Clustering as an example of optimizing arbitrarily chosen objective functions

Marcin Budka

Bournemouth University, BH12 5BB Poole, UK
mbudka@bournemouth.ac.uk, mbudka@gmail.com
WWW home page: http://www.budka.co.uk/

Abstract. This paper is a reflection upon a common practice of solving various types of learning problems by optimizing arbitrarily chosen criteria in the hope that they are well correlated with the criterion actually used for assessment of the results. This issue has been investigated using clustering as an example, hence a unified view of clustering as an optimization problem is first proposed, stemming from the belief that typical design choices in clustering, like the number of clusters or similarity measure can be, and often are suboptimal, also from the point of view of clustering quality measures later used for algorithm comparison and ranking. In order to illustrate our point we propose a generalized clustering framework and provide a proof-of-concept using standard benchmark datasets and two popular clustering methods for comparison.

Keywords: clustering, cluster analysis, optimization, genetic algorithms, particle swarm optimization, general-purpose optimization techniques

1 Introduction

Clustering, also known as cluster analysis, is a well-known unsupervised learning task in data mining. The application areas of clustering are virtually endless, from customer segmentation in Customer Relationship Management systems, to image processing, to information retrieval or mining unstructured data [15]. Although the research on clustering dates back to the mid-20^{th} century and there are now dozens of various clustering algorithms one can find in the literature [15], the problems we are facing today are essentially still the same, only on a larger scale due to the recent explosion of the amounts of generated data. Most typical of these problems concern the number of clusters (groups) in the data, the choice of an appropriate similarity measure or the choice of the clustering algorithm itself. We argue that these concerns stem from more fundamental issues of what constitutes a cluster or how to assess the quality of the obtained clustering result.

Clustering is also a good example of a common practice of optimizing arbitrarily chosen criteria in the hope that they are well correlated with the criterion actually used for assessment of the result. We perceive it as an interesting contradiction if for example one is using the mean absolute percentage error to select the best model, but mean squared error to fit the candidate models, while

neither of them adequately reflects the problem being addressed, in effect leading to spurious and puzzling outcomes [18]. This of course stems from the fact that the latter is much easier to optimize, but as we demonstrate in this study, direct optimization of the assessment criterion is often still a viable approach, especially that the computational power is nowadays cheap and abundant.

In the following sections we address the above problems, pointing out the gaps in the current body of research and proposing an alternative view on clustering, which unifies all the issues mentioned above under a comprehensive, generalized framework. This work should be perceived as a proof of concept or position paper and hence the experimental evaluation of the proposed approach is by no means exhaustive in terms of algorithms and datasets used.

2 The notion of a cluster

Clustering is the process of discovering structure in data by identifying homogenous (or natural) groups of patterns [8]. The whole procedure is hence based on a similarity or distance function. The general agreement is that the patterns within a cluster should be more similar to each other than to patterns belonging to different clusters [15].

Definition of a cluster as a homogenous or natural group of patterns is however still not precise enough, in a sense that it does not specify in detail the features that a clustering algorithm should have. Hence many popular notions of clusters exist e.g. well separated groups with low distances among cluster members (e.g. K-means clustering algorithm [19]), dense areas of the input space (e.g. Expectation Maximization clustering algorithm [6]) or groups of patterns having particular information-theoretic properties (e.g. Cauchy-Schwarz divergence clustering algorithm [16]). The central issue is that in most cases these notions of clusters result from the characteristics of the clustering algorithm rather than being consciously designed to fit a particular application or problem, which often stems from the lack or ignorance of domain knowledge.

The choice of a similarity measure, which is inherent in the definition of what constitutes a cluster, is a good example of the above. Suppose that in a database of driving licence candidates there are the following two binary attributes: 'blind in left eye' and 'blind in right eye'[1]. Without knowing the context, one might be tempted to use the Hamming distance [13] (i.e. the number of positions at which the corresponding input vectors are different) as a measure of (dis)similarity. This would however lead to a situation in which a person blind in left eye is more similar to a person blind in both eyes than to a person blind in right eye, producing spurious results[2].

In our view the notion of a cluster and other related choices (e.g. similarity measure) should therefore be application-driven. This is especially important in

[1] This excellent example has been taken from [8].

[2] In some countries this could also possibly lead to the developer of the system being sued for discrimination if people blind in the left eye and people blind in the right eye were not treated equally.

unsupervised learning as the obtained results are often used to define a supervised learning problem (e.g. customer segmentation for developing a classification system for new customers), where the usual GIGO[3] principle applies. Also it is difficult to think about a 'natural grouping of patterns' without a context of a particular application – in the driving licence candidate example, grouping people blind in left eye and blind in right eye together is certainly more 'natural' than assigning them to the 'blind in both eyes' cluster.

3 The number of clusters

Every standard clustering algorithm requires the user to choose the number of clusters to be identified in the data [15]. This in a sense contradicts the whole idea of discovering structure in data, as the choice of the number of clusters imposes a limit on what can actually be discovered, especially if this parameter needs to be specified in advance. Although in hierarchical clustering [20] the user at least has a chance to explore the results at different levels of granularity, this is only feasible when dealing with relatively small datasets.

Of course there are approaches which try to provide guidance in this respect (see [7] for example) but they are usually very specific heuristics, tailored towards particular clustering algorithms and certainly suboptimal.

Recognizing the need for a more systematic and general approach we hence propose to cast clustering as an optimization problem within a generalized framework, allowing to free the users of the burden of 'guesstimating' the initial parameters of clustering algorithms and effectively artificially and considerably limiting the space of possible solutions.

4 Clustering as an optimization problem

The central issue in solving the clustering problem (and in fact any other computational problem) is to be able to quantify how good or bad a given result is. This in turn allows to compare various results and choose the one(s), which meet the modeler's expectations, in the context of a particular application.

Clustering can thus be viewed as an optimization problem, where the set of parameters that can be manipulated is not limited to the cluster memberships of all input patterns, but rather also includes the number of clusters, the similarity measure as well as its parameters. The whole difficulty now comes down to designing an appropriate objective function (criterion), reflecting the knowledge of the application area and expectations of the designer. We have deliberately used the word 'design' to emphasize that the objective function shouldn't be chosen arbitrarily from the vast set of existing functions, but should rather be carefully engineered to have the required properties and capture relevant factors from the point of view of a given learning problem.

[3] Garbage In, Garbage Out.

Most popular general-purpose clustering methods have necessarily been designed the other way round, with the clustering algorithm as a starting point and the objective function as a result or by-product of the algorithm design choices. In some cases, it is even not clear what objective a given algorithm actually optimizes, although the authors usually argue and demonstrate (using synthetic data, carefully crafted for this occasion) the superiority of their approach.

As a result the choice of a clustering algorithm to solve a particular problem is often motivated by factors like the familiarity of a user with an algorithm, its popularity or speed. Although in many cases it is difficult to undermine this last motivation (i.e. speed), on a closer look it may turn out that an approach only marginally slower can produce much better results, because it better fits the problem at hand.

Through this kind of uninformed choices people often either optimize 'something' regardless of its fit to their problem, or optimize 'something else' hoping that it is somehow correlated with their objective. The latter issue is actually typical for other learning problems, with a number of recent data mining competitions, where the criteria of success are often quite different from the objectives that the methods used by the contestants try to optimize, being one example.

Another good and in fact somewhat paradoxical example is the routine use of various clustering quality measures (like the Davies-Bouldin index [5] or the Dunn index [10]) to compare and rank different clustering algorithms. Each of these algorithms optimize a different criterion, very often only loosely related to the clustering quality measure used. The paradox here is that the very quality measures used would themselves make good objective functions (as long as their calculation does not require to know the 'ground truth', which is for example the case with a popular Jaccard similarity coefficient [14]), especially that they have been designed to capture various, in one way or another relevant, properties of the clustering result. Going a step further, performing simultaneous, multi-objective optimization of a set of quality measures, which are then used to compare the algorithms, definitely seems like a better initial choice than starting with an arbitrarily selected, popular clustering algorithm. We return to this issue in Section 6, where the two cluster quality measures mentioned above are used as optimization criteria in the experimental validation of the presented ideas.

5 Generalized clustering framework

It is well known that there are no 'one-fits-all' solutions as stated by the 'no free lunch' theorem [8]. Hence the power to solve any problem efficiently lies in the possibility of using a variety of diverse tools or simply a framework. In this section we thus propose a generalized clustering framework, allowing to construct and solve clustering problems using a variety of clustering and optimization algorithms, suited to the level of knowledge about the application domain.

The framework, which has been depicted in Figure 1, consists of two libraries: (1) the Optimization Method Library (OpML) and (2) the Objective Function Library (ObFL). All objective functions in the ObFL are associated with at least

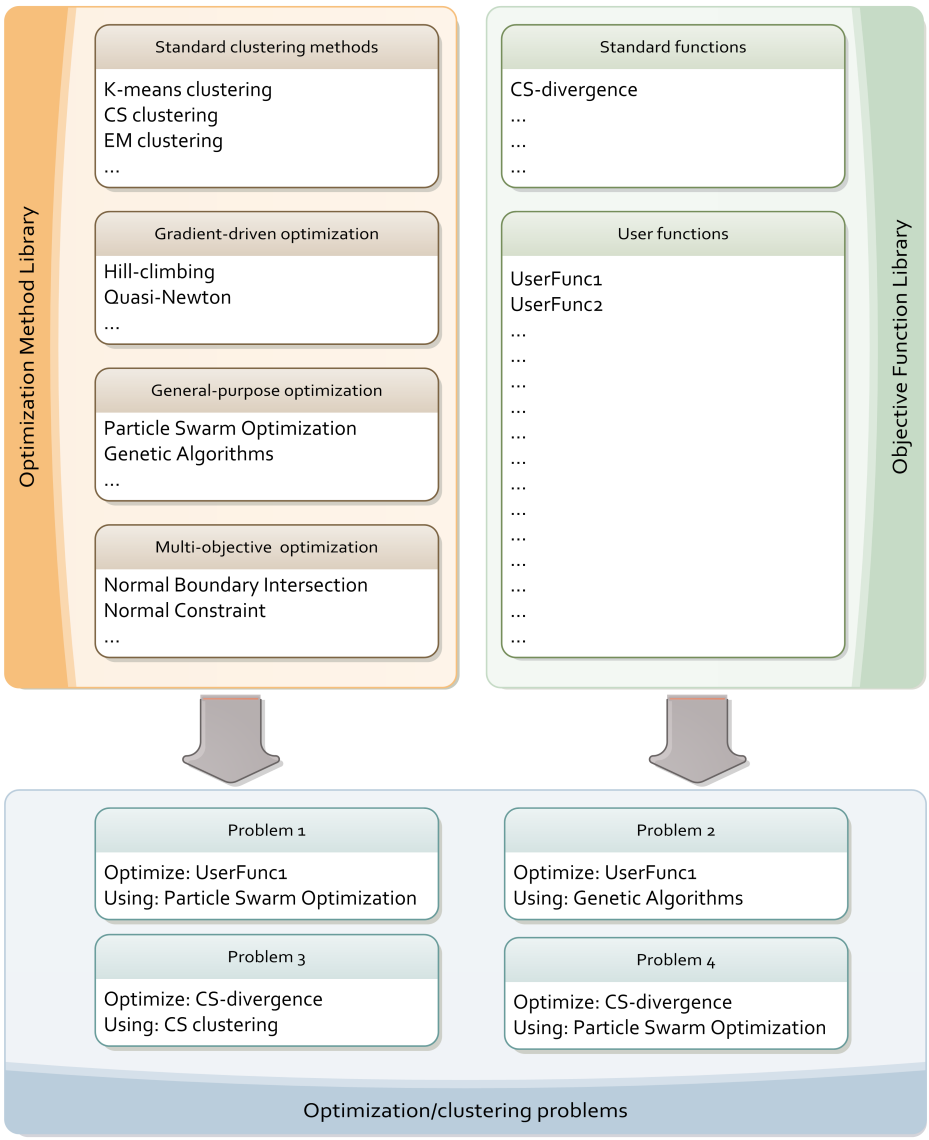

Fig. 1. Generalized clustering framework

one optimization method in the OpML – although this has not been shown in Figure 1 for clarity, not all optimization algorithms are suitable for all objective functions. The ObFL can also include user-defined objective functions, which can be designed freely, for example also as combinations of other functions.

Association of the user-defined functions with appropriate optimization methods depends on the properties of these functions. For objectives which are continuous and differentiable, any gradient-driven optimization technique can be used. If the gradient formula cannot be derived analytically, numerical gradient approximation might be the solution – the optimization literature is very rich (see [11] for example). If possible, an optimization scheme dedicated for the non-standard objective function can also be created. This usually takes on the form of a greedy optimization approach, which often performs surprisingly well in practice (see [3] and [4] for example), with the additional benefit of being deterministic and hence providing reproducibility of the results.

In more complex cases one should revert to the general-purpose optimization techniques, always having in mind that the goal is to optimize a consciously chosen criterion (and not any other criteria which happen to be easier to optimize) for the results to be meaningful. Due to the fact that the space of all possible cluster assignments, combined with all possible values of other parameters (e.g. cluster count) is immense, exhaustive search for an optimum value of a given objective function is computationally prohibitive. Even if it is possible to constrain this space, e.g. by specifying the minimal and maximal cluster count, the situation does not usually improve much. This however is an area in which algorithms based on the exploration-exploitation paradigm, like Genetic Algorithms (GA) [12] or Particle Swarm Optimization (PSO) [17], can prove their merit.

6 Experiments

We have performed the proof-of-concept experiments on two well-known datasets from the UCI machine learning repository [1]: IRIS and WINE, which have been chosen for illustrative purposes. In the experiments we compare the performance of a number of standard clustering methods with the approach proposed in this paper using the two clustering quality measures mentioned before i.e. the Davies-Bouldin index [5] and the Dunn index [10]. Apart from these two indexes, we made the algorithms also optimize the cluster count, which for K-means and EM-clustering meant an exhaustive search in the range between 2 and 9[4] for both datasets.

The Davies-Bouldin index is given by the following formula:

$$DB = \frac{1}{c}\sum_{i=1}^{c}\max_{i\neq j}\left(\frac{\sigma_i + \sigma_j}{\delta(C_i, C_j)}\right) \tag{1}$$

where c is the number of clusters, $\delta(C_i, C_j)$ is the distance between the i^{th} and j^{th} cluster (in our case the Euclidean distance between their centroids C_i and

[4] This number was chosen for convenience of binary representation in GA (3 bits).

C_j), and σ_i represents the average distance of all elements in the i^{th} cluster to its centroid (C_i). The Davies-Bouldin index favours clusterings with low intra-cluster and high inter-cluster distances and the lower its value, the better.

The Dunn index is given by:

$$DI_m = \min_{1\leq i\leq c} \left\{ \min_{1\leq j\leq c, j\neq i} \left\{ \frac{\delta(C_i, C_j)}{\max\limits_{1\leq k\leq c} \Delta_k} \right\} \right\} \tag{2}$$

where Δ_k represents the intra-cluster distance (in our case maximal pairwise Euclidean distance between the objects in the cluster). The Dunn index also favours clusterings with low intra-cluster and high inter-cluster distances, although the compactness of the clusters is assessed in a different way. The Dunn index should be maximized.

The experimental results have been reported in Table 1. Although as mentioned before the quality measures used should favour the K-means algorithm, as the idea behind it is to find compact, well separated clusters, it is not the case. As it can be seen, direct optimization of the clustering quality measures, which is the approach postulated in this paper, is vastly beneficial in terms of both DB and DI_m, for both datasets.

Table 1. Dunn and Davies-Bouldin index optimization

Clustering method	**IRIS**				**WINE**			
	DI_m		DB		DI_m		DB	
	c	**index**	c	**index**	c	**index**	c	**index**
'Ground truth'	3	0.3815	3	0.8635	3	0.3757	3	1.1985
K-means clustering[a]	2	0.7120	2	0.4595	2	0.5202	3	1.0589
EM-clustering[b]	7	0.5296	3	0.7563	6	0.4407	3	1.1411
Direct optimization with GA[c]	2	0.8142	2	0.4390	2	0.5412	3	**1.0381**
Direct optimization with PSO[d]	2	**0.8875**	2	**0.4385**	2	**0.5689**	3	**1.0381**

[a] PRTools [9] implementation has been used with the maximum number of iterations (i.e. random selections of the initial cluster centres) set to 1000; the experiment has been repeated 100 times and the best result has been reported; default values of the remaining parameters have been used.

[b] PRTools implementation has been used; the experiment has been repeated 100 times and the best result has been reported; default values of the remaining parameters have been used.

[c] Built-in MATLAB implementation has been used; the population size has been set to 1000; default values of the remaining parameters have been used.

[d] Particle Swarm Optimization Toolbox [2] for MATLAB has been used; the population size has been set to 1000; default values of the remaining parameters have been used.

The high values of the Dunn index and low values of the Davies-Bouldin index are out of reach of the standard clustering methods, with the Particle Swarm

Optimization (PSO) algorithm being an absolute leader here. Intuitively, for both GA and PSO this should come at the price of increased computational requirements as the number of times the optimization criterion alone had to be calculated is orders of magnitude higher than in the case of standard clustering methods. As it turns out, the computational time needed to calculate the criteria was negligible and in fact the running time of GA was comparable to that of EM-clustering, mostly due to the exhaustive search strategy for the optimal cluster count that had to be employed in the latter case. For the PSO, the clustering has been performed in about $70 - 80\%$ of the time required by GA. Although we recognize that some running time difference might be a result of differences in implementations, standard library versions have been used to alleviate this effect as much as possible.

An interesting thing to note are the values of DB and DI_m for the 'ground truth' i.e. the situation when the clusters correspond with the classes given with both datasets. Regardless of the clustering algorithm used, in all cases the obtained clusterings are scored better than the original classes in the data. This means that the original classes do not admit to the notions of compactness and separation, as defined by the two criteria used. In other words if one is after discovering the original classes in the data, neither DB nor DI_m is an appropriate choice in this case, as apparently neither of them is able to reflect the true class structure. This confirms the importance of choosing a proper, application-specific clustering criterion as postulated in this study.

7 Conclusions

In this paper we have proposed a unified view of clustering as an optimization problem. We have investigated the importance of conscious selection of an objective function, which is appropriate in the context of a particular application and should be used not only for assessing the clustering quality but also as an optimization criterion. We have also presented a generalized clustering framework and provided a proof-of-concept, which validated the above ideas.

During this study a number of challenges has been identified. The most notable of these challenges are:

1. Identification of exact objective functions optimized by popular clustering algorithms as well as properties of the clustering results produced. As mentioned before, it is not clear what objective functions are optimized by many popular clustering algorithms or if it is even possible to express those in a closed form. This kind of knowledge would however allow to better understand the characteristics of problems for which a given clustering algorithm is most suitable.
2. Development of advanced heuristics to constrain the search space and overcome high computational requirements of general-purpose optimization methods. The size of the search space when viewing clustering as an optimization problem is immense. On the other hand it is also immense in many other problems yet there are techniques which successfully deal with this issue. In

case of clustering, good performance of PSO proves that optimization is a viable approach and we believe that there still is a lot of room for improvement.
3. Development of new objective functions and clustering quality measures fitting different types of applications, together with optimization approaches dedicated for these objective functions. Although there is no 'one-fits-all' solution, there certainly exist groups of problems which have similar properties. Identification of these groups and development of tailored clustering objective functions is an interesting challenge.

We hope that this study, which is a first step towards designing of a new breed of clustering approaches will stimulate and inspire further research in this exciting area of data mining.

Acknowledgments

The research leading to these results has received funding from the EU 7^{th} Framework Programme (FP7/2007-2013) under grant agreement no. 251617.

References

1. A. Asuncion and D. Newman. UCI Machine Learning Repository, 2007.
2. B. Birge. PSOt – a particle swarm optimization toolbox for use with Matlab. *Proceedings of the 2003 IEEE Swarm Intelligence Symposium SIS03 Cat No03EX706*, pages 182–186, 2003.
3. M. Budka and B. Gabrys. Correntropy-based density-preserving data sampling as an alternative to standard cross-validation. In *Neural Networks (IJCNN), The 2010 International Joint Conference on*, pages 1–8, july 2010.
4. M. Budka and B. Gabrys. Ridge regression ensemble for toxicity prediction. *Procedia Computer Science*, 1(1):193–201, 2010.
5. D. Davies and D. Bouldin. A cluster separation measure. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (2):224–227, 1979.
6. A. Dempster, N. Laird, and D. Rubin. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977.
7. R. Dubes. How many clusters are best?-an experiment. *Pattern Recognition*, 20(6):645–663, 1987.
8. R. Duda, P. Hart, and D. Stork. *Pattern Classification 2nd ed.* John Wiley & Sons, New York, USA, 2001.
9. R. Duin, P. Juszczak, P. Paclik, E. Pekalska, D. de Ridder, D. Tax, and S. Verzakov. PR–Tools 4.1, A MATLAB Toolbox for Pattern Recognition, 2007. http://prtools.org.
10. J. Dunn. Well-separated clusters and optimal fuzzy partitions. *Journal of cybernetics*, 4(1):95–104, 1974.
11. R. Fletcher. *Practical methods of optimization, 2nd ed.* Wiley, 2000.
12. A. Fraser. Simulation of genetic systems by automatic digital computers vi. epistasis. *Australian Journal of Biological Sciences*, 13(2):150–162, 1960.

13. R. Hamming. Error detecting and error correcting codes. *Bell System technical journal*, 29(2):147–160, 1950.
14. P. Jaccard. *Etude comparative de la distribution florale dans une portion des Alpes et du Jura*. 1901.
15. A. Jain, M. Murty, and P. Flynn. Data clustering: a review. *ACM computing surveys (CSUR)*, 31(3):264–323, 1999.
16. R. Jenssen, D. Erdogmus, K. Hild, J. Principe, and T. Eltoft. Optimizing the Cauchy–Schwarz PDF distance for information theoretic, non–parametric clustering. In A. Rangarajan, B. Vemurl, and A. Yuille, editors, *Energy Minimization Methods in Computer Vision and Pattern Recognition*, volume 3257 of *Lecture Notes in Computer Science*, pages 34–45. Springer, 2005.
17. J. Kennedy and R. Eberhart. Particle swarm optimization. In *Neural Networks, 1995. Proceedings., IEEE International Conference on*, volume 4, pages 1942–1948. IEEE, 1995.
18. R. Kohavi, A. Deng, B. Frasca, R. Longbotham, T. Walker, and Y. Xu. Trustworthy online controlled experiments: Five puzzling outcomes explained. In *KDD 2012 Aug 12-16, 2012, Beijing China*, 2012.
19. J. MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, page 14. California, USA, 1967.
20. R. Sibson. Slink: an optimally efficient algorithm for the single-link cluster method. *The Computer Journal*, 16(1):30–34, 1973.