, 2016.

Dimitris Achlioptas and Frank McSherry. On spectral learning of mixtures of distributions. In *18th Annual Conference on Learning Theory*, pages 458–469, 2005.

Sanjeev Arora and Ravi Kannan. Learning mixtures of separated nonspherical gaussians. *The Annals of Applied Probability*, 15(1A):69–92, 2005.

Sanjeev Arora, Rong Ge, and Ankur Moitra. Learning topic models—going beyond SVD. In *2012 IEEE 53rd Annual Symposium on Foundations of Computer Science—FOCS 2012*, pages 1–10. IEEE Computer Soc., Los Alamitos, CA, 2012.

Sanjeev Arora, Rong Ge, and Ankur Moitra. New algorithms for learning incoherent and overcomplete dictionaries. In *COLT*, volume 35 of *JMLR Workshop and Conference Proceedings*, pages 779–806. JMLR.org, 2014.

Mikhail Belkin and Kaushik Sinha. Polynomial learning of distribution families. In *FOCS*, pages 103–112. IEEE Computer Society, 2010.

Srinadh Bhojanapalli, Behnam Neyshabur, and Nathan Srebro. Global optimality of local search for low rank matrix recovery. In *NIPS*, pages 3873–3881, 2016.

Anirban Dasgupta, John E. Hopcroft, Ravi Kannan, and Pradipta Prometheus Mitra. Spectral clustering by recursive partitioning. In *ESA*, volume 4168 of *Lecture Notes in Computer Science*, pages 256–267. Springer, 2006.

Anirban Dasgupta, John E. Hopcroft, Ravi Kannan, and Pradipta Prometheus Mitra. Spectral clustering with limited independence. In *SODA*, pages 1036–1045. SIAM, 2007.

Sanjoy Dasgupta. Learning mixtures of gaussians. In *FOCS*, pages 634–644, 1999.

Ilias Diakonikolas, Daniel M. Kane, and Alistair Stewart. List-decodable robust mean estimation and learning mixtures of spherical gaussians. *CoRR*, abs/1711.07211, 2017.

Ilir Gashi, Vladimir Stankovic, Corrado Leita, and Olivier Thonnard. An experimental study of diversity with off-the-shelf antivirus engines. In *Proceedings of The Eighth IEEE International Symposium on Networking Computing and Applications, NCA 2009, July 9-11, 2009, Cambridge, Massachusetts, USA*, pages 4–11, 2009.

Rong Ge, Furong Huang, Chi Jin, and Yang Yuan. Escaping from saddle points - online stochastic gradient for tensor decomposition. In *COLT*, volume 40 of *JMLR Workshop and Conference Proceedings*, pages 797–842. JMLR.org, 2015a.

Rong Ge, Qingqing Huang, and Sham M. Kakade. Learning mixtures of gaussians in high dimensions. In *STOC*, pages 761–770. ACM, 2015b.

Rong Ge, Jason D. Lee, and Tengyu Ma. Matrix completion has no spurious local minimum. *CoRR*, abs/1605.07272, 2016.

Rong Ge, Chi Jin, and Yi Zheng. No spurious local minima in nonconvex low rank problems: A unified geometric analysis. In *ICML*, pages 1233–1242, 2017.

Philippe Hamel and Douglas Eck. Learning features from music audio with deep belief networks. In *ISMIR*, pages 339–344. International Society for Music Information Retrieval, 2010.

Geoffrey E. Hinton and Sam T. Roweis. Stochastic neighbor embedding. In *NIPS*, pages 833–840, 2002.

Samuel B. Hopkins and Jerry Li. Mixture models, robustness, and sum of squares proofs. *CoRR*, abs/1711.07454, 2017.

Daniel Hsu and Sham M. Kakade. Learning mixtures of spherical Gaussians: moment methods and spectral decompositions. In *ITCS'13—Proceedings of the 2013 ACM Conference on Innovations in Theoretical Computer Science*, pages 11–19. ACM, New York, 2013.

Adam Tauman Kalai, Ankur Moitra, and Gregory Valiant. Efficiently learning mixtures of two gaussians. In *STOC*, pages 553–562. ACM, 2010.

Adam Tauman Kalai, Ankur Moitra, and Gregory Valiant. Disentangling gaussians. *Commun. ACM*, 55(2):113–120, 2012.

Ravindran Kannan, Hadi Salmasian, and Santosh Vempala. The spectral method for general mixture models. In *COLT*, pages 444–457, 2005.

Jon M. Kleinberg. An impossibility theorem for clustering. In *NIPS*, pages 446–453, 2002.

Pravesh K. Kothari and Jacob Steinhardt. Better agnostic clustering via relaxed tensor norms. *CoRR*, abs/1711.07465, 2017.

George C Linderman and Stefan Steinerberger. Clustering with t-sne, provably. *arXiv preprint arXiv:1706.02582*, 2017.

Dohyung Park, Anastasios Kyrillidis, Constantine Caramanis, and Sujay Sanghavi. Non-square matrix sensing without spurious local minima via the burer-monteiro approach. In *AISTATS*, pages 65–74, 2017.

Uri Shaham and Stefan Steinerberger. Stochastic neighbor embedding separates well-separated clusters. *arXiv preprint arXiv:1702.02670*, 2017.

Ju Sun, Qing Qu, and John Wright. Complete dictionary recovery over the sphere I: overview and the geometric picture. *IEEE Trans. Information Theory*, 63(2):853–884, 2017.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

Santosh Vempala. Spectral algorithms for learning and clustering. In *COLT*, volume 4539 of *Lecture Notes in Computer Science*, pages 3–4. Springer, 2007.

Santosh Vempala and Grant Wang. A spectral algorithm for learning mixture models. *Journal of Computer and System Sciences*, 68(4):841–860, 2004.

Izhar Wallach and Ryan Lilien. The protein-small-molecule database (psmdb), a non-redundant structural resource for the analysis of protein-ligand binding. *Bioinformatics, 25(5):615-20*, 2009.

ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/313737530

Review on Customer Segmentation Technique on Ecommerce

Article *in* Journal of Computational and Theoretical Nanoscience · October 2016

DOI: 10.1166/asl.2016.7985

CITATIONS
62

READS
27,884

4 authors:

Juni Nurma Sari
Caltex Riau Polytechnic
18 PUBLICATIONS **120** CITATIONS
SEE PROFILE

Lukito Nugroho
Universitas Gadjah Mada
231 PUBLICATIONS **2,085** CITATIONS
SEE PROFILE

Ridi Ferdiana
Universitas Gadjah Mada
230 PUBLICATIONS **1,494** CITATIONS
SEE PROFILE

Paulus Insap Santosa
Universitas Gadjah Mada
278 PUBLICATIONS **2,619** CITATIONS
SEE PROFILE

All content following this page was uploaded by Juni Nurma Sari on 15 July 2018.

The user has requested enhancement of the downloaded file.

Copyright © 2011 American Scientific Publishers
All rights reserved
Printed in the United States of America

Advanced Science Letters
Vol. 4, 400–407, 2011

Review on Customer Segmentation Technique on Ecommerce

Juni Nurma Sari[1,2], Lukito Edi Nugroho[1],Ridi Ferdiana[1] ,P. Insap Santosa[1]

[1]Department of Electrical Engineering and Information Technology, University of Gadjah Mada, Jogjakarta, Indonesia
[2]Department of Informatics Technology, Polytechnic Caltex Riau, Pekanbaru, Indonesia

Ecommerce transactions are no longer a new thing. Many people shop with ecommerce and many companies use ecommerce to promote and to sell their products. Because of that, overloading information appears on the customers' side. Overloading information occurs when customers get too much information about a product then feel confused. Personalization will become a solution to overloading problem. In marketing, personalization technique can be used to get potential customers in a case to boost sales. The potential customer is obtained from customer segmentation or market segmentation. This paper will review customer segmentation using data, methods and process from a customer segmentation research.The data for customer segmentation were divided into internal data and external data. Customer profile and purchase history were treated as the internal data while server log, cookies, and survey data were as the external data. These data can be processed using one of several methods: Business Rule, Magento, Customer Profiling, Quantile Membership, RFM Cell Classification Grouping, Supervised Clustering, Customer Likeness Clustering, Purchase Affinity Clustering and Unsupervised Clustering. In this paper, those methods were classified into Simple technique, RFM technique, Target technique, and Unsupervised technique and the process was generalized in determining the business objective, collecting data, data preparation, variable analysis, data processing, and performance evaluation. Customer behavior in accessing ecommerce when viewing a product on ecommerce was recorded in server log with time. Duration when seeing the product can be used as customer interest in the product so that it can be used as a variable in customer segmentation.

Keywords: Ecommerce, Customer Segmentation, Personalization

1. INTRODUCTION

Ecommerce development began when the internet is growing and growing until today, especially in B2C ecommerce (Business to Customer). When shopping use ecommerce, a user finds it easy and faster. The ease of using ecommerce encourages customers to buy using ecommerce. With these conditions the problem that comes up is the overloading information because of many products offered by ecommerce[1]. Overloaded information can be overcome by an implementation of personalization in ecommerce services such as providing product recommendation, links recommendation, ads or text and graphics that correspond to the users' characteristics and needs[2]. In addition to solving the problem of overloaded information, personalized services in ecommerce can maintain customer loyalty of existing customer[3], getting new customers by providing service to customers in accordance with their needs and characteristics. It will generate more profits for the company. Before the personalization is implemented, customer segmentation should be conducted because the result from customer segmentation process will be used as inputs to personalize ecommerce services, resulting in dynamic personalization ecommerce services based on current customer conditions.

Customer segmentation is currently performed by processing customer database, i.e. demographic data or purchase history. Several researchers discuss the customer segmentation method on their papers, such as Magento[4], who used several variables to perform customer segmentation, namely transaction variable, product variable, geographic variable, hobbies variable and page viewed variable; Baer[5] and Colica[6] discuss customer segmentation methods of Business Rule, Quantile membership, Supervised Clustering, Unsupervised Clustering, Customer Profiling, RFM Cell Classification Grouping, Customer Likeness Clustering and Purchase Affinity Clustering. Some of these methods have similarity. Other researchers discuss the implementation of customer segmentation. This paper will classify customer segmentation methods based on data processing.

*Email Address: juni.s3te14@mail.ugm.ac.id

1Adv. Sci. Lett. Vol. 4, No. 2, 2011 1936-6612/2011/4/400/008 doi:10.1166/asl.2011.1261

2.CUSTOMER SEGMENTATION

In marketing, one way to increase profits is to communicate with customers to determine customer wishes[5]. Communication is built according to the characteristics of the customer. Communication is very difficult to create using personal approaches. So it is necessary to divide customers into groups that have the same characteristics, and this is called customer segmentation. Schneider[7] also called market segmentation that divides potential customers into a group. Magento[4], an ecommerce platform, in its ebook mentions that customer segmentation is an activity to divide customers into groups that have the same characteristics. Customer segmentation has several benefits: it enables us to match between the customer and an offer of similar products; it changes the way we communicate with the customer based on customer data; it identifies the most profitable customers; and it enables us to update the products and services to meet customer needs. Baer[5] states that customer segmentation is the activity to categorize or to classify an item or subject to a group that has been identified to have in common. In his research, Baer discusses Customer Segmentation Intelligence to improve marketing in offering products or services that meet the needs of each customer group. Segmentation according to Collica[6] is the process to categorize or classify an item into a group that has a similarity in characteristic and in CRM (Customer Relationship Management) segmentation is used to classify customer based on some similarities by segmenting the records of customer database. This chapter will discuss the customer data for customer segmentation, customer segmentation methods and customer segmentation process and then the methods will be classified based on data processing.

A. Data for Customer Segmentation

Customer segmentation requires customer data from various sources. Magento[4] categorizes the data into internal data and external Data. Customer registration, customer profile, and purchase history are the internal data obtained from the database of an ecommerce. While external data are census data, media browsing, surveys and market search, cookies, web and social media analysis. Information about customer lifestyle, attitude, activity and shopping preferences are obtainable through surveys and market search and social media. Browsing history can be seen from server log or cookies. Baer[5] in his research, Customer Segmentation Intelligence, uses internal data by looking the demographic data from customer profile and purchase history. Likewise, Colica[6], uses the customer database and purchase history on customer segmentation methods.

B. Methods of Customer Segmentation

Customer segmentation can be performed using various approaches. Theoretically, Schneider[7] divides customer segmentation methods into geographic, demographic, psychographic, behavioral/occasion, usage-based market segmentation. Geographic segmentation is based on location. Demographic segmentation is based on age, gender, family size, income, education, religion or ethnic. Psychographic segmentation is based on social class, personality or their approach to living. Behavioral segmentation is based on customer behavior but when customer behavior occurs in specific time or occasion, Schneider called it Occasion segmentation. Usage-based Market segmentation is based on behavior pattern of each visitor, which includes a set of categories of customer namely browser, buyer and shopper. Browsers are visitors that just browse a site; buyers are visitors that make a purchase; and shoppers are customers that want to buy, but want to read product reviews and the list of features before buying.

Almost the same with Schneider, Magento divides customer segmentation methods into Profit Potential, Past Purchase, Demographic, Psychographic and Behavior. In Magento[4] there are several variables used:

1) *Profit Potential:* using variable transaction frequency, date of last purchase, average order value, customer lifetime value.
2) *Past Purchases:* using the variable of product type/attribute, product price, payment/shipping method used, product benefit sought (price, quality, prestige), product satisfaction.
3) *Demographic:* using the variable of geographic location (city state, country, region), age, gender, household size, income, occupation, education, ethnicity, browsing device (laptop, PC, tablet, smartphone) and type (vendor and model), traffic source (organic search, banner link, referral site).
4) *Psychographic:* using the variable of hobbies and interest, leisure and recreational activity, affiliations (religious, professional, cultural, political, institutional), personal traits (social vs. private; modern vs. traditional; spontaneous vs. cautious).
5) *Behavior:* using the variable of pages viewed, responses to offers and promotions, participation in reward programs, channel management.

Magento also performed an analysis of purchase history to get the best customer, unprofitable customers, potential customer profit. Best customer is when the customer is a frequent shopper and a repeated customer, with high average order value, low return, providing review and response customer. Unprofitable customer when the customer has high rate product return, low average order value, high rate customer service calls, wants the lowest price. Potential customer profit is determined by counting customer lifetime values.

Baer[5] segments customer using business rules method, quantile membership method, supervised clustering with decision tree method and unsupervised clustering method using k-means algorithm. Demography

data and purchase pattern are used to segment costumers. Here are Baer customer segmentation methods:

1) *Bussiness Rule:* in this method, customers are grouped into specific groups based on a predetermined class, such as:
   a) Grouping based on demographic data, such as age, gender, income and education, etc. This method has similarity with Magento and Schneider.
   b) Grouping based on customer interaction with the company based on data purchase pattern such as the type of product or service provided or RFM data, where R is Recency (when customer last shopped), F is Frequency (how often the customer shops) and M is Monetary (how much the customer spends)

   According to Baer, the lack of business rule does not reflect the actual customer behavior and a segment similar to another segment.
2) *Quantile Membership,* this method uses data Recency, Frequency, and Monetary. Here is the quantile membership methods:
   a) Recency divided into five groups of intervals, for example, starting from 0 days up to 730 days then classify it with label A until E, where A is very valuable customer and E is low-value customer. Also with Frequency and Monetary. When 3 RFM is combined, there is label AAA until EEE.
   b) Map two components of RFM to a table.
   c) Divided into two groups A, B with the classification most valuable customer and two groups D, E to the classification of least valuable customer. C is average value customer.
   d) The result can be inferred for example good frequency (A or B), good monetary (A or B) but poor recency (D or E), and then the advice that given is upgrade the promotion strategy to make the old customer come back
3) *Supervised Clustering with decision tree:* this method uses a specific target, or dependent variable and target would predict differences in independent variables (input). Data utilized in this method is previous purchase pattern and customer demographic.The algorithm that used is decision tree with the target on their nodes. According to Baer, although this method connects the target with the other customer attributes, it shows only one aspect of customer behavior.
4) *Unsupervised Clustering*: this method uses any number of customer attributes then measure the similarity among customer, each customer attribute use Euclidean distance[8] (1) then cluster the customer use k-means clustering[9] (2). If the distance is the shortest distance between customer data and cluster, then customer is included in that cluster.

$$\text{Euclidean distance}=\sqrt{(X_A - X_B)^2}+\ldots+(X_A - X_B)^2 \quad (1)$$

$$C(i)= \arg\min \sum_{i=1}^{k} \sum_{x \in C_i} ||x - \mu_i||^2 \quad (2)$$

Colica has several methods are almost the same. Colica has segmentation methods as follows: Customer Profiling, Customer Likeness Clustering, RFM Cell Classification grouping and Purchase Affinity Clustering. In Customer Profiling method, the required information about customer is the fourW's (who, what, where and when) from customer database.It can be done by using a query on the customer database or using the clustering algorithm when the data is huge. Customer likeness clustering method is used in franchise stores to know whether the profits and turnover of each product in each store are similar, then to review other variables such as demographics. Colica also uses a decision tree for simple clustering the same with Baer. Method RFM (Recency Frequency Monetary) Cell Classification Grouping uses three dimensions to classify each customer in one cell after labeling each level of RFM. Colica names it the Segmentation Using Cell-Based Approach. This method is similar to Baer's quantile membership. Another method used by Colica is Purchase Affinity Clustering. This method uses scoring on interesting in certain products then clusters customer database based on that score to get a similar group.

Table 1. Methods of Customer Segmentation

Paper	Method	Data	Advantage	Disadvantage
Magento (2014)	Magento	Demographic, Purchase History, Data Product, Data Media, Data Marketing, Server Log	Have clear variable customer segmentation	There is no data processing for each variable
Baer (2012)	Bussiness Rule	Demographic, Purchase history	Easy to apply, Use database query	Not focus on customer behavior
	Quantile membership	Purchase history	Can process small data, can be used with other data	Good result obtained when determining a good classification
	Supervised Clustering with decision tree	Demographic, Purchase history	Classify customers according to target	Use one variable to cluster
	Unsupervised Clustering	Purchase history	Use any number of customer attributes	Speed of computation depends on k values
Colica (2011)	Customer Profiling	Demographic, Purchase history	use database query if data is small	Not focus on behavior
	Customer Likeness Clustering	Demographic, Purchase History, Data product	classify customers according to the target	Problem arises when there are different unit in record
	RFM Cell Classification Grouping	Purchase history	Efficient three -dimensional mapping	Good result obtained when determining a good classification
	Purchase Affinity Clustering	Purchase history, Data product	know the products most in demand	Spesific to product segmentation

There are some researches that implement customer segmentation methods according to the table above such as Lieberman[10], who uses combination Business Rule, Customer Profiling, Magento to find how much customer spend money monthly on clothing and how many customers visit monthly; Dodwell[11], who uses RFM Analysis to segment email marketing for potential customer; Birant[12], who uses combination RFM Analysis and Data Mining (Classification Rules and Association Rules) to provide better product recommendation; Han[13], who uses Decision tree model to identify high-value customer; Ma[14], who uses Association Rules and Decision Tree to improve customer loyalty, attract new customer and expand the market effectively; Baer[15], who uses Market Base Analysis, K-means Clustering, and Doughnut Clustering to segments customer based on product, and Ezenkwu[9] and Venkatesan[8], who use K-means Clustering to segment customer.

Based on table and researches above, customer segmentation methods can be classified into: Simple technique, because this method uses database query and statistical data; RFM technique, because this method uses RFM analysis; Target technique, because this method must have target to segments customer, for instance, customer segmentation focus on product, focus on purchase; and Unsupervised technique, because this method uses dynamic data. Figure 1 describes Customer Segmentation Classification.

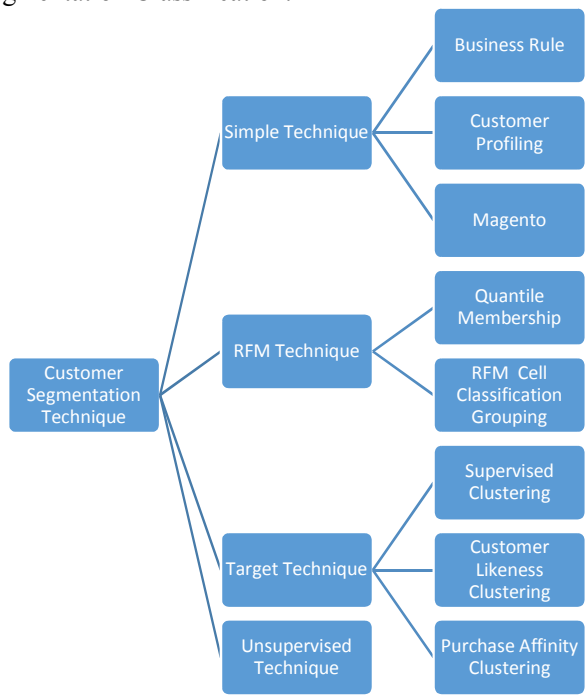


Figure 1. Customer Segmentation Classification

C. *Process of Customer Segmentation*

Customer Segmentation is associated with the business objective. The first step of segmentation is deciding business objective. Chen[16] discusses customer segmentation process begins by determining the business objective such as the identification of high profitable customer groups, improve product for that customer. The next step is collecting the necessary data such as demographic data, transaction data, and promotional data, then determining the method of customer segmentation and standardization measurement. After that, the next step is exploration data by analyzing the statistics and look for relationships between variables. Results of analysis can be used to measure the similarity among the customers using Euclidean distance to measure two points in a multidimensional space where the point is customer data. The cluster is validated by calculating the ratio of the between-cluster variantto within cluster variants(RSQ/1-RSQ).

Process Customer Segmentation on Lieberman[10] research begins with determining the business rule, collecting data spread the questioner, then data processing with logistic regression and waterfall and analyze statistic data. Birant[12] has a more complex process than Lieberman because he combines RFM Analysis and Data Mining to find product recommendation. Birant starts the process of defining the business objective, collecting data, and then data processing with the first method of RFM analysis that uses quantile membership to find customer level of Recency, Frequency, and Monetary. The second method is Clustering with RFM Cell Classification Grouping to find customer segmentation. After segmentation, there is prediction of customer behavior, it uses Association Rule method. Finally, the product recommendation uses Classification method. Process Customer Segmentation on Ma[17] research starts with defining the business objective, choosing variables that relate to purchasing then form data set, finding frequent item set use generalized association rule, cleaning non-interest rule, building tree process, prunning decision tree, extract rules from pruned decision tree in if-then format. Ezenkwu[9] process also starts with determining the business rule, choosing data variable, namely the amount of goods purchased by customer monthly and the average number of customers visiting monthly; the data processing with k-mean clustering which is normalization alongside centroids, initialization step, assignment step and updating step after that performance evaluation. Process of customer segmentation can be simplified into defining business objective, collecting data, data preparation, analyzing variable, data processing, and performance evaluation as describe in figure 2.

3. FUTURE WORK

One of the data used for customer segmentation is customer behavior in accessing ecommerce. Customer behavior data are obtainable from server log. Variables contained in server log are IP address of customer, date, time, HTTP request. Here an example of server log data:

```
05:09:49  GET /detail-item.php?item=ilford-delta-100 HTTP/1.0
05:09:53  GET /detail-item.php?item=ilford-pan-f-50 HTTP/1.0
```

Time shows when a customer accesses page, the difference of time between the customer's visit to the first page and the second page is the duration of customer's visit to the first page. The first data is page detail-item.php with the first product of ilford-delta-100 and the second data is page detail-item.php with the product of ilford-pan-f-50. Knowing data duration, we can determine the user's attention to the product. If the user's attention to the product is in long duration, then the customer has an affinity for product. It can also be used for customer segmentation based on the interest in the product. Such information can be utilized for the promotion of a product. The disadvantage of this method is when customer position isn't in front of computer but server still record the activity, so the solution is using an eye-tracker to record the customer's attention.

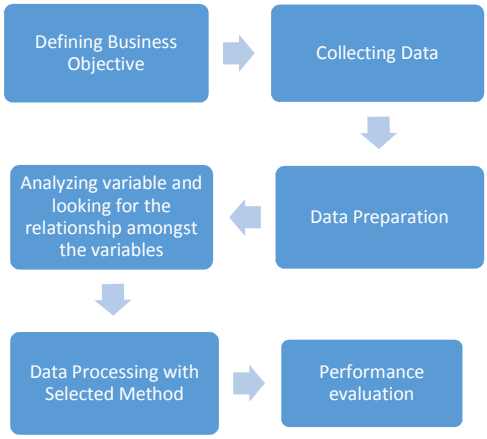


Figure 2. Process of Customer Segmentation

4. CONCLUSION

Customer segmentation is a way to improve communication with the customer, to know the wishes of the customer, customer activity so that appropriate communication can be built. Customer Segmentation needed to get potential customers used to increase profits. Potential customer data can be used to provide service the characteristics of customer including ecommerce services as a media buying and selling online.

This paper discusses several components to do customer segmentation, which is:

- Customer segmentation is an activity to divide customers or item into groups that have the same characteristics.
- Data that needed for customer segmentation are internal data and external data. The internal data include demographic data and data purchase history, while the external data include cookies and server logs. Internal data can be obtained from a database when customer do registration or transactions and external data can be obtained from web server or other source.
- Methods of Customer Segmentation can be classified into Simple technique, RFM technique, Target technique, and Unsupervised technique. On Target technique, researcher focus on one variable, it can be product or purchase. Unsupervised technique was used when clustering process reseacher have many variable
- Process of Customer Segmentation can be simplified into defining business objective, collecting data, data preparation, analyzing variable, data processing, and performance evaluation.

REFERENCES

[1] Al-Qaed F, Sutcliffe A. Adaptive Decision Support System (ADSS) for B2C E-Commerce. *2006 ICEC Eighth Int Conf Electron Commer Proc NEW E-COMMERCE Innov Conqu Curr BARRIERS, Obs LIMITATIONS TO Conduct Success Bus INTERNET*. 2006:492-503.

[2] Mobasher B, Cooley R, Srivastava J. Automatic Personalization Based on Web Usage Mining. *Commun ACM*. 2000;43(8).

[3] Cherna Y, Tzenga G. Measuring Consumer Loyalty of B2C e-Retailing Service by Fuzzy Integral: a FANP-Based Synthetic Model. In: *International Conference on Fuzzy Theory and Its Applications iFUZZY*.; 2012:48-56.

[4] Magento. An Introduction to Customer Segmentation. 2014. info2.magento.com/.../An_Introduction_to_Customer_Segmentation...

[5] Baer D. CSI : Customer Segmentation Intelligence for Increasing Profits. *SAS Glob Forum*. 2012:1-13. http://support.sas.com/resources/papers/proceedings12/103-2012.pdf.

[6] Colica R. Customer Segmentation And Clustering Using SAS Enterprise Minner Part I The Basics. 2011:1-14.

[7] Schneider G. *Electronic Commerce, 9th Edition*.; 2013:643. doi:10.1002/1521-3773(20010316)40:6<9823::AID-ANIE9823>3.3.CO;2-C.

[8] Venkatesan R. Cluster Analysis For Segmentation. 2007.

[9] Ezenkwu CP, Ozuomba S. Application of K-Means Algorithm for Efficient Customer Segmentation : A Strategy for Targeted Customer Services. 2015;4(10):40-44.

[10] Lieberman M. Target " golden egg " consumer to achieve maximum ROI. 2009;(May):50-51.

[11] Dodwell A. Effective Marketing email startegy _ Segmentation RFM. 2015. http://www.sailthru.com/marketing-blog/written-effective-email-marketing-strategies-segmentation-rfm/.

[12] Birant D. Data Mining Using RFM Analysis. *Knowledge-Oriented Appl Data Min*. 2011;(iii):91-108. doi:10.5772/13683.

[13] Hua S, Xiu S, Leung SCH. Expert Systems with Applications Segmentation of telecom customers based on customer value by decision tree model. *Expert Syst Appl*. 2012;39(4):3964-3973. doi:10.1016/j.eswa.2011.09.034.

[14] Ma H. A Study on Customer Segmentation for E-Commerce Using the Generalized Association Rules and Decision Tree. 2015;(December):813-818.

[15] Baer D, Ph D. Product Affinity Segmentation Using The Doughnut Clustering Approach. *Cust Intell SAS Glob Forum 2013*. 2013.

[16] Chen J. Retail Customer Segmentation. 2014;(April).

[17] Chan C, Swatman PMC. Management and business issues for B2B ecommerce implementation. *Proc 35th Annu Hawaii Int Conf Syst Sci*. 2002;00(c):1-11. doi:10.1109/HICSS.2002.994303.

View publication stats