Dataset comparison workflows

Marko Robnik-Šikonja

Faculty of Computer and Information Science,
University of Ljubljana,
Večna pot 113, 1001 Ljubljana, Slovenia
Email: Marko.Robnik@fri.uni-lj.si

Abstract: To assess similarity of two datasets from the point of view of data science, univariate statistical comparisons are mostly insufficient. We present a methodology which estimates similarity of datasets from the point of view of data mining tasks. For example, we provide a relevant information for a decision if a new/related dataset can be used with an existing supervised or unsupervised model or not. We propose several workflows which cover: (a) statistical properties of generated data; (b) distance based structural similarity and (c) predictive similarity of two datasets. We evaluate the proposed workflows on random splits of several datasets and by comparing original datasets with datasets produced by a generator of semi-artificial data. The results show that the proposed workflows can reveal relevant similarity information about datasets needed in many data mining scenarios.

Keywords: data analytics; data mining; machine learning; data similarity; clustering; classification.

Reference to this paper should be made as follows: Robnik-Šikonja, M. (2018) 'Dataset comparison workflows', *Int. J. Data Science*, Vol. 3, No. 2, pp.126–145.

Biographical notes: Marko Robnik-Sikonja is an Associate Professor of Computer Science and Informatics at Faculty of Computer and Information Science, University of Ljubljana. He received his PhD in Computer Science and Informatics from University of Ljubljana, Faculty of Computer and Information Sciences in 2001. His research interests include artificial intelligence, machine learning, text and data mining, as well as their applications. He is coauthor of approximately 60 scientific publications which received over 2000 citations, and three data analysis software packages in R.

1 Introduction

In data science practitioners often address problems incorporating multiple complex datasets with different characteristics and even within one dataset differences between subsets of data might be considerable. Reasons why subsets could be different are various. For example,

they could be collected in different company's departments, possibly located in different parts of the world, they could be extracted from different databases, created at different times, captured with different types of sensors, etc. An important question is whether these data (sub)sets are similar enough to be used in a common data analytics task, for example in prediction, development of a new analytical method or in tuning parameters of a given model. As there are practically no tools available to answer these relevant questions, the questions are frequently swept under the carpet. We describe an approach to empirically obtain an important information about the similarity of datasets in the context of data mining tasks. Our toolkit consists of three workflows containing several similarity indicators, incorporated into the R package semiartificial (Robnik-Šikonja, 2014b). The workflows compare statistical properties of datasets and analyse dataset similarity using clustering and classification. Below we shortly introduce their principles.

The question whether two datasets can be used (interchangeably) in the same data mining task is not the same question as whether (parts of) datasets come from the same underlying probability distribution. For many data (sub)sets used in practical analytical tasks the statistical tests assessing differences in the distributions would likely return negative results, nevertheless these data (sub)sets can be used together in e.g., classification, as data mining methods are designed to be robust and generalise the differences. Similarity/equality of probability distributions is therefore just a part of information we shall provide. Whether two samples are drawn from the same probability distribution is a well-researched topic for low dimensional datasets and is covered in textbooks e.g., Venables and Ripley (2002). For example, to compare two one-dimensional numeric data samples we can apply a statistical test such as the Kolmogorov-Smirnov test or, for nominal data, we can compute distance between two discrete distributions using Hellinger distance or Kullback-Leibler divergence. To estimate a similarity of distributions for two higher dimensional datasets no practically useful mechanism exist, but if any of their marginal, lower dimensional, distributions are different, we can conclude that the datasets are not drawn from the same distribution. The same reasoning goes for central moments of the distributions, which shall also be similar. We use the central moments in the workflow presented in Section 3.

In unsupervised setting we are interested if two data (sub)sets produce similar clustering, i.e., if their distance based structure is similar. Simple as it may look, this is not the question answered by clustering comparison methods (for example adjusted rand index (ARI)). These methods take two clusterings obtained on *the same dataset* and compare membership of instances in both clusterings. In our case we want to compare clusterings obtained on *two different datasets* which do not have common instances so direct application of clustering comparison methods is not possible. In Section 4, we propose an original method to first transform the datasets and then apply standard clustering similarity measures to get a similarity estimate.

Classification is one of the most frequently used tasks in data mining. In the context of dataset similarity, we want to establish if two datasets produce comparable classification models. If the answer is positive, we could use the datasets interchangeably or jointly in classification tasks. This can help us to estimate parameters of prediction models and is of great importance if instances of one dataset are easier to obtain than instances of the other dataset. In this case, the auxiliary dataset can save us valuable resources and/or improve predictive performance. Our original workflow for prediction based similarity is presented in Section 5.

A need for effective dataset comparison methodology is evident form related work, where several useful applications are described. Our motivation for development of dataset

comparison methodology lies in generators of semi-artificial data, where it is crucial to know if the generated datasets are suitable replacements for original data. As semi-artificial data can be used in several scenarios (Robnik-Šikonja, 2014a), it is important to provide a toolbox which can reveal many possible similarities and differences between datasets.

The rest of the paper is divided into six sections. In Section 2 we review related work. In Sections 3–5, we present the three workflows: statistical, clustering, and predictive, respectively. We empirically evaluate all three dataset comparison workflows in Section 6. We first use random splits of several UCI datasets to establish initial expectations of what the workflows return for similar datasets and to show validity of the workflows. In the second stage we test the similarity of original and generated datasets using the semi-artificial dataset generator (Robnik-Šikonja, 2014b). In Section 7 we conclude with the overview of the work and ideas for improvements of the proposed comparison workflows.

2 Related work

Similarity is one of the central notions in learning and generalisation. There exist many similarity measures to compare individual objects (numbers, vectors, strings, graphs etc.). For example in the context of recommender systems, Morozov and Saiedian (2013) use cosine similarity and Pearson's correlation coefficient to compare individual instances. The focus of this paper is not on assessing similarity of individual instances, but on similarity of entire datasets with a purpose to use them interchangeably in data mining tasks.

In statistics a question whether two samples come from the same probability distribution is an important and well-researched topic. For low dimensional datasets this question can be answered by statistical tests comparing the underlying (unknown) distributions, see e.g., Venables and Ripley (2002). Examples of such tests are Kolmogorov–Smirnov test, suitable for one dimensional numerical datasets, and χ^2-goodness of fit test for nominal data. For nominal data we can also compute the distance between two discrete distributions using Hellinger distance or Kullback–Leibler divergence.

In the context of association rule learning the attribute similarity between different databases was investigated in Das et al. (1998). The motivation was to find items with similar purchase patterns and to use clustering to form hierarchies of similar products. Authors define the external similarity of two attributes as similarity of projections of the attributes to a hand-selected subset of other attributes with the motivation that two items are similar if they are purchased together with the same set of items. Subramonian (1998) describes association rules based probability density estimation to detect when two datasets are significantly different and uses the approach to test if clustering of new data is different from clustering of old data. For efficiency reasons implementation of similarity in both Das et al. (1998) and Subramonian (1998) remains on the level of individual attributes. Parthasarathy and Ogihara (2000) use association rules to compare similarity of datasets. The definition of similarity uses the number of cooccurring association rules produced from two datasets. The potential application is in distributed computing, where similar datasets shall be distributed to the same processing node in order to get relevant rules or clustering for the assigned data. In Anwar et al. (2012) similar similarity measure is applied to datasets describing software measurements and defectiveness of software modules. Related to association rules, which try to find rules with sufficient support and confidence, contrasts sets are groups of instances with statistically significant differences. Bay and Pazzani (2001) present an algorithm for discovery and description of such groups. This line of work was later extended to subgroup

discovery algorithms (Lavrač et al., 2004), which try to find rules describing subsets of the population that are sufficiently large and statistically unusual. Association rule based similarity is comparable to our third workflow (similarity based on classification), but we do not use the structure of models (i.e. association rules) but the models' predictions, which means that any decision model (including association rules) can be used in our workflow.

In Sequeira and Zaki (2007), the motivation to find unusually similar (or dissimilar) datasets is to apply successful policies from one domain to the other. The compared datasets may be, for example, geographically distributed or collected at different times. The approach first discretises attributes and generates subspaces for each dataset separately. The constructed subspaces become nodes in the graph and several similarity measures are defined to weight the edges between nodes. Finally, several graph similarity algorithms are tested to compute similarities of the two constructed graphs and to detect matching subspaces. The approach was tested on artificial data and a high dimensional datasets from bioinformatics domain, where matching subspaces could potentially be exploited for discovery of gene functions. This approach tries to capture structural similarity of the graphs and weakly resembles our second approach (clustering similarity), but differs in a way how constructed subspaces are compared, as we do need explicit identification of matching clusters. Our approach also does not rely on any specific structural similarity metric (e.g., similarity of connections or components in the graph), but uses clustering and classification directly.

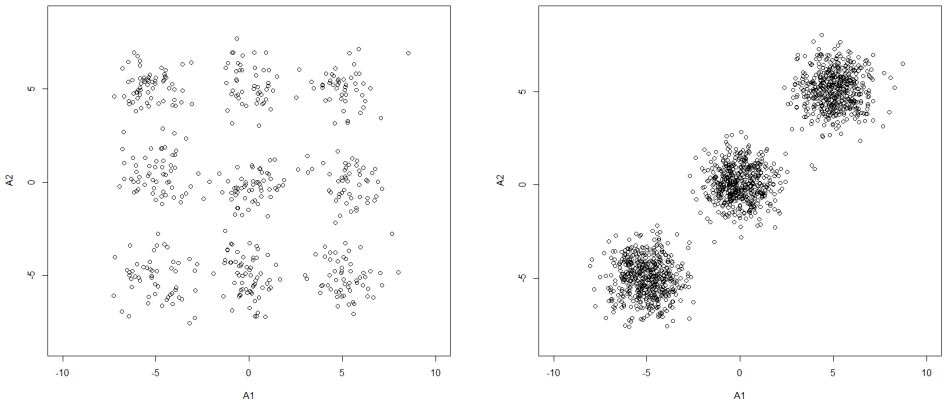
Ganti et al. (1999) present FOCUS framework for comparison of datasets based on association rules, clustering, or decision tree models induced from the data. The framework uses these models to partition the problem space of both compared datasets. The created partitions are merged (i.e., overlayed) in a process called greatest common refinement and then measures such as misclassification rate or statistical goodness of fit are used to compare matching partitions and aggregate the scores. The FOCUS can also focus on regions of interest defined by the user and aggregate scores only there. This was essential in one of the presented application scenarios where the differences between two datasets were interactively explored, similarly to drill-down and roll-up strategies in OLAP (online analytical processing) databases. Another application was analysis of required dataset size, i.e., how many instances are required to obtain predictive performance comparable to the full dataset. The approach used in this paper is conceptually simpler than in FOCUS and is presented in the form of workflows. Our clustering similarity workflow does not form matching partitions by overlaying two clusterings, but compares the originally produced clusterings by filling in instances from the compared datasets. By this our approach avoids matching errors. Also our predictive similarity workflow is not limited to matching regions and explicit space partitioning models, therefore we can use any classification model which is an important advantage as decision trees used in FOCUS are not competitive with state-of-the-art classifiers (Caruana and Niculescu-Mizil, 2006).

3 Statistical comparison

A commonly used approach to compare similarity of two low dimensional datasets which is also a part of our system is to compare statistics based on central moments: mean, median, standard deviation, skewness, and kurtosis. These statistics are computed separately for each attribute and then compared for matching attributes from both datasets. While these statistics offer an important introspection into the datasets, they are insufficient for datasets with several attributes. Visualisation of several statistics for each of attributes is

impractical and comparisons are difficult when the number of attributes increases (e.g., to more than 5). Another shortcoming of using these statistics is that they cannot provide a global view of the data. They are computed for each attribute separately and do not take possible interactions between attributes into account. The problem of attribute interactions is illustrated in Figure 1. The mean and median of attributes A_1 and A_2 are the same for left- and right-hand graph, but the datasets are quite different. The standard statistics also do not convey any information about similarity or interchangeability of datasets for typical data mining tasks. This issue is addressed in Sections 3 and 4.

Figure 1 Scatter plots of two dataset with the same mean and median for attributes A_1 and A_2

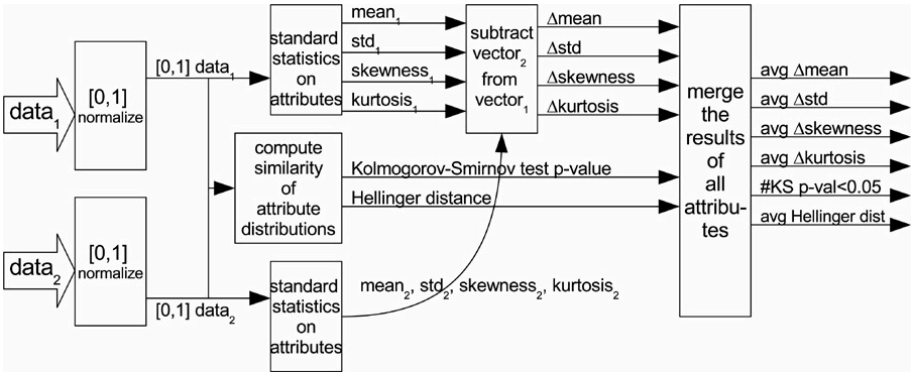


Standard, central moments based, skewness and kurtosis, frequently denoted with γ_1 and γ_2, have several drawbacks. They require symmetric distribution for interpretation, are sensitive to outliers, and their values are unrestricted (i.e., their span is from $-\infty$ to ∞). These properties make skewness and kurtosis difficult to summarise and compare across several attributes. We therefore provide also their robust variants proposed in Brys et al. (2006): medcouple (MC) and left/right medcouple (L/RMC). Nevertheless, the user is advised to check the values of skewness and kurtosis indicators with some knowledge of the domain.

The workflow of the whole process is illustrated in Figure 2. We normalise each numeric attribute to $[0, 1]$ to make comparison across attributes sensible. While value ranges do convey important information, they cannot be taken into account for comparisons across several attributes. Take for example medical records, where the scales of body temperature, mass, blood pressure, and cholesterol are very different and comparing differences in absolute numbers would not make sense. Normalisation of values is a common approach taken in machine learning for such cases, for example in attribute evaluation.

Input to each comparison workflow are two datasets. Comparison of attributes' statistics computed on both datasets is tedious, especially for datasets with large number of attributes. We therefore first compute the standard statistics on attributes and then subtract the statistics of the second dataset from statistics of the first dataset. To summarise the results we report only an verage difference for each of the statistics.

Figure 2 The workflow for comparing standard statistics between two datasets



To compare distributions of attribute values we use Hellinger distance for discrete attributes and Kolmogorov-Smirnov (KS) test for numerical attributes. The Hellinger distance between two discrete univariate distributions $P = (p_1, \ldots, p_k)$ and $Q = (q_1, \ldots, q_k)$ is defined as

$$H(P,Q) = \frac{1}{\sqrt{2}}\sqrt{\sum_{i=1}^{k}(\sqrt{p_i} - \sqrt{q_i})^2}.$$

The maximal Hellinger distance between two distributions is 1. There are several other measures comparing similarity of two univariate distributions, for example Kullback-Leibler distance, Bhattacharyya distance, and Jensen–Shannon distance. As they are all strongly related to Hellinger distance we do not report their scores in this work. For numerical attributes we use two sample KS test, which tests if two one-dimensional probability distributions differ. The test uses maximal difference between two empirical cumulative distribution functions $F_1(x, n_1)$ and $F_2(x, n_2)$, on samples of size n_1 and n_2, respectively:

$$D(n_1, n_2) = \sup_x |F_1(x, n_1) - F_2(x, n_2)|.$$

To get an overall picture we report only the average Hellinger distance over all discrete attributes, and percentage of numeric attributes for which p-value of KS test is below 0.05 using the null hypothesis that attributes' values in both datasets are drawn form the same distribution. For both Hellinger distance and KS test smaller values indicate greater similarity. The averages reported in the evaluation (Section 6) illustrate the univariate similarity of datasets.

4 Comparing clusterings

Clustering partitions the input instances into similar groups called clusters. The goal of clustering algorithms is to find such a partition of objects that objects in the same cluster are more similar to each other than objects in different clusters. The information about clustering offers an important introspection into the structure of the data. As no clustering algorithm is superior to others in all circumstances, the cluster validation measures asses

how well a given partitioning fits the structure underlying the data (Arbelaitz et al., 2013). Two main groups of cluster validation approaches exists, internal and external, based on availability of information in the validation process. The internal validation uses only the partition data to establish how appropriate the given partitioning is for a specified purpose. The idea of internal measures is to asses the balance between cluster compactness and between-cluster separation, where several definitions of compactness and separation exist. The external validation evaluates the degree of matching between two competing clusterings, mostly comparing the obtained clustering to a ground truth. From clustering validity research it is known that no clustering similarity index can outperform all the others in all scenarios (Jaskowiak et al., 2016), and in fact, many clustering similarity indexes are strongly correlated, in particular when they are adjusted for chance (Vinh et al., 2009; Jaskowiak et al., 2016).

Our aim is not to compare two clusterings on the same dataset, as both internal and external clustering validity measures do, but we want to use clusterings produced by one clustering method on two different datasets to asses similarity of these two datasets. The internal validation approaches are therefore not appropriate for our purpose, and even external validation measures are not directly applicable. In Section 4.2, we describe our novel approach which preprocess the two datasets so that standard external clustering comparison metrics (presented below) can be applied.

4.1 Similarity of two clusterings on the same dataset

Starting from the dataset $D = \{(\mathbf{x}_i)_{i=1}^n\}$ with n data points, we assume two different clusterings of D, namely $U = \{U_1, U_2, \ldots, U_u\}$ and $V = \{V_1, V_2, \ldots, V_v\}$, where $U_1 \cap U_2 \cap \cdots \cap U_u = \emptyset$, $U_1 \cup U_2 \cup \cdots \cup U_u = D$, $V_1 \cap V_2 \cap \cdots \cap V_v = \emptyset$, $V_1 \cup V_2 \cup \cdots \cup V_v = D$. The information about overlap between clusters of U and V can be expressed with $u \times v$ contingency table as shown in Table 1.

Table 1 The contingency table of clusterings overlap, $n_{i,j} = |U_i \cap V_j|$

U/V	V_1	V_2	$\ldots$	V_v	*Sum*
U_1	$n_{1,1}$	$n_{1,2}$	$\ldots$	$n_{1,v}$	u_1
U_2	$n_{2,1}$	$n_{2,2}$	$\ldots$	$n_{2,v}$	u_2
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
U_u	$n_{u,1}$	$n_{u,2}$	$\ldots$	$n_{u,v}$	u_u
Sum	v_1	v_2	$\ldots$	v_v	n

There are several measures comparing clusterings based on counting the pairs of points on which two clusterings agree or disagree (Vinh et al., 2009). Any pair of data points from the total of $\binom{n}{2}$ distinct pairs in D falls into one of the following four categories.

- N_{11}, the number of pairs that are in the same cluster in both U and V
- N_{00}, the number of pairs that are in different clusters in both U and V
- N_{01}, the number of pairs that are in the same cluster in U but in different clusters in V
- N_{10}, the number of pairs that are in different clusters in U but in the same cluster in V.

The values N_{11}, N_{00}, N_{01}, and N_{10} can be computed from contingency table (Hubert and Arabie, 1985). Values N_{11} and N_{00} indicate agreement between clusterings U and V, while values N_{01} and N_{10} indicate disagreement between U and V. The rand index (RI) (Rand, 1971) is defined as

$$RI(U,V) = \frac{N_{00} + N11}{\binom{n}{2}}.$$

The RI lies between 0 and 1 and takes the value 1 when the two clusterings are identical, and the value 0 when no pair of points appear either in the same cluster or different clusters in both U and V. It is desirable that a clustering similarity indicator would take value close to zero for two random clusterings, which is not true for RI. The ARI (Hubert and Arabie, 1985) fixes this by using generalised hypergeometric distribution as a model of randomness and computes expected number of entries in the contingency table. It is defined as

$$\begin{aligned} \text{ARI} &= \frac{RI - E[RI]}{\max RI - E[RI]} \\ &= \frac{\sum_{i=1}^{u}\sum_{j=1}^{v}\binom{n_{i,j}}{2} - [\sum_{i=1}^{u}\binom{u_i}{2}\sum_{j=1}^{v}\binom{v_j}{2}]/\binom{n}{2}}{\frac{1}{2}[\sum_{i=1}^{u}\binom{u_i}{2} + \sum_{j=1}^{v}\binom{v_j}{2}] - [\sum_{i=1}^{u}\binom{u_i}{2}\sum_{j=1}^{v}\binom{v_j}{2}]/\binom{n}{2}}. \end{aligned} \tag{1}$$

The ARI has expected value of 0 for random distribution of clusters and value 1 for perfectly matching clusterings. ARI can also be negative.

Similar approach is taken in Fowlkes and Mallows (1983), who originally designed a measure to compare hierarchical clusterings. The Fowlkes-Mallows (FM) clustering similarity index is defined as:

$$FM = \sqrt{\frac{N_{11}}{N_{11}+N_{01}} \cdot \frac{N_{11}}{N_{11}+N_{10}}}. \tag{2}$$

The FM is 1 for perfectly matching clusterings and approaches zero for random clusterings.

Another frequently used similarity criterion is Jaccard index (Jaccard, 1901), defined as

$$J = \frac{N_{11}}{N_{11}+N_{01}+N_{10}}. \tag{3}$$

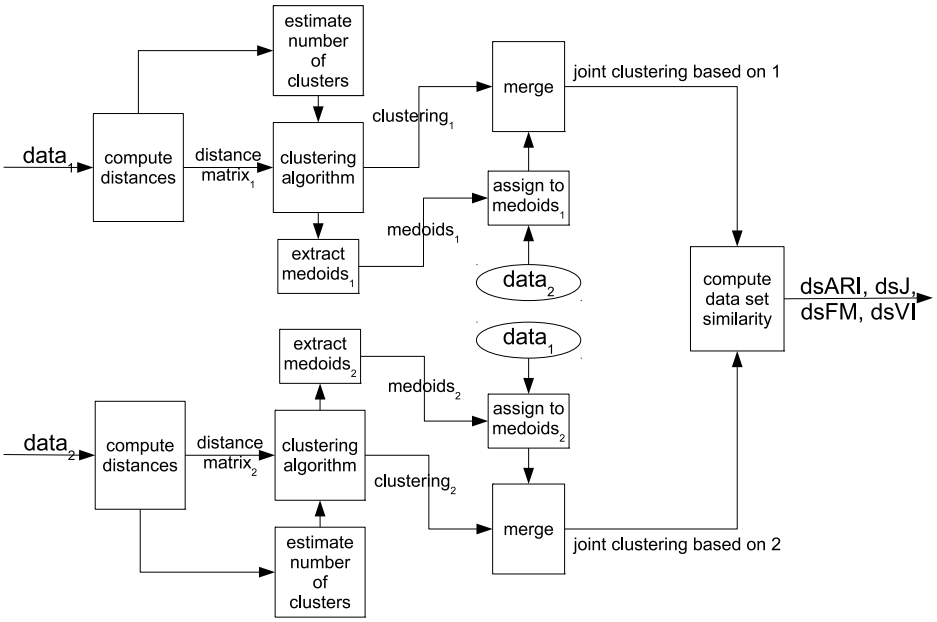
Somewhat different approach is taken in variation of information (VI) index (Meilă, 2007), where membership of an instance in a cluster is viewed as a random variable and then mutual information between two clusterings is computed as the information one clustering contains about the other. The VI measures the amount of information lost and gained in changing from one clustering to the other.

Following Arbelaitz et al. (2013), who tested validity of 30 internal clustering validity measures using ARI, Jaccard and VI index to compare produced clusterings to the predefined ground truth, we computed the same three measures and additionally the FM index. In our dataset comparison scenario the similarity results based on ARI, FM, Jaccard, and VI turned out to be strongly correlated (Pearson's correlation coefficient between ARI and FM, Jaccard, and VI is 0.96, 0.97, and –0.91, respectively), so we report only ARI scores.

4.2 A workflow for comparing clusterings on two datasets

Standard external clustering similarity measures (like ARI, FM, J, and VI) compare two different clusterings on the same set of instances, while we want to compare similarity of two different sets of instances. To overcome this obstacle, we first cluster both datasets separately and extract medoids of the clusters for each clustering. The medoid of a cluster is an existing instance in the cluster whose average similarity to all instances in the cluster is maximal. For each instance in the first dataset, we find the nearest medoid in the second clustering and assign it to that cluster, thereby getting a joint clustering of both datasets based on the cluster structure of the second dataset. We repeat the analogous procedure for the second dataset and get a joint clustering based on the first dataset. These two joint clusterings are defined on the same set of instances (i.e., union of both datasets), therefore we can use standard external clustering similarity measures to assess similarity of the clusterings and compare structure of both datasets. The workflow of cluster based comparison of two datasets is presented in Figure 3.

Figure 3 The workflow for comparing two datasets based on clustering similarity



As we need to assign new instances to existing clustering we, as a matter of convenience, use partitioning around medoids (PAM) clustering algorithm (Kaufman and Rousseeuw, 1990), which, besides partitions, outputs also medoids. Other clustering algorithms can be used, for example Jin et al. (2014), Reddy and Jana (2014) and Ashour and Fyfe (2014), but in this case centres of clusters have to be computed separately. Distance to the medoids is the criterion we use to assign new instances to existing clusters. PAM clustering is implemented in R package cluster (Maechler et al., 2013). To use it we first computed distances between instances of each dataset using Gower's method (Gower, 1971). The method normalises numeric attributes to $[0, 1]$ and uses 0-1 scoring of dissimilarity between nominal attributes

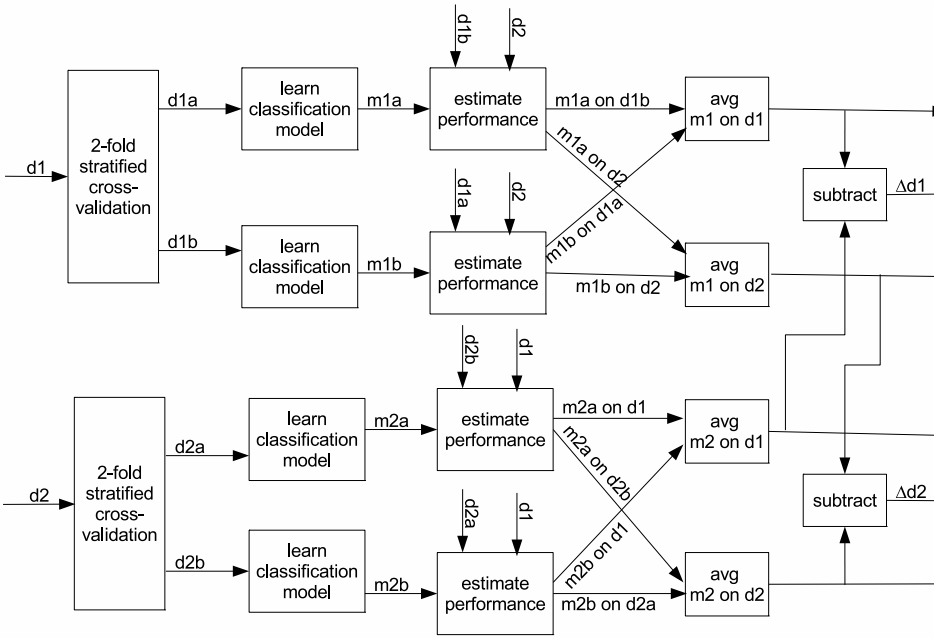
(0 for the same, 1 for different categories). The distance is a sum of dissimilarities over all attributes.

The number of clusters is set to an estimated optimal value separately for both compared datasets. Several heuristics exist to estimate the optimal number of clusters. We selected the optimum average silhouette width method, implemented in R package fpc (Hennig, 2013). The silhouette width method is an internal cluster validity measure which returns the best result in many contexts according to the large scale comparison of clustering validity indices conducted in Arbelaitz et al. (2013). Once instances from one datasets are assigned to the compared clustering and vice-versa, standard measures ARI, J, and FM are computed with R package flexclust (Leisch, 2006) and VI is computed with mcclust package (Fritsch, 2012).

5 Comparing classifications

The classification is probably the most important task in machine learning and data mining. As such the judgement weather two datasets can be used together or one dataset can be a substitute for the other in classification is an important question even without general interest in dataset similarity. We propose a dataset similarity workflow based on classification performance shown in Figure 4.

Figure 4 The workflow for comparing two datasets based on classification performance



The basic idea is to train classification models on both datasets. The two trained models are tested on yet unseen data from both datasets and the performances are compared. If the performance of a model trained on the first dataset is comparable for the first and the second dataset, this is an indication that the second data is within the first dataset distribution (i.e.,

there are not many outliers and all the aspects of the first dataset are captured). In this case the second dataset can, for example, represent a good substitute for the first dataset concerning machine learning and data mining tasks. If the model trained on the first dataset achieves better performance on the second dataset than on the first, this indicates that the second dataset may be a simplified version of the first one and may not contain all peculiarities of the first dataset. The analogous reasoning can be applied for the model trained on the second dataset.

In our workflow (Figure 4) we start with two datasets d1 and d2 (the first and the second one) and split them randomly but stratified into two halves (d1 produces d1a and d1b, d2 is split into d2a and d2b). Each of the four splits is used to train a classifier, and we name the resulting models m1a, m1b, m2a, and m2b, respectively. We evaluate the performance of these models on data unseen during training, so m1a is tested on d1b and d2, m1b is tested on d1a and d2, m2a uses d2b and d1, and m2b uses d2a and d1 as the testing set. Each test produces a performance score (e.g., classification accuracy, AUC, (area under the ROC curve i.e., area under the receiver operating characteristic curve), misclassification cost, etc.) which we average as in a 2-fold cross-validation to get the following estimates:

- performance of m1 on d1 (model built on first dataset and tested on the first dataset) is the average performance of m1a on d1b and m1b on d1a
- performance of m1 on d2 (classifier built on first dataset and tested on the second dataset) is the average performance of m1a on d2 and m1b on d2
- performance of m2 on d2 (model built on the second dataset and tested on the second dataset) is the average performance of m2a on d2b and m2b on d2a
- performance of m2 on d1 (model built on second dataset and tested on the first dataset) is the average performance of m2a on d1 and m2b on d1.

These estimates already convey an important information as discussed above, but we can subtract performances on the same dataset e.g., of m1 on d1 and m2 on d1 (models built on first and second dataset, both tested on the first dataset) to get Δd1 (and analogously Δd2). This differences, in our opinion, are the most important indicators how suitable is the second dataset as a replacement for the first dataset in the development of classification methods. If, in ideal case, this differences would be close to zero, the second dataset would be good substitute for possibly small first dataset, and we can expect that performance of methods developed using the second dataset will be comparable when used on the first dataset.

6 Evaluation

We want to verify if the proposed comparison methods return results which are consistent with our expectations. We also try to determine working conditions of the comparison methods: on which datasets they work and where they fail.

To evaluate the comparison workflows, we perform an empirical evaluation using 15 datasets from UCI (University of California Irvine) repository (Bache and Lichman, 2014) with great variability in the number of attributes, types of attributes and number of class values. We use R package readMLData (Savicky, 2012) which provides a uniform interface for manipulation of UCI datasets. To provide sufficient data to make comparisons statistically reliable but to keep the computational load low, we limit our study to datasets

with the number of instances between 500 and 1000. Taking these conditions into account we extracted 15 classification datasets from a collection of 92 datasets kindly provided by the author of the package readMLData. The characteristics of these datasets are presented in Table 2.

Table 2 The characteristics of datasets used. The column are: n – number of instances, a – number of attributes, numeric – number of numeric attributes, discrete – number of discrete attributes, v/a – average number of values per discrete attribute, C – number of class values, majority % – proportion of majority class in percent, missing % – percentage of missing values

Dataset	n	a	*Numeric*	*Discrete*	*v/a*	C	*Majority* (%)	*Missing* (%)
Annealing	898	38	6	32	2.4	5	76.2	0.00
Balance-scale	625	4	4	0	0.0	3	46.1	0.00
Breast-cancer-WDBC	569	30	30	0	0.0	2	62.7	0.00
Breast-cancer-Wisconsin	699	9	9	0	0.0	2	65.5	0.25
Credit-screening	690	15	6	9	4.4	2	55.5	0.64
Cylinder-bands	540	37	20	17	8.7	2	57.8	5.00
PIMA-Indians-diabetes	768	8	8	0	0.0	2	65.1	0.00
Soybean-large	683	35	0	35	2.8	19	13.5	9.77
Spectrometer	531	101	101	0	0.0	48	10.4	0.00
Statlog-Australian	690	14	6	8	4.5	2	55.5	0.00
Statlog-German	1000	20	7	13	4.2	2	70.0	0.00
Statlog-German-numeric	1000	24	24	0	0.0	2	70.0	0.00
Statlog-vehicle	846	18	18	0	0.0	4	25.8	0.00
Tic-tac-toe	958	9	0	9	3.0	2	65.3	0.00
Vowel-context	990	10	10	0	0.0	11	09.1	0.00

With the first set of experiments, described in Section 6.1, we want to test the validity of the proposed workflows, so the workflows compare random subsets of datasets in Table 2. We expect that random subset are highly similar, subject to variation by chance. The second set of experiments, described in Section 6.2, puts the proposed methodology into practical use and we compare similarity of original datasets from Table 2 with datasets returned by generator of semi-artificial data. An efficient and reliable dataset similarity evaluation is of crucial importance for usability of semi-artificial data, as it establishes users' confidence in the generated data.

6.1 Comparing random subsets

We first check if comparison workflows return expected results. We randomly split each dataset into two equal parts and pronounce them data1 and data2. We expect that produced pairs of datasets will be highly similar and that no significant differences will be found by the three workflows described in Sections 3–5. The results of comparisons are presented in Table 3. Before commenting on the results, we first give details of computed scores.

- *Statistics of attributes*: We compare standard statistics of numeric attributes (see workflow in Figure 2) – the mean, standard deviation, skewness, and kurtosis. The interpretations of skewness and kurtosis is difficult and relevant only per each dataset

and attribute separately, so we exclude skewness and kurtosis from the summary comparisons in this section. For numeric attributes we compute p-values of KS tests under the null hypothesis that attribute values from both compared datasets are drawn from the same distribution. We report the percentage of numeric attributes where this hypothesis was rejected at 0.05 level (lower value reported indicates higher similarity). For discrete attributes we compare similarity of value distributions using Hellinger distance. We report average Hellinger distance over all discrete attributes in each dataset.

- *Clustering*: The structure of two datasets is compared with k-medoids clustering computed separately on data1 and data2. In both cases the instances from dataset unused in clustering are assigned to produced partitions based on their nearest medoids. In these way each clustering contains all the instances from both data1 and data2 and we can use ARI (equation (1)) as presented in workflow on Figure 3 to asses clustering similarity. The class variables are excluded from the datasets. For many datasets computed ARI scores exhibit high variance, so we report the average ARI over 100 repetitions (each repetition randomly splits the data into two parts, named data1 and data2).
- *Classification performance*: We compare the predictive similarity of the two datasets using classification accuracy of random forests as illustrated in the workflow on Figure 4. We selected random forests algorithm to build prediction models due to its robust performance under various conditions (Verikas et al., 2011). The implementation used comes from R package CORElearn (Robnik-Šikonja and Savicky, 2015). The default parameters are used: we built 100 random trees with the number of randomly selected attributes in nodes set to square root of the number of attributes. We report 5×2 cross-validated performances of models trained and tested on both datasets.

The results of all three dataset similarity workflows are presented in Table 3. The column labelled with $=$ gives the percentage of instances exactly equal in both datasets. This happens only in datasets which contain several copies of the same instance: annealing has 5%, breast-cancer-Wisconsin has 20.2%, and soybean-large has 9.4% of replicas in two random halves.

Columns labelled $\overline{\Delta mean}$ and $\overline{\Delta std}$ report average differences in mean and standard deviation for pairs of attributes normalised to $[0, 1]$. In all cases the differences are below 0.05, which shows that moments of distributions for individual attributes in both subsets are close. The distributions of individual attributes are compared with KS test for numeric attributes and with Hellinger distance for discrete attributes. Column labelled KSp gives the percentage of p-values below 0.05 in KS tests comparing numeric attributes using the null hypothesis that pairs of attributes from two halves of datasets are drawn from the same distribution. For 9 out of 13 datasets the KS-test detects not a single difference in distributions, while for remaining datasets small differences are detected. Column labelled $\overline{H}$ presents average Hellinger distance for matching discrete attributes. For all datasets the distances are low (below 0.10).

The similarity of datasets may be a perquisite in development, simulation, and benchmarking. We evaluate this type of similarity via clustering and classification workflows. The column labelled ARI presents ARI. We can observe that the clustering similarity is considerable for many datasets (ARI > 0.5 for 8 out of 15 datasets) but there

are also some datasets where ARI is low (for 4 datasets the ARI is below 0.2). The numbers given are averages of 100 repetitions of workflow from Section 4 using random splits of dataset into data1 and data2. Given that datasets compared are random splits of the same dataset, we can conclude that in general clustering based similarity can be unreliable. This is not surprising given that most clustering algorithms are approximations for NP-hard (non-deterministic polynomial-time hard) problems and tend to find local minima of their objective functions. As stated in Ben-David et al. (2006) stability of clustering algorithms is fully determined by the behaviour of the objective function: if it has a unique global minimiser, the algorithm is stable, otherwise it is unstable. Taking this into account and knowing that in practice clustering algorithms strongly depend on the set of instances and initial choice of clusters, we recommend to treat ARI scores with caution: high ARI score is a sign that compared datasets are indeed similar while low ARI score may not be a sign that they are dissimilar. In the light of theoretical result of Ben-David et al. (2006) the reliability of this workflow could be improved if we find the right objective function i.e., the right clustering algorithm for each dataset. Such challenge is beyond the scope of this work.

Table 3 The comparison of two randomly selected subsets of data. The columns are: $=$ – proportion of second dataset instances exactly equal to first dataset instances, $\overline{\Delta\mathrm{mean}}$ – average difference in means for normalised numeric attributes, $\overline{\Delta\mathrm{std}}$ – average difference in standard deviation for normalised numeric attributes, KSp – percentage of p-values below 5% in KS tests comparing numeric attributes, $\overline{H}$ – average Hellinger distance for discrete attributes, ARI – Adjusted Rand index, mXdY – classification accuracy in percents for model trained on data $\times$ and tested on data Y. The dash means that given comparison is not applicable to the dataset

Dataset	$=$	$\overline{\Delta mean}$	$\overline{\Delta std}$	KSp	$\overline{H}$	*ARI*	*m1d1*	*m1d2*	*m2d1*	*m2d2*
Annealing	5.0	0.01	0.01	0.0	0.02	0.859	96	97	96	97
Balance-scale	0.0	–0.01	0.00	0.0	–	0.181	83	84	82	84
Breast-cancer-WDBC	0.0	0.02	0.02	3.3	–	0.924	94	94	96	94
Breast-cancer-Wisconsin	20.2	–0.01	–0.00	0.0	–	0.945	96	96	96	96
Credit-screening	0.0	–0.03	–0.04	0.0	0.03	0.139	86	85	85	87
Cylinder-bands	0.0	–0.00	0.01	0.0	0.07	0.489	74	75	74	76
PIMA-Indians-diabetes	0.0	0.00	0.01	0.0	–	0.518	76	74	75	75
Soybean-large	9.4	–	–	–	0.03	0.565	88	88	89	85
Spectrometer	0.0	0.01	–0.00	2.0	–	0.964	44	42	41	47
Statlog-Australian	0.0	–0.01	0.00	0.0	0.05	0.706	86	86	86	87
Statlog-German	0.0	0.01	–0.00	14.3	0.04	0.308	76	72	76	71
Statlog-German-numeric	0.0	–0.00	–0.00	0.0	–	0.249	77	71	74	75
Statlog-vehicle	0.0	0.01	0.01	0.0	–	0.944	71	72	71	73
Tic-tac-toe	0.0	–	–	–	0.03	0.095	86	85	85	86
Vowel-context	0.0	0.00	0.00	10.0	–	0.106	74	77	76	75

The columns m1d1, m1d2, m2d1, and m2d2 in Table 3 report 5x2 cross-validated classification accuracy of random forest models trained on either the first (m1) or the second (m2) data split and tested on both splits (d1 and d2). We can observe that no difference

in performance on d1 (i.e., between models m1d1 and m2d1) is larger than 3% and no difference on d2 (i.e., between models m1d2 and m2d2) is larger than 6%. These differences are statistically insignificant using Wilcox paired two sample test with p-values being 0.55 and 0.67, respectively. We conclude that comparisons of classifications give stable and expected results, therefore we can use the workflow from Section 5 to assess classification similarity of datasets.

6.2 Semi-artificial data

Several data mining tasks need large amounts of data for simulation, development of domain specific adjustments and setting of models' parameters. One possibility to gain sufficient number of adequate instances is to prepare a problem-specific data generator which mimics the decision process. Such a generator has to be prepared separately for each dataset which is not trivial and may require significant effort. Another more general approach was taken in Robnik-Šikonja (2014a), where radial based function (RBF) networks are used to learn properties of given datasets and then learned models are turned into generators able to produce semi-artificial data. Users of these generators require an empirical evaluation in order to be sure that generated datasets are similar enough to the original datasets. The presented workflows allow such an evaluation and give users information whether the generated data can be used instead/beside the original dataset in many data mining tasks.We use the three presented workflows to compare the original datasets with the datasets returned by the generators. The generators are available from R package semiartificial (Robnik-Šikonja, 2014b).

For each original dataset from Table 2 the instances are randomly split into two equal parts. One part of the original dataset is used to create problem specific semi-artificial data generator. Default parameters of the RBF-based generator are used. The produced generators create the same number of instances as in the original datasets with the same distribution of classes. We compare the original datasets with the generated ones using the three workflows described in Sections 3–5.

The results of comparisons are presented in Table 4. The column labelled G presents the number of Gaussian kernels in the constructed generators. Relatively large number of Gaussian units are needed to adequately represent the training data which is an indication of sparse nature of the data and fitting local specifics in RBF generator. The column labelled t gives time in seconds needed to construct the generators. The time to generate new instances is negligible (below 1 s in all cases), so we do not report it.

The percentage of generated instances exactly equal to the original instances (column $=$) is nonzero only for datasets with solely discrete attributes where the whole problem space is rather small and identical instances are to be expected.

Average differences in mean (column $\overline{\Delta mean}$) and standard deviation (column $\overline{\Delta std}$) for attributes normalised to $[0, 1]$ are in all cases below 0.10 which shows that moments of the distributions for individual attributes are close to the originals. For almost all datasets and most of the attributes the KS-test (column KSp) detects the differences in distributions of matching numerical attributes. For most datasets the Hellinger distances (column $\overline{H}$) are low (i.e., below 0.11).

Table 4 The comparison of original and generated datasets. The columns are the same as in Table 3, except for G – the number of Gaussian kernels in generative model and t – generator construction time in seconds. In classification comparison for models mXdY 1 stands for the original data and 2 for the generated data

Dataset	G	t	$=$	$\overline{\Delta mean}$	$\overline{\Delta std}$	KSp	$\overline{H}$	*ARI*	*m1d1*	*m1d2*	*m2d1*	*m2d2*
Annealling	143	12.0	0	–0.098	0.024	100	0.029	0.721	98	92	97	93
Balance-scale	212	4.5	0	0.010	0.037	100	–	0.288	84	85	83	88
Breast-WDBC	151	6.0	0	–0.021	–0.047	97	–	0.951	95	97	95	98
Breast-WISC	122	3.9	0	–0.011	0.017	100	–	0.968	96	98	97	98
Credit-screening	342	13.4	0	–0.091	–0.063	83	0.032	0.115	87	92	87	94
Cylinder-bands	355	27.1	0	–0.015	–0.063	100	0.108	0.602	78	73	73	86
PIMA-diabetes	506	12.3	0	–0.021	–0.024	100	–	0.295	76	81	76	84
Soybean-large	300	41.6	28	–	–	–	0.058	0.385	92	94	83	96
Spectrometer	473	33.1	0	–0.008	–0.032	97	–	0.969	49	57	38	88
Statlog-Austr	333	9.6	0	–0.082	–0.053	100	0.030	0.939	87	91	87	93
Statlog-German	845	39.1	0	–0.042	0.020	100	0.060	0.253	75	83	76	96
German-num	723	24.5	0	–0.023	0.019	100	–	0.159	75	80	76	83
Statlog-vehicle	571	17.5	0	–0.035	–0.031	100	–	0.954	74	72	70	80
Tic-tac-toe	859	28.6	78	–	–	–	0.090	0.164	95	98	74	99
Vowel-context	291	12.4	0	0.004	–0.016	0	–	0.441	88	77	85	76

The clustering similarity (column ARI) is considerable for many datasets (ARI > 0.5 for 7 of 15 datasets) but there are also some datasets where it is low (ARI < 0.2 for 3 out of 15 datasets). When comparing these numbers to the ones reported for random splits of the same dataset in Table 3, we can notice that clustering based similarity is only slightly worse for the generated data, which is an indication that the generated data is a good approximation of originals in unsupervised learning.

The columns m1d1, m1d2, m2d1, and m2d2 report 5x2 cross-validated classification accuracy of random forest models trained on either original (m1) or generated (m2) data and tested on both original (d1) and generated (d2) data. A general trend observed is that on majority of datasets model trained on original data (m1) performs better on generated data than on the original data (m1d2 is larger than m1d1 for 11 of 15 datasets). This indicates that some of the complexity of the original data is lost in the generated data. This is confirmed also by models built on generated data (m2) which mostly perform better on the generated than on original data (m2d2 is larger than m2d1 in 13 out of 15 datasets). Nevertheless, the generated data can sometimes be a satisfactory substitute, namely, models build on generated data outperform models built on original data when both are tested on original data (m2d1 is larger than m1d1 in 3 cases out of 15, in 8 cases m1d1 is larger and in 4 case there is a draw). The results show that classification workflow can distinguish between successfully and unsuccessfully generated semi-artificial datasets. The gives users of semiArtificial and other custom built data generators an important information about the usability of the generated datasets.

6.3 Computational complexity of the workflows

The computational complexity of the three proposed workflows partially depends on used components. The most complex operation in the statistical workflow is counting the number of equal instances in both datasets which is upper bounded by $O(an^2)$, where a is the number of attributes and n the number of instances. The implementation using hash map would reduce this to $O(an)$. Computation of statistics mean, standard deviation, skewness, kurtosis, medcouple, L/R medcouple, Hellinger distance, and KS test takes $O(n)$.

The clustering workflow executes a chosen clustering algorithm and heuristics to estimate an appropriate number of clusters. The expected running time of k-medoid clustering we use is $O(\mathrm{ank})$, where k is the number of clusters. The silhouette heuristics requires $O(\mathrm{ank}^2)$. We use several repetitions of the whole process to reduce the variance of the reported means.

The computational complexity of classification workflow is the same as complexity of the used learning algorithm. For random forests we used, the expected running time is upper bounded by $O(tmn \log n)$ where t is the number of trees, m is the number of randomly selected features in each node, and n is the number of instances.

We measured the actual running times of all three proposed workflows on a single core of 2.67GHz Intel I7 Windows 7 machine with R 3.2 and 12GB RAM. We report average times for comparisons of 15 random data subsets as reported in Table 3. The executions take 0.2 s for statistical workflow (computing mean, standard deviation, skewness, kurtosis, medcouple, L/RMC, KS test, and Hellinger distance), 1.5 s for one repetition of clustering workflow (using preprocessing with k-medoid clustering, silhouette heuristics, and similarity measures ARI, FM, J, and VI), and 72s for classification workflow (using 5×2 cross-validation with random forests algorithm building 100 trees in each fold). As computational complexity is (super)linear in the number of attributes and number of

instances considerably larger datasets can be compared in practice (with expected linear increase in time). All components of the workflows can get significant speedup on parallel and distributed architectures and this is a recommend approach for really large datasets.

7 Conclusions and further work

We presented three practically useful dataset comparison workflows which help to answer the question how similar two datasets are. The first workflow compares matching pairs of attributes in the two datasets. It computes differences in their statistics: mean, standard deviation, skewness, kurtosis, medcouple and L/R medcouple. It also assesses similarity of attribute distributions using Kolmogorov-Smirnov test for numeric attributes and Hellinger distance for discrete attributes. The second workflow tests structural similarity of datasets based on clustering. We preprocess compared datasets so that standard external clustering validation methods can be used. We use ARI, Fowlkes-Mallows index, Jaccard index and Variance of Information index. As the results of clustering algorithms tend to have large variance, low similarity score returned by this workflow has to be taken with caution. The third workflow compares similarity of classifications, which is important for many practical uses.

We evaluated the three workflows on a collection of UCI datasets. By testing similarities of random halves of datasets we established the baseline of expected results. Testing samples generated by the RBF networks based data generators we were able to distinguish between successful and unsuccessful generators. This can help users decide if their generated semi-artificial dataset is a usable substitute for the original dataset.

Many improvements are possible and the proposed workflows could be parameterised in several ways. Statistical workflow could be extended with robust statistics based on percentiles as well as other statistical tests for distributions of univariate data. Clustering workflow could incorporate heuristics to suggest the most appropriate clustering method which might reduce clustering variability. For hierarchical clustering specific similarity measure adaptations could be introduced. Classification workflow could be extended to regression and multi-labelled classification.

References

Anwar, S., Rana, Z.A., Shamail, S. and Awais, M.M. (2012) 'Using association rules to identify similarities between software datasets', *Eighth International Conference on the Quality of Information and Communications Technology QUATIC*, Lisbon, Portugal, pp.114–119.

Arbelaitz, O., Gurrutxaga, I., Muguerza, J., Pérez, J.M. and Perona, I. (2013) 'An extensive comparative study of cluster validity indices', *Pattern Recognition*, Vol. 46, No. 1, pp.243–256.

Ashour, W. and Fyfe, C. (2014) 'Improving bregman k-means', *International Journal of Data Mining, Modelling and Management*, Vol. 6, No. 1, pp.65–82.

Bache, K. and Lichman, M. (2014) *UCI Machine Learning Repository, 2014*, http://archive.ics.uci.edu/ml

Bay, S.D. and Pazzani, M.J. (2001) 'Detecting group differences: mining cotrast sets', *Data Mining and Knowledge Discovery*, Vol. 5, No. 3, pp.213–246.

Ben-David, S., Von Luxburg, U. and Pál, D. (2006) 'A sober look at clustering stability', *Computational Learning Theory, COLT'06*, Springer, Pittsburgh, Pennsylvania, USA, pp.5–19.

Brys, G., Hubert, M. and Struyf, A. (2006) 'Robust measures of tail weight', *Computational Statistics and Data Analysis*, Vol. 50, No. 3, pp.733–759.

Caruana, R. and Niculescu-Mizil, A. (2006) 'An empirical comparison of supervised learning algorithms', *Proceedings of the 23rd International Conference on Machine Learning, ICML '06*, Pittsburgh, Pennsylvania, USA, pp.161–168.

Das, G., Mannila, H. and Ronkainen, P. (1998) 'Similarity of attributes by external probes', *Proceedings of Knowledge Discovery in Databases, KDD 1998*, AAAI Press, New York, USA, pp.23–29.

Fowlkes, E.B. and Mallows, C.L. (1983) 'A method for comparing two hierarchical clusterings', *Journal of the American Statistical Association*, Vol. 78, No. 383, pp.553–569.

Fritsch, A. (2012) *mcclust: Process an MCMC Sample of Clusterings*, R package version 1, https://CRAN.R-project.org/package=mcclust

Ganti, V., Gehrke, J. and Ramakrishnan, R. (1999) 'A framework for measuring changes in data characteristics', *Proceedings of the 18th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, Philadelphia, Pennsylvania, USA, pp.126–137.

Gower, J.C. (1971) 'A general coefficient of similarity and some of its properties', *Biometrics*, Vol. 27, No. 4, December, pp.857–871.

Hennig, C. (2013) *fpc: Flexible Procedures for Clustering*, R package, version 2.1-7, http://CRAN.R-project.org/package=fpc

Hubert, L. and Arabie, P. (1985) 'Comparing partitions', *Journal of Classification*, Vol. 2, No. 1, pp.193–218.

Jaccard, P. (1901) 'Distribution de la florine alpine dans la bassin de dranses et dans quelques regiones voisines', *Bulletin de la Societe Vaudoise des Sciences Naturelles*, Vol. 37, pp.241–272.

Jaskowiak, P.A., Moulavi, D., Furtado, A.C., Campello, R.J., Zimek, A. and Sander, J. (2016) 'On strategies for building effective ensembles of relative clustering validity criteria', *Knowledge and Information Systems*, Vol. 47, No. 2, May, pp.329–354.

Jin, H., Wang, S., Zhou, Q. and Li, Y. (2014) 'An improved method for density-based clustering', *International Journal of Data Mining, Modelling and Management*, Vol. 6, No. 4, pp.347–368.

Kaufman, L. and Rousseeuw, P.J. (1990) *Finding Groups in Data: An Introduction to Cluster Analysis*, John Wiley & Sons, Hoboken, New Jersey, USA.

Lavrač, N., Kavšek, B., Flach, P. and Todorovski, L. (2004) 'Subgroup discovery with CN2-SD', *The Journal of Machine Learning Research*, Vol. 5, pp.153–188.

Leisch, F. (2006) 'A toolbox for k-centroids cluster analysis', *Computational Statistics and Data Analysis*, Vol. 51, No. 2, pp.526–544.

Maechler, M., Rousseeuw, P., Struyf, A., Hubert, M. and Hornik, K. (2013) *cluster: Cluster Analysis Basics and Extensions*, R Package, version 1.14.4, http://CRAN.R-project.org/package=cluster

Meilă, M. (2007) 'Comparing clusterings – an information based distance', *Journal of Multivariate Analysis*, Vol. 98, No. 5, pp.873–895.

Morozov, S. and Saiedian, H. (2013) 'An empirical study of the recursive input generation algorithm for memory-based collaborative filtering recommender systems', *International Journal of Information and Decision Sciences*, Vol. 5, No. 1, pp.36–49.

Parthasarathy, S. and Ogihara, M. (2000) 'Exploiting dataset similarity for distributed mining', *Parallel and Distributed Processing*, Springer, Cancun, Mexico, pp.399–406.

Rand, W.M. (1971) 'Objective criteria for the evaluation of clustering methods', *Journal of the American Statistical Association*, Vol. 66, No. 336, pp.846–850.

Reddy, D. and Jana, P.K. (2014) 'A new clustering algorithm based on voronoi diagram', *International Journal of Data Mining, Modelling and Management*, Vol. 6, No. 1, pp.49–64.

Robnik-Šikonja, M. (2014a) *Data Generator based on RBF Network*, Technical Report, University of Ljubljana, Faculty of Computer and Information Science, http://arxiv.org/abs/1403.7308v1

Robnik-Šikonja, M. (2014b) *semiArtificial: Generator of Semi-Artificial Data*, R package version 1.2.0, http://cran.r-project.org/package=semiArtificial

Robnik-Šikonja, M. and Savicky, P. (2015) *CORElearn – Classification, Regression, Feature Evaluation and Ordinal Evaluation*, R package version 1.47.01, http://cran.r-project.org/package=CORElearn

Savicky, P. (2012) *readMLData: Reading Machine Learning Benchmark Data Sets from Different Sources in their Original Format*, R package, version 0.9-6, http://cran.r-project.org/package=readMLData

Sequeira, K. and Zaki, M. (2007) 'Exploring similarities across high-dimensional datasets', *Research and Trends in Data Mining Technologies and Applications*, Vol. 3, pp.53–85.

Subramonian, R. (1998) 'Defining diff as a data mining primitive', *Proceedings of Knowledge Discovery in Databases, KDD 1998*, New York, USA, pp.334–338.

Venables, W.N. and Ripley, B.D. (2002) *Modern Applied Statistics with S*, 4th ed., Springer Verlag, New York.

Verikas, A., Gelzinis, A. and Bacauskiene, M. (2011) 'Mining data with random forests: A survey and results of new tests', *Pattern Recognition*, Vol. 44, No. 2, pp.330–349.

Vinh, N.X., Epps, J. and Bailey, J. (2009) 'Information theoretic measures for clusterings comparison: is a correction for chance necessary?', *Proceedings of the 26th International Conference on Machine Learning, ICML 2009*, Montreal, Canada, pp.1073–1080.