Data Mining and Knowledge Discovery manuscript No.
(will be inserted by the editor)

SimUSF: An efficient and effective similarity measure that is invariant to violations of the interval scale assumption

Thilak L. Fernando · Geoffrey I. Webb

Received: date / Accepted: date

Abstract Similarity measures are central to many machine learning algorithms. There are many different similarity measures, each catering for different applications and data requirements. Most similarity measures used with numerical data assume that the attributes are interval scale. In the interval scale, it is assumed that a unit difference has the same meaning irrespective of the magnitudes of the values separated. When this assumption is violated, accuracy may be reduced. Our experiments show that removing the interval scale assumption by transforming data to ranks can improve the accuracy of distance-based similarity measures on some tasks. However the rank transform has high time and storage overheads. In this paper, we introduce an efficient similarity measure which does not consider the magnitudes of inter-instance distances. We compare the new similarity measure with popular similarity measures in two applications: DBScan clustering and content based multimedia information retrieval (CBMIR) with real world datasets and different transform functions. The results show that the proposed similarity measure provides good performance on a range of tasks and is invariant to violations of the interval scale assumption.

Keywords Similarity Measure · Interval Scale · Clustering · CBMIR

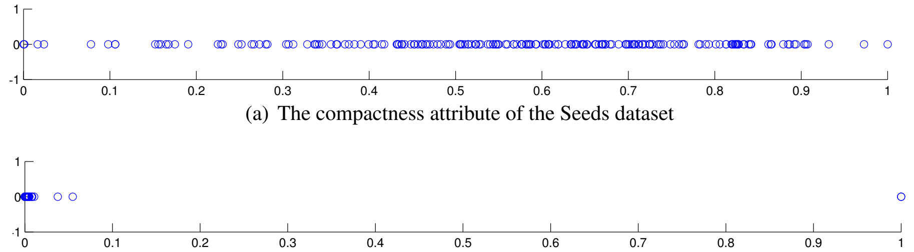
1 Introduction

Many machine learning algorithms rely on similarity calculations between instances. Clustering algorithms group instances that are most similar. Information retrieval ranks instances on similarity to a query. No single measure can capture all notions of similarity that may be relevant to all different applications. Hence, a variety of similarity measures are used for different applications and data.

A numeric attribute of a dataset can be interpreted in one out of four scales—nominal, ordinal, interval and ratio. Unfortunately, unless there are only few values, which is sugges-

Thilak L. Fernando
Monash University, Australia.
E-mail: thilak.fernando@monash.edu

Geoffrey I. Webb
Monash University, Australia.
E-mail: geoff.webb@monash.edu



(a) The compactness attribute of the Seeds dataset

(b) Inverse of the compactness attribute of the Seeds dataset

Fig. 1: Effect of the attribute representation on the distances between instances of the Seeds dataset. (The values are min-max normalized.)

tive of a nominal or ordinal scale, the data usually contain few clues as to which is the highest order scale applicable to an attribute. Often the data analyst is given little or no information about the scale that is appropriate for a given attribute. When an attribute is interpreted as interval or ratio scale, a given difference between two values is treated identically irrespective of the magnitudes of those values.

Often the units in which quantities are measured are arbitrary (e.g. *miles per gallon* or *gallons per mile*). For example, the compactness in the Seeds dataset (Lichman, 2014) is defined differently to the compactness in the Breast Cancer Wisconsin dataset (Lichman, 2014), such that each is the inverse form of the other. Figure 1(a) shows the distribution of the original compactness values in the Seeds dataset. If the inverse is used (as defined in the Breast Cancer Wisconsin dataset), the compactness values of the Seeds dataset will be distributed as shown in Figure 1(b). There is a huge difference in inter-instance distances between the two forms.

Often the only way to determine which transform of a numeric attribute will produce the best results is to try them. For example, in Figure 3 we show that when the Euclidean distance is used as a similarity measure within the DBScan clustering, using the square roots of the attributes produces the best clustering for the Seeds dataset out of the original representation and 6 transformations tested. As there are an infinite number of possible transformations, finding the best one by trial and error is infeasible.

There is often much uncertainty about the true scale of data. We show that making poor assumptions can greatly harm the accuracy of a learning algorithm. In consequence, in some applications it is advantageous to minimize the assumptions that are made. One way of doing so is to make no stronger assumption than that the data are ordinal. The assumptions of ordinality hold for all higher order scales, but the reverse is not true.

One simple approach to this end is to replace numeric values by their ranks before applying traditional (dis)similarity measures such as the Euclidean distance. Our experiments show that for most tasks examined this approach can produce results that are at least competitive with the best outcomes of the corresponding similarity measures applied to the original data, while being impervious to transformations of the data that preserve order. However, using the rank transform is inefficient when previously unseen instances are used in algorithms.

Another well-known measure which is invariant to violations of the interval scale assumption is the random forest. The random forest is very successful in supervised learning. The *Addcl1* sampling based unsupervised random forest produces a similarity measure (SimURF) which is invariant to violations of the interval scale assumption. However, in our experiments it was more effective than the cityblock distance on ranked data only in two out of the twelve tested cases of clustering and information retrieval. Further, Addcl1 unsupervised random forest is very inefficient, as it needs high execution time and space to generate decision trees from intermingled real and synthetic data.

In this paper we introduce a new formalism, unsupervised stochastic forest (usForest), and define a new similarity measure (SimUSF), based thereon. SimUSF is more efficient than replacing data values with their ranks. With respect to cityblock, Euclidean, cosine, Chebychev distance measures and unsupervised random forest, SimUSF provides greater accuracy in DBScan clustering and very competitive accuracy in CBMIR. SimUSF is invariant to violations of the interval scale assumption.

The rest of the paper is organized as follows. Section 2 contains a summary of related work. The use of popular distance based similarity measures with rank transformed data is discussed in Section 3. Section 4 includes details of the unsupervised random forest. We introduce the unsupervised stochastic forest (usForest) and the new similarity measure (SimUSF) in Section 5. Section 6 details similarity based ranking, DBScan clustering and CBMIR experiments to support the claims of this paper. The computational complexities and execution times are compared in Section 7. We summarize the results in Section 8. The conclusion is given in Section 9.

2 Related work

Similarity is a concept that is used extensively, not only in machine learning, but also in many other fields such as psychology (Ashby and Ennis, 2007) and biology (Altschul et al, 1990). Different types of similarity measures are best suited to different tasks. No similarity measure works equally well in all cases. We refer the reader to (Cha, 2007) and (Zezula et al, 2006) for a rich collection of similarity measures. The assessment and comparison of similarity measures are mostly empirical.

An attribute of a dataset can be interpreted in one of four scales: *nominal*, *ordinal*, *interval* and *ratio* (Stevens, 1946; Han et al, 2011). In the nominal scale, values imply categories. The comparison between two values in the nominal scale is limited to identifying whether the two values are the same or different. The magnitudes of differences are meaningless in this scale. In the ordinal scale, attribute values represent orders. The ordinal difference represents how many values lie between two given values of the given attribute. In the interval scale, differences between values represent distances in the particular dimension. The absolute zero is not defined in the interval scale and therefore the ratio between two values is meaningless. As absolute zero is defined, the ratio between two values is meaningful in the ratio scale. The information contained in a value increases in the order nominal, ordinal, interval and ratio. Hence these scales are also called the levels of measurement. An attribute in a higher scale can be transformed to an attribute in a lower scale. However, a lower scale attribute cannot safely be transformed to or interpreted in a higher scale because a higher scale attribute requires additional information to the corresponding lower scale one.

Often, there is no standard for the representation of attributes of a dataset. Some are defined by the mechanism of attribute extraction or the output of the sensors or other processes that generate them. Some are transformed into a particular range in the pre-processing stage. Osborne (2002, 2010) discusses the use of data transforms, highlighting that data transformed inappropriately may produce anomalous conclusions and can complicate interpretations.

In the literature we found two similarity judgment solutions which are invariant to violations of the interval scale assumption. One solution to the problem is to use standard distance measures on rank transformed data (Conover, 1980). The most popular solution to the problem is using a distance measure derived from the random forest. In the trees of a random forest (Breiman, 2001), split points are chosen from values of instances. Tree generation and deployment use only ordinal scale operations where values of instances are compared with a split value to test whether they are less than, greater than or equal to the split value. Hence, random forest based algorithms are invariant to strictly monotonic transforms. Shi and Horvath (2006) introduced a random forest based similarity measure for unsupervised learning.

3 Rank transformation

One way to convert a standard distance measure so as to remove any assumptions relating to the data being in a higher scale than ordinal is to replace each value in the dataset by its tied rank. The tied ranks are calculated independently for each attribute. Let us take a dataset of n instances where in attribute a, which has k^a distinct values, $Dom(a) = \{v_1^a, v_2^a, v_3^a, \ldots, v_{k^a}^a\}$. The value of attribute a of an instance x is denoted by V_x^a, where $V_x^a \in Dom(a)$. We denote the number of instances with value v_i^a by n_i^a, $n_i^a = \sum_{j=1}^{n} I(V_j^a = v_i^a)$, where I is the indicator function and $n = \sum_{i=1}^{k^a} n_i^a$. Let us use N_x^a to represent the number of instances that share the value of instance x. $N_x^a = \sum_{y=1}^{n} I(V_y^a = V_x^a) = \sum_{i=1}^{k^a} n_i^a I(v_i^a = V_x^a)$. Then, the tied rank of instance x in attribute a, $TR(x,a)$ is given by Equation 1.

$$\begin{aligned} TR(x,a) &= \sum_{i=1}^{k^a} n_i^a I(v_i^a < V_x^a) + \frac{1}{2}\{1 + \sum_{j=1}^{k^a} n_j^a I(v_j^a = V_x^a)\} \\ &= \sum_{i=1}^{k^a} n_i^a I(v_i^a < V_x^a) + \frac{1}{2}\{1 + N_x^a\} \end{aligned} \tag{1}$$

If $V_x^a < V_y^a$, Equation 2 shows the difference between tied ranks of two instances x and y in attribute a. The difference between the tied ranks of two instances x and y counts the number of instances between x and y assuming that they are each located in the middle of their duplicates.

$$RD(x,y,a) = TR(y,a) - TR(x,a) = \frac{1}{2}\{N_y^a - N_x^a\} + \sum_{i=1}^{k^a} n_i^a I(V_x^a < v_i^a < V_y^a) \tag{2}$$

Tied ranks for the training data can be calculated and stored in memory. However, to use rank transformed data with many algorithms, the rank differences between a new instance, k and the instances in the training dataset have to be calculated. First the sorted values of the training data are searched to find whether there is an exact value V_p^a such that $V_p^a = V_k^a$. If it does not exist, the largest V_p^a and the smallest V_q^a are found such that $V_p^a < V_k^a < V_q^a$. The rank difference between the new instance, k and existing instance, x can be calculated as given in Equation 3.

$$RD(k,x,a) = \begin{cases} |TR(p,a) - TR(x,a)| + 0.5 & \text{if } V_p^{(a)} = V_k^{(a)} \text{ and } V_p^{(a)} \neq V_x^{(a)} \\ 0 & \text{if } V_p^{(a)} = V_k^{(a)} \text{ and } V_p^{(a)} = V_x^{(a)} \\ |TR(p,a) - TR(x,a) + 0.5| + 0.5 & \text{otherwise} \end{cases} \tag{3}$$

If the dataset has n instances and d attributes, the rank transform needs $O(dn\log(n))$ time. It takes $O(d\log(n))$ to find V_p or/and V_q values in d attributes by binary search. To calculate the similarity between two instances takes $O(d)$ time. Thus, even after the ranks of the training dataset are pre-calculated, calculation of the similarity between an unseen instance and an instance in the dataset still requires $O(d\log(n))$ time. Therefore the rank transform is inefficient, as the time taken for similarity calculation with each previously unseen instance depends on the number of instances in the training dataset.

The rank transform converts the data to ranks. As the transform loses information, there is no inverse transform which can be used to recover the original data from the transformed data. Therefore there is an additional memory overhead of retaining the sorted original data in order to calculate the ranks of previously unseen instances.

4 Unsupervised random forest

A random forest (Breiman, 2001) consists of decision trees in each of which data instances are separated into classes. Shi and Horvath (2006) introduced a random forest for unsupervised learning. For the unsupervised random forest, a synthetic dataset of the same size as the real dataset is generated. The two datasets are combined to form a training set and the instances from the real dataset are identified as one class and the instances from the synthetic dataset are identified as another class. Decision trees are generated to separate these two classes. Similar to the supervised random forest, bootstrap samples from the training set are used to generate trees.

To measure the similarity between two instances x and y, they are parsed through the out-of-bag trees, i.e. the trees which are generated without using x or y. If $T(x,y)$ and $L(x,y)$ represent the number of out-of-bag trees and the number of common leaves for x and y on the out-of-bag trees respectively, the unsupervised random forest based similarity (SimURF) is defined as given in Equation 4. Equation 5 shows the definition of the unsupervised random forest based dissimilarity, DissimURF.

$$SimURF(x,y) = \frac{L(x,y)}{T(x,y)} \tag{4}$$

$$DissimURF(x,y) = \sqrt{1 - SimURF(x,y)} \tag{5}$$

The authors have introduced two different methods to generate the synthetic data. The first, $Addcl1$ randomizes the values of each attribute in real dataset to obtain a synthetic dataset. In the second method, $Addcl2$, the synthetic dataset is generated by uniform random sampling from the hyper rectangle which encloses the real dataset. Out of the two methods, $Addcl1$ produces better results than $Addcl2$ in most cases (Shi and Horvath, 2006). $Addcl1$ based random forests are tolerant to violations of the interval scale assumption as the $Addcl1$ sampling randomly arranges the existing values of each attribute of the real data to generate the synthetic data. The $Addcl2$ sampling is affected by violations of the interval scale assumption as in a given attribute the probability of finding a sample value between given two real data values is proportional to the distance between the two values.

The unsupervised random forest has not been designed to learn different real data clusters. It rather segregates the synthetic data from the real data. As the real and synthetic instances are intermingled, the unsupervised random forest cannot successfully segregate real data from the synthetic data without growing trees to greater heights that require high execution time and memory. Further, when the trees are large, smaller numbers of instances end up at each individual leaf. As the similarity is defined based on the shared leaves, as the number of leaves increases the number of pairs of instances with no shared leaves increases and increasing numbers of similarity assessments return the maximum level of dissimilarity.

5 Unsupervised stochastic forest

Distance based similarity measures are invariant to violations of the interval scale assumption when used with the rank transformed data. However, they require high execution time when calculating the similarity with previously unseen data. The other solution, which does not depend on the interval scale assumption is the SimURF, which is calculated based on the $Addcl1$ unsupervised random forest. As we identified, the use of synthetic data is the main draw back in the unsupervised random forest. The unsupervised random forest focuses on segregating the synthetic data from the real data instead of trying to identify the real data clusters. Further, it requires high execution time and memory resources as the decision trees in unsupervised random forests are exceptionally tall. As an alternative solution which is invariant to violations of the interval scale assumption, we propose Unsupervised Stochastic Forest (usForest) and a similarity measure (SimUSF) based on that. usForest does not use synthetic data. It splits small data samples to generate shorter trees which require less time and memory compared with the unsupervised random forest. SimUSF is more efficient than using distance based similarity measures with ranked data when previously unseen instances are involved in similarity calculations.

Algorithm 1: usForest(D,T,H)

```
Input: $D$ - data, $T$ - number of trees, $H$ - tree height
Output: $usForest$
Initialize $usForest$
for $i = 1 \rightarrow t$ do
    $\mathcal{D} \leftarrow$ select $2^H$ instances from $D$ without replacements.
    $T \leftarrow usTree(\mathcal{D})$
    $usForest \leftarrow usForest \cup T$
end for
return $usForest$
```

Algorithm 2: usTree($\mathcal{D}$)

```
Input: $\mathcal{D}$ - input data
Output: $usTree$
if $|\mathcal{D}|$ is 1 then
    return $ExternalNode$
end if
Let $A$ be the set of attributes
$a \leftarrow$ Randomly selected attribute from $A$
$V \leftarrow (\frac{|\mathcal{D}|}{2})^{th}$ largest value of $a$ in $\mathcal{D}$
$\mathcal{D}_l \leftarrow filter(\mathcal{D}, \mathcal{D}^{(a)} \leq V)$
$\mathcal{D}_r \leftarrow filter(\mathcal{D}, \mathcal{D}^{(a)} > V)$
return $InternalNode\{$
    $LeftChild \leftarrow usTree(\mathcal{D}_l)$,
    $RightChild \leftarrow usTree(\mathcal{D}_r)$,
    $SplitAttribute \leftarrow a$, $SplitValue \leftarrow V$ $\}$
```

A usForest consists of T usTrees. We create usTrees such that in a given usForest all the usTrees have the same height, H. Each usTree has exactly 2^H external nodes and each external node is located at height H from the root. To create a usTree we first randomly select 2^H instances without replacements from the training dataset. Then at each internal node (including the root) an attribute, a, is randomly selected from all attributes in the dataset. Where h is the height of the node, with the height of the root being 0, the split value, V is the $2^{(H-h-1)}$ th largest value of attribute a out of the $2^{(H-h)}$ sample instances at the node. Half of the sample instances have attribute a values less than or equal to V and they are referred to the left child node. The other half of the instances are referred to as the right child node. This is repeated for each child node and stops at height H, where each node has exactly one sample instance. The process is explained in Algorithms 1 and 2.

It may not be possible to split some samples into two equal halves using some attributes because of duplicate values. Then another attribute is randomly selected from the remaining attributes. If a split point cannot be found on any attribute the tree is discarded and a new tree is built from a new sample. We did not include this exception handling in Algorithm 2 for the sake of clarity.

Following the form of analysis used for the random forest in (Breiman, 2001), we use the variable θ to denote a usTree. By the definition of the usTree, θ is identically and independently distributed and we denote the i^{th} usTree by θ_i. If $L(x,\theta_i)$ represents the leaf traced by instance x on usTree θ_i and I is the indicator function, we define the similarity between x and y, $SimUSF(x,y)$ and dissimilarity between x and y, $DissimUSF(x,y)$ as shown in Equations 6 and 7 respectively.

$$SimUSF(x,y) = \lim_{T\to\infty} \frac{1}{T} \sum_{i=1}^{T} I(L(x,\theta_i) = L(y,\theta_i)) \tag{6}$$

$$DissimUSF(x,y) = 1 - SimUSF(x,y) \tag{7}$$

Following the method used by Breiman (2001), we can represent usTree θ_i by a set of non-overlapping k hyper-rectangles $S_{(i,1)}, S_{(i,2)}, S_{(i,3)} \ldots S_{(i,k)}$. Then, $SimUSF(x,y)$ can be written as shown in Equation 8.

$$\begin{aligned} SimUSF(x,y) &= \lim_{T\to\infty} \frac{1}{T} \sum_{j=1}^{k} \sum_{i=1}^{T} I((x \in S_{(i,j)}) \text{ and } (y \in S_{(i,j)})) \\ &= \sum_{j=1}^{k} \{ \lim_{T\to\infty} \frac{1}{T} \sum_{i=1}^{T} I((x \in S_{(i,j)}) \text{ and } (y \in S_{(i,j)})) \} \\ &= \sum_{j=1}^{k} P(x,y,j,\theta) \end{aligned} \tag{8}$$

Where,

$$P(x,y,j,\theta) = \lim_{T\to\infty} \frac{1}{T} \sum_{i=1}^{T} I((x \in S_{(i,j)}) \text{ and } (y \in S_{(i,j)})) \tag{9}$$

By the law of large numbers $P(x,y,j,\theta)$ is the probability of finding x and y in hyper rectangle S_j generated by stochastic process θ. By design k $(= 2^H)$ is a constant. Hence, the number of usTrees needed to estimate $SimUSF(x,y)$ with a sufficient accuracy depends on θ.

In a usForest, θ consists of random processes for selecting k instances without replacements from the dataset, D and for selecting a split attribute, a at each internal node. There are $\binom{n}{k}$ ways of selecting k out of n instances. As there are $k-1$ internal nodes in a usTree, the split attributes can be selected in d^{k-1} different ways. Therefore, the maximum number of different trees is $d^{k-1}\binom{n}{k}$.

Similar analysis can be done for the unsupervised random forest. k in the unsupervised random forest is in the order of $O(n)$ having the maximum value $2n$ which is much larger than k in usForest. If values are unique in each attribute there are $(n!)^d$ ways to generate $Addcl1$ synthetic data. As $2n$ instances are selected with replacements from a dataset having n real instances and n synthetic instances, $\binom{4n-1}{2n}$ different samples can be generated. If we assume that only one attribute is used in split point selection, the number of different trees that can be generated from a given sample takes the order of $O(d^{2n})$. As k of usForest $\ll 2n$ and $\binom{n}{k}$ of usForest $\ll \binom{4n-1}{2n}(n!)^d$ we can expect the usForest based similarity, SimUSF to converge faster than the similarity calculated from the unsupervised random forest, SimURF to respective expected values. Section 6.2.1 provides experimental results to support this analysis.

The SimUSF takes $O(TH2^H)$ time to generate T usTrees. For an unseen instance, it takes $O(TH)$ to traverse the trees to find the leaves. To calculate the similarity between two instances takes $O(T)$ time. Thus, after a usForest is created it takes $O(TH)$ to calculate similarity between a pair of unseen instances. In contrast to the rank transform, it is independent from the number of instances in the original dataset. Once the usForest is created the original dataset is not required for further processing and the memory requirement, $O(T2^H)$ is small and independent from the number of instances in the original dataset.

6 Empirical Evaluation

In this section we empirically evaluate SimUSF in similarity based ranking, DBScan clustering and content based multimedia information retrieval (CBMIR). We used similarity based ranking to compare SimUSF with other similarity/dissimilarity measures as ranking is one of the main purposes of using a similarity measure. DBScan clustering (Ester et al, 1996) was chosen to compare the similarity measures as it can identify arbitrary shaped clusters and is a highly regarded and widely used algorithm (SIGKDD, 2015). Then we tested the similarity measures in CBMIR as it is one of the active research areas with number of recent publications.

It should be noted that SimUSF is not directly applicable to k-means clustering, which requires the computation of the mean of objects in the probability space constructed by the similarity measure. SimUSF is a distance measure and constructing a mean relative to it might be challenging because there is no direct access to the associated probability space. The most common approach to k-means assumes the instances can be interpreted as points in Euclidean space, implicitly assuming an interval scale, and hence not appropriate for our purposes. Note that this observation about the difficulty of calculating an average object with regard to a measure has been studied for different measures, such as Dynamic Time Warping (Petitjean and Gançarski, 2012). A work-around could have been to use the k-medoids algorithm, but its drawbacks (eg: potential oscillation of the results, computational complexity, inferior clustering results) have led us to consider the more popular DBScan algorithm.

6.1 Experimental set-up

We designed the experiments to study how similarity measures are affected by violations of the interval scale assumption. To this end, we applied a number of strictly monotonic transforms[1] to change the inter-instance distances.

A strictly monotonic transform can be order preserving or order reversing. They can also be linear or non-linear. We tested common non-linear[2] order preserving and order reversing transforms: $e^X, X^2, \sqrt{X}, ln(X), \frac{1}{X}$ and $e^{\text{-}X}$, where $X = b(x+a)$ and $X > 0$. A value of a min-max normalized attribute is represented by x. Since the functions $ln(X)$ and $\frac{1}{X}$ are not defined for $X = 0$, a small positive value a is used. We employed a positive value b to transform the values into a wide range which considerably change the inter-instance distances within the data-type limits. Hence, the values $a = 0.0001$ and $b = 100$ were used with all datasets. Some real world datasets are already min-max normalized. Hence, to induce the same effect on all the datasets the original datasets were first min-max normalized, then subjected to one of the transformations; and then renormalized using min-max before calculating the similarity values.

The lines in the figures should be used only as aids to identify and compare the corresponding points. They are useful for the clarity of the figures as there are many overlapping points. The lines do not represent functional relationships.

6.2 Similarity based ranking

Let us define a list $R(i, Sim)$ which ranks the instances in the dataset D based on the similarity calculated between a given instance, $i \in D$ and each other instance in D using the similarity measure Sim. As in (Faith et al, 1987) we used Spearman's rank correlation coefficient, $\rho(R(i, Sim^m), R(i, Sim^n))$ to calculate the similarity between two such lists $R(i, Sim^m)$ and $R(i, Sim^n)$. As $\rho(R(i, Sim^m), R(i, Sim^n))$ values approximated the Gaussian distribution the average, $\rho(Sim^m, Sim^n, D)$ and standard deviation, $\sigma(Sim^m, Sim^n, D)$ of $\{\rho(R(i, Sim^m), R(i, Sim^n)) | i \in D\}$ were used to asses how similar the ranking produced by Sim^m and Sim^n.

6.2.1 Comparison between SimUSF and SimURF estimations

For a tree based similarity measure, we calculate $\rho(Sim^{(T,1)}, Sim^{(T,2)}, D)$ and $\sigma(Sim^{(T,1)}, Sim^{(T,2)}, D)$, where $Sim^{(T,1)}$ and $Sim^{(T,2)}$ use the same similarity measure, Sim and they are independently calculated from two independently built forests each with T trees and identical parameter settings. When T increases $\rho(Sim^{(T,1)}, Sim^{(T,2)}, D)$ increases and $\sigma(Sim^{(T,1)}, Sim^{(T,2)}, D)$ decreases as Sim calculations converge to the expected value.

To compare the convergence of SimUSF and SimURF calculations we used a dataset which has 1000 instances and 4 dimensions. Each value in the dataset was independently selected from a uniform distribution having the range [0,1]. The calculations were done for T values 100,1000 and 10000. The $\rho(Sim^{(T,1)}, Sim^{(T,2)}, D)$ and $\sigma(Sim^{(T,1)}, Sim^{(T,2)}, D)$ values are given in Table 1. With a given number of usTrees SimUSF produced a much

[1] A monotonic transform, $f : \mathbb{R} \to \mathbb{R}$ is either $\forall x > y \Leftrightarrow f(x) \geq f(y)$ or $\forall x > y \Leftrightarrow f(x) \leq f(y)$. Such a transform can produce ambiguities in the order. A strictly monotonic transform is either $\forall x > y \Leftrightarrow f(x) > f(y)$ or $\forall x > y \Leftrightarrow f(x) < f(y)$. Hence, a strictly monotonic transform can guarantee either order preservation or order reversal.

[2] Datasets are generally subjected to min-max normalization. As a result, linear order preserving transforms do not alter the similarity scores.

Table 1: $\rho(Sim^{(T,1)}, Sim^{(T,2)}, D)$, $\widetilde{\rho}$ and $\sigma(Sim^{(T,1)}, Sim^{(T,2)}, D)$, $\widetilde{\sigma}$ values produced by SimUSF and SimURF

T	SimUSF		SimURF	
	$\widetilde{\rho}$	$\widetilde{\sigma}$	$\widetilde{\rho}$	$\widetilde{\sigma}$
100	0.951	0.016	0.418	0.078
1000	0.994	0.002	0.694	0.035
10000	0.999	0.000	0.867	0.018

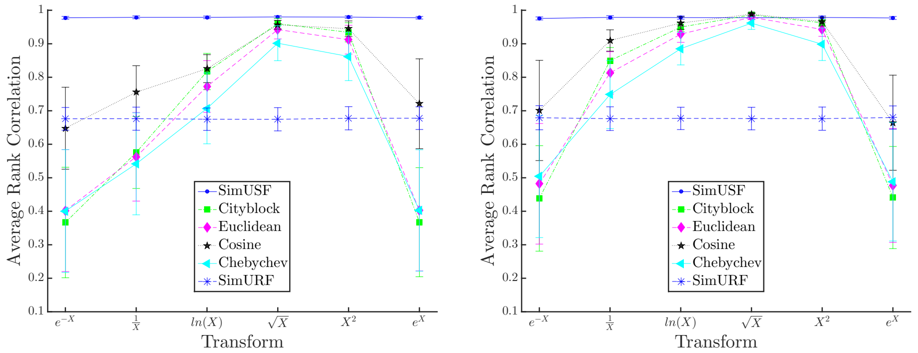
closer estimation for its expected value than SimURF estimated its expected value with the same number of decision trees. This observation supports our analysis given in Section 5.

Increasing the number of decision trees in an unsupervised random forest for a better estimate is not a viable option in many practical cases as they have high execution time and memory overheads. The unsupervised random forests needed taller trees to segregate the synthetic data from the real data. As a result, fewer instances were traced to individual leaves. In consequence, in these experiments SimURF produced zero values for 96.0%, 85.5% and 71.0% out of the total number of similarity calculations between the instance pairs when 100, 1000 and 10000 decision trees were used respectively. In contrast, SimUSF produced 0.00% zero similarity values when calculated using 100 usTrees. Having non-zero similarity values are important when a large number of instances are to be ordered with respect to a given instance as zero (or non-unique) values do not differentiate the instances. Thus, a large number of zero values (or non-unique) may affect the precisions of some algorithms. As the SimURF produces a huge number of zero values and non-zero values having very small magnitudes, the SimURF mean square error values convey a false notion of convergence. Therefore, we used similarity based ranking to compare the convergence of the SimURF with that of the SimUSF.

6.2.2 Effects of violations of the interval scale assumption

$\rho(Sim^{(o)}, Sim^{f(x)}, D)$ and $\sigma(Sim^{(o)}, Sim^{f(x)}, D)$ are used in this section to assess the effects of violations of the interval scale assumption when the similarity measure, *Sim* is used for ranking. $Sim^{(o)}$ represents the similarity calculation for the original data and $Sim^{f(x)}$ represents the similarity calculation after the data are transformed with the function $f(x)$. The transforms were discussed in Section 6.1. Two four-dimensional synthetic datasets were used in the experiments. Their characteristics are described along with the plots in Figure 2. SimUSF was compared with four popular similarity/distance measures: cityblock, Euclidean, cosine and Chebychev. It was also compared with the unsupervised random forest based similarity (SimURF). The SimUSF values were calculated from usForests each with 1000 height 5 trees. The SimURF values were calculated from Addcl1 unsupervised random forests each with 1000 decision trees.

SimUSF consistently showed a high correlation across all the tested transforms whereas the other similarity measures except SimURF were considerably affected by the transforms. SimUSF always had very small standard deviation values which indicate that it performed equally well for all the instances in the dataset after all the transforms. The other similarity measures showed high standard deviation values in most of the cases. In such cases the rankings for different instances were differently affected even within the same transformed dataset. Based on our experiments we can conclude that the rankings produced by SimUSF and SimURF are invariant to violations of the interval scale assumption and the rankings



(a) Uniform distribution: 1000 instances, 4 dimensions, $minimum = 0, maximum = 1$

(b) Standard normal distribution: 1000 instances, 4 dimensions, $\mathcal{N}(0, I)$

Fig. 2: Effects of violations of the interval scale assumption on similarity based ranking

Table 2: Clustering Datasets

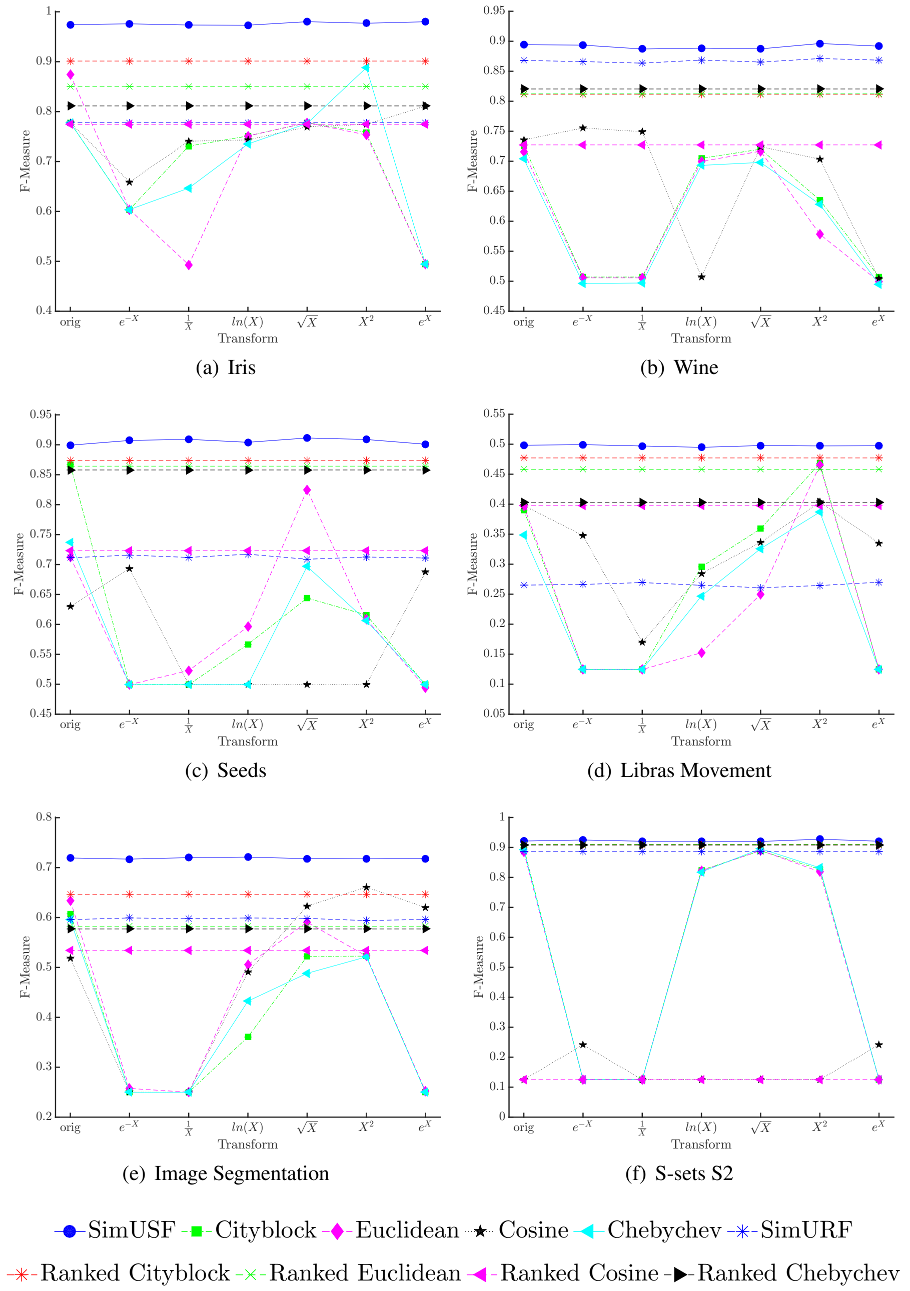
Dataset	Instances	Dimensions	Classes
Iris	150	4	3
Wine	178	13	3
Seeds	210	7	3
Libras Movement	360	91	15
Image Segmentation	2,310	19	7
S-set S2	5,000	2	15

produced by cityblock, Euclidean, cosine and Chebychev distances are not tolerant to such violations. Despite the fact that both SimUSF and SimURF use trees that are invariant to violations of the interval scale assumption, neither could produce average rank correlation value = 1, as those are ensemble measures where different iterations produce different sets of trees. SimUSF showed a high average rank correlation value than SimURF as it could produce a better estimation for its expected value when compared with SimURF with the given number of trees. This is in-line with the discussion in Sub section 6.2.1.

6.3 DBScan clustering

DissimUSF was compared with the cityblock, Euclidean, cosine, Chebychev distance measures and unsupervised random forest based dissimilarity measure (DissimURF) in DBScan clustering. Each dissimilarity measure was used as the distance measure in the DBScan and the best F-Measure (F1 measure) values found in the experiments were compared. The details of the datasets used in the experiments are given in Table 2. S-sets S2 dataset was taken from the University of Eastern Finland (2015). The remaining datasets were downloaded from the UCI repository (Lichman, 2014).

1000 usTrees, each generated with height 5 ($H = 5$) were used in all the usForest calculations for the DBScan clustering. 1000 decision trees were generated for the unsupervised random forests. In the DBScan parameter estimation, for a given K, a K-distance graph was used as a clue to find the start and end ε values to search. The ε value where the first cluster started to form was identified as the starting ε. The end ε was identified as the minimum ε



(a) Iris

(b) Wine

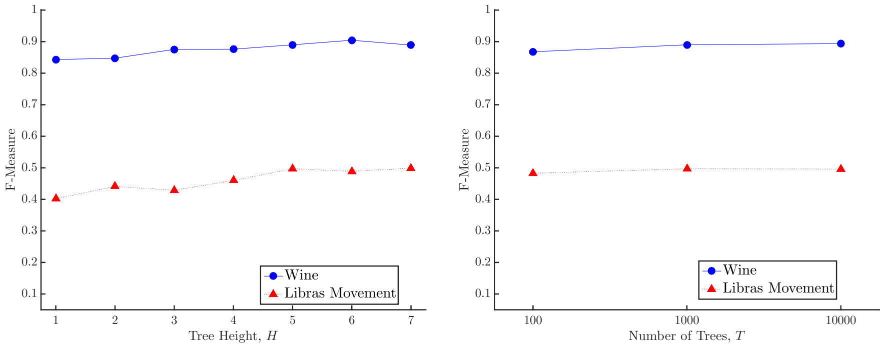
(c) Seeds

(d) Libras Movement

(e) Image Segmentation

(f) S-sets S2

Fig. 3: Effects of violations of the interval scale assumption on F-Measure of DBScan clustering



(a) Tree Height, H

(b) Number of Trees, T

Fig. 4: Effects of usForest parameters on DBScan F-Merasure

value where all instances gather to form a single cluster. The range was searched in 20 equal size steps. Then the search was focused on the maximum F-Measure found so far for the given K. The new minimum ε was set to one step less than the best ε and the new maximum was set to one step more. Then the search was done again in 20 steps covering the new ε range. This process was repeated to find the best F-measure till ε values were explored to three significant digits. The entire process was repeated for all K values from 2 to 25. The best F-Measure found in the entire search were used in the comparisons. The results of the DBScan clustering experiments are shown in Figure 3.

When cityblock, Euclidean, cosine and Chebychev distances were used in DBScan clustering algorithm there were number of cases where each of the four distance based similarity measures produced better F-Measure values after some transforms than with the original data. Therefore, in such cases, the original data representations were not the optimal to use in DBScan clustering with the respective similarity measures. Even though out of the tested transforms some transforms produced better F-Measure values with some datasets, we cannot conclude that they produced the best results as there are infinite number of possible transforms and testing all of them is infeasible.

The cityblock and Euclidean distance measures with the rank transformed data often produced better or similar F-Measure values when compared with the best F-Measure values produced by respective similarity measures with the original data or data transformed using the six functions. With the rank transformed data, the cityblock distance always performed better than the Euclidean distance. Except for the Image Segmentation dataset in the DBScan clustering, the cityblock distance with the rank transformed data produced better or equally good results as the best results produced by all four distance based similarity measures: cityblock, Euclidean, cosine and Chebychev with the original or the data transformed by the other six functions. Therefore, based on these results we can argue that using the cityblock distance with rank transformed data is a better alternative to using the cityblock, Euclidean, cosine and Chebychev distances with original data. In contrast to using cityblock, Euclidean, cosine and Chebychev distances, the cityblock distance with rank transformed data produces results that are invariant to violations of the interval scale assumption.

In the DBScan clustering experiments, only with the Wine dataset, the unsupervised random forest based similarity, SimURF produced a better F-Measure value than the F-Measure value produced by the cityblock distance with rank transformed data. In all other 5 cases (i.e. 83.33%) it did not produce competitive results. As shown in the plots of Figure 3 the new similarity measure, SimUSF which is based on our newly introduced unsupervised stochastic forest, usForest always produced better F-Measure values than all other tested measures in the DBScan clustering. This is because, SimUSF successfully distinguishes nearest neighbors from the rest of the instances, as they are gathered at tree leaves of the usForest. The unsupervised random forest attempts to segregate real data from synthetic data rather than collecting neighboring instances together. Hence, SimURF proves less effective than SimUSF for the purposes of clustering.

The effects of parameter settings: the tree height (H) and the number of trees (T) of the usForest on the best F-measures in the DBScan clustering were studied in the following experiments. First, T was fixed to 1000 and the best F-measure values were recorded for H values from 1 to 7. Then H was fixed to 5 and the best F-measure values were recoded for T values 100, 1000 and 10000. Figure 4 shows the results of the experiments done with the Wine and Libras Movements datasets. For the Wine and Libras Movements datasets $H = 6$ and $H = 5$ produced the best F-measures respectively. For both datasets the F-Measure values slowly increases with the number of trees, T. We used the parameter values $H = 5$ and $T = 1000$ when DissimUSF was compared with the other similarity measures as those parameter values could always produce good results in our experiments.

6.4 CBMIR

An information retrieval system fetches relevant instances to a user query from a given database. The instances are ranked based on the similarity to the query. In relevance feedback, the user selects a few relevant and irrelevant instances from the top k results and the system uses them to fetch an improved result set from the database. The relevance feedback process is continued several times.

We studied the effect of violations of the interval scale assumption on CBMIR precision@k results[3] for standard benchmark datasets. A simplified version of Rocchio's algorithm (Rocchio, 1971; Manning et al, 2008) was used in the CBMIR experiments with different dissimilarity measures. Equation 10 describes how query, positive and negative feedbacks were used to find the distance of an instance $\mathbf{x}$ with respect to a composite query $\mathcal{Q} = \mathcal{P} \cup \mathcal{N}$. $\mathcal{P}$ represents the initial query and the positive feedbacks. $\mathcal{N}$ represents the negative feedbacks. γ is the weighing parameter for the negative feedbacks.

$$dist(\mathbf{x}, \mathcal{Q}) = \frac{1}{|\mathcal{P}|} \sum_{\mathbf{y} \in \mathcal{P}} dist(\mathbf{x}, \mathbf{y}) - \gamma \frac{1}{|\mathcal{N}|} \sum_{\mathbf{z} \in \mathcal{N}} dist(\mathbf{x}, \mathbf{z}) \tag{10}$$

Zhou et al (2012) shows that the ReFeat, which is a also a tree based CBMIR system, can produce higher precision than CBMIR systems manifold learning method (MRBIR) (He et al, 2004), instance-based relevance feedback method (InstRank) (Giacinto and Roli, 2005), Bayesian learning method (BALAS) (Zhang and Zhang, 2006) and query-sensitive ranking method (Qsim) (Zhou and Dai, 2006). Therefore, the SimUSF, SimURF and city-block, Euclidean, cosine, Chebychev distances were also compared with the ReFeat in our experiments.

Similar to the experiments with ReFeat (Zhou et al, 2012), we used query and five feedback rounds. In each feedback round, up to two positive feedbacks and two negative feedbacks were randomly selected as available in the top 50 results from the previous round. This

[3] Only the top k instances are important to the user in information retrieval.

process was repeated 5 times starting with different randomly selected unseen queries from each class. The entire process was independently repeated 20 times to calculate the mean and standard deviation. The standard deviation values were represented as the error bars in the figures. Initial queries were not used in the tree generations in the usForest, unsupervised random forest and ReFeat. The query and feedbacks were not included in the results used to compute precision@50.

We tested with γ values 0, 0.25, 0.5, 0.75 and 1 on every dataset. In addition to that, sample sizes 2, 4, 8, 16, 32, 64 and 128 were used to build the trees for the usForest. Sample sizes 4, 8, 16, 32, 64 and 128 were used for the ReFeat[4]. The SimUSF, SimURF and ReFeat were tested with 1000 trees in each case. The optimal results were used in the comparisons. Table 3 shows the publicly available datasets used in our experiments. They were also used before in ReFeat (Zhou et al, 2012) and other similar type of experiments. The feedback round 5 precision@50 values are shown in Figure 5.

Table 3: CBMIR Datasets

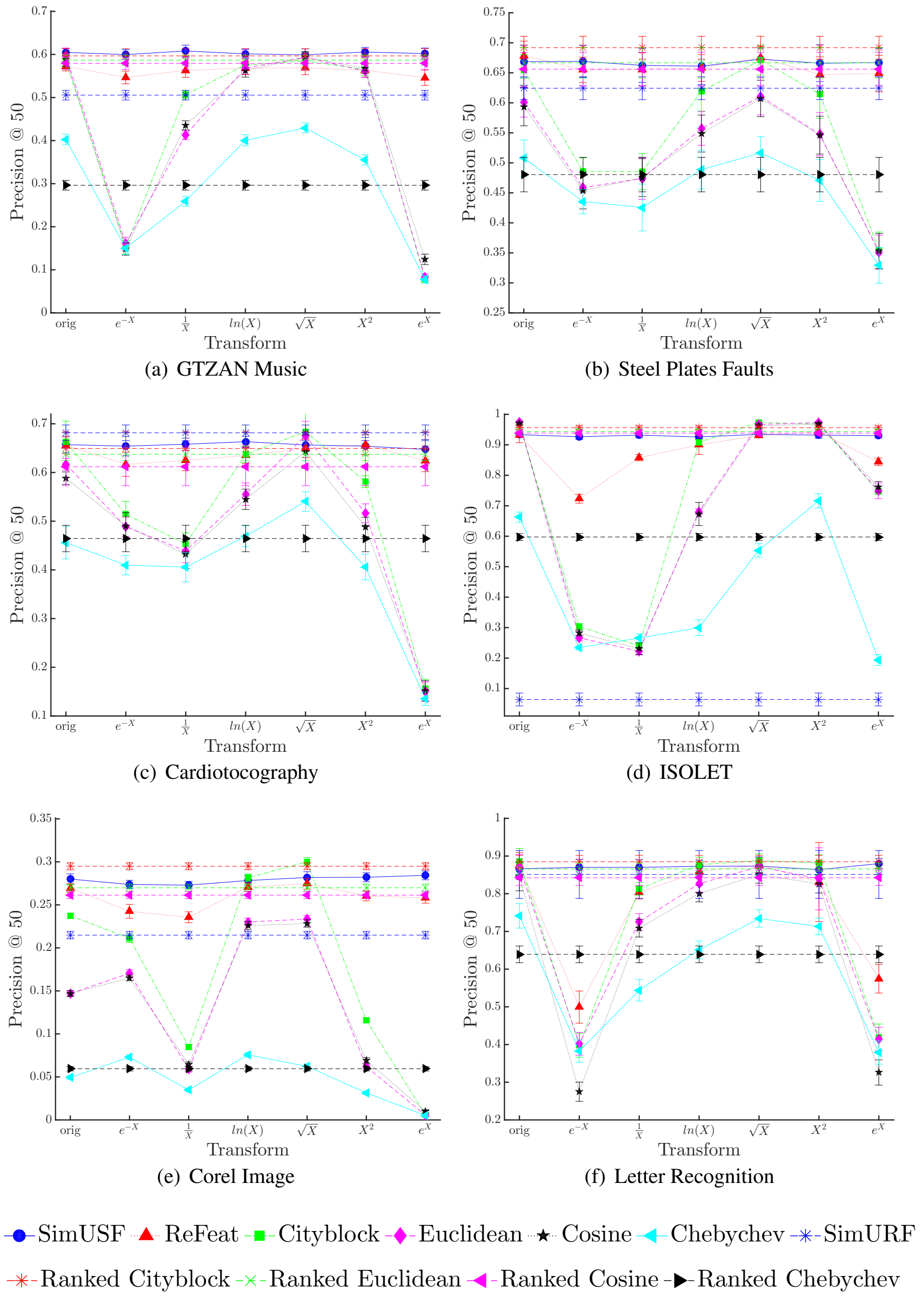
Dataset	Instances	Dimensions	Classes
GTZAN Music	1,000	230	10
Steel Plates Faults	1,941	27	7
Cardiotocography	2,126	21	10
ISOLET	7,797	617	26
Corel Image	10,000	64	100
Letter Recognition	20,000	16	26

Similar to the observations in DBScan clustering,in our CBMIR experiments we observed number of cases where the four distance based similarity measures: cityblock, Euclidean, cosine and Chebychev produced better results after some transforms than with the original data. Therefore in those cases the original data representations were not the optimal to use in CBMIR with the corresponding similarity measures. As it is impossible to test all the possible transforms we cannot be conclusive on the best transforms to use with a given dataset. However, we can argue that the original form may not be optimal and an undesirable transform may produce lower precision@50 values.

The cityblock and Euclidean distance measures with the rank transformed data in almost every case resulted in higher or similar precision@50 when compared with the best results produced by respective similarity measures with the original data or data transformed using the six functions. One notable exception is the $\sqrt{X}$ transform on the Cardiotocography dataset, which improves both cityblock and Euclidean performance, underscoring the importance of recognizing that the measure in which the data are expressed is not necessarily optimal with respect to judging similarity under an interval scale assumption. With the rank transformed data, the cityblock distance always performed better than the Euclidean distance. Except for the Cardiotocography dataset, the cityblock distance with the rank transformed data produced better or equally good results as the best results produced by all four distance based similarity measures: cityblock, Euclidean, cosine and Chebychev with the original or the data transformed by the other six functions. Based on these results we can conclude that the cityblock distance with rank transformed data is a better alternative to using cityblock, Euclidean, cosine and Chebychev with original data in CBMIR as the cityblock distance with rank transformed data being invariant to violations of the interval scale assumption, has an additional advantage over the rest.

ReFeat was affected by violations of the interval scale assumption and it did not produce better CBMIR precision than the cityblock distance with the rank transformed data or

[4] ReFeat works only with imbalanced trees. Sample size 2 can only produce balanced trees.



(a) GTZAN Music

(b) Steel Plates Faults

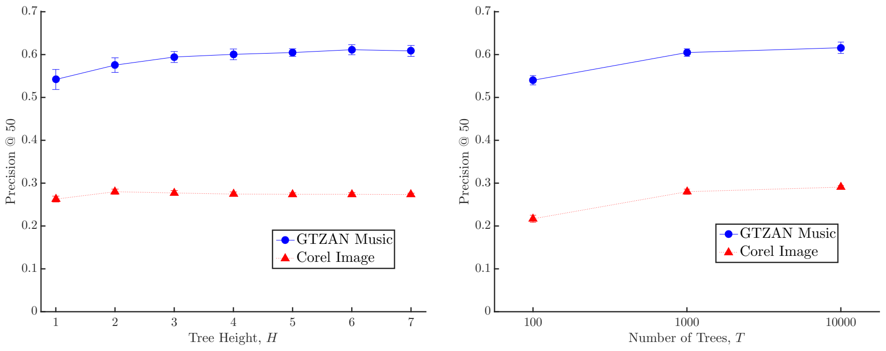
(c) Cardiotocography

(d) ISOLET

(e) Corel Image

(f) Letter Recognition

Fig. 5: Effects of violations of the interval scale assumption on feedback round 5 of CBMIR



(a) Tree Height, H (b) Number of Trees, T

Fig. 6: Effects of usForest parameters on feedback round 5 of CBMIR

SimUSF. In addition, ReFeat is not based on a similarity measure and therefore it cannot be used with different machine learning applications other than CBMIR.

The unsupervised random forest based similarity, SimURF produced competitive results only with the Cardiotocography and Letter datasets. In the other four cases it did not produce good results when compared with the cityblock distance with rank transformed data and SimUSF. It could beat the SimUSF results only in one case, the Cardiotocography out of the six tested cases of CBMIR. But the error bars show that the difference was not significant. The plots in Figure 5 show that, the SimUSF is very competitive with the best precisions produced by the other similarity measures in all six tested CBMIR cases. Further, SimUSF is invariant to violations of the interval scale assumption.

Figure 6 shows the results of the CBMIR experiments with the GTZAN Music and Corel Image datasets to study how parameters: the tree height (H) and number of trees (T) affect the precision@50 achieved by the SimUSF. In order to assess the effect of the tree height we fixed the number of trees to 1000 and tests were carried out for tree heights from 1 to 7. SimUSF produced good precision@50 over the tested range of H. When the tests were carried out with tree height 5, the precision@50 increased with the number of trees though the improvement was small when the number of trees increased from 1000 to 10000.

7 Computational complexity comparison

Table 4 shows the time complexities of calculating similarity between pairs of unseen instances. The complexities of the cityblock, Euclidean, cosine and Chebychev distances are identical. They usually require $O(nd)$ time for min-max normalization and they require $O(d)$ time for calculation of similarity between two instances. The tied rank transform and SimUSF do not require min-max normalization. They have the ranking and usForest building as preprocessing overheads respectively. The unsupervised random forest is not compared in this section as it could not produce good results in most of the cases. The ReFeat is also excluded as it does not calculate the pairwise similarity values between the instances.

Table 5 shows the average times taken to calculate the similarity between two unseen instances. The similarity values between 100 unseen instance pairs were calculated and the mean values were rounded to three significant digits. The experiments were carried out

Table 4: Time complexities of cityblock distance (CB), cityblock distance with ranked data (RCB) and SimUSF

	CB	RCB	SimUSF
Preprocessing	nd	$dnlog(n)$	$TH2^H$
Similarity Calculation	d	$dlog(n)$	$T2^H$

Table 5: Time taken in seconds for preprocessing (PP) and similarity calculation (SC) between a pair of unseen instances using cityblock distance, cityblock distance with ranked data (Cityblock-Ranked) and SimUSF.

Data		Cityblock		Cityblock-Ranked		SimUSF	
n	d	PP	SC	PP	SC	PP	SC
1000	10	1.76×10^{-3}	1.11×10^{-4}	1.76	3.17×10^{-2}	2.90	2.01×10^{-2}
1000	100	1.58×10^{-2}	1.12×10^{-4}	16.7	3.07×10^{-1}	2.90	2.01×10^{-2}
1000	1000	1.53×10^{-1}	1.36×10^{-4}	201	3.06	2.90	2.01×10^{-2}
10000	10	2.76×10^{-3}	1.11×10^{-4}	17.0	3.28×10^{-1}	2.91	2.02×10^{-2}
10000	100	2.54×10^{-3}	1.12×10^{-4}	176	3.01	2.90	2.02×10^{-2}
10000	1000	2.74×10^{-3}	1.38×10^{-4}	2090	32.0	3.10	2.01×10^{-2}

on a Windows 8.1 machine with 4GB memory and a dual core Pentium i5 processor. The programs were written in Perl and the compilation times were excluded.

As shown in Table 4 in contrast to using distance measures with rank transformed data time requirement of the SimUSF calculation is independent from the size of the training dataset. Table 5 shows that SimUSF needs less execution time than a distance based similarity measure used with ranked data when previously unseen instances are involved in calculations.

8 Discussion

In previous sections, the effects of violations of the interval scale assumption on a few popular distance based similarity measures were demonstrated. Two solutions which are impervious to such violations: using distance based similarity measures with rank transforms and the unsupervised random forests were discussed. Then, we introduced a novel similarity measure SimUSF, which is invariant to violations of the interval scale assumption. We analyzed SimUSF with respect to SimURF and performed similarity based ranking, DBScan clustering and CBMIR experiments to compare SimURF with the other similarity measures.

Out of the DBScan clustering and CBMIR experiments discussed in Section 6, in all but two cases, one in the DBScan clustering and the other in the CBMIR, the cityblock distance with the rank transformed data produced better or equally good results as the best results produced by all four distance based similarity measures: cityblock, Euclidean, cosine and Chebychev with the original or the data transformed by the other six functions. The results produced by rank transformed data are not affected by different data representations that violate the interval scale assumption. Therefore, we can argue that using the cityblock distance with rank transformed data is a better alternative to using the cityblock, Euclidean, cosine and Chebychev distances with original data if the time and storage requirements can be met. The major drawback of using rank transformed data is that when a previously unseen instances are involved in calculations the rank revaluation is computationally expensive and depends on the size of the dataset.

Though the unsupervised random forest based similarity, SimURF, is invariant to violations of the interval scale assumption, it did not produce competitive results in our DBScan clustering and CBMIR experiments except for 3 out of 12 experiments. Even in those cases it could not significantly beat the newly introduced unsupervised stochastic forest based similarity, SimUSF. As discussed in Section 4 SimURF cannot produce good results because SimURF attempts to segregate real instances from synthetic instances instead of attempting to collect nearest neighbors together to form clusters. Further SimURF produces large trees and it is therefore highly time and memory consuming. SimURF needs large number of trees to produce a good estimate of its expected value when compared with the SimUSF.

The new similarity measure, SimUSF which is based on our newly introduced unsupervised stochastic forest, usForest produced better results than all other tested measures in the DBScan clustering. This is because, the SimUSF successfully distinguishes nearest neighbors from the rest of the instances, as they are gathered at tree leaves of the usForest. The results produced by SimUSF was very competitive with the best results produced by the other similarity measure in the CBMIR in all six tested cases. Both in DBScan clustering and CBMIR, SimUSF could produce better results with usForests having 1000 usTrees with heights 4, 5 and 6. Hence, H and $H2^H$ in the SimUSF time complexities given in Table 4 are small. As shown under similarity based ranking SimUSF converges with its expected value with small number of trees when compared with the SimURF. It also produced equally high average rank correlation values with the two tested datasets where one was uniformly distributed and the other was normally distributed. This is because SimUSF uses ordinal scale calculations and therefore it is expected to produce equally good results irrespective of the underlying distribution of the original data in the interval scale. However, as number of instances and number of attributes of the dataset increase, SimUSF will need more usTrees in usForests to produce reliable estimate of its expected value. usTrees are expected to be smaller compared with the decision trees in unsupervised random forest. In contrast to using distance measures with rank transformed data the time and memory requirements of SimUSF are independent from the size of the training dataset. The SimUSF needs less execution time than a distance based similarity measure used with ranked data when previously unseen instances are involved in calculations. Therefore the SimUSF is a better alternative to using distance measures with rank transformed data.

9 Conclusion

Most distance measures assume that the numeric data are represented in the interval scale. However, rarely does data provide any clues as to whether this assumption holds. In this research we studied how common distance measures are affected by violations of the interval scale assumption. The results of our experiments suggest that the cityblock distance with tied rank transformed data can often produce good results. It is specially useful when violations of the interval scale assumption are present. However, using the rank transform has high time and space overheads when previously unseen instances are introduced. Though the $Addcl1$ unsupervised random forest based similarity is invariant to violations of the interval scale assumption it could not produce good precision values in the DBScan clustering and CBMIR experiments. Further, it has high time and memory overheads.

We introduced a novel unsupervised learning algorithm, unsupervised stochastic forest (usForest) and a similarity measure, SimUSF, on top thereof. With lower time and space overheads than the rank transform and unsupervised random forest, the SimUSF outperforms all other measures in our experiments on DBScan clustering. It produces very competitive results in CBMIR. SimUSF is also invariant to violations of the interval scale assumption. Therefore the SimUSF is the most successful out of the tested similarity measures in our experiments.

Acknowledgements

We are grateful to Francois Petitjean for valuable feedback and suggestions. This research has been supported by the Australian Research Council under grant DP140100087.

References

Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ (1990) Basic local alignment search tool. Journal of molecular biology 215(3):403–410

Ashby FG, Ennis DM (2007) Similarity measures. Scholarpedia 2(12):4116

Breiman L (2001) Random forests. Machine learning 45(1):5–32

Cha SH (2007) Comprehensive survey on distance/similarity measures between probability density functions. International Journal of Mathematical Models and Methods in Applied Sciences 1(4):300–307

Conover WJ (1980) Practical nonparametric statistics. Wiley, New York

Ester M, Kriegel HP, Sander J, Xu X (1996) A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proceedings of the Second ACM International Conference on Knowledge Discovery and Data Mining, pp 226–231

Faith DP, Minchin PR, Belbin L (1987) Compositional dissimilarity as a robust measure of ecological distance. Vegetatio 69(1-3):57–68

Giacinto G, Roli F (2005) Instance-based relevance feedback for image retrieval. Advances in Neural Information Processing Systems 17:489–496

Han J, Kamber M, Pei J (2011) Data mining: concepts and techniques, 3rd edn. Morgan Kaufmann

He J, Li M, Zhang HJ, Tong H, Zhang C (2004) Manifold-ranking based image retrieval. In: Proceedings of the 12th Annual ACM International Conference on Multimedia, ACM, New York, NY, USA, MULTIMEDIA ’04, pp 9–16

Lichman M (2014) UCI machine learning repository. URL http://archive.ics.uci.edu/ml. Accessed 22 Oct 2014

Manning CD, Raghavan P, Schütze H (2008) Introduction to Information Retrieval. Cambridge University Press, New York, NY, USA

Osborne J (2002) Notes on the use of data transformations. Practical Assessment, Research & Evaluation 8(6):1–8

Osborne JW (2010) Improving your data transformations: Applying the box-cox transformation. Practical Assessment, Research & Evaluation 15(12):1–9

Petitjean F, Gançarski P (2012) Summarizing a set of time series by averaging: From steiner sequence to compact multiple alignment. Theoretical Computer Science 414(1):76–91

Rocchio JJ (1971) Relevance feedback in information retrieval. In: Salton G (ed) The SMART Retrieval System: Experiments in Automatic Document Processing, Prentice-Hall, Englewood Cliffs NJ, pp 313–323

Shi T, Horvath S (2006) Unsupervised learning with random forest predictors. Journal of Computational and Graphical Statistics 15(1)

SIGKDD (2015) 2014 SIGKDD test of time award winners. http://www.kdd.org/awards/view/2014-sikdd-test-of-time-award-winners. Accessed 16 May 2015

Stevens S (1946) On the theory of scales of measurement. Science 103(2684)

University of Eastern Finland (2015) Clustering datasets. https://cs.joensuu.fi/sipu/datasets/. Accessed 19 Nov 2015

Zezula P, Amato G, Dohnal V, Batko M (2006) Similarity Search - The Metric Space Approach, Advances in Database Systems, vol 32. Springer

Zhang R, Zhang ZM (2006) BALAS: Empirical bayesian learning in the relevance feedback for image retrieval. Image and Vision Computing 24(3):211–223

Zhou G, Ting K, Liu F, Yin Y (2012) Relevance feature mapping for content-based multimedia information retrieval. Pattern Recognition 45(4):1707–1720

Zhou ZH, Dai HB (2006) Query-sensitive similarity measure for content-based image retrieval. In: Proceedings of the Sixth International Conference on Data Mining, IEEE Computer Society, Washington, DC, USA, ICDM ’06, pp 1211–1215