The Instance Easiness of Supervised Learning for Cluster Validity

Vladimir Estivill-Castro[*]

Griffith University, QLD 4111, Australia
v.estivill-castro@griffith.edu.au
http://vladestivill-castro.net

Abstract. “The statistical problem of testing cluster validity is essentially unsolved” [5]. We translate the issue of gaining credibility on the output of un-supervised learning algorithms to the supervised learning case. We introduce a notion of instance easiness to supervised learning and link the validity of a clustering to how its output constitutes an easy instance for supervised learning. Our notion of instance easiness for supervised learning extends the notion of stability to perturbations (used earlier for measuring clusterability in the un-supervised setting). We follow the axiomatic and generic formulations for cluster-quality measures. As a result, we inform the trust we can place in a clustering result using standard validity methods for supervised learning, like cross validation.

Keywords: Cluster validity, Supervized Learning, Instance easiness.

1 Introduction

From its very beginning, the field of knowledge discovery and data mining considered validity as a core property of any outcome. The supervised case assumes a function $c(\boldsymbol{x})$ and attempt to fit a model $F(\boldsymbol{x})$ given the training set (or data points) $\{(\boldsymbol{x}_i, c(\boldsymbol{x}_i))\}_{i=1,\ldots,n}$. One can evaluate the quality of the fit with many solid alternatives [7,13]. However, in the un-supervised setting we are only presented with the set of cases $\{\boldsymbol{x}_i\}_{i=1,\ldots,n}$. Most likely we are performing such learning with no solid grounds for what is the actual (real-world) generator of these examples and any assumption on our part may actually constitute a far too large unjustified bias. What in fact constitutes learning and what is the goal?

How can we establish some confidence (or “credibility” in the language of Witten and Frank [13, Chapter 5]) on the result delivered by a clustering algorithm? This constitutes a fundamental question. The very well known distance-based clustering algorithm k-means is among the top 10 most used algorithms in knowledge discovery and data mining applications [14], however it is statistically inconsistent and statistically biased (converging to biased means even if the input is generated from k spherical multi-variate normal distributions with equal proportions). How do the users of such a method derive any trust in their results? Or in the credibility of any other clustering methods?

[*] Work performed while hosted by Universitat Popeu Fabra, Barcelona, Spain.

L. Cao et al. (Eds.): PAKDD 2011 Workshops, LNAI 7104, pp. 197–208, 2012.

This paper proposes two new measures of cluster quality. The fundamental idea is that no matter what clustering algorithms is used, in the end one desires to obtain a model that can accurately answer the question "are $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ in the same cluster?" When clusterings are partitions, this questions has only two disjoint answers (yes or no), and thus, the results of a clustering algorithm can be scrutinized by the facility by which supervised learning algorithms can discover a suitable classifier. We show that these two measures have mathematical properties that have been considered desirable by several authors. The measures are inspired by the intuition that if the clustering results does identify classes that are well-separated, these results constitute an easy problem in the supervized-learning sense. This implies formalizing a notion of "instance easiness". We measure how easy is to learn in the supervized-learning case by using a similar approach or easiness previously introduced for unsupervised learning. That is, we will draw on the notions of "clusterability" [2] to suggest the mechanisms to achieve this.

2 Instance Easiness

In the machine learning literature, instance easiness has been applied with the notion of *clusterability* [2] to the un-supervised learning (or clustering) case. That is, Ackerman and Ben-David introduced notions to measure how easy is to cluster a particular instance X into k clusters.

2.1 Generic Definitions

We now introduce formal definitions and nomenclature for the clustering problem (un-supervised learning) that follow the general formulations of Ackerman and Ben-David [1] since this is a generic form that is widely applicable.

Let X be some domain set (usually finite). A function $d : X \times X \rightarrow \Re$ is a *distance function* over X if

1. $d(\boldsymbol{x}_i, \boldsymbol{x}_i) \geq 0$ for all $\boldsymbol{x}_i \in X$,
2. for any $\boldsymbol{x}_i, \boldsymbol{x}_j \in X$, $d(\boldsymbol{x}_i, x_j) > 0$ if and only if $\boldsymbol{x}_i \neq \boldsymbol{x}_j$, and
3. for any $\boldsymbol{x}_i, \boldsymbol{x}_j \in X$, $d(\boldsymbol{x}_i, \boldsymbol{x}_j) = d(\boldsymbol{x}_j, \boldsymbol{x}_i)$ (symmetry).

Note that a distance function is more general than a metric; because the triangle inequality is not required[1].

A *k-clustering* of X is a k-partition, $C = \{C_1, C_2, \ldots, C_k\}$. That is, $\bigcup_{j=1}^{k} C_i = X$, $C_j \neq \emptyset$, for all $j \in \{1, \ldots, k\}$; and $C_i \cap C_j = \emptyset$ for all $i \neq j$. A *clustering* of X is a k-clustering of X for some $k \geq 1$. A clustering is *trivial* if $|C_j| = 1$ for all $j \in \{1, \ldots, k\}$ or $k = 1$. For $\boldsymbol{x}_i, \boldsymbol{x}_j \in X$ and a clustering C of X, we write $\boldsymbol{x}_i \sim_C \boldsymbol{x}_j$ if $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are in the same cluster of clustering C, and we write $\boldsymbol{x}_i \not\sim_C \boldsymbol{x}_j$ if they are in different clusters.

[1] Most authors prefer to call these functions *dissimilarities* and use *distance* as synonym to *metric*, but here we keep this earlier use of *distance* so our notation follows closely the notation in the clustering case [1,2].

A *clustering function* for some domain set X is a function (algorithm) that takes as inputs a distance function d over X, and produces as output a clustering of X. Typically, such clustering function is an algorithm that attempts to obtain the optimum of a loss function that formalizes some induction principle. In this form, a clustering problem is an optimization problem. For example, if we minimize the total squared error when we chose a set of k representatives in an Euclidean space (that is, d is the Euclidean metric *Eucl*) we obtain the problem that k-means attempts heuristically to solve. Given X with $|X| = n$ and $k > 1$, minimize $ErrorSQ_{Eucl}(R) = \sum_{i=1}^{n}[Eucl(\boldsymbol{x}_i, rep[\boldsymbol{x}_i, R])]^2$ where R is a set of k representatives and $rep[\boldsymbol{x}_i, R]$ is the nearest representative to $\boldsymbol{x}_i$ in R. This problem can be solved exactly by first enumerating all k-clusterings of X, and by taking the mean of each cluster as a representative and finally by evaluating the loss. However, the number of k-clusterings corresponds to the Stirling numbers of the second kind, and this approach has complexity at least exponential in n. A discrete version is usually referred as medoids where we also require that $R \subseteq X$. In this case, the problem remains NP-hard as it reduces to the p-median problem; however, we can now exhaustively test all subsets $R \subset X$ with $|R| = k$. The complexity of this exhaustive search algorithm is now at least proportional to $\binom{n}{k}$. This approach would have complexity $O(n^{k+1})$ and would now be polynomial in n. While Ackerman and Ben-David [1,2] refer to this as "polynomial" for some of their easiness results alluded earlier, it is perhaps more appropriate to refer to it as *polynomial in n for each fixed k* and thus our use of quotation marks (this class is also known as the class *XP*).

2.2 Instance Easiness for Supervised Learning

We introduce here a notion of instance easiness for the supervised learning problem. To the best of our knowledge, this is the first use of *instance easiness* applied to supervised learning. It also will be the building block for our presentation of cluster-quality measures. Our approach follows the notion of stability to perturbation of a problem (this approach was used for the unsupervised case by Ackerman and Ben-David [2]). Consider an instance of the supervised learning problem given by

1. a set of n pairs $\{(\boldsymbol{x}_i, c(\boldsymbol{x}_i))\}_{i=1}^{n}$, where $X = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ is the training set of labeled examples, and Y is a finite[2] set of labels (thus, $c(\boldsymbol{x}_i) \in Y$),
2. a family of models $\mathcal{F}$, so that if $F \in \mathcal{F}$, then $F : X \to Y$, and
3. a real valued loss function $\mathcal{L}$.

The goal is to find $F_O \in \mathcal{F}$ that optimizes the loss function. For brevity we will just write $[X, Y]$ for an instance of the supervised learning problem. For example, the family of models could be all artificial neural networks with a certain number of layers and neurons per layer and the loss function could be the total squared

[2] In this paper we consider $|Y| \in \mathbb{N}$ and small. Thus we focus on classification and not on interpolation/regression.

error. Then, back-propagation can be seen as a gradient-descent approach to the corresponding optimization problem.

We also make some generic observations of what we require of a loss function. First, the loss function in the supervised learning setting is a function of the instance $[X, Y]$ and the classifier $F : X \rightarrow Y$, thus, we write $\mathcal{L}([X, Y], F)$. We expect F to always be a mathematical function (and not a relation). In the supervised learning setting, the instance $[X, Y]$ is typically formed by data points for a mathematical function c (that for each element in the domain, associates one and no more than one element in the codomain). However, in practice, it is not unusual to have contradictory examples; that is, it is not uncommon for data sets derived from practice to have two (or more) contradictory examples $(\boldsymbol{x}_i, c)$ and $(\boldsymbol{x}_i, c')$ with $c \neq c'$. Nevertheless, what we will require is that the loss function cannot be oblivious to the requirement that the classifier be a function in the following sense. Given an instance $[X, Y]$, at least for every classifier F_O that optimizes the loss function $\mathcal{L}([X, Y], F)$ the optimal value $\mathcal{L}([X, Y], F_O)$ cannot be the same to $\mathcal{L}([X', Y], F_O)$ when X' is the same as X except that X' contains one or more additional contradictory examples (perturbations can cause more contradictory examples, and that cannot improve the loss).

What we propose is that, if there is a distance function d over X, then we can consider an instance of supervised learning as easy if small perturbations of the set X result also in small perturbations of the loss. More formally, we say that two set X and X' are *ϵ-close* (with respect to a distance function d over $X \cup X'$) if there is a bijection[3] $\pi : X \rightarrow X'$ so that $d(\boldsymbol{x}_i, \pi(\boldsymbol{x}_i)) \leq \epsilon$, for all $i = 1, \ldots, n$. With these concepts we introduce our first fundamental definition.

Definition 1. *Let $[X, Y]$ and $[X', Y']$ be two instances of the supervised learning problem, we say they are* ϵ-close *if*

1. *$Y' \subseteq Y$ (no new classes are introduced),*
2. *X and X' are ϵ-close, and $c(\boldsymbol{x}_i) = c(\pi(\boldsymbol{x}_i))$ where $\pi : X \rightarrow X'$ provides the ϵ-closeness.*

That is, the training sets are ϵ-close and there are no more class labels.

Now, let $OPT_{\mathcal{L}}(X, Y)$ be the optimum value of the loss function $\mathcal{L}$ for the instance $[X, Y]$; that is, $OPT_{\mathcal{L}}(X, Y) = \min\{\mathcal{L}([X, Y], F) \mid F \in \mathcal{F}\} = \mathcal{L}(F_O)$.

Definition 2. *We say that the instance $[X, Y]$ is (ϵ, δ)-easy if*

1. *there is $F_0 : X \rightarrow Y$ a classifier that optimizes the loss, and*
2. *for all instances $[X', Y]$ that are ϵ-close to $[X, Y]$, we have*

$$\mathcal{L}([X', Y], F_0) \leq (1 + \delta) OPT_{\mathcal{L}}(X, Y).$$

The loss does not depend on any distance function on X. We also assume that the loss is based on the categorical/nominal nature of the set Y, and thus the loss value does not change if we rename the classes with any one-to-one function. We say such loss functions are isomorphism-invariant. Common loss functions are isomorphism-equivalent; that is, they do not depend on the name of the classes.

[3] Originally ϵ-closeness was defined [2] with a one-to-one mapping, but we ensure the relation "X is ϵ-close to X'" is symmetric, but this is not necessary for what follows.

2.3 Illustration

We here illustrate the concepts introduced earlier. For visualization purposes, we consider a two-dimensional data set, and for simplicity, we assume we are using a clustering algorithm like k-means and, because of some intuition, we are seeking two clusters[4], i.e. $k = 2$. The data set in Fig. 1 consist of 4 normal distributions in a mixture with equal proportions (1/4). The respective means $\boldsymbol{\mu}_1 = (0, 0)$, $\boldsymbol{\mu}_2 = (20, 20)$, $\boldsymbol{\mu}_2 = (-25, 25)$, $\boldsymbol{\mu}_2 = (-5, 45)$. All have diagonal covariance matrices and all elements of the diagonal are equal to 10. While this data is not challenging for k-means, there are at least two local minima for the clustering loss function. Therefore, depending on its random initialization, k-means produces two clusterings. One with centers $M = \{(-15, 35), (10, 9)\}$ and another with centers $M' = \{(-12, 11), (7, 33)\}$. We used WEKA's SimpleKMeans [8] and the first set is obtained with 7 iterations on average (see Fig. 1) while the second one required 29 iteration on average (see Fig. 1).

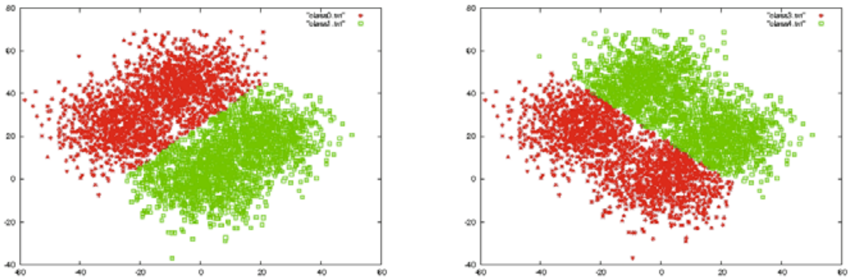

Fig. 1. Data with 4,000 points and two clusterings resulting from centers M and M'

For each of this clusterings we can draft a supervised learning problem. We argue that the corresponding supervised learning problem that results from this two clusters are different in terms of how easy they are. Note however, that Fig. 1 illustrates the corresponding supervised learning problems; and therefore, they correspond to linearly separable classes (if fact, k-means classes are a Voronoi partition of the universe and therefore, always separable by k-hyperplanes). This suggest that k-means always produces what can be regarded as an "easy" supervised-learning problem (we would expect linear discriminant, support vector machines, CART and many classifiers to do very well).

We suggest that if the clusters do reflect genuine structure and we have discovered meaningful classes, the corresponding job of using these concepts for learning a classifier should be "easier", and we measure "easier" by how stable the supervised learning instance is to perturbations. Obtaining very accurate classifiers for the two supervised learning problems in Fig. 1 can be achieved with the simplest of WEKA's algorithms. Using WEKA's stratified cross-validation to

[4] Although here we know the ground-truth, in a practical clustering exercise we would not know much about the data and identifying the value of k would be part of the challenge; one approach is to test if there are clusters by evaluating $k = 1$ vs. $k = 2$.

evaluate accuracy, NaiveBayes achieves 99.7% accuracy, only misclassifying 11 instances for M while it achieves 99.1% accuracy, only misclassifying 36 instances for M'. Similarly, WEKA's NNge (Nearest-neighbor-like algorithm using non-nested generalized exemplars which are hyper-rectangles that can be viewed as if-then rules) achieves 99.2% accuracy, only misclassifying 32 instances for M while it achieves 99.2% accuracy, only misclassifying 33 instances for M'. However, if one takes the data set in Fig. 1 and associates all those points that have one or more negative coordinates to one class and those points that have both positive coordinates to another class, we obtain a supervised learning problem Problem: N $= x_1 > 0 \wedge x_2 > 0$ that is also "easy" because the two classifiers above can also obtain high accuracy. In fact, (also evaluated by WEKA's stratified cross-validation) NNge achieves 99.95% accuracy only missing 2 instances and NaiveBayes achieves 91% accuracy only missing 361 instances.

However, clearly the last supervised learning problem is not the result of a good clustering. What we do now is keep those classifiers learned with the unperturbed data, and use perturbed data as test data. We see that those classifiers that come from less quality clusterings degrade their accuracy more rapidly in proportion to the perturbation. We perturbed the 3 supervised learning problems by adding a uniformly distributed random number in $[-1, 1]$ to the attributes (but the class remains untouched). Now in M, NaiveBayes is 99.5% accurate (19 errors on average), NNge is 99.6% accurate (14 errors on average). In M' NaiveBayes is 99% accurate (40 errors on average), NNge is 99% accurate

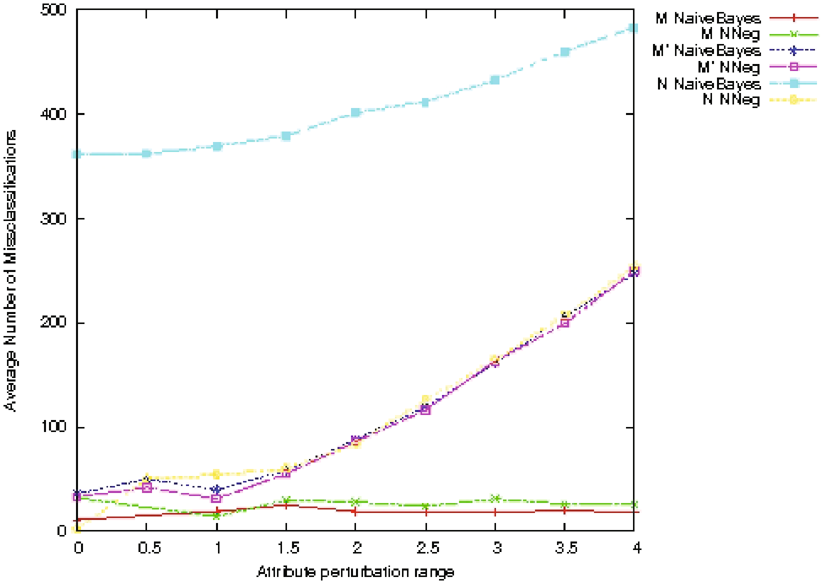

Fig. 2. Deterioration of performance of classifiers as perturbations on the attributes is larger. But classifiers derived from good clusters preserve accuracy.

(26 errors on average), In PROBLEM N: $x_1 > 0 \wedge x_2 > 0$ NAIVEBAYES is 90% accurate (369 errors on average), NNGE is 99% accurate (44 errors on average),

Fig. 2 shows points (ϵ, y) where misclassification rate y in the test-set corresponds to data that has suffered a perturbations with uniformly distributed random noise is $[-\epsilon, \epsilon]$. The behavior (effects on the loss) is essentially independent of the classifier, with the two classifiers for M remaining stable to perturbation. However, for the other supervised learning problems, the misclassification rate rapidly grows (classification accuracy deteriorates).

In fact, while the growth rate is important, we will be interested on the stability. That is, we focus on the largest $\epsilon > 0$ where the problem remains easy since in this region the loss suffers small impact. In Fig. 1, this correspond to how far right does the misclassification line remain flat before it starts to increase.

This experimental observation will be formalized in the next section to construct measures of cluster quality. Note that PROBLEM N has a simple boundary but not a good clustering. Although M' is a good clustering, the M clustering is the best because the groups are essentially the clouds at $(0, 0)$ and $(20, 20)$ on one side and the clouds at $(-25, 25)$ and $(-5, 45)$ on the other. These pairing has more separation that the paring by M'.

We emphasize that this example is mainly for illustration purposes. By no reasonable standard data of 4 normal distributions with equal cylindrical covariance in two dimensions and equal proportions corresponds to a challenging clustering exercise. In fact, with 4, 000 points, it hardly corresponds to a challenging data mining setting. However, we believe this illustrates our point further. The most widely used clustering algorithm (k-means) even on this data set (which is supposed to be suitable for k-means since clusters are spherical and separated) can provide wrong answers. Clearly, part of the problem is the inappropriate value $k = 2$. Can the loss function for k-means indicate the better clustering between M and M'? This example is so simple that such is the case. For example, WEKA standard evaluation with SIMPLEKMEANS indicates a better loss function for M than for M'. But this example is for illustration only.

2.4 The Clustering-Quality Measure

A *clustering-quality measure* (*CQM*) is a function that is given a clustering C over (X, d) (where d is a distance function over X) and returns a non-negative real number. Many proposals of clustering-quality measures have been suggested for providing some confidence (or ensuring validity) of the results of a clustering algorithm. Before we introduce two new *CQM*, we need a bit of notation. If d is a distance function over X and $\lambda > 0$ with $\lambda \in \Re$, then the *λ-scaled version of d* is $d' = (\lambda d)$ and is defined by $(\lambda d)(\boldsymbol{x}_i, \boldsymbol{x}_j) = \lambda \cdot d(\boldsymbol{x}_i, \boldsymbol{x}_j)$. If d is a distance function over X, the *normalized* version of d is denoted by $nor(d)$ and is defined as the $1/\lambda$-scaled version of d when $\lambda = \max\{d(\boldsymbol{x}_i, \boldsymbol{x}_j) \mid \boldsymbol{x}_i, \boldsymbol{x}_j \in X\}$; that is, $nor(d) = (d/\lambda) = (d/\max\{d(\boldsymbol{x}_i, \boldsymbol{x}_j) \mid \boldsymbol{x}_i, \boldsymbol{x}_j \in X\})$.

Definition 3. *Given a clustering $C = \{C_1, \ldots, C_k\}$ of (X, d), the* CQM by classification *m_c is the largest $\epsilon > 0$ so that, if we construct a supervised learning* *instance* $[X, C]$ *derived from the clustering by* $Y = C$ *and* $c(\boldsymbol{x}_i) = C_j$ *such that* $\boldsymbol{x}_i \in C_j$*, then* $[X, C]$ *is* $(\epsilon, 0)$*-easy with respect to nor*(d).

Definition 4. *Given a clustering* $C = \{C_1, \ldots, C_k\}$ *of* (X, d)*, the* CQM by pairing m_p *is the largest* $\epsilon > 0$ *so that, if we construct a supervised learning instance* $[X \times X, \{\text{Yes}, \text{No}\}]$ *derived from the clustering* C *by*

$$c(\boldsymbol{x}_i, \boldsymbol{x}_j) = \begin{cases} \text{Yes} & \textit{if } \boldsymbol{x}_i \sim_C \boldsymbol{x}_j \\ \text{No} & \textit{if } \boldsymbol{x}_i \not\sim_C \boldsymbol{x}_j, \end{cases}$$

then the instance $[X \times X, \{\text{Yes}, \text{No}\}]$ *is* $(\epsilon, 0)$*-easy with respect to nor*(d).

We are now in a position to prove the four (4) properties required by Ackerman and Ben-David of a *CQM*. These properties were inspired by the 3 axioms suggested by Kleinberg [9]. Kleinberg proved that although the axioms are desirable of all clustering functions, they were inconsistent. This is usually interpreted as the impossibility of defining what clustering is. However Ackerman and Ben-David properties are sound, thus suggesting it is feasible to describe what is good clustering. Scale invariance means that the output is invariant to uniform scaling of the input.

Definition 5. Scale invariance: *A* CQM m *satisfies* scale invariance *if* $\forall C$ *a clustering of* (X, d) *and* $\lambda > 0$*, we have* $m[C, X, d] = m[C, X, (\lambda d)]$.

Lemma 1. *On a bounded study region, The* CQM m_c *and the* CQM m_p *satisfy invariance.*

Since $nor(d) = nor(\lambda d)$ for all $\lambda > 0$ and the definitions of m_p and m_c use the normalized version of d the lemma follows. Thus, we now assume all distance functions are normalized to the largest ball that includes the study region.

For the next property we need to introduce the notion of isomorphic clusters, denoted $C \approx_d C'$. A distance-preserving isomorphism $\phi : X \to X$ satisfies that $\forall \boldsymbol{x}_i, \boldsymbol{x}_j \in X$, $d(\boldsymbol{x}_i, \boldsymbol{x}_j) = d(\phi(\boldsymbol{x}_i), \phi(\boldsymbol{x}_j))$. Two clusters C and C' of the same domain (X, d) are *isomorphic* if there exists a distance-preserving isomorphism such that $\forall \boldsymbol{x}_i, \boldsymbol{x}_j \in X$, we have $\boldsymbol{x}_i \sim_C \boldsymbol{x}_j$ if and only if $\phi(\boldsymbol{x}_i) \sim_{C'} \phi(\boldsymbol{x}_j)$.

Definition 6. Invariant under isomorphism: *A* CQM *is* invariant under isomorphism *(*isomorphism-invariant*) if* $\forall C, C'$ *non-trivial clusterings over* (X, d) *where* $C \approx_d C'$*, we have* $m[C, X, d] = m[C', X, d]$.

Lemma 2. *The* CQM m_c *and* m_p *are isomorphism-invariant.*

Definition 7. Richness: *A* CQM *satisfies* richness *if for each non-trivial partition* C *of* X *there is a distance function* $\hat{d}$ *such that* C *maximizes* $m[C, X, \hat{d}]$ *when considered as a function of* C.

Lemma 3. *The measures* m_c *and* m_p *satisfy richness.*

The final property of a *CQM* is *consistency.* Given a clustering C over (X, d) (that is, d is a distance function over X), we say that another distance function d' over X is a C*-consistent variant* of d if

1. $d'(\boldsymbol{x}_i, \boldsymbol{x}_j) \leq d(\boldsymbol{x}_i, \boldsymbol{x}_j)\ \forall \boldsymbol{x}_i \sim_C \boldsymbol{x}_j$, and
2. $d'(\boldsymbol{x}_i, \boldsymbol{x}_j) \geq d(\boldsymbol{x}_i, \boldsymbol{x}_j)\ \forall \boldsymbol{x}_i \not\sim_C \boldsymbol{x}_j$.

Definition 8. Consistency: *A* CQM *satisfies* consistency *if* $\forall$ *C a clustering of (X, d) and d' that is C-consistent variant of d, we have $m[C, X, d'] \geq m[C, X, d]$.*

Lemma 4. *The* CQM *m_c and the* CQM *m_p satisfy consistency.*

3 Discussion

We have introduced two *CQM* and mathematically demonstrated fundamental properties that have several favorable implications. First, *CQM* that satisfy richness, scale invariance, consistency and isomorphism-invariance can be combined to produce new measures with also these properties [1]. Thus, we have not only enriched the set of *CQM* since m_c and m_p become generators to produce *CQM.*

Secondly, the methods to verify accuracy in supervised learning are now well established and many strong and solid implementations exist (like WEKA [8]). Therefore, the issue of cluster quality can now be simplified as we did in the earlier illustration. Before the proposal here, it is not surprising to find statements like: "Evaluation of clusterers is not as comprehensive as the evaluation of classifiers. Since clustering is unsupervised, it is a lot harder determining how *good* a model is" [4]. Our proposal here shows that the machinery for evaluating supervised learning can be useful to tackle the issue of cluster validity without the need of already classified (supervised) instances. We should aim for cluster validity methods that are as close as possible to the "comprehensive" landscape we have for supervised learning. Our proposal here suggests this direction.

Our proposal is applicable to the issue of alternative clusterings. The outputs from these algorithms are several alternative clusterings, because the data-miner believes there may be several meaningful ways to create such clusterings [3]. While this approach needs to resolve the issue of cluster similarity or dissimilarity (as in external cluster validity), it is also guided by a measure of cluster quality. That is, the approach also needs to provide some credibility for each of the multiple answers provided to an unsupervised learning problem.

Traditionally, cluster validity has taken three avenues: *internal cluster validity*, *external cluster validity* [7], and *experimental cluster validity.* A comprehensive discussion of the issues and challenges with each appears elsewhere. Approaches to cluster validity since then continue along these lines. But, typically there is an admission that evaluating a model built from a clustering algorithm is challenging [7]. Proposals like comparing a matrix of two clusterings [7] still face many problems, and lead to the challenges of measures of similarity between clusters. In the handbook of Data Mining and Knowledge Discovery, Chapter 15 [11] has a discussion of clustering methods, and some material on cluster evaluation and validity. M. Halkidi and M. Vazirgiannis [7, Chapter 30] also offer a discussion of cluster validity. The fact of the matter all remain variations of earlier methods. Our approach is perhaps most similar to other experimental approaches [6,15] which have justification in the intuition that the boundary of clusters should show sparsity. Since implicitly, support vector machines offer to find margins that are as wide as possible, one can hypothesize about outliers and boundaries and filter them out [6]. If removing few boundary items and repeating the clustering passes an external validity test that shows the clustering is robust to this change, then we can raise our confidence that the clustering has good quality. Otherwise the clustering is suspicious. A similar idea is derived from mathematical properties of proximity-graphs [15]. Here as well, a candidate clustering can be polished on the boundaries of clusters simply by removing those data points that the proximity graph suggests are on fringes of cluster. If repeating the alternation of polishing and clustering offers stable clusters (clusters do not change with respect to some external clustering validity measure), trust in the clustering is raised.

Both of these mechanisms [6,15] have a notion of robustness, and the foundations are derived from proximity structure reflected in the clusterer (clustering function) itself. However, they are computationally costly as clustering needs to be repeated, external cluster-validity functions need to be computed and the test can only be one of similarity between pairs of clusters. Our approach here is mathematically more formal, and it is easy to implement by the availability of supervised learning techniques and their implementations.

Consensus clustering or ensemble of clusters [10,12,16] is an extension of these earlier ideas [6,15] of agreement between clusterings. Although initially proposed for problems in bio-informatics, the concept seems quite natural, since in fact, many clustering algorithms will produce different clusters if initialized with different parameters. So, the same clustering approach leads to multiple answers. Proponents of consensus clustering argue it is sensible to produce a clustering that maximizes the agreement (a similar idea occurs with multiple classifiers or a classifier ensemble).

Our approach here also enables to give some assessment of the clustering participating in the ensemble as well as the resulting combined clustering. Our approach does not need to deal with the issues of cluster similarity. But, we illustrate we can apply our approach to consensus clustering with the data set made available by Dr. A. Strehl `x8d5k.txt` [12]. This is a mixture of 8 non-symmetrical Gaussians in 8 dimensions. The data consist of 1000 points, and the `Original` cluster labels from the mixture are provided. These original labels provide 200 data points from each cluster. Also, 5 clusterings `V1`, `V2`, `V3`, `V4`, and `V5` are provided and the consensus clustering (`Combined`) of these five is the 6-th clustering. It corresponds to "the best known labels in terms of average normalized mutual information (ANMI)". We applied our approach to these 7 clustering and present our results in a similar way to our earlier illustration in Fig. 3. Because we have the `Original` clustering (sometimes referred as the *true* clustering), we can see that the alternative clusters are in fact weaker that the truth. However, the `Combined` cluster is extremely satisfactory and our approach shows that it is essentially equivalent for our *CQM* to the `Original`. These conclusions are also in agreement if the supervised learner is WEKA's `NaiveBayes` or `NNge`. This is what we would expect.

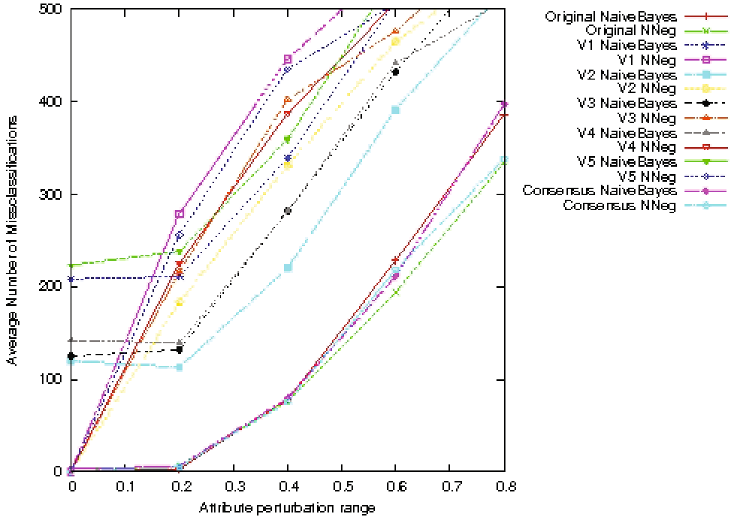

Fig. 3. Deterioration of performance of classifiers as perturbations on the attributes are larger (data set is `x8d5k.txt`[12]. But classifiers derived from true clusters and consensus cluster preserve accuracy.

4 Summary

Despite decades of research the fundamental issue of clustering validity has remained unsolved. So much so that widely accepted software like WEKA has minimal tools and resources for it. This contrasts with the large set of tools for validity for the supervised learning case. This paper enables to use the set of tools for the supervised case in the unsupervised case.

The intuition of our work is a simple idea. When we have to discover groups (classes) in a data set where we have no information regarding this, whatever results must be assessed for validity. A clustering algorithm's output must be evaluated and external validity approaches are out of the question, since if we had knowledge of the *true* clustering, why would be trying to find it? However, we would expect that the classes obtained by the clustering function are in some way separable and constitute meaningful concepts. They should be robust to small perturbations. A classifier obtained from corresponding supervised learning result should have performance that degrades rather slowly when presented with data that is close. Such data can be obtained by perturbations and then the robustness of the classifier measured by the now standard approaches of supervised learning.

We have provided illustrations that this idea is manifested in practical clustering scenarios including consensus clustering. We have also provided theoretical foundations by formalizing a notion of instance easiness for supervised clustering and then deriving measures of cluster quality. We have shown that these measures satisfy the generic properties of richness, scale invariance, isomorphism-invariance and consistency that are common to many measures (however, some of these other measures are very costly to compute). Thus, our approach enables a practical and theoretical useful mix.

References

1. Ackerman, M., Ben-David, S.: Measures of clustering quality: A working set of axioms for clustering. In: Advances in Neural Information Processing Systems 22 NIPS, Proceedings of the Twenty-Second Annual Conference on Neural Information Processing Systems, pp. 121–128. MIT Press, Vancouver (2008)
2. Ackerman, M., Ben-David, S.: Clusterability: A theoretical study. In: Proceedings of the Twelfth Int. Conf. on Artificial Intelligence and Statistics AISTATS, Clearwater Beach, Florida, USA, vol. 5, JMLR:W&CP (2009)
3. Bae, E., Bailey, J.: Coala: A novel approach for the extraction of an alternate clustering of high quality and high dissimilarity. In: Proceedings of the 6th IEEE Int. Conf. on Data Mining (ICDM), pp. 53–62. IEEE Computer Soc. (2006)
4. Bouckaert, R.R., Frank, E., Hall, M., Kirkby, R., Reutemann, P., Seewald, A., Scuse, D.: WEKA Manual for Version 3-6-2. The University of Waikato (2010)
5. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification, 2nd edn. John Wiley & Sons, NY (2001)
6. Estivill-Castro, V., Yang, J.: Cluster Validity using Support Vector Machines. In: Kambayashi, Y., Mohania, M., Wöß, W. (eds.) DaWaK 2003. LNCS, vol. 2737, pp. 244–256. Springer, Heidelberg (2003)
7. Halkidi, M., Vazirgiannis, M.: Chapter 30 — quality assessment approaches in data mining. In: The Data Mining and Knowledge Discovery Handbook, pp. 661–696. Springer, Heidelberg (2005)
8. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA data mining software: an update. SIGKDD Explorations 11(1), 10–18 (2009)
9. Kleinberg, J.: An impossibility theorem for clustering. In: The 16th conference on Neural Information Processing Systems (NIPS), pp. 446–453. MIT Press (2002)
10. Monti, S., Tamayo, P., Mesirov, J., Golub, T.: Consensus clustering: A resampling-based method for class discovery and visualization of gene expression microarray data. Machine Learning 52(1-2), 91–118 (2003)
11. Rokach, L., Maimon, O.: Chapter 15 — clustering methods. In: The Data Mining and Knowledge Discovery Handbook, pp. 321–352. Springer, Heidelberg (2005)
12. Strehl, A., Ghosh, J.: Cluster ensembles – a knowledge reuse framework for combining multiple partitions. J. on Machine Learning Research 3, 583–617 (2002)
13. Witten, I., Frank, E.: Data Mining — Practical Machine Learning Tools and Technologies with JAVA implementations (2000)
14. Wu, X., et al.: Top 10 algorithms in data mining. Knowledge and Information Systems 14(1), 1–37 (2008)
15. Yang, J., Lee, I.: Cluster validity through graph-based boundary analysis. In: Int. Conf. on Information and Knowledge Engineering, IKE, pp. 204–210. CSREA Press (2004)
16. Yu, Z., Wong, H.-S., Wang, H.: Graph-based consensus clustering for class discovery from gene expression data. Bioinformatics 23(21), 288–2896 (2007)