Finding True Clusters: On the Importance of Simplicity in Science

March 2, 2019

Contents

1 Introduction

Machine learning is a scientific discipline that can be divided into two main branches: supervised machine learning and unsupervised machine learning. Generally speaking, an algorithm is supervised when it learns to make predictions based on examples. For instance, an algorithm can be trained to predict y given x based on several instances of (x,y) vectors.

In a supervised setting, one of the main challenges is to avoid a phenomenon known as overfitting, *i.e.*, to avoid choosing models that make poor predictions on data that have never been used before even though they make excellent predictions with the data used to train them. One of the key characteristics of overfitting models is that they are too complex. They have too many adjustable (or effective) parameters (Rochefort-Maranda 2016). This fact has led some philosophers to study the importance of parametric simplicity in model selection (See Sober 2002; Forster and Sober 1994; Forster 2007; Hitchcock and Sober 2004).

In unsupervised contexts however, things are fairly different. There is no such thing as ”learning from examples” or ”overfitting models”. That is because we do not make predictions when we use unsupervised algorithms. This does not mean that parametric simplicity is not important. It just means that it takes a different form. In this paper, we aim to show just how simplicity matters in such contexts. This is important because unsupervised machine learning algorithms have barely received any attention in philosophy. Yet, there is a direct link between simplicity and truth in unsupervised contexts that we do not find in their supervised counterparts. This has thus far evaded philosophical discussions on simplicity.

Unsupervised algorithms are mainly used to find geometrical shapes within a data set. This is the truth that we are aiming for. We show how those spatial structures, if they exist, can be found with the help of Ockham's razor. The resulting groups can be used for various ends (*e.g.*, market segmentation, classification of diseases, or species identification).

There are various kinds of clustering algorithms and each comes with its own challenges. But here we will focus on one popular algorithm called the K-means algorithm. It is fairly easy to grasp, it is used in current machine learning practice (e.g., Samanta and Khan 2018), and it shows the importance of simplicity in unsupervised settings. By the same token, we will also show how other epistemic virtues, such as theoretical coherence, makes a difference in cluster selection.

The main point of this paper is to underscore the link between parametric simplicity and truth in an unsupervised context. We shall also show how dimensional simplicity, as it is defined in Rochefort-Maranda 2016 can help with that goal. The topic of this paper is not the K-means algorithm or its possible extensions (like the sparse, fuzzy or kernel K-means). The K-means algorithm is taken as a mere case study to make a philosophical point -one that has evaded philosophers so far.

This paper contains two main sections. In the first section, we give an introduction to the K-means algorithm and we explain how we can choose the number of clusters. We show how simplicity matters for this task and we expound on the link between simplicity and truth. This link is absent in supervised contexts and this is why it is valuable to bring the philosophical discussion to the unsupervised side of machine learning. In the

second section, we underscore the weaknesses of the K-means algorithm and explain how theoretical coherence and practical concerns play a crucial role in determining the final clusters, regardless of the algorithm we decide to use.

2 The K-means Algorithm and the Truth

K-means is an iterative algorithm that partitions each observation of a data set into a predetermined number of clusters. For example, if we want to create 8 clusters (K=8), then the algorithm will created 8 non-overlapping sets and each observations will belong to one and only one of those sets. The mathematical implementation of this algorithm is expounded here.

The algorithm can be implemented with three main steps (James et al. 2013, p.388). Variations and sophistications exist but here is a typical/basic description.

1. Specify the number of groups K that we want to create.
2. Randomly assign a group to each observation.
3. Repeat the following two steps until the assignment does not change.
    (a) Assign each observation to the closest centroid.
    (b) For each cluster K, compute a mean.

In fact, the algorithm aims to find the K clusters such that the Euclidean distances between every two points inside each cluster is minimal. In other words, the algorithm aims to minimise the within cluster variance.

This concept can be defines as follows where K is the number of clusters, C is a cluster, and p is the number of features (clustering variables).

$$\underset{C_1,C_2,\ldots,C_k}{\text{minimise}}\{\sum_{k=1}^{K}\frac{1}{|C_k|}\sum_{i,i'\in C_k}\sum_{j=1}^{p}(x_{ij}-x_{i'j})^2\}$$

The formula above is called a cost function. By looking to minimise this function, the algorithm is therefore ideal to create spherical shaped clusters of similar sizes and similar density (SeeFahim et al. 2008). Knowing this much, we also know that the k-means algorithm is a perfect tool to find such geometrical structures in a dataset. If this is the truth that we are looking for, then this is the algorithm to use. In other words, this algorithm will label truthfully each observations in our dataset if the dataset is made of spherical shaped clusters of similar sizes and similar density.

When we use a clustering algorithm with the purpose of finding geometrical structures, we need to make such an assumption right from the beginning. This will determine the choice of the clustering algorithm. Whether or not this is the right choice will usually depend on how well we can interpret the clusters (See section 3).

Different algorithms are able to find different kinds of geometric structures. Although we are now discussing the k-means algorithm, the authors of this paper are fully aware that there are other clustering algorithms that can latch onto different geometric shapes.

Having thus decided on the k-means algorithm, we now need to specify the number of clusters K before we can even run the algorithm, which brings the question of how to chose K in the first place. If K=8, the algorithm will create 8 clusters as long as there are at least 8 observations. But

it could also create 7 or 6 clusters. We need to find the true number of clusters with spherical shapes of similar size and similar density.

2.1 How to Choose K: Simplicity and Its Link with Truth

In this section we shall discuss three adequate ways to solve this problem and explain how parametric simplicity plays a crucial role. I will show how we need to take into account parametric simplicity if we want to find all the clusters of spherical shapes with similar size and similar density (the truth).

2.1.1 The Elbow Method

A naive attempt at solving this problem might be to choose the number of clusters such that the within cluster variance is the smallest possible. Unfortunately, that variance inevitably diminishes as the number of clusters K increases (in the extreme case where there is one observation per cluster, there is no variance).

A better idea, and one that is used in practice, is to plot the within cluster variance in function of the number of clusters and look for an inflexion point (for the "elbow"). In other words, the idea is to look for the smallest number of clusters that is associated with a sharp drop in within cluster variance.

Here is an example taken from a toy dataset that we have created. It is important to use a toy dataset in this context since we aim at showing that we can truthfully label observations with simplicity and the k-means

algorithm under the assumption that those observations belong to clusters of spherical shapes with similar size and similar density.

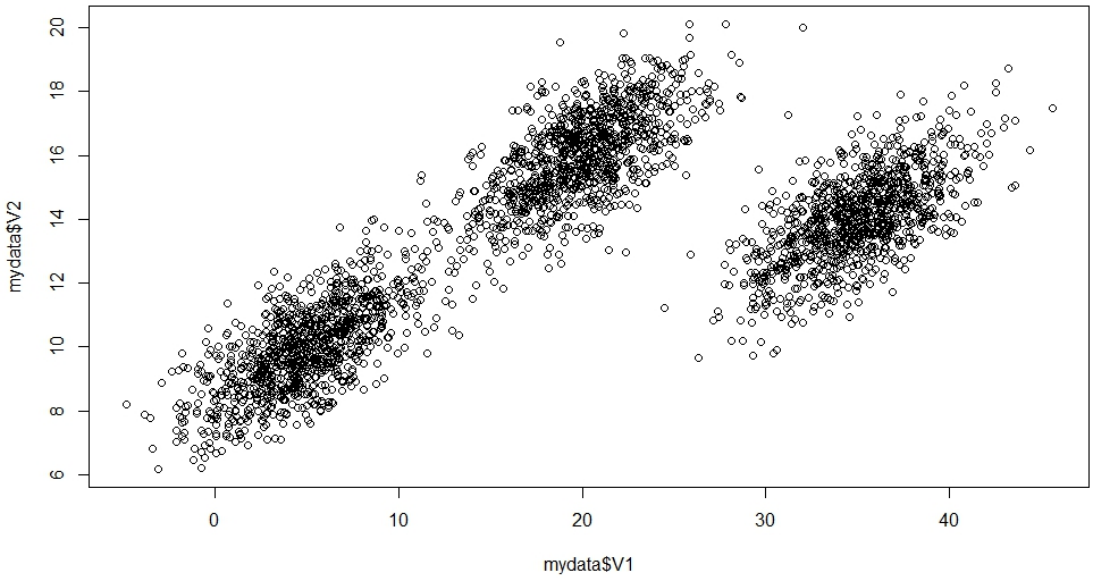

Figure 1: Toy Dataset

Figure 1 show three obvious groups. By using a K-means algorithm and by plotting the within cluster variance from k=1 to k=15, we can obtain the results shown in Figure 2.

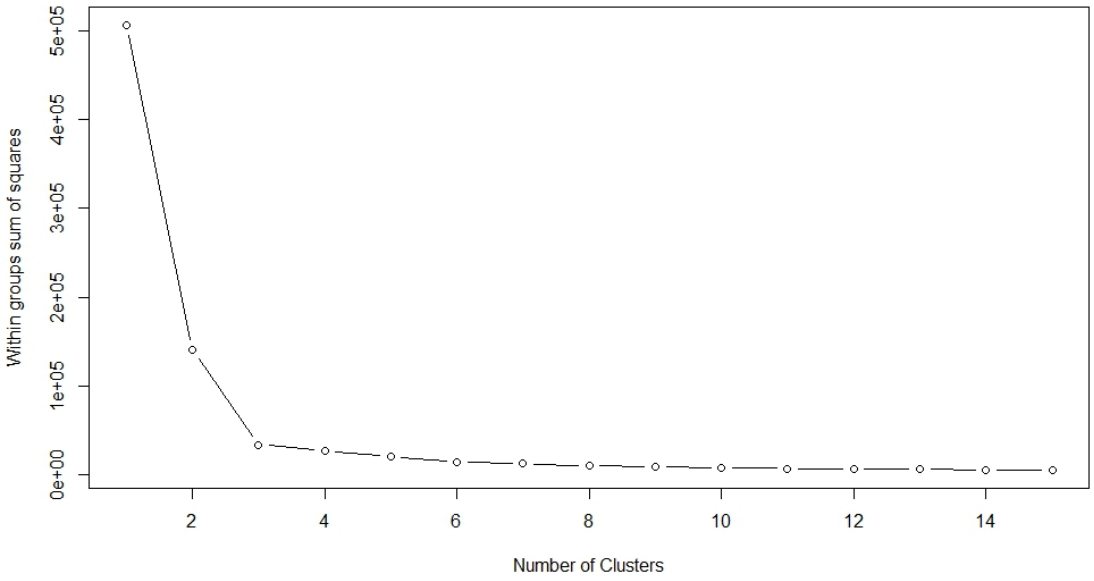

Figure 2: Within Cluster Sum of Squares in Function of the Number of Clusters

As we can see, the within cluster variance decreases with k. However, the point of inflection (the smallest number of clusters that is associated with a sharp drop in within cluster variance) is found for k=3. This suggests that there are 3 groups (surprise!) and we can see the result of the clustering with the K-means algorithm in Figure 3.

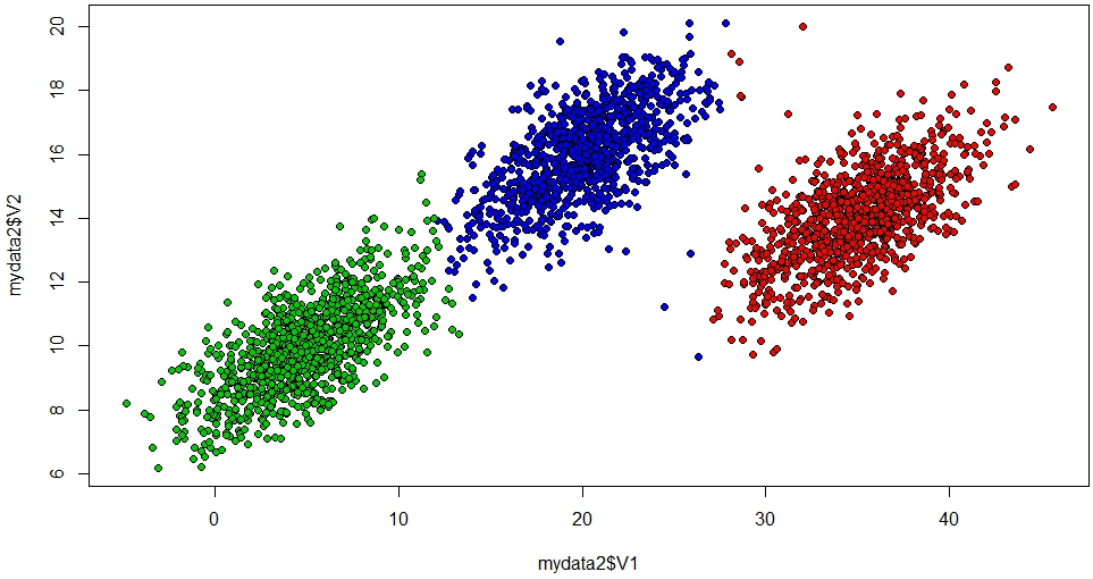

Figure 3: Final Clustering with k=3. The colors represent the clustering done by the algorithm.

The search for a point of inflexion in a graph like the one pictured in Figure 2 shows the importance of parametric simplicity for finding all the clusters of spherical shapes with similar size and similar density (the truth). Parametric simplicity here is defined by the number of parameters that an algorithm is trying to adjust by using a cost function. The less parameters we have to adjust, the simpler is the result. As the elbow method shows, if we want to find the true number of clusters of spherical shapes with similar size and similar density in the dataset, we need to find the

simplest clustering (defined in terms of parametric simplicity) such that any further minimisation of the cost function would be an artefact of the complexity (defined in terms of parametric simplicity) of the clustering.

The more clusters we try to fit onto a data set, the lower will be the cost function that we try to minimise with the K-means algorithm (see p.5). However, we will fail to latch onto real structures if we ask the algorithm to find too many clusters. In this context, Ockham's razor is used in a way that makes sure that we will not specify more clusters that we need to, in order to minimise adequately the cost function.

Of course here there are only 3 clusters in the toy dataset. What if we had 100 clusters instead? Would we not end up choosing 100 parameters such that simplicity would not matter? The answer is no. 100 would be the smallest number of parameters (e.g. the simplest clustering) such that any further minimisation of the cost function would be an artefact of the complexity of the clustering.

The argument is this: If we want to find the true number of clusters of spherical shapes with similar size and similar density in a dataset with the k-means algorithm, we need to make sure that the number of parameters that we are trying to adjust is minimal such that any further minimisation of the cost function would be an artefact of the complexity of the clustering. The true (finite) number of clusters can be as large as one wishes.

2.1.2 The Gap Statistic and Cross-validation

Now, the elbow method is quite rudimentary. We can use other methods to choose the correct number parameters, such as maximising the Gap Statis-

tic or a cross-validation score. But the argument for simplicity remains the same. If we want to find the true number of clusters of spherical shapes with similar size and similar density with the k-means algorithm, we need to make sure that the number of parameters that we are trying to adjust is minimal such that any further minimisation of the cost function (See p.5) would be an artefact of the complexity of the clustering.

The Gap Statistic was first developed in 2000 by Tibshirani, Whalter, and Hastie (Tibshirani et al. 2001). Give that D is a Euclidean distance, the statistic is defined as follows:

$$D_r = \sum_{i,i' \in C_r} D_{ii'}$$

$$W_k = \sum_{r=1}^{k} \frac{1}{2n_r} D_r$$

$$Gap_n(k) = \mathbf{E}_n^*\{log(W_k)\} - log(W_k)$$

$\mathbf{E}_n^*$ denotes expectation under a sampling of size n from a reference distribution (uniform). Intuitively, the Gap statistic measures the distance between a given clustering and the clustering that we would have obtained under a reference distribution such as the uniform distribution. The larger the distance, the better the clustering. However, a large Gap can be an artefact of the complexity of the clustering. Therefore, it is recommended to choose the smallest number of clusters such that its corresponding Gap statistic is larger than the next Gap statistic minus its standard deviation

(Tibshirani et al. 2001, p.415). Once again, we can see the importance of parametric simplicity.

If we plot the Gap statistic and it's variance for different number of clusters on the toy dataset, here is what we obtain: (The choice is clear. There are 3 clusters in the dataset.)

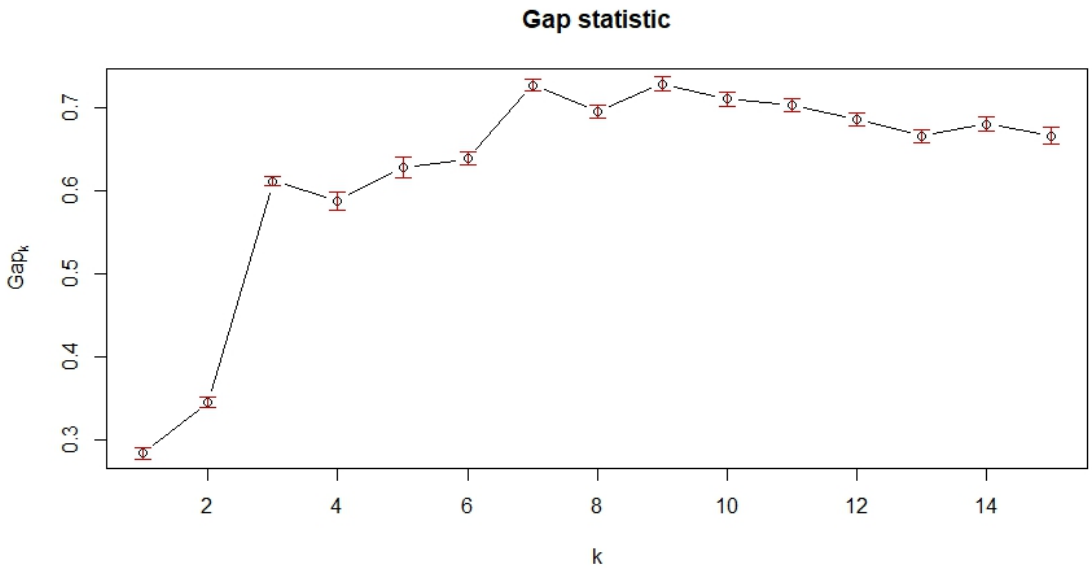

Figure 4: Gap Statistic in Function of the Number of Clusters applied on the Toy Dataset.

We could also compute a cross-validation score, to choose the right number of clusters. Cross-validation in an unsupervised context comes in many shapes and forms (See Wang 2010 and Tibshirani and Walther 2005). Here is one way to do it as defined in Wang 2010 :

1. Divide the data set into three: one validation set, two training sets.
2. Cluster the two training sets separately with the same clustering algorithm and the same number of clusters K.
3. For each training set, use the centers found with the clustering algorithm to cluster the observations in the validation set. This will result into two different sets of clusters containing only the observations of the validation set
4. We can measure the stability of the clustering by counting how many pairs of observations are clustered together in one set of clusters but not in the other. The more pairs we find, the least stable is the clustering.
5. Find the optimal number of clusters K such that the clustering displays the most stability.
6. Repeat the experiment. Choose the number of clusters K that is voted the optimal number of clusters most of the time.

When we use such a technique, we also find 3 clusters in the toy dataset. Any other number of clusters is less stable. This is a direct consequence of the fact that a better minimisation of the cost function described on p.5 is an artefact of the complexity of the clustering.

2.2 Unsupervised VS Supervised Contexts

In a supervised learning scenario, we need parametric simplicity if we want to have a good predictive model. That kind of simplicity is necessary

to reach the goal of predictive accuracy. But even if we find the best predictive model, it does not mean that we will find the true model. A good predictive model does not imply that the model is true or close to the truth in any way. Simple models can be false and make excellent predictions. In fact, they can make better predictions than the true model. That is why some have underscored the gap between parametric simplicity and truth. Parametric simplicity in supervised context aim for predictive accuracy, not truth.

> Perhaps the most interesting of the standard arguments in favor of simplicity is based upon the concept of 'overfitting' [...] although this argument is sound and compelling, so far as using an equation for predictive purposes is concerned, it is also irrelevant to the question at hand, which concerns finding the true theory rather than using a false theory for predictive purposes (Kelly 2007a, p.113).

On the other hand, in an unsupervised context like the one presented above, we need parametric simplicity if we want to have a chance at discovering specific geometric shapes within our dataset. Parametric simplicity is essential for the algorithm to latch onto true structures (if they exist). That is what sets apart parametric simplicity in unsupervised context. The idea that we wish to underscore here is that parametric simplicity, when using the K-means algorithm, is not an indication of truth but the methodological principle that has been explained in details here is truth conducive.

The idea is not that we can find real kinds with the K-means algorithm and Ockham's razor. This is a totally different question that is addressed in (Hennig 2015). In this paper, we are talking about finding true patterns inside the data set. Whether or not those patterns point to something "real" or "constructed" is a different matter.

"Is a set of young millennials, living with their parents and who like to buys clothes online a natual or a constructed kind?" is not the type of question that we are interested in. What we are interested in here is whether or not the representation of those individuals inside a data set forms a specific structure of points in space and whether the K-means algorithm can truly "catch" that structure. We have shown that the algorithm can if we pay attention to parametric simplicity.

What are those true patterns? They are geometrical shapes such as spherical shaped clusters of similar sizes and similar density. If those specific clusters are in our dataset, the k-means algorithm will find them only if we pay attention to parametric simplicity as we try to minimise the cost function associated with that algorithm:

$$\underset{C_1,C_2,\ldots,C_k}{\text{minimise}}\{\sum_{k=1}^{K}\frac{1}{|C_k|}\sum_{i,i'\in C_k}\sum_{j=1}^{p}(x_{ij}-x_{i'j})^2\}$$

2.3 Simplicity Strikes Again

Besides parametric simplicity as defined by the number of clusters K that we choose to partition our data with the K-means algorithm, we must also be parsimonious with the number of variables that we choose in order to

partition the data. This is sometimes referred to as dimensional simplicity (Rochefort-Maranda 2016). Not every variable in a data set are useful in order to cluster specific shapes in a dataset. Some of them might just add meaningless noise.

This is why another algorithm, called sparse k-means, can be used when we work in high dimensions (Witten and Tibshirani 2010). It is an extension of the K-means algorithm with an emphasis on the importance of dimensional simplicity.

The idea behind this algorithm is quite simple, as we previously mentioned, K-means tries to minimise the within cluster sum of squares. This is equivalent to maximising the between cluster sum of squares (BCSS).

$$BCSS = \sum_{j=1}^{p}(\sum_{i=1}^{n}(x_{ij} - \mu_j)^2 - \sum_{k=1}^{K}\sum_{i \in C_k}(x_{ij} - \mu_{kj})^2)$$

Now we can assign weights w_j to each variables as we try to maximise BCSS.

$$\max_{C_1 \ldots C_k, w} \{\sum_{j=1}^{p} w_j(\sum_{i=1}^{n}(x_{ij} - \mu_j)^2 - \sum_{k=1}^{K}\sum_{i \in C_k}(x_{ij} - \mu_{kj})^2)\}$$

$$||w||^2 \leq 1$$

$$||w||_1 (norm1) \leq s$$

$$w_j \geq 0 \forall j$$

The idea here is that unimportant variables will have no or very little weight such that we can eliminate them from the clustering procedure.

This will simplify and hopefully improve the quality of the clustering. This is yet another way that simplicity can help an algorithm to find existing geometrical shapes in the dataset.

3 The Weaknesses of the K-means Algorithm and the Importance of Theoretical Coherence and Usefulness

So far, the argument for simplicity when using the k-means algorithm as been that we need to pay attention to it if we wish to find spherical shaped clusters of similar sizes and similar density. But we normally do not know if there are such shapes in the dataset because we are in an unsupervised context. The K-means algorithm performs rather poorly when the clusters have, for example, an elongated shape like the ones presented in Figure 5

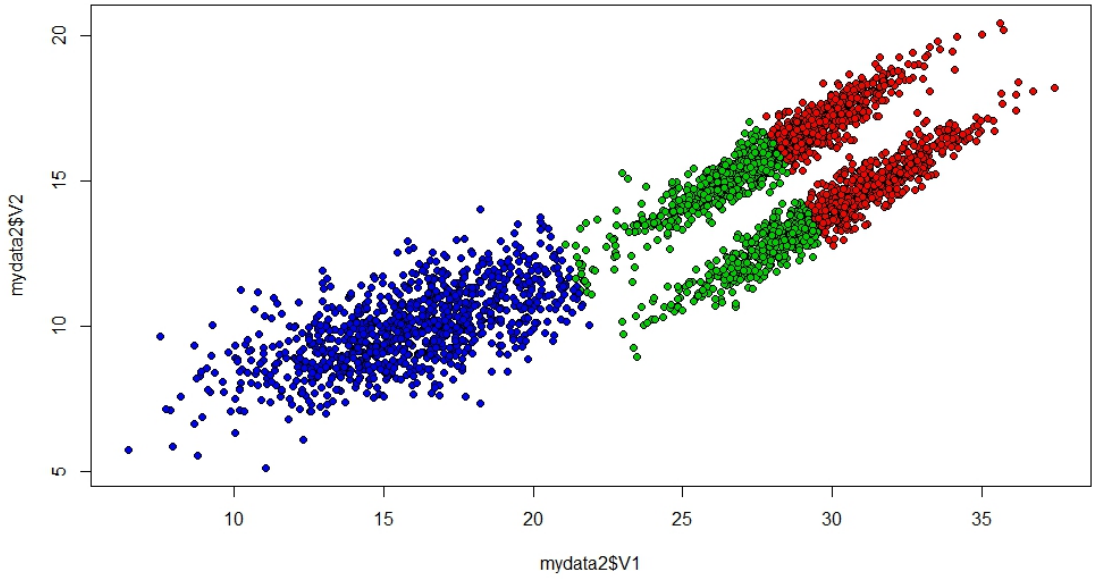

Figure 5: Elongated Clusters: The colors represent the clustering done by the algorithm with k=3

Other algorithms such as the EM (expectation-maximization) algorithm and the kernel K-means clustering (and many other algorithms), can outperform K-means in that respect but the problem is that it is usually impossible to visualise the clusters that we have produced in order to validate the performance of any clustering algorithm. In the previous examples it was easy to do so because there were only 2 clustering variables such that we could know in advance if the data could be separated into groups and if the algorithm could find the real structures. However, in high di-

mensional spaces, we cannot visually validate the result of a clustering algorithm and we do not even know how many clusters there are.

That is why we always have to study the properties of a given clustering to see if the clusters are interpretable and useful. Theoretical coherence and usefulness will ultimately determine the choice of a given clustering and provide evidence as to whether or not we have captured true structures (specific geometrical shapes) with a method like K-means.

As stated in a study on clustering fMRI time series with the K-means algorithm, the authors point out that the choice of the number of clusters must be justified by the interpretation of the clusters: "When the chosen number is not reflected in the data, the results might end up being essentially meaningless"(Goutte et al. 1999, p.300).

In fact, there might be more than one possible grouping with one data set Looking at figure 4, for example, we could decide to group the two elongated clusters together since they are so close. We might not need to make very detailed groups. Ultimately, we will chose the one that suits our practical goals.

> The difficulty with unsupervised clustering is that there is a huge number of possibilities regarding what will be done with it and (as yet) no abstraction akin to a loss function which distils the end-user intent. Depending on the use to which a clustering is to be put, the same clustering can either be helpful or useless. (Guyon et al. 2009, p.66).

In a market segmentation project for example, perhaps it is more useful to group individuals of similar generations, jobs and location and not

individuals with the same ethnic background and salary range. However, interpretability and practicality are no guarantee that we have carved the data at its joints. Being able to use and give an interpretation to a particular set of clusters is not a substitute for the quantitative work that has been presented in section 2. If we wish for our work to be replicable, then it has to be grounded in real structural properties of the data.

4 Conclusion

In this paper we have expounded on a neglected side of machine learning within the philosophical literature which is called "unsupervied machine learning". Unlike their supervised counterparts, unsupervised algorithms are not evaluated with respect to their predictions. They are usually assessed in function on the quality and usefulness of the clusters that they produce.

We have shown that simplicity (parametric and dimensional), coherence and usefulness are all part of the epistemic toolbox that is used in order to determine the quality of unsupervised algorithms such as the K-means algorithm. This has given us the opportunity to underscore the link between parametric simplicity, dimensional simplicity, and truth in unsupervised contexts. The main take-away message is the following:

- *Parametric and dimensional simplicity are not indicators of truth but the methodological principle that urges us to pay attention to such notions of simplicity is truth conducive. The truth that we are looking for are specific geometrical shapes and we know which algorithm can find which shape*

provided that we pay attention to parametric and dimensional simplicity.

Its meaning and justification have been expounded in details in section 2.

Ockham's razor can be used in a way that makes sure that we will not specify more clusters (parameters) that we need to in order to minimise a cost function. This is essential if we want this algorithm to latch onto real structures. It can also be used in order to prevent the clustering of noise by cleverly reducing the number of clustering variables. We have made this point by showing how the sparse K-means algorithm works.

References

Baker, A. (2013). Simplicity. In E. N. Zalta (Ed.), *The Stanford Encyclopedia of Philosophy* (Fall 2013 ed.).

Fahim, A., G. Saake, A. Salem, F. Torkey, and M. Ramadan (2008). K-means for spherical clusters with large variance in sizes. *Journal of World Academy of Science, Engineering and Technology*.

Forster, M. and E. Sober (1994). How to Tell When Simpler, More Unified, or Less *Ad Hoc* Theories will Provide More Accurate Predictions. *The British Journal for the Philosophy of Science 45*(1), 1–35.

Forster, M. R. (2007). A Philosopher's Guide to Empirical Success. *Philosophy of Science 74*(5), 588–600.

Friedman, J., T. Hastie, and R. Tibshirani (2001). *The Elements of Statistical Learning*, Volume 1. Springer series in statistics Springer, Berlin.

Goutte, C., P. Toft, E. Rostrup, F. Å. Nielsen, and L. K. Hansen (1999). On clustering fmri time series. *NeuroImage 9*(3), 298–310.

Guyon, I., U. Von Luxburg, and R. C. Williamson (2009). Clustering: Science or art. In *NIPS 2009 workshop on clustering theory*, pp. 1–11.

Hennig, C. (2015). What are the true clusters? *Pattern Recognition Letters 64*, 53–62.

Hitchcock, C. and E. Sober (2004). Prediction Versus Accommodation and the Risk of Overfitting. *The British Journal for the Philosophy of Science 55*(1), 1–34.

James, G., D. Witten, T. Hastie, and R. Tibshirani (2013). *An Introduction to Statistical Learning*. Springer.

Kelly, K. T. (2007a). How Simplicity Helps You Find the Truth Without Pointing at it. In *Induction, algorithmic learning theory, and philosophy*, pp. 111–143. Springer.

Kelly, K. T. (2007b). Ockhams Razor, Empirical Complexity, and Truth-Finding Efficiency. *Theoretical Computer Science 383*(2), 270–289.

Rochefort-Maranda, G. (2016). Simplicity and model selection. *European Journal for Philosophy of Science 6*(2), 261–279.

Samanta, A. K. and A. A. Khan (2018). Computer aided diagnostic system for automatic detection of brain tumor through mri using clustering based segmentation technique and svm classifier. In *International Con-*

ference on Advanced Machine Learning Technologies and Applications, pp. 343–351. Springer.

Sober, E. (2002). Instrumentalism, Parsimony, and the Akaike Framework. *Philosophy of Science 69*(S3), S112–S123.

Tibshirani, R. and G. Walther (2005). Cluster validation by prediction strength. *Journal of Computational and Graphical Statistics 14*(3), 511–528.

Tibshirani, R., G. Walther, and T. Hastie (2001). Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 63*(2), 411–423.

Wang, J. (2010). Consistent selection of the number of clusters via cross-validation. *Biometrika 97*(4), 893–904.

Witten, D. M. and R. Tibshirani (2010). A framework for feature selection in clustering. *Journal of the American Statistical Association 105*(490), 713–726.