INTERNSHIP RESEARCH REPORT

Cognition-based clustering and its use in relevant descriptions

September 1, 2021

Hanady Gebran (X2018)

TELECOM Paris

IP PARIS

ÉCOLE POLYTECHNIQUE

IP PARIS

PLAGIARISM INTEGRITY STATEMENT

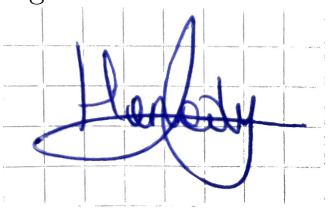
Je soussigné Hanady Gebran certifie sur l'honneur :

- Que les résultats décrits dans ce rapport sont l'aboutissement de mon travail.
- Que je suis l'auteur de ce rapport.
- Que je n'ai pas utilisé des sources ou résultats tiers sans clairement les citer et les référencer selon les règles bibliographiques préconisées.

Je déclare que ce travail ne peut être suspecté de plagiat.

Date: 18/08/2021

Signature:



ABSTRACT

0.1 English version

In artificial intelligence, cognitive reasoning can be overshadowed by mathematical considerations that are generally probabilistic. Moreover, these calculations are greedy in data and the results obtained, although satisfactory, can be very different from the average human response. The clustering process is a major challenge and although it has been extensively studied, the proposed models often require conditions concerning the distribution or the amount of data, as well as external parameters that are often quite cryptic. This study aims to consider clustering as a cognitive process and to exploit human reasoning to achieve it, the ultimate goal being to reach optimal classification with few examples and no parameters. To this end, we rely on heuristic approaches beyond K-means, based on Kolmogorov complexity (which measures the intuitiveness of a piece of information) and contrast (which helps to compare an object to a given prototype), with the objective of mimicking human decision-making considerations. Finally, we explored a cognitively advanced task which is the generation of relevant descriptions, using our clustering approaches coupled with reflections on the problem itself as a novel contribution to solve this task.

0.2 Version française

En intelligence artificielle, le raisonnement cognitif peut être supplanté par des considérations mathématiques généralement probabilistes. De plus, ces calculs sont gourmands en données et les résultats obtenus, bien que satisfaisants, peuvent être très différents de la réponse humaine moyenne. Le processus de clustering est un défi majeur et bien qu'il ait été largement étudié, les modèles proposés nécessitent souvent des conditions concernant la distribution ou la quantité de données, ainsi que des paramètres externes souvent assez obscurs. Cette étude vise à considérer le "clustering" comme un processus cognitif et à exploiter le raisonnement humain pour y parvenir, le but ultime étant d'atteindre une classification optimale avec peu d'exemples et sans paramètres. Pour ce faire, nous nous appuyons sur des approches heuristiques au-delà de la méthode "K-means", basées sur la complexité de Kolmogorov (qui mesure l'intuitivité d'une information) et le contraste (qui permet de comparer un objet à un prototype donné), dans le but de reproduire les critères décisionnels humains. Enfin, nous avons exploré une tâche cognitivement avancée qui est la génération de descriptions pertinentes, en utilisant nos approches de "clustering" couplées à des réflexions sur le problème lui-même afin de contribuer de façon novatrice à la résolution de cette tâche.

ACKNOWLEDGEMENTS

Throughout my internship, I was given a great deal of assistance and support, therefore I would like to extend my gratitude to all the people who made this whole experience possible.

I am deeply indebted to my internship mentor, Mr. Jean-Louis Dessalles, for his steady and relentless guidance during the internship, for his enlightenment on areas I was totally unfamiliar with prior to this research internship, along with his many suggestions, insights and constructive criticism.

I would like to extend my thanks to the administration of Télécom Paris for allowing me to complete this internship. I would especially like to thank the DIG team, as well as the other interns of Mr. Dessalles (Kanvaly Fadiga, Etienne Houzé, Emmanuel Lagrée, Etienne Li, Julien Lie-Panis) who helped to broaden the scope of my understanding by presenting their own research projects and by raising their concerns about mine.

CONTENTS

1
ABOUT THE INTERNSHIP

1.1 Télécom Paris

I realized my internship in the INFRES department (Computer Science and Networks) of Télécom Paris. It is a research and academic department whose research activities cover computer science, communication and information. As part of this department, the DIG (Data, Intelligence and Graphs) team is active in research on databases, graph algorithms, learning techniques, cognitive models... The subject of this internship touches upon several areas of research investigated in this department.

1.2 Covid-19

The health crisis and the lockdown had an impact on the course of this research preparation but did not impede its smooth running. Initially, the weekly meetings were held by videoconference. Afterwards, the meetings were held at Télécom Paris in person. In addition, every Thursday, a member of the team presented his or her research topic and this presentation, at first remote, then became face-to-face. I was lucky to be able to present my subject during one of these presentations as well as in one of my tutor's trainees' private seminars.

2
INTRODUCTION AND CONTEXT

2.1 Motivation & Contribution

Artificial intelligence stems from the human desire to reproduce our thinking in order to delegate certain tasks, especially when those are repetitive and require too many human resources. Strangely enough, clustering, which is an easily describable problem: dividing data points into a number of groups, so that data points in the same groups are similar to each other and different from data points in other groups, has not really induced heuristics based on cognitive reasoning. Indeed, apart from K-means which is based on a heuristic based on a human thinking of putting together what is close, most approaches have become quite far from the human classifying methods and often require a lot of data.

Therefore, our primary goal is to produce a clustering that is as close as possible to human cognitive reasoning. Our ideas are very heuristic and can be compared to the K-means method.

One consequence of this incentive is that we aim at developing a methodology that does not require large datasets: we strive to learn from few examples given that this is a core feature of human reasoning. A further purpose is to avoid the need for external parameters. Finally, in order to render our input in a pertinent way, we try to use our clustering directly to describe films while keeping the approach very similar to human behavior.

2.2 Classical clustering methods and their limitations

In order to situate our approach in the landscape of current methods, we describe 3 families of approaches. Our proposed design is part of the first family (centroid-based).

2.2.1 • Centroid based clustering [1]

Description: these consist of an **iterative** clustering algorithm in which the concept of similarity is derived by the proximity of a data point to the centroid of the clusters. The K-means clustering algorithm is a popular algorithm that falls into this category. These models work iteratively to find local optima. In the case of K-means we need to minimize the function $\arg\min_{\mathbf{S}} \sum_{i=1}^{k} \sum_{\mathbf{x}_j \in S_i} \|\mathbf{x}_j - \boldsymbol{\mu}_i\|^2$ with x_j being the data points, μ_i the centroids and S_i the clusters

Limitation: a major concern arises from the use of **holistic distances**. This is a critical flaw as a small noise on many dimensions can be considered as important as a huge deviation on a single dimension which is cognitively absurd. Furthermore, the different coordinates must have the same order of magnitude so that a 5% difference on one coordinate is not considered more important than a 50% difference on another. In addition, a point is associated with a cluster without any real explanation of **its link with this cluster**. Moreover, the K-means method requires the user to specify the optimal number of clusters and we ideally wish to avoid giving any parameter.

2.2.2 • Model-based clustering [2]

Description: these consider the data as coming from a distribution that is a mixture of two or more clusters, model-based clustering uses a soft assignment, where each data point has a probability of belonging to each cluster.

Limitation: that clustering does not work well if the given dataset contains few observed data points: big datasets are needed whereas cognitively a few examples are enough to create a cluster. In addition, we lose the link between the point and its cluster because one point is tied to several independent clusters, "hard" clustering is therefore preferred. They also differ too much from human cognitive processes.

2.2.3 • Density-Based clustering [3]

Description: a cluster is seen as a region of high density in the dataspace separated by an area with little or no data. The idea behind this method is to detect clusters through density drops.

Limitation: these methods use parameters that are difficult to identify in order to deem what is a density drop, and like model-based clustering, these methods also differ too much from human cognitive processes. Moreover, there are occasionally significant differences between objects of the same type; with a density method these objects would most certainly be in different clusters.

2.3 Relevant description

The way an artificial system presents information differs from the paradigm that humans use for verbal interaction. In task-oriented communication, it is common for speakers to produce distinctive referring expressions. The main purpose of these expressions is to help identify uniquely a specific object while maintaining the listener's interest. This is one of the most explored tasks in natural language generation [4], and it seems to be an interesting use of clustering. Given a set of clusters categorizing cities for example, that categorization is likely to be an important indication of how we will describe a new city to the hearer.

3
FUNDAMENTAL CONCEPTS

3.1 Motivation

We introduce two concepts: **Kolmogorov's complexity** [5] [6] and **contrast** [7] [8] [9], because we truly believe that these are key concepts in the process of human reasoning. Kolmogorov's complexity is useful to create an informational criterion to translate a deeply human impression of what is "complicated", "intuitive", "weird". Contrast gives us a mathematical criterion of what seems "atypical" in an entity and what we will "retain".

3.2 Kolmogorov's complexity

Complexity can be measured as the length of a compressed representation of something. The downside is that the minimum possible compression of a given thing may never be known.

The Kolmogorov complexity $K_M(x)$, or algorithmic complexity, of a finite sequence x of characters for a machine M is defined by the length of the smallest program written for the machine M that generates the sequence x. This complexity can be approximated by finding a short binary code p(s) that represents s and then measuring the length of this binary representation of s. $K_M(x) = \min_{p \in P_M}\{l(p), s(p) = x\}$

This notion is interesting because it allows to have an informative criterion of selection. A simple example is an object in a set of N entities, any object in this list has a complexity close to $log_2(N)$ but if the object to be described is the biggest one we can consider that its complexity is 1, because a minimal binary description is 1 as it has the rank 1. This object is less complex by this definition like our understanding.

This notion is valuable for a cognitive approach because humans seek **intuitiveness** which is found in **low complexity characteristics**. In addition, this complexity **depends on the context**; which is similar to the human behavior of finding something more intuitive after learning other relevant information. Computationally, $K(x, y) = K(x) + K(y|x) + O(\log(K(x, y)))$

3.3 Contrast

Contrast is a congnitive human operation. Intuitively, it consists in comparing an object with a prototype (a mental representation of a group as a basic object). If this object differs a little from the prototype, we can consider that this object belongs to the group (cluster) represented by this prototype. If this object differs a lot from the prototype on a very reduced number of dimensions, we can consider that this object belongs to the group represented by this prototype but that it is "particular" and "notable". And if none of these 2 cases occurs, the object does not belong within, the cluster represented by this prototype.

Mathematically we create a contrast vector C, this vector is a difference vector between the object O and a given prototype P. **If the difference is below a certain threshold θ we cancel the difference**. $C_j = (O_j - P_j) \times 1_{|O_j - P_j| > \theta_j}$

In the standard cases, where the data has a quasi-Gaussian representation, this threshold will be the standard deviation on the considered dimension and we will divide this discrepancy by the standard deviation in a way to compute the difference in number of Standard Deviation of difference. $C = \left\lfloor \frac{O-P}{\sigma} \right\rfloor$

In some cases, we will have a precision which depends on the desired precision integrated in the definition of the cluster.

Whatever the case, this notion allows us to **create a direct link** between the object and its cluster and opens up perspectives of interpretation and description.

4
CLUSTERING

Our starting point for clustering was K-means. We then thought about ways to improve the clustering with new heuristics strongly inspired by our way of thinking.

4.1 Clustering on subspaces

4.1.1 • Motivation

This heuristic approach is based on the fact that, as human beings, we can only compare values in the same units. By creating clusters on spaces of comparable dimensions, we reduce the final clustering space by already associating each entity to a lower dimensional cluster, which itself corresponds to a label that a human could have put. For example, if the vector subspace is composed of the average speed and the maximum speed, a cluster with a low average speed and a high maximum speed corresponds to a show-off or someone who does not manage his cardio well.

4.1.2 • Results

This technique was interesting because it allowed us to have simpler interpretations. However, this idea was not adopted in the end because it sacrificed too many points if we did not want to blow up the number of clusters.

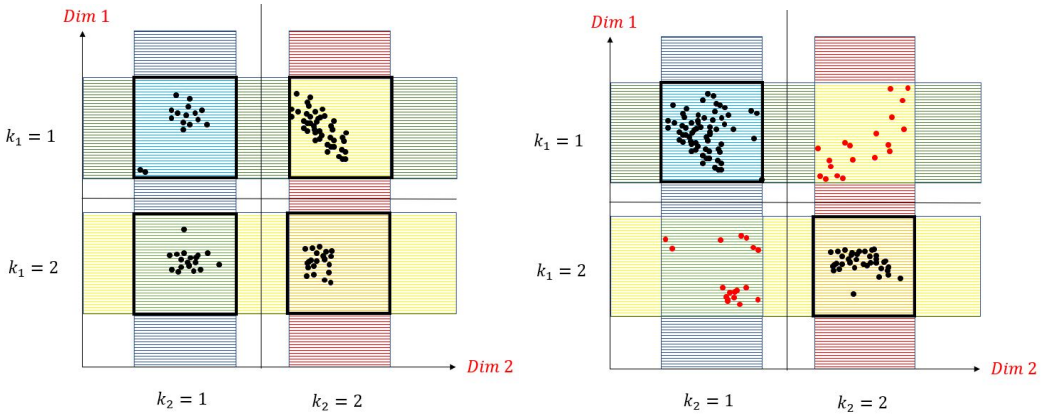


Figure 1: if the number of clusters can't grow exponentially, red points are removed. With no constraint, the number of clusters blows up (4 clusters instead of 2 here).

4.2 Hierarchical amnesiac clustering

4.2.1 • Motivation

Let's consider a baby who sees an animal for the first time, the baby will create a (null depth) cluster of average this animal. After a few animals this infant starts to have a notion of what an animal is with typical deviations on characteristic dimensions. After about a dozen animals, this infant will see an ant and will compare it to the prototype animal of the null depth cluster and he will perceive a significant difference on the "scale" feature of the animal. Thus he will create a sub-cluster of the null-depth cluster which will contain the "small" animals. After hundreds of observed animals, this sub cluster will have itself sub clusters which will allow to refine the understanding of what an animal is.

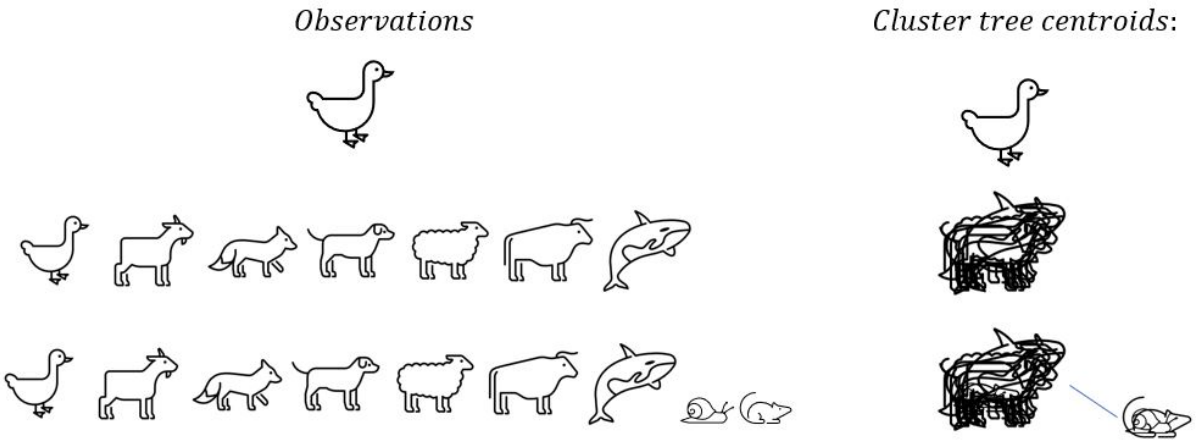


Figure 2: dynamical creation of a cluster tree after animal observations.

Intuitively at the adult age our brain does not see much "animal" objects (we see "colibri", "labrador", "golden retriever"..), our notion of animal does not change anymore it is sub categories of sub categories which change. Having a cluster tree in this fashion, when we add a new object it will have a very local influence and increasingly less significance with a large number of points in the sub cluster. We can already predict that contrast will play an important role: when we observe a cat for the first time we will retain it as it is and it will influence our perception of "animals" a little, if we observe another cat it will influence our perception of "cats" a little, if we observe a Siamese cat it will influence our perception of "cats" on all 0-contrasted dimensions but it will be noticeable and we will create a new subcluster!

The following sub-sections show how we implemented this replicated human logic.

4.2.2 • Framework for clustering

Locally each cluster (besides the **null-depth cluster**) has in memory its pseudo centroid, the standard deviation, its depth, the number of points it contains locally as well as the reference to

the sub-clusters of **singular points**. The **singular points** are the points such that at least one of the dimensions has a value farther from the pseudo centroid than the standard deviation. The **null-depth cluster** has for standard deviation the standard deviation of a number of points given in input and its centroid is the origin, it is technically the father of all our clusters. The **null-depth cluster** is necessary because we cannot make up a unit for the precision of the measurement of the data points, so we consider in a somewhat human approach that we already have a small number of data (5-15% of the total number) whose standard deviation is known and we take it as our unit. The purpose of the cluster of depth 0 is also to have all the functions work recursively from it. We then store the values of the data points at depth 1 with a precision equal to the deviation value retained at depth=0. And for all clusters of depth ≥ 1, the pseudo centroid is obtained by a calculation with the **contrasted vector** with the pseudo centroid of the parent cluster. This heuristic is therefore very strongly inspired by the example of animals in the previous section.

4.2.3 • Qualitative description of the point addition within the online structure

When adding a point we primarily reflect **locally**, the question is to know where to put the point with regard to this current cluster. We have 3 options. First we have to determine if the point will stop at this level (a **local addition**) or if it should be added to a sub-cluster of the current cluster (an addition in the descendance). To do this, we calculate the cost of storing the point at the current level C1 as well as the cost of storing the point if it was placed in the descendancy of any one of the sub-clusters of the current cluster C2 (C2 is obtained by recurrence).
If C1<C2 we have two possible cases. The **first case** is the simplest: the **contrasted vector** between the representation of the point at this level and the pseudo centroid is the null vector. In this case, the point will just change the pseudo centroid and the standard deviation of the current cluster as well as the number of points. In the **second case**, we say that the point is a **singular point**, in addition to modifying the pseudo centroid and the standard deviation of the current cluster as well as the number of points, the point will be added to the current cluster as a sub-cluster, of pseudo centroid initialized to the value of the **contrasted vector**.
If C2>C1, it is the **third case**, we have to apply local recursion to all the sub-clusters of the current cluster until we obtain a **local addition**. This result is surely reached because the deepest descendants don't have any sub-clusters. By applying this reasoning starting from the **null-depth cluster**, we get the exact position of the point in the hierarchical cluster tree.

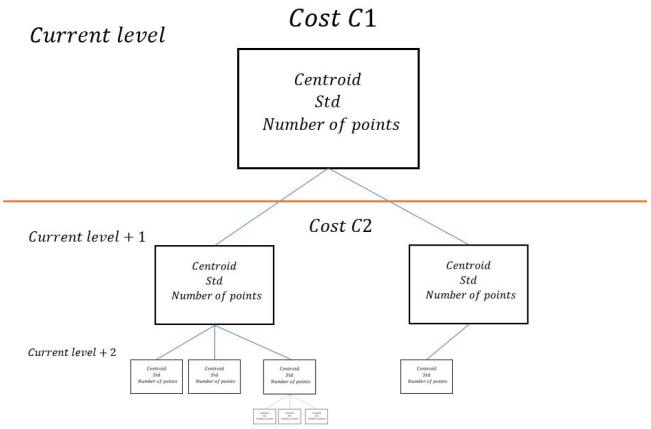


Figure 3: structure of the cluster tree at a level before a local addition.

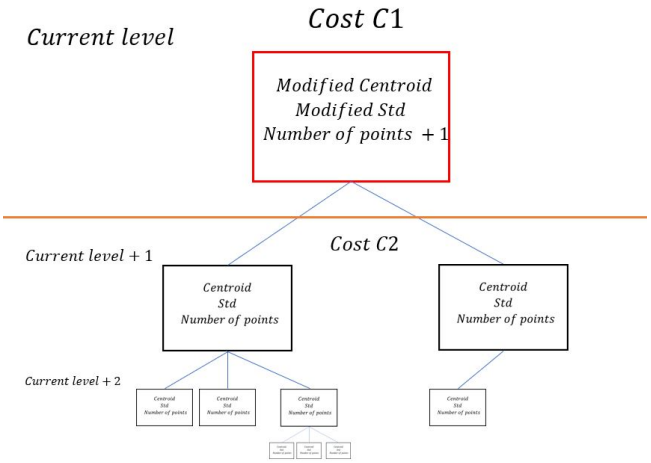


Figure 4: structure of the cluster tree at a level after a local addition, first case.

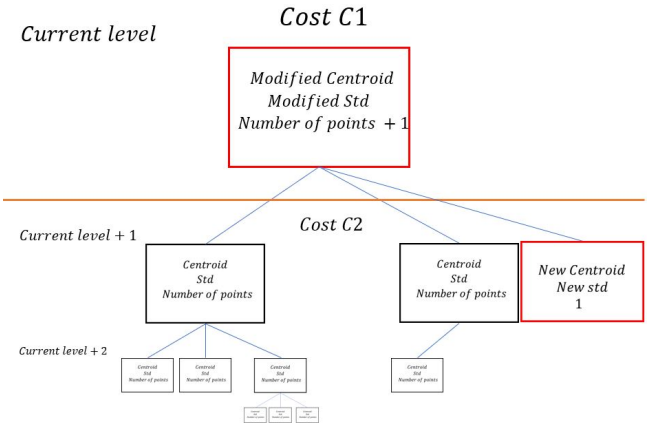


Figure 5: structure of the cluster tree at a level after a local addition, second case.

4.2.4 • Computation of components within the online environment

Null-depth cluster:

Centroid: null vector. We want the exact representation at depth=1 with a precision = std.
Std: standard deviation obtained from a sample of size one order of magnitude smaller than the total number of points. (5 to 15% of all datapoints).
Number of points: not important.
Addresses of the sub-clusters: all clusters of depth=1 are its descendants, it is the parent cluster of our whole hierarchical cluster tree.

Local addition:

Point representation: The space in which we work remains unchanged, but depending on the path followed by recursion in the hierarchical cluster tree, the representation of the point changes.

For the **calculation of complexity of a point**, the calculation is done by considering the representation of the point in the form: [[number of standard deviations, feature index], [number of standard deviations, feature index], ...], but with a cost of log_2(Number of dimensions+1) for all indices in order not to create a priority order between dimensions. With this approach, a point [0,0,0,5,0,0,-2] will be represented as [[5,4],[-2,7]] and will have a complexity cost = $log_2(7+1)+log_2(5+1)+1+log_2(7+1)+log_2(2+1)+1$ which is the cost of saying that in the 4th dimension there are 5 standard deviations of difference in the positive direction and in the 7th dimension there are 2 standard deviations of difference in the negative direction.

Std: We can calculate the new standard deviation using the old standard deviation, the pseudo centroid, and the point representation. The exact formula is obtained as follows:

$$\sigma_{n+1}^2 = \frac{1}{n+1}\sum_{i=1}^{n+1}(x_i - \overline{x_{n+1}})^2$$

$$\sigma_{n+1}^2 = \frac{1}{n+1}\sum_{i=1}^{n+1}(x_i - \overline{x_n} + \overline{x_n} - \overline{x_{n+1}})^2$$

$$\sigma_{n+1}^2 = \frac{1}{n+1}\sum_{i=1}^{n+1}((x_i - \overline{x_n})^2 + 2 \times (x_i - \overline{x_n})(\overline{x_n} - \overline{x_{n+1}}) + (\overline{x_n} - \overline{x_{n+1}})^2)$$

$$\sigma_{n+1}^2 = \frac{1}{n+1}(n\sigma_n^2 + (x_{n+1} - \overline{x_n})^2) + 2 \times (\overline{x_n} - \overline{x_{n+1}})^2 + (\overline{x_n} - \overline{x_{n+1}})^2)$$

$$\sigma_{n+1}^2 = \frac{n}{n+1}\sigma_n^2 + \frac{1}{n+1}(x_{n+1} - \overline{x_n})^2 + \frac{3}{(n+1)^2}(x_{n+1} - \overline{x_n})^2$$

Having σ_{n+1} the std after the local addition, x_{n+1} is the point representation and $\overline{x_n}$ the pseudo centroid before the local addition and n the number of points in the cluster locally.

Contrasted vector:

$$C = \left\lfloor \frac{\overline{x_n} - x_{n+1}}{\sigma_{n+1}} \right\rfloor$$

In the case where one of the coordinates of σ_{n+1} is zero, i for example, if we are in the case where $(\overline{x_n})_i = (x_{n+1})_i$ then $C_i = 0$, otherwise we have a maximum value which is placed at C_i in order not to have infinite values.

If we are in the **second case**, C is taken as the initialisation of the pseudo centroid of the new subcluster.

In the **third case** (not included in the current local addition), C is the new representation of the point. We explore the subclusters of the current cluster with this new representation.
Centroid:

$$(\overline{x_{n+1}})_i = 1_{|\overline{x_n}_i - x_{n+1\,i}| < \sigma_{n+1\,i}} \times \frac{n(\overline{x_n})_i + (x_{n+1})_i}{n+1} + 1_{|\overline{x_n}_i - x_{n+1\,i}| \geq \sigma_{n+1\,i}} \times (\overline{x_n})_i$$

Indeed, we modify the value of one of the coordinates of the pseudo centroid only if the value of the point on this coordinate is not aberrant. This is rather consistent because the singular points influence the pseudo centroid only on the coordinates where they are reasonable, they already directly influence the standard deviation on all coordinates. This approach allows to have a reliable method against anomalies.

Number of points: The number of points is increased by 1

Addresses of the sub-clusters: If we are in the **first case**, we do not change anything. In the **second case**, we create a cluster that we initialize with the pseudo centroid of value the contrasted vector C. We then add its adress.

4.2.5 • Results

To interpret the hierarchical cluster tree, we should read it in the following way, which depends on the **reading depth**. If we read at level 0, all points are in the same cluster. If we read at level k, all child clusters of depth>k are considered to be included in their parent cluster of level k and on the other hand all clusters of level k are independent. However, we had to reconsider slightly this way of reading the tree and consider that all the points without descendants at a given level are grouped with the forgotten points of the father cluster, in order to avoid having many singleton clusters. The greater the reading depth, the more clusters we have.

We also created a new dataset with the contrasting vectors. Without repetition, on the sport dataset for example, after clustering with our hierarchical cluster tree method, we notice that two clusters of contrasting vectors stand out because they contain 98% of the points. And after inspection, we understand that one corresponds to the case where there is no contrast on

the "source2" dimension and the other to the case where there is a contrast on the "source2" dimension.

We conclude that we have been able to **capture a meaning** through contrast, which was one of our challenges. However, we regret that the results are not as good theoretically as K-means on the original dataset, although the interpretation remains open. We have an artificial classification thanks to the type of sport on this particular dataset but a particularly sporty racing instance may be closer to a cycling instance than a racing instance which wrongly assesses the success of the approach.

One of the big limitations of this approach is the strong assumption that the **standard deviation is a relevant unit of measurement of the data**. However, some datasets have distributions that are not at all Gaussian, which makes this clusterization heuristic problematic. Moreover, even if the standard deviations do not vary greatly when the clustering becomes adult, it is risky to have a non-stationary unit.

Finally, we sticked to an optimization problem that is more faithful to cognitive reasoning than the K-means objective function.

4.3 Decimal precision clustering

4.3.1 • Motivation

This new approach is centered on the concept of complexity and is designed to be independent of the distribution of the data. Indeed, instead of a global optimization function with a holistic distance as for K-means. We stay on a greedy approach which is based on a minimization of the complexity of an object added to our system. We deviate a little bit from the human behavior described in the previous heuristic as we do not consider standard deviations but rather take **dimensional accuracies** which makes this approach a more reliable one.

4.3.2 • Clustering framework

Each cluster stores its centroid in memory locally. In this heuristic the centroid is a generalization of the points in the cluster. In dimension 1 let us consider the point 2.8156, we create a cluster of centroid 2.8156 from this point. At each addition of points, we calculate the **least generalization** (lgg) between the centroid and the added point. For example, if we add 2.81 the lgg is 2.81, if we then add 2.79 the lgg is 2.8. The link with the complexity calculation is therefore immediate. If we represent the numbers in base 10, we count the number of digits necessary to complete or modify the development of the prototype so that the object is determined without ambiguity. For 2.81 we have 2 times the cost of an addition while with 2.79 we have the cost of a deletion and 2 additions. We therefore automatically have a penalty for

centers that are too imprecise because we will have to add many bits and a penalty for centers that are too precise because we will have to remove many bits.

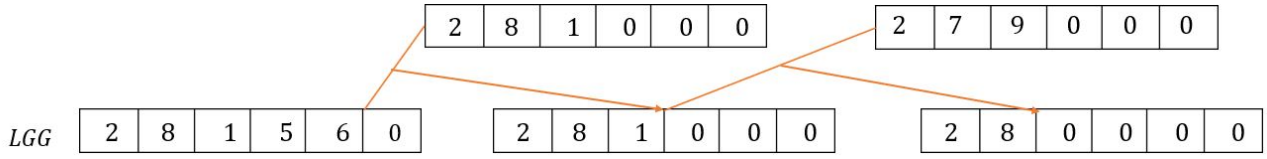


Figure 6: least generalization example.

4.3.3 • Observations

This heuristic is a bit too simplistic: when we change the centroid after adding a point, we do not take into account that the representation of the other points will change, which changes the total complexity if the clustering was not amnesiac. This amnesia was also partially present in the previous heuristic, but the " forgetfulness " is more consequential in this approach. Moreover, this algorithm is interesting in an online clustering dynamic where we get the data progressively and there is no way to do multiple passes on the data with the interesting function being the current complexity. If we are interested in the overall complexity, this heuristic is too simplistic and rather unfavorable. Moreover, we have no way of going back and making a centroid more accurate. Remarkably, we get pretty decent clusters with this heuristic. Moreover, in this case no artificial parameters are needed.

4.4 Non-amnesiac binary precision clustering

4.4.1 • Motivation

In this heuristic, we enable computing and repeating calculations, this version is more suitable for a **predominantly offline clustering scenario**, for which we cannot afford to forget points are permitted to do multiple passes on the data.Therefore, it is a method that is a step away from human functioning, albeit also based on complexity and informational criteria. This approach is also driven by the **concept of precision** across dimensions but in this case it is combined with the requirement to be able to **rescale the precision as we go along**. The aim is to approach the behavior of the standard deviation: it decreases when a point is close to the cluster (so we want the accuracy to increase) and it increases when a point is distant from the cluster (so we want the accuracy to drop). Moreover, the **optimization function is the total complexity**, therefore we have to take into account the effect of a point on the complexity of the integrality of the points of the selected cluster.

4.4.2 • Data representation

We make sure that all data coordinates are less than 1 by dividing by the largest value on each dimension. To be able to deal directly with the problem in bits, we convert the decimal

number to its binary representation on max_bits (=8 for example), in this case the number 0.8 is converted to 0.11001100. The reason why we choose to convert numbers to a value below 1 is that for decimal numbers without an integer part, **two close numbers will have a close binary representation**. For example, 0.78 and 0.8 have the same binary representation on 4 bits.

4.4.3 • Clustering framework

Every cluster is constituted of a centroid in binary representation and a precision vector. The centroid is, as for the K-means algorithm, the average of all the points in the cluster. The precision vector is the number of bits seen by the other points of the cluster. At initialization, the precision is equal to **max_bits** on all dimensions. The initial description cost is therefore max_bits*number of dimensions. This precision is higher when the cluster expects the point to be close to the centroid on a certain dimension.

4.4.4 • Qualitative description of the point addition within the structure

We want to add a point to our clustering. If it's the first point, we create directly a cluster of precision max_bits on all the dimensions and of center the binary representation of the point to add.

If there are already existing clusters, we need to calculate the cost of adding the point Pt to every cluster and choose the cluster that minimizes this cost. For each considered cluster C, let new_centroid_binary be the new centroid of C in binary representation if we add Pt to it. For each dimension d, we have several cases following the comparison of len(new_centroid_binary[d] ∩ centroid_binary[d]) with precision[d].

If len(new_centroid_binary[d] ∩ centroid_binary[d]) = precision[d]: This case is the simplest, it confirms that **our precision is appropriate**. In this situation, the complexity of adding Pt[d] to this cluster is easily computable, it is the cost of storing Pt[d] knowing new_centroid_binary[d] and precision[d].

If len(new_centroid_binary[d] ∩ centroid_binary[d]) < precision[d]:

In this situation, **the precision is perhaps too high**, we will reconsider the precision on the dimension d and see if it is more interesting to reduce the precision. For all the precisions between len(new_centroid_binary[d] ∩ centroid_binary[d]) and precision[d] we calculate the cost of adding Pt[d] until having a new_precision[d]=p+1 such that the cost is higher than for new_precision[d]=p. On reaching this case we fix precision[d]=p. Beware, in this case the computation of the cost of adding Pt[d] is more complicated, because changing precision[d] impacts the cost of storing all the points of the cluster: the cost of adding Pt[d] with precision p is therefore equal to the cost of storing all the points (plus Pt) of the cluster in the dimension d with the new precision minus the cost of storing the cluster in dimension d prior to the addition of Pt[d].

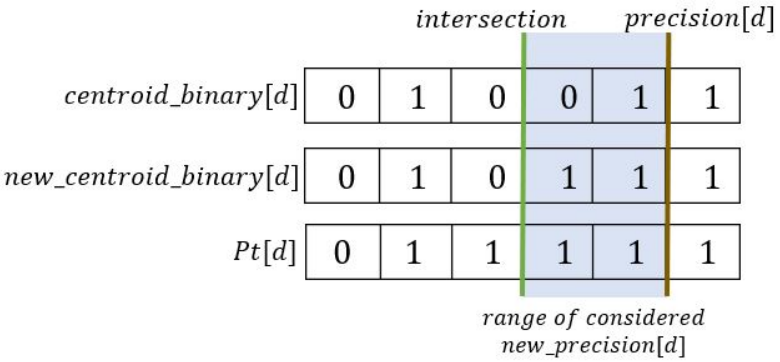


Figure 7: example of the case where len(new_centroid_binary[d] ∩ centroid_binary[d]) < precision[d].

Finally if len(new_centroid_binary[d] ∩ centroid_binary[d]) > precision[d]:

In this situation, **the precision is perhaps too low**, we will reconsider the precision on the dimension d and see if it is more interesting to increase the precision. For all the precisions between precision[d] and len(new_centroid_binary[d] ∩ centroid_binary[d]) we calculate the cost of adding Pt[d] until having a new_precision[d]=p+1 such that the cost is higher than for new_precision[d]=p. On reaching this case we fix precision[d]=p. In this case the computation of the cost of adding Pt[d] is similar to the previous case.

We do not directly compare Pt[d] and centroid_binary[d]: if a bit discrepancy does not affect the new average, a precision that takes this divergence into account is sub-optimal.

4.4.5 • Computation and definitions

In this part, we are interested in a simple calculation which is the storage cost of a binary sequence S1 (a coordinate of a point in binary representation for example) knowing another binary sequence S2 (a coordinate of the centroid in binary representation for example) as well as the precision on S2. This calculation is relatively simple if len(S1 ∩ S2) ≥ precision, the storage cost of S1 is max_bits-precision (in other words the storage cost is the cost to complete S1[:precision] in order to have S2). If len(S1 ∩ S2) < precision the cost is (max_bits-precision)+(precision-len(S1 ∩ S2)*2 which is the cost of subtracting false bits before precision and then completing the bits first to precision and then to max_bits.

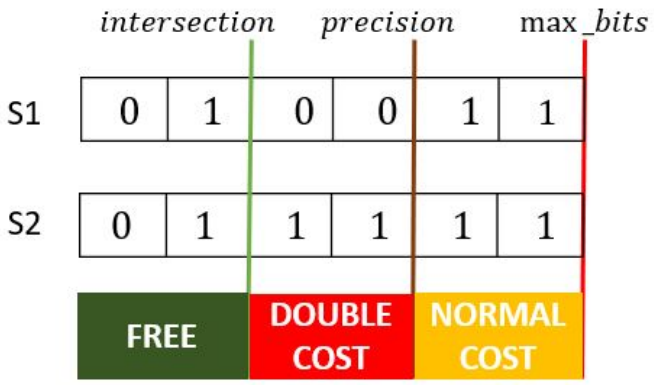


Figure 8: computation cost.

4.4.6 • Observations

This heuristic is the **most expensive**, moreover it needs a rather critical parameter which is max_bits. However, we can see that the idea of precision is quite federating because it allows to achieve a fairly robust complexity-based clustering. Moreover, this approach is more attractive than the decimal precision clustering because the function to be minimized is more global. Nevertheless, we fall again in the problem of **holistic distances** as a big difference on one coordinate can be penalized in the same way as a small one on several coordinates.

5 DESCRIPTION

5.1 Motivation

We investigate generating descriptions because it is an attractive application of clustering. For this purpose, we are limited to a description formed from a database of entities of the same type that have characteristics in the same space. This framework is well suited to clustering points but is only a very specific framework for generating a description. We will start by introducing some considerations and thoughts on this subject before addressing in the following part how clustering can be applied.

5.2 Discriminatory description

In our framework each entity has a number of characteristics. In order to describe the entity we have to put forward some characteristics hoping to give a "good" description.The definition of a “good description” is merely vague. In fact, associated research [10] [11] has shown that a relevant criterion is taking a rather **discriminating** characteristic to easily identify the object after description. For example, mentioning the main religion of a country to describe it may be more useful for the description of Japan which is the only one practicing Shintoism than for the description of Indonesia which is one of many Muslim countries. Sometimes it means choosing a "less interesting" feature, the interest being increased by the distinctiveness of the feature. In our case, i.e. we have a database of objects with features as columns, choosing the most discriminating feature f for an object O in the database D is to take the column C such as C=argmax$_{c \in columns_{\mathcal{D}}}(\#rows - \#rows(c = O[c]))$

5.3 Intuitive description

We want to produce intuitive descriptions, which could be considered human-produced. One way to quantify the **intuitiveness of a description** is to compute its complexity [12], in particular an approximation of its Kolmogorov complexity . The higher the complexity of a description, the less intuitive it seems. However, this criteria can be in conflict with the previous one. Indeed, if we want to describe a country by the spoken languages, choosing English is not discriminatory. However, a language often used in the database will have a **lower complexity** because we will give it a **reduced-length binary mapping**. In our compression, we classify the words by frequency of appearance in the database and the length of the binary representation is log_2(rank), English being frequent, its rank is small and hence its binary representation is short which makes the use of the word English less complex.

5.4 Contextualized description

In real life, the context is often essential when we want to describe something. In our case, two aspects pertain to the notion of context: the **previous discussing context** and the **knowledge level** of the individual.

5.4.1 • Previous discussing context

In this case, our complexity calculation allows us to take this background into account. Intuitively, having a previous discussion allows to make some information less complex. For example, if we have to describe Brad Pitt after having mentioned Angelina Jolie, we would like our approach to choose the information "ex-husband of angelina jolie". For this purpose, our

method considers that an already seen information obtains **a second non-fixed representation**. The first, fixed, representation is determined by our database and the second one is **created from the moment the information has been mentioned**, and it is the binary representation of its rank of appearance in decreasing order. As the conversation progresses, this rank will change and so does this second representation. For example, consider that in a conversation 3 artists were mentioned after Angelina Jolie, the second binary representation of Angelina Jolie will be 100 which is 4 in binary base.

5.4.2 • Knowledge level

We introduce a new but meaningful invented indicator, the **complexity threshold E**. We consider that each person has a level of knowledge of a given subject. For example, a person P1 is passionate about geography and knows all 195 countries, while someone else, P2, knows only 10 countries. This is important because if we mention Tonga as a way to describe a character, it is not relevant for P2 while it is useful for P1. For this, E is log_2(number of entities known by the person for this category). Indeed, 10 countries are known it is probably the 10 countries which have the strongest binary compression (the simplest ones). E is defined as the **maximum accepted length of the binary representation of the entity** used for the description. In our example, when describing a country to P1 there is no limitation because the length of the binary representation of a country is less than log2(195) whereas a country with a binary length greater than 4 cannot be used to describe something to P2. E will therefore depend on the person, and for a given person on the subject.

6
BUILDING THE MODEL

6.1 Motivation

We wish to describe films from a rather complete database (IMDb) containing various numerical (such as the budget) and non-numerical (such as the country of creation) information. In order to make the most out of our heuristics and experiments, we proceed in 2 steps. The first step is offline: it consists in clustering the films present in the database and then associating a relevant description to each of these clusters. The second step is online: it consists in receiving a request for a description of an entity and sending a description. In order to do so, we have to associate the request to a cluster, then use the description of the cluster and finally the relationship of the point with the cluster.

Indeed, when we are asked to describe a country for example, we often think: what kind of country is it? Third-world country with over population? developed with good diplomatic relations? Then we try to characterize this country to make it distinguishable from other points

in its cluster, is it the most touristic country? The only one to have a female president?

The construction of the model will simply be using heuristics and considerations seen previously. Therefore, we will explain how to fit these different elements together to be able to give a relevant description of movies requested by the user.

6.2 Offline clustering and cluster description

The first step is an offline clustering. Indeed, we use all available data in our database and we examine the obtained clusters. All heuristic clustering methods can be used at this step, the one we selected for the final method is the non-amnesiac binary precision clustering.

Then it is time to generate descriptions of the obtained clusters. By viewing some clusters, we can manually describe the types of movies they represent (the number of clusters being much lower than the total number of points, this is not very constraining, especially since this part is done offline). For example, a cluster such as the centroid has a high complexity actors (not well-known), a low budget, a high income and high scores is an unexpected success; we can therefore give this cluster the description "unexpected success".

However, it is also possible to find an automatic description of the clusters themselves, for this we calculate the complexity of each component of the centroid and we classify them in increasing order of complexity. The calculated complexity is an approximation of **Kolmogorov's complexity**: for numbers it is the binary representation of their rank in the appropriate order and for non-numerical entities it is the binary representation of their rank in terms of frequency of occurrence in decreasing order. An artificial parameter **P** can be added to decide how many features are needed to describe a cluster and then we take the **P lowest complexity features** for each cluster to describe it.

We introduce a way to get rid of P based on adjectives. For this purpose, we create a **description scale based on standard deviations**. We have a total of 5 categories: very low (-max δ σ), low (between -max δ σ and -σ), normal (between -σ and + σ), high (between +σ and +max δ σ) and very high (+max δ σ). The centroids and the inner standard deviation of the cluster are converted into the number of standard deviations within the database, then a description is associated based on the adjectives scale. The description is terminated when the cluster is fully defined with respect to the others. For example, let's consider cluster 1 (centroid=(σ_1, $2\sigma_2$, $3\sigma_3$), std=($0.3\sigma_1$, $0.2\sigma_2$, $0.3\sigma_3$)), cluster 2 (centroid=($-\sigma_1$, $2\sigma_2$, $-5\sigma_3$), std=($0.25\sigma_1$, $0.1\sigma_2$, $0.35\sigma_3$)) and cluster 3 (centroid=($2\sigma_1$, $-4\sigma_2$, $3\sigma_3$), std=($0.2\sigma_1$, $0.15\sigma_2$, $0.5\sigma_3$)). Let's consider that we want to describe cluster 1 knowing that its complexity cost vector C=(10,9,11). We take the second characteristic (the less complex one because C_2=9<C_1=10<C_3=11) first. It gets the adjective "high", however cluster 2 has the same adjective for this feature, so we continue the description. We take the first characteristic (the second less complex C_1=10). It gets the adjective high, being the only cluster that can be described by the description (high charac-

teristic 2, high characteristic 1) the description is finished. In our database of films for example high rank actors = little known actors (because the rank is the inverse of the frequency of appearance), very low rank budget = very important budget and therefore we obtain intuitive descriptions of the clusters.

6.3 Online clustering and description of contextualized objects

At this point we have clusters and their descriptions obtained offline. Now, the interlocutor gives the name of a movie he wants to be described and optionally his personal complexity threshold E which can be general or according to the different categories. We add the movie in the nearest cluster according to the chosen clustering method. Once the requested movie is in its cluster, we use the principle of **discriminatory description** to make a description complement D2. We add the characteristics that make the movie distinctive: we add the features by order of discriminating power while rejecting those for which the complexity > E until the description imprecision cost (log_2(number of movies that have the same description)) is less expensive than the cheapest available feature in terms of complexity. The description returned to the caller is the concatenation of the description of the cluster containing only features of complexity < E and D2.

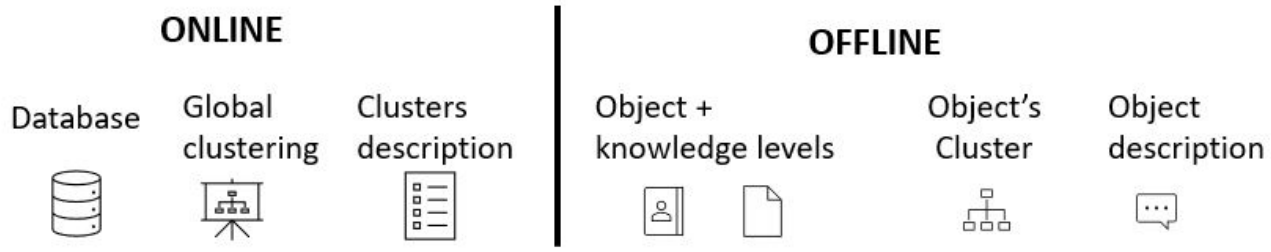


Figure 9: pipeline.

7 PERSPECTIVES

The present report is stemming from an alternative view on clustering. It can be used as a basis for the creation of methods for **more elaborate tasks** related to cognitive processes or that are intended to give results that are as close as possible to a human output. Some ideas such as **precision** or **complexity threshold E** are completely new and therefore pave the way for a new type of heuristics and new insights. Moreover, even if some of the proposed approaches had an underlying complexity despite their apparent simplicity, the reflection is very transparent

throughout the work. For a future study we could generalize the framework in which we make descriptions. For instance, our hierarchical clustering method does not require a lot of data and is very malleable. Therefore, a description model following a growing database is very interesting because it is representative of the modification of our way of describing over time with the accumulation of knowledge. In addition, the core of our ideas have been expressed through **Kolmogorov's complexity** and **contrast**, and the results obtained, although difficult to evaluate, are in agreement with human reasonings conveyed via these two operators, one can thus wonder if other mechanisms can also be derived from similar explicit operations. We also briefly investigated the behavior of the vectors obtained by contrast but are confident that this topic requires more research. Finally I hope this report is a new and exciting toolbox.

REFERENCES

[1] Santosh Kumar Uppada : Centroid based clustering algorithms-a clarion study. 2014.

[2] Shi Zhong et Joydeep Ghosh : A unified framework for model-based clustering. *J. Mach. Learn. Res.*, 4:1001–1037, décembre 2003.

[3] Rashi Chauhan, Pooja Batra et Sarika Chaudhary : A survey of density based clustering algorithms. *www.ijcst.com*, 5, 05 2014.

[4] Emiel Krahmer et Kees van Deemter : Computational Generation of Referring Expressions: A Survey. *Computational Linguistics*, 38(1):173–218, 03 2012.

[5] Peter D. Grünwald et Paul M.B. Vitányi : Algorithmic information theory. *Philosophy of Information*, 8:281–317, 2008.

[6] Ming LiPaul et Vitányi : Algorithmic complexity. *An introduction to Kolmogorov complexity and its applications*, Ch. 1.

[7] Jean-Louis Dessalles : *From Conceptual Spaces to Predicates*, pages 17–31. Springer International Publishing, Cham, 2015.

[8] Giovanni Sileno, Isabelle Bloch, Jamal Atif et J-L. Dessalles : Computing Contrast on Conceptual Spaces. *In International Workshop on Artificial Intelligence and Cognition*, pages 11–25, Palermo, Italy, 2018.

[9] Giovanni Sileno, Isabelle Bloch, Jamal Atif et Jean-Louis Dessalles : Similarity and contrast on conceptual spaces for pertinent description generation. *In* Gabriele Kern-Isberner, Johannes Fürnkranz et Matthias Thimm, éditeurs : *KI 2017: Advances in Artificial Intelligence*, pages 262–275, Cham, 2017. Springer International Publishing.

[10] Alexandre A. J. Denis : Generating Referring Expressions with Reference Domain Theory. *In INLG 2010*, pages 27–35, Dublin, Ireland, juillet 2010.

[11] Ehud Reiter et Robert Dale : A fast algorithm for the generation of referring expressions. *In COLING 1992 Volume 1: The 14th International Conference on Computational Linguistics*, 1992.

[12] Luis Galárraga, Julien Delaunay et Jean-Louis Dessalles : REMI: mining intuitive referring expressions on knowledge bases. *CoRR*, abs/1911.01157, 2019.