KCRS: KClustering Recommender System for Component Configuration

Maurizio Monticelli[1], Ricardo Anibal Matamoros Aragon[2][0000−0002−1957−2530], Francesco Epifania[2], Luca Marconi[2][0000−0002−0236−6159], and Antonio De Simone[2][0000−0001−5217−7586]

[1] Nathan Instruments srl, Italy
monticelli@nathan.it
[2] Social Things SRL, Italy
{ricardo.matamoros,francesco.epifania,luca.marconi,antonio.desimone}@socialthingum.com

Abstract. In recent years, the exponential growth in the number of items and products handled by e-commerce sites has led to the introduction of intelligent systems aimed at supporting users during the decision-making proces.
Making the choice of a product among thousands of items becomes complicated for consumers, and in response to this problem, recommender systems (RS) are born. These systems are a set of algorithms based on the concept of information filtering and make it possible to reduce the cognitive effort required of users.
In this paper we present a model-based RS, belonging to the collaborative filtering (CF) category, for the e-commerce website of the company Nathan Instruments (NI). Thus, the main objective of this paper is to provide an intelligent approach for recommending configurations of hardware components for Computers. This configurator uses clustering algorithms to address the problems associated with small dataset sizes. Finally, in the experimentation and conclusion sections it is reported how the proposed model simplifies the decision process related to the required computer customization in terms of hardware and software components.

Keywords: Recommender Systems · Clustering · E-commerce · Machine Learning · Software Components.

1 Introduction

This article aims to present the intelligent model created by the NI team, which optimises the process of recommending computer configurations. This RS can be used both within NI's e-commerce and by the company's own technicians who assist customers in ordering and purchasing products in physical shops. [1]
Introducing the RS into NI's sales process optimises the customisation of the computer required in terms of hardware, software and various options. [18] At the same time, it is able to reduce the time and cost of producing the offers themselves[8], which were initially made from a catalogue offer and managed by human resources who had to take into account the compatibility of the basic components available.
Thus, the sales process without RS support was complex: Given the large number of hardware components, the cardinality of possible configurations grew exponentially and required the vendor to memorise the compatibility and budget constraints specified by the customer (PCs intended for office automation rather than processing large amounts of data, preference for a particular operating system, etc).
The introduction of the RS [6], on the other hand, automates the collection of requirements and technical specifications that the final configuration must present, and it is the customer who can enter this information in a simple and intuitive way [5]. The following sections describe state-of-the-art methodologies in the field of artificial intelligence (AI)[9,10], such as Machine Learning (ML) [7], used to continuously update knowledge on all the technical specifications of the various components.
In particular, "clustering methodologies" are also reported, which allow to homogeneously group similar configurations and components and thus suggest those that are less distant from user requirements. The storage of knowledge about the technical specifications of the hardware and software products in the catalogue and their processing was carried out through cloud-computing (CC). The article also reports on the experimentation phase carried out to identify the best type of clustering in the case study described, as well as the results and conclusions related to the performance of the RS.

2 Clustering Methodologies

For the realisation of the RS it was decided to analyse some clustering methodologies in order to identify similar products on the basis of predefined characteristics, the aim being to make efficient and compatible recommendations with respect to user requirements[12].
Two different methodologies were adopted for the subdivision of the products:

- clustering on the hardware components of the products within each user category, so as to search for different types of objects within the same user type;

- double clustering on the entire range of available products, with the aim of searching for substantial differences between the various objects and classifying them on the basis of the components of each.

Both methods follow using the k-means algorithm, based on vector distance. Although this methodology is the one applied by default, it was decided to study and implement other types of clustering through which similar solutions are reached.
In particular, the different methodologies analysed are:

2.1 K-Means

Defined $C_1, ..., C_K$ a set of groups containing the observations 1, ..., n, such a set of groups must satisfy the following properties[2]:

- $C_1 \cup C_2 \cup ... \cup C_n = \{1, ..., n\}$
- $C_k \cap C_{k'} = 0$, con k $\neq k^{'}$

In K-Means a good grouping of observations is one for which the intra-cluster variation, $W(C_K)$, is minimal, i.e. the following problem must be solved[13]:

$$min_{C_1,...,C_K} \sum_{K=1}^{K} W(C_K) \tag{1}$$

where, through the quadratic Euclidean distance, we define

$$W(C_K) = \frac{1}{|C_k|} \sum_{i,i^{'} \in C_K} \sum_{j=1}^{p} (x_{ij} - x_{i^{'}j})^2 \tag{2}$$

where $|C_K|$ is the number of observations in cluster K.
Thus, the algorithm performs the following steps:

1. A value, from 1 to K, is randomly assigned to each observation;
2. The value of the centroid is calculated. The centroid of the k-th cluster is the vector of dimension p that contains the averages of the variables for the observations in the $k - th$ cluster;
3. Each observation is assigned to the cluster for which the centroid is closest. The proximity is determined by the value of the Euclidean distance.
4. Repeat steps 2. and 3. until convergence is achieved.

2.2 Hierarchical clustering

In this type of method, data are associated with a tree structure in such a way that the leaves of the tree correspond to observations and the nodes to subsets of observations. The very nature of the tree introduces a hierarchy into the subsets associated with the branches.
There are two broad families of hierarchical methods:

1. Agglomerative methods [AGNES]: starting from an initial state of n groups, in which each observation is a separate group, one proceeds by making successive mergers of groups with high similarity between them, until k = 1, i.e. all observations belong to the same group;
2. Divisive Methods [DIANA]: starting from an initial state with $k = 1$ groups, i.e. a single group, one proceeds by successive subdivisions until arriving at n groups[4].

Both procedures operate on a dissimilarity matrix. The main feature of this type of method is that once two groups have been joined they will no longer be separated later, and similarly[11], once two groups are separated they will no longer be part of the same cluster. Furthermore, applying an algorithm of this type, the same tree is used for all values of k, each time referring to a different level of the tree. It is therefore a rigid structure.

2.3 Mini Batch K-Means

In the case of large amounts of data, it is known that the computational complexity of the K-means algorithm increases considerably. Consequently, it was decided to modify K-means clustering for research purposes in order to reduce the computational costs of the algorithm.

Mini-Batch K-means clustering is a variant of K-means clustering in which the size of the data set considered at each interaction is limited. Standard K-means clustering operates on the entire dataset at once, while mini-batch K-means clustering operates on the entire dataset[17].

The mini-batches are randomly sampled from the entire dataset and for each new iteration a new random sample is selected and used to update the position of the centroids.

In K-Means clustering by minbatch, clusters are updated by a combination of the mini-batch values and the learning rate. The learning rate decreases during iterations and is the inverse of the number of elements inserted in a specific cluster.

The effect of reducing the learning rate is that the impact of new elements decreases and convergence is achieved when, after several iterations, there are no changes in the clusters. The results of studies on the effectiveness of mini-batch clustering suggest that it can reduce computation time with a slight trade-off with respect to the quality of the final clusters.

3 Kcluster recommender system - KCRS

The Kcluster algorithm is able to recommend to the user, whether new or not, products that, based on his past purchases or similarity with other users, can satisfy his need, therefore, the proposed algorithm solves the problem known as cold start, which occurs when a new user registers to the system and has not yet provided any interaction, so it would not be possible to offer him personalised

recommendations with so little data[15].
The defined KCRS model is mainly based on similarities between products. In particular through simulated data, as explained in [ref section] according to precise patterns of computer configurations, two products can be recommended:

1. the first one will be extracted on the basis of the similarities that a user (belonging to a specific consumer category) has with other members belonging to the same category;
2. The second recommendation is obtained by a double subdivision of the products, again based on the similarity between them, so it is as similar as possible to the products previously purchased by the customer making the request. This second recommendation is futile when the customer is new and has never purchased products from NI. In this specific case, the products extracted will both come from the first search methodology.

All recommendations also respect certain constraints, such as lower price, stock availability and year of publication of the product. The Kcluster algorithm allows you to choose the type of clustering method you prefer from the three implemented K-means, Hierarchical and Mini Batch K-means. Naturally, a default method is set, namely, K-means[16].

4 NI simulated dataset

Initially, the data available was numerically insufficient for the application of the KCRS algorithm. In particular, there was no identifier to link the different data sets, price list and billing system. As a result, there was insufficient data to validate the recommendations of the products sold. To deal with this problem of lack of data, it was decided to simulate data that followed as closely as possible the real data in the price list and Sales system.
In the domain of computer sales, the relevant data relates to hardware components and therefore it was decided to simulate only three main components, namely:

- Processor
- Ram
- Storage

With the criterion of being able to apply the algorithm to real data, in was decided to simulate again with the following variables:

- User Id
- Product Id
- Selling Price
- Availability
- Date of sale
- Year of production

The statistical process of simulating the dataset is based on the knowledge of the team of developers and on a survey carried out on a sample of 100 individuals using Google forms, through which the probability of the presence of a certain component within a specific computer was defined.
For this reason, the simulation is based on different patterns of configurations, each with a different extraction probability.

5 Evaluation KCRS

Since these are unsupervised algorithms, the evaluation component is not as immediate as for other types of supervised clustering, for which the accuracy of an algorithm is given by the difference of the prediction with respect to the observed value. The choice of the k-optimum (K^*) is a fundamental procedure to obtain the best product partitioning in both methodologies followed; in particular, the objective is to maximise the distance between clusters (BSS) and minimise the distance between clusters (WSS). To do so, it was decided to implement using two evaluation metrics through which the optimal K^* can be chosen[3].

5.1 Silhouette

Each cluster is represented by a silhouette, which indicates the belonging of a specific point to the reference cluster.
In particular, the closer the value is to 1, the more the point has been "well" classified, and vice versa the further away the point is from unity.
The process to apply this metric consists of the following steps:

1. consider each observation i in the dataset, assuming A as the cluster of i;
2. calculate the average distance a_i of i with all other observations in A

$$a_i = \frac{1}{N_A} \sum_{j \in A, j \neq i} d(i,j) \tag{3}$$

3. consider each cluster C different from A and define the average distance $d(i,C)$ of i with the observations in C

$$d(i,C) = \frac{1}{N_C} \sum_{c \in C} d(i,c) \tag{4}$$

4. calculate $d(i,C) \ forall \ C \neq A$, and select the one with $b_i = min \ d(i,C), C \neq A$

The silhouette of i is defined as:

$$Sil_i = \frac{b_i - a_i}{max(a_i, b_i)} \tag{5}$$

Whereas the one used in the current project is the so-called Silhouette Media:

$$sil_{av} = \sum_i \frac{sil_i}{N} \tag{6}$$

The number of clusters k can be determined by choosing the value of k that leads to the highest average silhouette value.

5.2 Index of Calinski-Harabasz

For any number of clusters $k \geq 2$, the Calisnki Harabasz index is defined as:

$$ch(k) = \frac{tr(B_k)/(k-1)}{tr(W_k)/(n-k)} \tag{7}$$

Where n and k are the total number of observations and the number of clusters, respectively. $tr(B_k)$ is the trace of the matrix of clusters between groups, while $tr(W_k)$ is the trace of the matrix of clusters within groups. The optimal number of clusters is obtained at k for which there are large dissimilarities between clusters and large similarities within clusters: the solution is then the value of k that maximises $ch(k)$.

Both metrics give similar results in terms of k^* so we decided to use only one evaluation measure, in particular, we apply the average Silhouette as the default metric since it is the one with the highest level of explainability[14].

6 Experimentation and results

In this section we report the results obtained with the KCRS algorithm. In the first results we simulate a given user who decides to do a search through his user id taking into account purchases made at the same retailer. The algorithm outputs a summary of past purchases "Previous Purchases" and then lists the two recommendations "Recommendations" (Table 1):

- The first recommendation refers to the product searched for within the same user category;
- The second recommendation refers to the product being similar to those previously bought by the user.

The two recommended products (Table 2) in this case are very similar to those previously purchased by the user, which means that the KCRS algorithm recommends correctly as it captures the similarity between purchased and recommended products.
In the second table the user enters both their id x, belonging to user category y, and a user category z within which the search is carried out. As can be seen from the table, this is similar to the first case, the difference being that the recommendation no longer refers to past purchases, but to the user category entered.

In the last example (Table 3), where only user category y is entered, the recommendation is single. This is because, without the user id, the only recommendation proposed refers to similarity between users, i.e. products sold within the same user category. Consequently, the recommendation does not take into account products that have not been sold in the past, for the simple reason that a product is only assigned to a user category once it has been purchased.

These three short tables containing the results obviously cannot be defined as systematic verifications of the algorithm, but are intended to briefly summarise the functionalities and recommendations of the KCRS algorithm, which, as can be seen from the results just shown works and meets the initial needs of the customer.

Previous Purchases							
items	Id	Processor	Ram	Memory	Price	Quant.	year
item 1	5447	i7	32	3072	$1.249,57	1	2019
item 2	9861	R5	8	1024	$938,66	0	2019

Recommendations							
items	Id	Processor	Ram	Memory	Price	Quant.	year
item 1	9794	i5	4	1024	$704,27	1	2018
item 2	98	i5	4	256	$948,23	1	2019

Table 1. Recommendation by user Id

Previous Purchases							
items	Id	Processor	Ram	Memory	Price	Quant.	year
item 1	3033	i7	16	3072	$1.424,44	1	2019
item 2	4821	R9	16	2048	$2.372,5	1	2019

Recommendations							
items	Id	Processor	Ram	Memory	Price	Quant.	year
item 1	3528	R3	8	256	$456,24	1	2018
item 2	94	i7	16	2048	$1.407,21	1	2019

Table 2. Recommendation by user Id and user category.

7 Conclusions

The work described in this paper provides an interesting case of using clustering algorithms within RS, in particular through an RS combining both Content Based and CF approaches.

Starting from a simulated dataset for the domain of e-commerce sites, in particular in the electronics and IT category, the KCRS algorithm allows, within

Previous Purchases							
items	Id	Processor	Ram	Memory	Price	Quant.	year
-	-	-	-	-	-	-	-

Recommendations							
items	Id	Processor	Ram	Memory	Price	Quant.	year
item 1	5222	i9	64	2048	$1.837,73	1	2019

Table 3. Recommendation by user category.

catalogues with considerable quantities of products, to make the most suitable recommendations for the customer's needs. The decision to use simulated data stems from the fact that NI's billing system is not yet linkable with the price lists of computer suppliers.

The simulated data has the sole purpose of testing and validating the developed algorithm, and then in the future applying KCRS to real data. In particular using K-Means, Hierarchical and Mini-Batch together with evaluation metrics for cluster separation, Silhouette and Calinski-Harabasz Index, it was concluded that the clusters found within the datasets are adequately separated to make personalised recommendations.

Some improvements and future developments consist in analysing further characteristics of the users e.g. the metadata associated with their profile. A user is described not only by job category, but can be categorised differently within the platform and, consequently, recommendations could achieve better accuracy. In addition, a test phase is planned to apply the Precision@K, Recall@K, F-measure@K metrics, which will allow a better assessment of the model's performance against recommendations.

References

1. Sebastian Thrun, Wolfram Burgard Dieter Fox: Probabilistic Robotics (2005)
2. J. B. MacQueen (1967): ”Some Methods for classification and Analysis of Multivariate Observations”, Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability, Berkeley, University of California Press
3. Darius Pfitzner, Richard Leibbrandt, David Martin Ward Powers June: Characterization and evaluation of similarity measures for pairs of clusterings, 2009 Knowledge and Information Systems
4. K. Leonard, R. Peter J: Finding groups in data: an introduction to cluster analysis. New York etc : Wiley, 1990
5. Fernando Ortega e Angel Gonzalez-Prieto: Recommender Systems and Collaborative Filtering, (2020).
6. Ricci, Francesco and Rokach, Lior and Shapira, Bracha: Introduction to recommender systems handbook, ricci2011introduction, Springer 2011.
7. Thomas M. Mitchell. 1997. Machine Learning (1st. ed.). McGraw-Hill, Inc., USA
8. Michael Scholz, Verena Dorner, Guido Schryen, Alexander Benlian, A configuration-based recommender system for supporting e-commerce decisions, European Journal of Operational Research, Volume 259, Issue 1, 2017

9. Melville, P., & Sindhwani, V. (2010). Recommender systems. Encyclopedia of machine learning, 1, 829-838.
10. Zhao, Q., Harper, F. M., Adomavicius, G., & Konstan, J. A. (2018, April). Explicit or implicit feedback? Engagement or satisfaction? A field experiment on machine-learning-based recommender systems. In Proceedings of the 33rd Annual ACM Symposium on Applied Computing (pp. 1331-1340).
11. E. B. Fowlkes, C. L. Mallows A Method for Comparing Two Hierarchical Clusterings, Journal of the American Statistical Association 78.383 pp.553-69, 1983
12. Davies, D.L., Bouldin, D.W.: A cluster separation measure. IEEE Trans. Pattern Anal. Mach. Intell. 2, 224–227 (1979)
13. Likas, Aristidis, Nikos Vlassis, and Jakob J. Verbeek. "The global k-means clustering algorithm." Pattern recognition 36.2 (2003): 451-461.
14. S. Łukasik, P. A. Kowalski, M. Charytanowicz and P. Kulczycki, "Clustering using flower pollination algorithm and Calinski-Harabasz index," 2016 IEEE Congress on Evolutionary Computation (CEC), 2016, pp. 2724-2728, doi: 10.1109/CEC.2016.7744132.
15. U. Kuzelewska and A. Kuryłowicz, "Multi-Clustering Applied to Collaborative Recommender Systems," 2018 Thirteenth International Conference on Digital Information Management (ICDIM), 2018, pp. 118-123, doi: 10.1109/ICDIM.2018.8847141.
16. R. M. Esteves, T. Hacker and C. Rong, "Competitive K-Means, a New Accurate and Distributed K-Means Algorithm for Large Datasets," 2013 IEEE 5th International Conference on Cloud Computing Technology and Science, 2013, pp. 17-24.
17. Z. Wang, Y. Zhou and G. Li, "Anomaly Detection by Using Streaming K-Means and Batch K-Means," 2020 5th IEEE International Conference on Big Data Analytics (ICBDA), 2020, pp. 11-17.
18. Liang Wang and Huan Ruan, "An improved recommender system in publications e-commerce based on TOPSIS Algorithm," 2011 International Conference on Computer Science and Service System (CSSS), 2011, pp.
19. Schröder, G., Thiele, M., & Lehner, W. (2011, October). Setting goals and choosing metrics for recommender system evaluations. In UCERSTI2 workshop at the 5th ACM conference on recommender systems, Chicago, USA (Vol. 23, p. 53).