Similarity to a Single Set

Lee Naish

Computing and Information Systems,
The University of Melbourne, Melbourne 3010, Australia
lee@unimelb.edu.au,
http://people.eng.unimelb.edu.au/lee/

Abstract. Identifying patterns and associations in data is fundamental to discovery in science. This work investigates a very simple but fundamental instance of the problem, where each data point consists of a vector of binary attributes. For example, each data point may correspond to a person and the attributes may be their sex, whether they smoke cigarettes, whether they have been diagnosed with lung cancer, etc. Our primary application is spectral based fault localisation (SBFL), in which each point represents a test case for a computer program and the attributes are whether the program failed the test and whether certain parts of the program were used during the test. Measuring similarity of attributes in the data is equivalent to measuring similarity of sets. Furthermore, there is one identified "base" set and only similarity to that set is considered—the other sets are just ranked according to how similar they are to the base set. For example, if the base set represents lung cancer sufferers, the set of smokers may well be high in the ranking. In SBFL the base set represents the failed tests and the ranking is used to help find bugs. Identifying set similarity or correlation has many uses and is often the first step in determining relationship or causality. Set similarity is also the basis for comparing binary classifiers such as diagnostic tests for any data set. More than a hundred set similarity measures have been proposed in the literature but there is very little understanding of how best to choose a similarity measure for a given domain. This work discusses numerous properties that similarity measures can have and identifies important forms of symmetry which have not previously been considered. It gives alternative versions of various previously defined properties so they are no longer incompatible, defines ordering relations over similarity measures and shows how some properties of a domain can be used to help choose a similarity measure which will perform well for that domain.

Keywords: binary similarity measure, set similarity, STASS, data mining, clustering, classification, diagnostic test, spectral based fault localization

1 Introduction

Recognition of associations in data is an important contributing factor to progress in science. To give an (over simplified) example, the correlation between cigarette smoking and lung cancer was recognised, prompting the hypothesis that a causal relationship exists and this has lead to further research and eventual acceptance of the hypothesis. Refinement of our understanding of the mechanism and appropriate social policy is ongoing. Spectral based fault localisation (SBFL) has recently been a very active research area in software engineering [1–7] and uses the same kinds of associations in data. There is a definite causal relationship between buggy code being executed and incorrect behaviour of a program so correlations between code execution and test case failure can be used to help locate bugs. With the phenomenal growth in volume of data, automated discovery of patterns in data is becoming more important. This paper considers one instance of the problem that is particularly simple but has been applied in many different domains. Specifically, it is the case where data points are vectors of binary attributes, thus attributes can be viewed as sets. Furthermore, there is one distinguished "base" set that all other sets are compared to, resulting in a ranking of all sets. Here this is called the "similarity to a single set" or *STASS* problem. Even when the raw data is not binary, we are often interested in binary classifications such as diagnostic tests, and their relative merit. Each possible classification or test can be considered a set (albeit inferred rather than directly measured), with the base set being the gold standard or ground truth. Thus comparison of binary classifiers for any data set is an instance of the STASS problem. Furthering our understanding of the STASS problem has been helpful in SBFL [5] and may well be helpful in many other domains. Furthermore, some of the insights may be applicable to discovering associations in more general cases.

There are three areas in which this paper contributes. The first is formally defining the STASS problem and distinguishing it from other more general problems of similarity. STASS has direct applicability. For example, much of the SBFL literature is simply finding better ways to compute similarity to the set of failed tests, so that bugs appear higher in a ranking of program components. STASS also has some characteristics that are not shared with more general problems of similarity. In studying the more general problems we miss opportunities for refinement that are available for STASS. The second contribution is a comprehensive examination of properties that set similarity measures may have, in the STASS context. This includes several refinements of properties that have been proposed for more general contexts and properties related to forms of symmetry that have not previously been proposed. The third contribution is a way to use domain knowledge to help choose a set similarity measure that performs well for the domain. The choice of set similarity measure determines performance but there are no established ways to help choose a measure that performs well. Many authors choose similarity measures based on what measures other authors in the same field choose [8] or experiment with many measures and empirically determine which ones perform well for their data—see Section 2.5. This paper does not solve the difficult problem of choosing a set similarity measure but does provide some guidance.

The paper is structured as follows. Section 2 gives a brief introduction SBFL (the running example used here) and to set similarity. It precisely defines the problem addressed, discusses set similarity measures, how similarity to a single set relates to other problems concerning similarity and discusses some related papers that investigate larger collections of set similarity measures and properties of these measures. Section 3 defines properties measures may have, and some relationships between these properties. In several cases these are weaker (more widely applicable) versions of properties proposed elsewhere in the literature and new forms of symmetry are introduced. Section 4 focusses on one such form of symmetry. It discusses how measures vary according to how much importance or weight they give to "true positives" versus "false positives", and how this relative weight can be used for ordering measures. Section 5 uses this ordering to propose a general way in which domain knowledge can be used to help choose a measure that performs well for that domain. The approach is validated by experiments from the software debugging domain. Section 6 concludes.

2 Set similarity

Section 2.1 reviews how set similarity is used in SBFL. The STASS problem is then defined formally and the rest of this Section discusses how measuring set similarity can be viewed and briefly discusses more general similarity problems.

2.1 Spectral based fault localisation

SBFL runs a program on each of a set of test cases for which the correct program behaviour is known. Tests either *succeed* (the program behaves correctly) or *fail* (the program behaves incorrectly). The program is instrumented so that information about its execution ("program spectra") is obtained. The examples used in this paper use "statement spectra"—information about which statements or lines of code are executed in each test. Thus we can count the total number of passed and failed tests and, for each statement, the number of passed and failed test for which the statement was (and wasn't) executed. A numeric function is applied to these values to estimate how likely it is that each statement is buggy and this is used to rank the statements. The ranking can be used to guide the search for bugs. Performance of SBFL is most often measured by the position of the top-ranked bug within the ranking as a percentage of the number of statements.

Table 1 gives an example with just five test cases and three statements. The results of the test cases can be represented as a binary vector, where 1 means the test case failed and 0 means the test case passed. The statement spectra can be represented as a binary matrix with a row for each statement, where 1 means the statement was executed in the test case and 0 means it was not. Equivalently, the results can be represented as the set of test cases which failed, $\{C_1, C_2\}$ and the spectra for each statement can be represented as the set of test cases in which the statement was executed. For statements S_1, S_2 and S_3

these are $\{C_1, C_4\}$, $\{C_1, C_2, C_4\}$ and $\{C_1, C_2, C_3, C_5\}$, respectively. The numeric function used to produce the ranking can be considered a measure of similarity between the set of test cases which fail and the set of test cases in which the statement was executed—it ranks these three sets according to how similar they are to $\{C_1, C_2\}$. In this example it seems reasonable to rank S_2 most highly as it is executed in the maximal number of failed tests and the minimal number of passed tests. Deciding the relative ranking of S_1 and S_3 is harder because S_3 is executed in more failed tests and more passed tests. The set similarity measure is defined in terms of the aggregate information on the right of Table 1. This gives the cardinalities of the sets, their complements, intersections, etc. For example, the number of passed test cases in which a statement was executed is the cardinality of the intersection of the set of test cases in which it was executed and the complement of the set of test cases that failed.

Table 1. Spectra for statements $S_1 \ldots S_3$ with test cases $C_1 \ldots C_5$

	C_1	C_2	C_3	C_4	C_5	exec. & failed	exec. & passed
S_1 executed	1	0	0	1	0	1	1
S_2 executed	1	1	0	1	0	2	1
S_3 executed	1	1	1	0	1	2	2
					$\vdots$		
Test case failed	1	1	0	0	0	Tot. failed = 2	Tot. passed = 3

The terminology used below follows the machine learning and data mining literature. Often data points called *instances* are classified according to their *features*. SBFL does not classify statements as buggy or correct, but instead produces a ranking of statements from mostly likely buggy to most likely correct. Thus the instances are the rows of the table—statements and test results. The test results have a special status as all the other instances are compared to this instance. The features are the columns of the table, corresponding to test cases. For statements this is whether the statement is executed and for test results it is whether the result is failure. For SBFL the number of instances is fixed by the program we are debugging whereas the number of features is unbounded because we can always come up with new test cases. In other domains the features can be fixed first and the number of instances can be unbounded. SBFL is also unusual in that we know a causal relationship exists between execution of buggy statements and test case failure. We hope there is a corresponding correlation in the data. In many domains we may find a correlation in the data and hope it corresponds to a relationship.

2.2 The STASS problem

Consider a universe of *features* C of finite cardinality T and some *instances* which are subsets of C. One instance is identified as the *base instance* (or *base set*) B and there are an additional K instances. For each additional instance A_k, we compute a numeric measure of the similarity of A_k and the base instance. These numeric measures are used to rank the instances according to how well they correlate with the base instance. A strong positive correlation is often of interest and in some domains a high negative correlation is also of interest. The STASS problem is formally defined as follows:

Definition 1 (STASS problem). *Given a finite set C, a set $B \subseteq C$ and a finite number of sets $A_k \subseteq C$, $1 \leq k \leq K$, the STASS problem is to find a ranking of the sets A_k, $1 \leq k \leq K$ (in injection from $\{1 \ldots K\}$ to $\{1 \ldots K\}$) which gives the relative similarity of each A_k to B.*

Note that there may be ties in the ranking. This paper uses M to denote the cardinality of the base instance (the number of *members*) and N the cardinality of its complement (the number of *non-members*), so $N = T - M$. In SBFL M is the number of failed tests and N is the number of passed tests. For each additional instance k, m_k is the number of features in instance k and the base instance (the number of *matches*), and n_k is the number of features in instance k but not the base instance (the number of *non-matches*). In SBFL m_k is the number of failed tests that execute statement k and n_k is the number of passed tests that execute statement k. Similarity measures are evaluated separately for all K instances, so typically the subscripts are left implicit.

Because the raw data can be viewed as a binary matrix, the role of instances and features can be swapped (equivalent to swapping the rows and columns of the matrix)—a form of symmetry. For the SBFL domain this would result in a ranking of test cases. Although this may be of some interest, it does not solve the problem addressed in SBFL, which is to produce a ranking of statements. Several forms of symmetry are important for the STASS problem but this is not one of them (see Section 3); it is always instances that are ranked. In the lung cancer example given earlier, we want to rank attributes such as being a smoker, so these attributes are the instances. Specifically, the set of people who are smokers is one instance and the people in the set are features (depending on how the STASS problem is applied to a particular data set, the terminology can be counter-intuitive). Throughout, the measurements of similarity are between a set of features that is the base instance and a set of features that is another instance. The following notation is used for comparing measures of set similarity (and other numbers):

Definition 2 (result of comparison). *The* result of comparison *of two numbers x and y, $C(x, y)$, is 1 if $x > y$, 0 if $x = y$ and -1 if $x < y$.*

2.3 Set similarity measures

It is common to present a single set comparison as a two by two *contingency table* as follows, where B is the base instance and A some other instance:

	B	$\bar{B}$
A	m	n
$\bar{A}$	o	p
Total	M	N

The table is also known as a confusion matrix, with m, n, o and p representing counts of "true positive", "false positive" (type I error), "false negative" (type II error) and "true negative" features, respectively. Set similarity measures are often defined by functions over m, n, o and p, or these values divided by T, giving the relative frequency (it is assumed $T > 0$). In this paper there is no importance placed in the absolute measures of similarity (they are often arbitrary in any case), just whether one pair of sets is more or less similar to another pair of sets. Furthermore, only the relative similarity of sets to a fixed base set is important: multiple contingency tables with different instances A_k but the same base instance B. Thus M and N are the same in all contingency tables where similarity measures are compared. For this reason, measures are defined in terms of M, N, m and n (the information content is the same since $o = M - m$ and $p = N - n$). The pair (M, N) is a *domain* and the pair (m, n) a *point* in the domain ($0 \leq m \leq M$ and $0 \leq n \leq N$). In a STASS problem the base instance fixes the domain and each instance corresponds to a point.

Though these variables are all natural numbers, in some circumstances we are interested in how similarity measures scale. Section 4.1 requires similarity measures to be defined for some points where m and n are non-integral rationals. Here measures are generalised further to allow real numbers, thus supporting the familiar definitions and properties associated with functions over reals. Similarity measures are often defined over rationals (in terms relative frequencies) and all proposed measures the author is aware of can be generalised to functions over reals.

Name	Formula	Name	Formula
Jaccard	$\frac{m}{M+n}$	Tarantula	$\frac{\frac{m}{M}}{\frac{m}{M}+\frac{n}{N}}$
Russell and Rao	$\frac{m}{M+N}$	Zoltar	$\frac{m}{M+n+\frac{10000M-m*n}{m}}$
Simple Matching	$\frac{m+N-n}{M+N}$	Ochiai	$\frac{m}{\sqrt{M*(m+n)}}$
Faith	$\frac{m+\frac{1}{2}(N-n)}{M+N}$	Pearson	$\frac{Nm-Mn}{\sqrt{MN(m+n)(M+N-m-n)}}$
Ample2	$\frac{m}{M} - \frac{n}{N}$	Ample	$\left\lvert \frac{m}{M} - \frac{n}{N} \right\rvert$
Op	$m - \frac{n}{N+1}$	Added Value	$\frac{m}{\max(M,m+N-n)}$
Wong3	$m-h, \text{ where } h = \begin{cases} n & \text{if } n \leq 2 \\ 2+0.1(n-2) & \text{if } 2 < n \leq 10 \\ 2.8+0.001(n-10) & \text{if } p > 10 \end{cases}$		

Fig. 1. Some of the many proposed set similarity measures

Definition 3 (Set similarity measure). *A* (set similarity) measure *is a partial function from a pair of natural numbers (M, N) and a pair of non-negative real numbers (m, n) to a real number. It is undefined if $m > M$ or $n > N$ or $M = N = 0$. The application of a measure f will be written $f_N^M(m, n)$ to emphasise that the domain (M, N) is the same for all instances that appear in the same ranking and (m, n) is a point in the domain.*

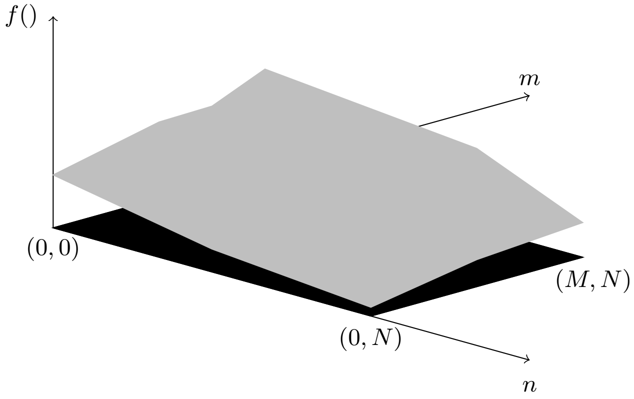

Fig. 2. Domain and plot of a measure for fixed M and N

Figure 1 defines a small sample of measures proposed in the literature. All these have been evaluated for SBFL [2] and some, such as Tarantula, Zoltar, Wong3, Ample, Ample2, and Op were developed specifically for debugging. Other measures have been developed for and used in many different domains. Jaccard [9] was developed for botany, Ochiai [10] was developed for marine zoology (it is also known as Cosine in other areas) and both have been used in many other domains. As well as botany and software engineering, Jaccard has been used in disciplines such as ecology [11], chemistry [12], genetics [13], paleontology [14], physics/mathematics [15] and psychology [16], to give just a few examples.

Different measures can be equivalent in that they produce identical rankings [2, 17–19]. For example, Tarantula gives the same ranking as m/n and $m/(m+n)$ (known as precision or positive predictive value and called error detection accuracy in software debugging). Note that many proposed measures are undefined for some points, particularly for $m = n = 0$, due to division by zero or taking logarithms of zero. It practice it may be necessary to add special cases to deal with such exceptions [20], and/or add a small constant [5]. Also, some literature uses measures of dissimilarity or "distance" rather than similarity or closeness. For example, the Jaccard distance is one minus the Jaccard similarity measure. To measure similarity we can take any distance measure and negate it (or apply any monotonically decreasing function).

Given a domain (M, N), a measure f can be viewed as a surface in three dimensions m, n and $f()$, within a rectangle with $0 \leq m \leq M$ and $0 \leq n \leq N$ (see Figure 2)—points in the domain are ranked according to their f value, or height of the surface at that point. In general, there are sixteen distinct symmetric variants of such a surface, which can be obtained by using subsets of the following operations.

1. Reflection in a plane with constant f value. This paper refers to this as antisymmetry rather than symmetry. It inverts the surface and reverses the ranking. Since only the relative f values are important in STASS, it makes no difference which plane of constant f value is used.
2. Reflection in the plane $m = M/2$. This "inverts" the m values while keeping the domain the same. Our notation uses $\bar{m}$ rather than m to indicate this.
3. Reflection in the plane $n = N/2$. This "inverts" the n values while keeping the domain the same. Our notation uses $\bar{n}$ rather than n to indicate this.
4. Reflection in the plane $m = n$. This effectively swaps m with n and M with N. Our notation has n appearing before m in a prefix or superscript to indicate this.

For example, $\bar{m}\bar{n}$-antisymmetry refers to a combination of the first three operations and nm-antisymmetry refers to a combination of the first and last. These and other symmetries are discussed more in Section 3.

2.4 Related similarity problems

What this paper calls a domain corresponds to a "coverage space" in the PN analysis of [21]. Receiver operating characteristic (ROC) analysis (see [22]) uses a version of this space scaled to the unit square. ROC curves plot the true positive rate (TPR, m/M, "hit rate", "sensitivity", d' or "recall") against the false positive rate (FPR, n/N, "false alarm rate", "fall-out" or $1 -$ "specificity"). Any set similarity measure can be used to derive a binary classifier by simply providing a threshold for the value of the measure. The threshold corresponds to a single contour or iso-metric of the surface which divides the domain in two, and each instance or set is mapped to true or false depending on whether the similarity measure exceeds the threshold:

Definition 4 (Set Similarity Classifier). *Given a set similarity measure f and real number threshold α, the* set similarity classifier *f_c^α is defined as follows. For domain (M, N) (corresponding to a base instance B) and natural numbers m and n (corresponding to another set or instance),*

$$f_c^\alpha(M, N, m, n) = f_N^M(m, n) > \alpha$$

ROC analysis is widely used to compare and visualise the effectiveness of classes of binary classifiers as the threshold is adjusted. Often the area under the ROC curve is used as a measure of effectiveness. ROC analysis is discussed further in Section 5.4. As mentioned in the Introduction, comparison of binary classifiers of a data set is also an instance of STASS.

The STASS problem is closely related to the problem of measuring evidential support or confirmation: determining the extent to which evidence E confirms a hypothesis H. Statistical hypothesis testing is the cornerstone of much of science. For each instance A_k in a STASS problem we can have a null hypothesis that A_k and B are independent and an alternative hypothesis that there is some (perhaps causal) relationship between A_k and B. For example, A_k may be a buggy statement which causes at least some of the test case failures represented by B. Hypotheses are typically tested by choosing a test statistic t, computing its value t_{obs} from the data and comparing this to some threshold α (often 0.05 or 0.01). The value t_{obs} can typically be interpreted as the probability of the null hypothesis holding, and the null hypothesis is rejected if and only if t_{obs} is below α. Note that when testing multiple hypotheses, α should be lowered to ensure there is a sufficiently small probability that *none* of the corresponding null hypotheses are rejected purely by chance. In the STASS context there is no threshold α but the test statistic t can be viewed as a set distance measure and used to rank the instances based on the t_{obs} for the corresponding hypotheses. It is tempting to think the highest ranked instances are those with the most plausible alternative hypotheses, though it is actually those with the least plausible null hypotheses.

STASS is an instance of the more general problem of comparing similarity of arbitrary pairs of sets. Sets can be viewed as points in T-dimensional space and various related mathematical abstractions can be used. For example, some (dis)similarity measures obey the axioms of *metrics* in metric spaces [23]. However, human perception of similarity does not appear to be compatible with similarity measures being metrics [16]. For some applications such as image retrieval, human perception of similarity is all important, whereas for others it is not. Unsurprisingly, there is little concensus on what axioms or properties should (or should not) apply to similarity measures in general. For STASS, whether a similarity measure is a metric (or Euclidean) is not important (the "triangle inequality" axiom is irrelevant because all comparisons involve a single set).

There are two main ways in which the problem of comparing arbitrary pairs of sets can be generalised further. The first is measuring correlation where features are not binary. They can be one of a relatively small number of values for which no natural ordering exists. Alternatively, they can have a larger number of values, often with some natural ordering, allowing them to be mapped to integers or real numbers, for example. Thus an instance may be a vector of real numbers rather than a vector of binary numbers or set. An example is the use of *fuzzy sets* [17, 24–26], where elements have a degree of membership of a set anywhere in the range 0–1. Many ways of measuring correlations for such instances have been devised and they can be applied to the simpler case of binary data. Some measures that are distinct in the general case are equivalent in the binary case, and measures that are distinct for arbitrary pairs of sets may be equivalent for STASS.

The second way the set similarity problem can be generalised is by structuring the sets in a more complex way, rather than just ranking them. This can be done for both binary and non-binary instances. For example, we may want to identify "clusters" of instances where the similarity of pairs of instances within a cluster is relatively high and the similarity of pairs of instances in different clusters is relatively low. Additionally, we may want a hierarchical structure such as a dendrogram or decision tree for classification or other purposes. Or we may want to extract interesting "association rules" which relate different instances.

For association rules, there is generally a distinction made between *discriminant* rules and *characteristic* rules—see [27], for example. Discriminant rules are of the form $E \to B$, where E is a combination of one of more instances. They can be seen as a form hypothesis about a cause for the base instance B (in general, the conclusion may also be a combination of instances). Characteristic rules are the converse, $B \to E$, and can be seen as a way of describing the base instance. The STASS problem is often used to rank possible causes of the base instance, and hence it can naturally be viewed as ranking discriminant rules, where E is also restricted to be a single instance. However, STASS can equally be used to rank possible effects. For example, the base instance may be taking some new drug and the other instances may be possible effects. Thus the STASS problem can apply to both discriminant and characteristic rules.

Problems such clustering, hierarchical classification and association rule mining all have some notion of similarity at their core. Furthermore, the algorithms proposed to solve such problems often have steps that measure and rank similarity and a deeper understanding of the STASS problem may have implications for these more general problems. One characteristic of the approach used here that makes it simpler than many other approaches to problems of similarity is that all features are given equal importance (a set similarity measure is applied to each instance separately in order to obtain the ranking). Features can be given varying importance for both information-theoretic and domain specific reasons. A feature that is in nearly all instances or very few instances provides little discrimination between instances and has little information content, whereas a feature in exactly half the instances has maximal discrimination and information content. Rather than simply counting the number of features in a particular instance, it may be desirable to give more weight to features with more information content.

There are also application areas where domain knowledge suggests quite different relative importance. In SBFL a failed test case that executes very few statements is particularly helpful for locating a bug and it is rational to give it high weight [1], whereas naive use of information theory would give it low weight. Our research group has experimented with many more sophisticated machine learning algorithms and tools using software debugging data sets, including various forms of support vector machines, decision trees, nearest neighbour and association mining. The results have been poor compared with the simple SBFL methods. This may be in part due to the mis-match between information theory and domain knowledge noted above. Another factor may be the very uneven "class distribution"—in practice there are always *far* more correct statement than buggy statements and the most extreme case, where programs have a single bug, dominates many data sets. Many machine learning algorithms are known to perform relatively poorly under these circumstances. Thus although the simple approach of treating all features equally used here seems naive from the general machine learning perspective, there are domains where it performs relatively well and the insights gained from the STASS problem may well be useful more widely.

2.5 Collections of similarity measures

There have been numerous papers from a variety of disciplines that survey similarity measures for sets and (in some cases) non-binary data, discuss properties of such measures and/or compare them empirically in various ways. Here several of them are discussed but the list is not exhaustive. [28] [29] and [30] consider several similarity measures used for evidential support (or confirmation), and discuss various properties such as forms of symmetry. [31] discusses 21 measures used for association rule mining, properties of those measures including symmetries, how several measures can become equivalent if contingency tables are normalised in various ways, and suggests how a relatively small set of representative tables can be generated that can help a domain expert choose between different measures. [27] discusses properties of measures of "interestingness" of association rules, focussing on the difference between discriminant and characteristic rules and compares 11 measures. [32] discusses 20 measures used for association rule mining, properties of those measures and how they can be compared. [33] surveys measures of "interestingness" of association rules and properties of those measures. Thirty eight measures for ranking rules are discussed; methods for filtering rules are also discussed. Both "objective" and "subjective" (more domain dependent) measures are discussed. [18] discusses 14 set similarity measures and various properties. [20] discusses and compares 22 set similarity measures and several properties and gives dendrograms derived empirically from data sets. [34] discusses 51 similarity measures used in the (more general) non-binary case, and compares/classifies them, including a dendrogram. [35] discusses 76 set similarity measures and compares/classifies them, including a dendrogram. [8] investigates performance of 18 similarity measures for comparing structural similarities in software source code entities. [7], expanding on [2] and other work compares 157 similarity measures for SBFL. Similarity measures have also been developed and evaluated for SBFL automatically, using methods such as genetic programming [4, 6], leading to numbers of similarity measures in the hundreds of thousands at least.

3 Properties of set similarity measures

This Section discusses properties of set similarity measures in the context of the STASS problem. Many of the properties of similarity measures previously proposed and discussed [27, 31–33] are overly strict for the STASS problem. For STASS we are only concerned with relative measures of similarity for different

A_i rather than the absolute measure computed for a single A_i. Furthermore, we are only concerned with relative similarity within a single domain—M and N are the same in any comparison that is relevant to a STASS problem.

3.1 Uniform scalability

Many measures have the intuitive property that the ranking is preserved if M, N, m and n are all multiplied by some scaling factor s. For example, if we collect data from T features to get some ranking and independently collect data from another T features that happens to be identical, combining the $2T$ features should (one would expect) result in the same ranking. This idea can be expressed by saying that a similarity measure is invariant under scaling of all parameters. The following is a specialised version of the definition of [31] (the more general version is discussed in the next section).

Definition 5 (absolute uniform scalable measure). *A measure f is* absolute uniform scalable *if for all points where f is defined and all positive s*

$$f_N^M(m,n) = f_{sN}^{sM}(sm,sn)$$

Although this definition captures the intuition in a reasonable way, it is stricter than necessary for STASS because it uses equality of measures of single points in different domains. The following refinement is a weaker definition, which says that the result of comparison of similarity measures for two points is invariant under uniform scaling:

Definition 6 (uniform scalable measure). *A measure f is* uniform scalable *if for all points where f is defined and all positive s*

$$C(f_N^M(m,n), f_N^M(m',n')) = C(f_{sN}^{sM}(sm,sn), f_{sN}^{sM}(sm',sn'))$$

It is clear that any absolute uniform scalable measure is a uniform scalable measure. Uniform scalability effectively reduces degrees of freedom by one—we can fix one of the parameters by choosing an appropriate s value. Existing measures are defined using formulas in which the variables are not restricted to be natural numbers, so the scaling can result in fractional numbers without problems (if parameters must be natural numbers then the fixed parameter value must have appropriate factors). Similarity measures are often defined in terms of relative frequencies (m/T, n/T, etc.), but this can only be done for uniform scalable measures (dividing by T is not the only way of normalizing a contingency table [31, 25]). All definitions in [31] use relative frequencies, as do all but Laplace Correction in [33] (thus the table in [33] showing they are all absolute uniform scalable is unsurprising). In [32] there are definitions for all measures in terms of m, n, etc. and also in terms of relative frequencies where possible. Some of the latter formulas also use the total number of features, T, resulting in measures that are uniform scalable but not absolute uniform scalable.

There are also measures proposed that are not uniform scalable, such as Wong3. If such measures are used, the number of features T should be carefully considered as it generally affects the ranking produced. More data may not give more reliable results, for example. Wong3 is a piecewise-linear function and the constants were (at least in part) chosen to get good performance for a particular data set. The best choice of constants depends on both the nature of the programs and test cases and the size of the data set. In [2] the influence of the test suite size on performance of various measures is investigated and although the Wong3 performance is close to optimal for large test suites the performance for smaller test suites is relatively poor. In [6] another class of measures is proposed and machine learning is used to determine various constants. Because measures in this class are uniform scalable the constants learned and the performance should be less dependent on the size of the data set.

3.2 General scalability

For some measures, the base instance and its complement can be scaled separately without affecting the ranking. Absolute uniform scalability can be generalised as follows; in [31] this is called "row/column scaling invariance".

Definition 7 (absolute general scalable measure). *A measure f is* absolute general scalable *if for all points where f is defined and all positive integers s and t*

$$f^M_N(m,n) = f^{sM}_{tN}(sm,tn)$$

Absolute general scalability effectively reduces degrees of freedom by two. Any such measure can be defined in terms of m/M and n/N. Alternatively, we can fix both M and N by suitable scaling, and then just have a function over m and n. As before, we can refine this to have a weaker condition that avoids comparison of measures for different M and N.

Definition 8 (general scalable measure). *A measure f is* general scalable *if for all points where f is defined and all positive integers s and t*

$$C(f^M_N(m,n), f^M_N(m',n')) = C(f^{sM}_{tN}(sm,tn), f^{sM}_{tN}(sm',tn'))$$

Commonly used measures are typically uniform scalable but not general scalable. For example, of the 21 measures investigated in [31], all are uniform scalable but only three ("odds ratio" and two variations of it) are general scalable. Using measures that are not general scalable should only be done with careful consideration of the relative M and N values (the class distribution or skew). Often there is an arbitrary relationship between M and N due to the way data is collected. For example, if we are attempting to identify possibly causes for are rare disease, M is likely to be limited to the number of individuals with the disease who can be contacted and are willing to participate in the study and N is likely to be chosen to be some "reasonable" size, somewhat similar to M but constrained by cost and an attempt to make the set of controls similar to those with the disease in terms of age etc. The ratio of M to N is nothing like the ratio of people with the disease to those without the disease in the general population, for example, and changing this ratio by adjusting the number of controls may systematically influence the results for measures that are not general scalable. In [2] the influence of the proportion of failed tests on performance of various measures is investigated. Most measures considered are not general scalable and their performance steadily increases as the proportion of failed tests increases but this is not the case for Ample2 (which is general scalable).

ROC analysis deliberately scales the domain to eliminate the effects of class distribution when comparing classifiers. However, class distribution can contain valuable information for some domains. For example, in SBFL, a very small proportion of test cases failing suggests there few bugs and/or execution of a buggy statement rarely leads to failure, whereas a large proportion of test cases failing suggests there are multiple bugs and/or execution of a bug leads to failure relatively frequently. This information can be useful for finding a similarity measure that performs well for locating bugs [5]. The class of measures in [6] is not general scalable and our experiments indicate similar classes that are general scalable do not perform as well.

3.3 Null invariance

Adding more features that are neither in the base instance or any other instance arguably (in some domains) should not affect the ranking. In SBFL we may avoid gathering spectra from "trusted" code such as libraries and it seems rational that adding additional test cases which succeed and only execute trusted code should not affect the ranking. An "absolute" definition is given in [31] (such measures are called "Type I" by some authors [18]) and this can be refined to a weaker version:

Definition 9 (absolute null-invariant measure). *A measure f is* absolute null-invariant *if for all points where f is defined and all k*

$$f_N^M(m,n) = f_{N+k}^M(m,n+k)$$

Definition 10 (null-invariant measure). *A measure f is* null-invariant *if for all points where f is defined and all k*

$$C(f_N^M(m,n), f_N^M(m',n')) = C(f_{N+k}^M(m,n+k), f_{N+k}^M(m',n'+k))$$

Advantages of our weaker definitions of null-invariance and general scalability are discussed after Proposition 2. Absolute null-invariant measures can be defined in terms of m, n and o (independent from p), as done in [16, 26, 25, 24], or m, n and M in our notation (this is not generally the case for null-invariant measures). Independence from p or a form of monotonicity with respect to p is suggested in [19].

3.4 Monotonicity

STASS measures *similarity* of sets (rather than dissimilarity or distance). Thus we can expect measures to be (strictly) increasing in m and (strictly) decreasing in n. In SBFL, statements which are executed in more failed tests or fewer passed tests should be considered more likely to be buggy.

Definition 11 (monotone measure). *A measure f is* monotone *if it is monotonically increasing in m and monotonically decreasing in n: for all points where f is defined we have*

$$\begin{array}{ll} C(m,m') &= C(f_N^M(m,n), f_N^M(m',n)) \\ C(n,n') &= -C(f_N^M(m,n), f_N^M(m,n')) \end{array}$$

This can be separated into two separate properties, as in [36, 31, 37]. In [3] the term "strictly rational" is used and a weaker definition of "rational" is given where measures must be increasing in m and decreasing in n, but not strictly so (this is not sufficient for the theoretical results in [3]). Similar non-strict definitions are given elsewhere [25, 26, 17]. In [19] strictness is required except for minimal and maximal values and there are alternative strict formulations that assume absolute null-invariance [16]. Monotonicity implies that where a measure f is differentiable, the partial derivative with respect to m is positive and the partial derivative with respect to n is negative. It also implies that the base instance itself, $(M, 0)$, has the highest similarity measure (called maximality in [18]) and its complement, $(0, N)$, has the lowest.

Several proposed measures are monotone for nearly all their domain. For example, the Jaccard measure is monotone with the exception of when $m = 0$, in which case it is always zero, rather than strictly decreasing in n (so it complies with the definition of [19]). By slightly modifying its definition it can be made monotone. For example, we can tweak the numerator so it is never quite zero using the following function, where ϵ is some sufficiently small constant, such as 10^{-9}:

$$tnz(x) = \begin{cases} \epsilon \text{ if } x = 0 \\ x \text{ if } x \neq 0 \end{cases}$$

If we define Jaccard-m as $tnz(m)/(M+n)$ we obtain a monotone measure which is the same as Jaccard except when $m = 0$. tnz is also useful for adapting some proposed measures to avoid division by zero and taking logarithms of zero. Such apparently minor adjustments to make measures monotone can lead to significantly improved performance for SBFL in some cases [5].

Certain other proposed properties of measures are incompatible with monotonicity.

Proposition 1. *If f is both an absolute general scalable and absolute null-invariant measure then for all points where f is defined,*

$$f_N^M(m,n) = f_N^M(m,n')$$

Proof. We assume w.l.o.g. that $n' > n$. $f_N^M(m,n) = f_{tN+k}^M(m, tn+k)$, since f is absolute general scalable and absolute null-invariant. Let $t = (N - n')$ and $k = N(n' - n)$. Thus $tN + k = N(N - n') + N(n' - n) = N(N - n)$, and $tn + k = n(N - n') + N(n' - n) = n'(N - n)$. Thus $f_N^M(m,n) = f_{N(N-n)}^M(m, n'(N - n)) = f_N^M(m,n')$, due to absolute general scalability.

Thus if these two "absolute" properties hold, the measure is independent of n—given parameters M and N, it is a function of the single variable m. [25] assumes null-invariance and normalises contingency tables, leading to measures of "satisfiability" that have a single parameter. Satisfiability is similar to STASS in that there is a fixed base set or class that other sets are compared to, but fuzzy sets are used and absolute measures of similarity are considered instead of just the ranking.

Corollary 1. *If f is both an absolute general scalable and absolute null-invariant measure, f is not monotone.*

Proof. If $n > n'$ then $C(n, n') = 1$ whereas $C(f_N^M(m, n), f_N^M(m, n')) = 0$.

Although these results hold for the "absolute" definitions of scalability and null-invariance, they do not hold for our weaker variants. There are measures that are general scalable, null-invariant and monotone. For example:

Proposition 2. *Measure Op is general scalable, null-invariant and monotone.*

Proof. Straightforward, since $Op_N^M(m, n) > Op_N^M(m', n')$ if and only if $m > m'$ or $m = m'$ and $n < n'$.

To explore the difference in the two versions of general scalability and null-invariance, let us consider *Op* in more detail. It is designed so that the factor for n is much smaller than that of m, so any change in m is more significant than the maximum possible change in n. We can have a similar absolute general scalable measure such as $m/M - \epsilon n/N$, where ϵ is very small (but positive, to ensure monotonicity). Alternatively, we could use an absolute null-invariant measure such as $m/M - \epsilon n$. However, in both cases there will be some N and M (or scaling factors) where m does not dominate over n. There is no absolute general scalable measure or absolute null-invariant measure that results in the same rankings as *Op* in all cases. *Op* has been shown to be optimal for SBFL of programs with a single bug—no other monotone measure produces a better ranking [3, 2]. For this problem, no absolute general scalable or absolute null-invariant measures are optimal.

3.5 Other forms of monotonicity

Other forms of monotonicity have been suggested in the context of measuring interestingness of association rules in data mining. The *reliability* (precision, positive predictive value or confidence) of a rule is defined as $m/(m+n)$ and the *cover* of a rule is $m+n$. It is suggested in [38], and also adopted by [27, 37], that for rules of the same reliability, interestingness should monotonically increase in cover:

Definition 12 (cover-monotone measure). *A measure f is* cover-monotone *if for all points where f is defined and $m/(m+n) = m'/(m'+n')$ we have $C(m+n, m'+n') = C(f_N^M(m, n), f_N^M(m', n'))$.*

Although cover-monotonicity may be desirable for large m and small n, in general it is incompatible with monotonicity (take $m = m' = 0$, for example), and several other properties discussed later.

Geng [33] suggests two other forms of monotonicity: for constant $m + n$ and $o + p$, the measure should be increasing in *support*, $\frac{m}{T}$, and *confidence*, $\frac{m}{m+n}$. In the STASS context, M and N are fixed and both these forms of monotonicity are guaranteed by monotonicity (Definition 11).

3.6 Symmetry under variable permutation

Similarity of set A to set B is intuitively the same as similarity of set B to set A. This is called symmetry under variable permutation in [31] and commutativity symmetry in [28]. It is equivalent to swapping the rows with the columns of the contingency table. For STASS the base instance is fixed, so this property is not really relevant. For association rules, it is generally argued that interestingness of discriminant and characteristic rules should be computed in different ways [27, 32, 37], thus symmetry under variable permutation is typically not advocated in the data mining literature, and [28] also argues against it. It is also not supported by studies of human perception of similarity [16].

3.7 $\bar{n}\bar{m}$-symmetry

The following forms of symmetry discussed are related the discussion in Section 2.3 (also discussed in [39, 28–31]). Only one of these symmetries preserves monotonicity, and it is discussed first. Two others preserve monotonicity if measures are negated (they are forms of antisymmetry). All others preserve monotonicity in m or n but not both, so even if measures are negated, they are not monotone. For this reason these other forms of symmetry are not considered here. The three forms of symmetry that can preserve monotonicity are discussed in the context of SBFL in [5]. Here these forms of symmetry are also adapted to allow for a form of scaling. Section 3.13 provides a graphical summary of all six of these forms of symmetry; readers may wish to refer to this section, particularly Figures 3 to 5.

The first form of symmetry is based on the intuition that if two sets are similar then their complements are also similar. Thus if we take the complement of each instance (including the base instance) we may expect the ranking to remain unchanged. In SBFL, instead of measuring similarity of the set of failed tests with the set of tests in which a statement is executed, we can measure the similarity of the set of passed tests with the set of tests in which a statement is not executed. Taking the complement of the base instance means swapping M with N and m with n. Taking the complement of the other instances means replacing m with $M - m$ and n with $N - n$. This is equivalent to swapping ones and zeros in the encoding of all sets or swapping both the rows and columns of the contingency table. In [31] an “absolute” version of this is defined, called inversion invariance and in [28] it is called commutative symmetry.

Definition 13 (inversion invariant measure). *A measure f is* inversion invariant *if for all points where f is defined, $f_N^M(m, n) = f_M^N(N - n, M - m)$.*

The domains on the two sides of the equation are only the same when $M = N$. In general, the domains are a reflection of each other in the line $n = m$ or a 90° rotation. The crux of this (and indeed any) form of symmetry is how a single point is mapped to its reflected/rotated position; this paper uses the term "dual". This can be used to define the dual of a measure (the reflected/rotated surface), and symmetry can be defined in terms of the result of comparison of measures applied to pairs of points. For this form of symmetry, the dual of a point (or measure) is the reflection in the three vertical planes $n = m$, $m = M/2$ and $n = N/2$. Reflection in the latter two planes is equivalent to a 180° rotation around the vertical line at the center of the domain. When $M = N$, the three reflections collectively are equivalent to a reflection in the single vertical plane $n = M - m$, through points $(M, 0)$ and $(0, N)$.

Definition 14 ($\bar{n}\bar{m}$-duals and symmetry). *Given a domain (M, N), the* $\bar{n}\bar{m}$-dual *of a point (m, n), written $\mathcal{P}^{\bar{n}\bar{m}}(M, N, m, n)$, is $(N - n, M - m)$.*
The $\bar{n}\bar{m}$-dual *of a measure f, written $\mathcal{D}^{\bar{n}\bar{m}}(f)$, is defined as follows:*

$$\mathcal{D}^{\bar{n}\bar{m}}(f)_N^M(m, n) = f_M^N(m^d, n^d), \text{ where } (m^d, n^d) = \mathcal{P}^{\bar{n}\bar{m}}(M, N, m, n)$$

A measure f is $\bar{n}\bar{m}$-symmetric *if for all points where f is defined*

$$C(f_N^M(m, n), f_N^M(m', n')) = C(\mathcal{D}^{\bar{n}\bar{m}}(f)_N^M(m, n), \mathcal{D}^{\bar{n}\bar{m}}(f)_N^M(m', n'))$$

More explicitly, for a $\bar{n}\bar{m}$-symmetric measure f we have

$$C(f_N^M(m, n), f_N^M(m', n')) = C(f_M^N(N - n, M - m), f_M^N(N - n', M - m'))$$

Taking the $\bar{n}\bar{m}$-dual of a measure preserves monotonicity and a measure being its own $\bar{n}\bar{m}$-dual is a sufficient (though not necessary) condition for it to be $\bar{n}\bar{m}$-symmetric. Many measures are not $\bar{n}\bar{m}$-symmetric yet their $\bar{n}\bar{m}$-duals are rarely used as similarity measures. For example, the Jaccard measure, $m/(M + n)$ is common but its $\bar{n}\bar{m}$-dual, $(N - n)/(M + N - n)$ is rarely (if ever) used, even though it has similar attributes to Jaccard. Distinguishing between them is hardly intuitive: $J_{10}^{10}(5, 5) < J_{10}^{10}(6, 6)$ whereas $D_{10}^{10}(5, 5) > D_{10}^{10}(6, 6)$, where J is the Jaccard measure and D its $\bar{n}\bar{m}$-dual. Note that the $\bar{n}\bar{m}$-dual of the Jaccard similarity measure is quite distinct from the Jaccard distance.

3.8 *nm*-antisymmetry

The more similar a set is to the base instance, the less similar it is to the complement of the base set. Thus if we take the complement of the base instance we may expect the ranking will be inverted. This is like swapping ones and zeros in our encoding of just the base instance, or swapping just the rows of the contingency tables. In SBFL we can measure the similarity of the set of passed tests with the set of tests in which a statement is executed, and negate it. An "absolute" definition called hypothesis symmetry is given in [28] and a definition that relies on measures having a particular range is called antisymmetry for normalised measures in [31].

Definition 15 **(nm-duals and antisymmetry).** *Given a domain (M, N), the* nm-dual *of a point (m, n), written $\mathcal{P}^{nm}(M, N, m, n)$, is (n, m).*
The nm-dual *of a measure f, written $\mathcal{D}^{nm}(f)$, is defined as follows:*

$$\mathcal{D}^{nm}(f)^M_N(m, n) = -f^N_M(m^d, n^d), \text{ where } (m^d, n^d) = \mathcal{P}^{nm}(M, N, m, n)$$

A measure f is nm-antisymmetric *if for all points where f is defined*

$$C(f^M_N(m, n), f^M_N(m', n')) = C(\mathcal{D}^{nm}(f)^M_N(m, n), \mathcal{D}^{nm}(f)^M_N(m', n'))$$

Note that the nm-dual of a measure negates the measure (inverts the surface). A constant could be added to preserve the minimum and maximum values over the domain, but for STASS we are only interested in relative rather than absolute values so there is no advantage in doing so. The nm-dual of a measure preserves monotonicity. It is an inverted reflection of the surface in the plane $n = m$.

3.9 $\bar{m}\bar{n}$-antisymmetry

As with nm-antisymmetry, if we take the complement of every set *except* the base instance we may expect the ranking will be inverted. This is like swapping ones and zeros in our encoding of everything except the base instance, or swapping just the columns of the contingency tables. In SBFL we can measure the similarity of the set of failed tests with the set of tests in which a statement is not executed, and negate it. The definition of antisymmetry for normalised measures [31] combines an "absolute" version of both $\bar{m}\bar{n}$-antisymmetry and nm-antisymmetry. An absolute version of nm-antisymmetry is called evidence symmetry in [28]. The $\bar{m}\bar{n}$-dual of a measure is the inverted 180° rotation of the surface (or inverted reflection in the planes $m = M/2$ and $n = N/2$).

Definition 16 **($\bar{m}\bar{n}$-duals and antisymmetry).** *Given a domain (M, N), the* $\bar{m}\bar{n}$-dual *or* rotation-dual *of a point (m, n), written $\mathcal{P}^r(M, N, m, n)$, is $(M - m, N - n)$.*
The $\bar{m}\bar{n}$-dual *or* rotation-dual *of a measure f, written $\mathcal{D}^r(f)$, is defined as follows:*

$$\mathcal{D}^r(f)^M_N(m, n) = -f^N_M(m^d, n^d), \text{ where } (m^d, n^d) = \mathcal{P}^r(M, N, m, n)$$

A measure f is $\bar{m}\bar{n}$-antisymmetric *or* rotation-antisymmetric *if for all points where f is defined*

$$C(f^M_N(m, n), f^M_N(m', n')) = C(\mathcal{D}^r(f)^M_N(m, n), \mathcal{D}^r(f)^M_N(m', n'))$$

As with the nm-dual of a measure, a constant could be added and monotonicity is preserved. When there are multiple forms of symmetry, all forms are satisfied.

Proposition 3. *If a measure f has any two of the properties $\bar{n}\bar{m}$-symmetry, nm-antisymmetry and $\bar{m}\bar{n}$-antisymmetry, it has all three properties.*

Proof. Straightforward.

A special case of this is proved in [28] and the combination of all forms of "absolute" symmetry is referred to as total symmetry. [30] also discusses all these forms of absolute symmetry, plus the additional forms obtained by symmetry under variable permutation.

3.10 Correlation consistency

The first property suggested in [36] is that pairs of instances that are statistically independent should have a zero measure of similarity. The property of Bayesian confirmation [30] says a measure should be less than, equal or greater than zero dependent on whether the conditional probability of a hypothesis H given evidence E is greater, equal or less than the probability of H, respectively. In our context, only relative values are important, but it seems reasonable to say that positively correlated instances are more similar than instances with zero correlation, which are more similar than instances with negative correlations. Instances have zero correlation with the base instance when $Mn = Nm$. In SBFL there is zero correlation when the proportion of failed tests in which a statement is executed is the same as the proportion of passed tests in which it is executed. A larger m (or smaller n) value corresponds to positive correlation and a larger n (or smaller m) value corresponds to negative correlation.

Definition 17 (correlation-consistent measure). *f is* correlation-consistent *if for all points where f is defined we have, if $Nm > Mn$, $Nm' = Mn'$ and $Nm'' < Mn''$ then $f^M_N(m, n) > f^M_N(m', n') > f^M_N(m'', n'')$.*

3.11 Correlation antisymmetry

Correlation consistency suggests a form of antisymmetry may exist between the positively correlated and negatively correlated halves of the domain. When $M = N$, this is the same as nm-antisymmetry: the line of zero correlation, $Mn = Nm$, is the same as $n = m$ and in this special case, a monotone nm-antisymmetric measure must be correlation-consistent. *Correlation-antisymmetry* is defined below; it is a form of antisymmetry around $Mn = Nm$ and it coincides with nm-antisymmetry for $M = N$. It can be seen as scaling so the domain is a square, then reflection in $n = m$, then scaling back to the original domain. All general-scalable nm-antisymmetric measures are correlation-antisymmetric (see Proposition 4), but not all correlation-antisymmetric measures are general-scalable or nm-antisymmetric. Correlation-symmetry maps a point (m, n) to the point $(Mn/N, Nm/M)$. A point with integral coordinates may thus have a dual with non-integral coordinates, which is why this paper defines measures over reals (rationals would be sufficient). An alternative is to assume uniform scalability and multiply everything by MN to obtain integers.

Definition 18 (**nm-scaled-duals and antisymmetry**)**.** *For domain (M, N), the nm-*scaled-dual*, or* correlation-dual*, of point (m, n), written $\mathcal{P}^c(M, N, m, n)$, is $(Mn/N, Nm/M)$.*
*The nm-*scaled-dual*, or* correlation-dual*, of a measure f, written $\mathcal{D}^c(f)$, is defined as follows:*

$$\mathcal{D}^c(f)_N^M(m, n) = -f_N^M(m^d, n^d), \text{ where } (m^d, n^d) = \mathcal{P}^c(M, N, m, n)$$

*A measure f is nm-*scaled-antisymmetric*, or* correlation-antisymmetric*, if for all points where f is defined*

$$C(f_N^M(m, n), f_N^M(m', n')) = C(\mathcal{D}^c(f)_N^M(m, n), \mathcal{D}^c(f)_N^M(m', n'))$$

Note that the correlation-dual of f uses the same domain as f, even though m and n are swapped; this is due to the scaling.

Proposition 4. *If f is a general scalable nm-antisymmetric measure then f is correlation-antisymmetric.*

Proof. (sketch) It is straightforward to show that the nm-dual of a general scalable measure is general scalable. f can be scaled, multiplying M and m by N/M, the nm-dual can be taken and the result scaled in the same way by M/N (N and n are multiplied by this factor because the nm-dual has been taken) to obtain $\mathcal{D}^c(f)$.

Proposition 5. *If f is a monotone correlation-antisymmetric measure then f is correlation-consistent.*

Proof. A proof that positively correlated points have higher values than points with zero correlation is given here; the proof for negatively correlated points is similar. Let $Nm > Mn$ and $Nm' = Mn'$; we need to show that $f_N^M(m, n) > f_N^M(m', n')$. $m > Mn/N$ and $n < Nm/M$, so by monotonicity, $f_N^M(m, n) > f_N^M(Mn/N, n) > f_N^M(Mn/N, Nm/M)$. We have

$$\begin{array}{ll} C(f_N^M(m, n), f_N^M(m', n')) & \\ = -C(f_N^M(Mn/N, Nm/M), f_N^M(Mn'/N, Nm'/M)) & \text{by correlation-antisymmetry} \\ = C(f_N^M(Mn'/N, Nm'/M), f_N^M(Mn/N, Nm/M)) & \text{by the definition of } C \\ = C(f_N^M(m', n'), f_N^M(Mn/N, Nm/M)) & \text{since } Nm\text{'} = Mn\text{'} \end{array}$$

This must equal 1, because $f_N^M(m, n) > f_N^M(Mn/N, Nm/M)$.

3.12 Error symmetry

In a similar way to nm-scaled-antisymmetry, we can defined a scaled version of $\bar{n}\bar{m}$-symmetry. Instead of symmetry around the line $n = M - m$, we have scaled symmetry around the line $Mn = MN - Nm$, between points $(M, 0)$ and $(0, N)$. Here is is called *error symmetry* as this line is where the false positive rate, $n/(n + p)$, equals the false negative rate, $o/(m + o)$. Taking the dual of a point swaps the false positive and false negative rates. This form of symmetry is particularly important for the STASS problem. It is related to the ordering developed for measures in Section 4 and to method to help choose a measure given domain knowledge in Section 5.

Definition 19 ($\bar{n}\bar{m}$-scaled-duals and symmetry). *For domain* (M, N)*, the* $\bar{n}\bar{m}$-scaled-dual*, or* error-dual*, of point* (m, n)*, written* $\mathcal{P}^e(M, N, m, n)$*, is* $(M - Mn/N, N - Nm/M)$*.*
The $\bar{n}\bar{m}$-scaled-dual*, or* error-dual*, of a measure* f*, written* $\mathcal{D}^e(f)$*, is defined as follows:*

$$\mathcal{D}^e(f)^M_N(m, n) = f^M_N(m^d, n^d), \text{ where } (m^d, n^d) = \mathcal{P}^e(M, N, m, n)$$

A measure f *is* $\bar{n}\bar{m}$-scaled-symmetric*, or* error-symmetric*, if for all points where* f *is defined*

$$C(f^M_N(m, n), f^M_N(m', n')) = C(\mathcal{D}^e(f)^M_N(m, n), \mathcal{D}^e(f)^M_N(m', n'))$$

Scaled versions of rotation-duals and antisymmetry can be defined but they are the same as the basic (non-scaled) versions because M and N are not swapped. As with the basic symmetries, any two scaled symmetries implies the third.

3.13 Further discussion of symmetries

Here the forms of symmetry presented are first depicted graphically then discussed further. Figure 3 illustrates the different forms of symmetry for the special case of $M = N$, where the domain is a square, error-symmetry is the same as $\bar{n}\bar{m}$-symmetry and correlation-antisymmetry is the same as nm-antisymmetry. A reference measure and each of its three duals are drawn as arrows. Each arrow depicts two representative points on the surface (the tail and head of the arrow) and their relative height, the arrow head being lower (for monotone measures this means smaller m value and/or greater n value). It can be seen that the nm-dual of the reference measure is the reflection in the $n = m$ line, except that the direction of the arrow is reversed, which indicates antisymmetry. The rotation-dual is the inverted rotation, or double reflection in the lines $m = M/2$ and $n = N/2$. Inverting the rotation-dual and reflecting in the $n = m$ line gives the $\bar{n}\bar{m}$-dual. This is also the reflection of the reference measure in the line $m = M - n$.

When $M \neq N$, the line $m = n$ remains the line of symmetry for the basic (non-scaled) symmetries: between the reference and nm-antisymmetry, and also between $\bar{n}\bar{m}$-symmetry and $\bar{m}\bar{n}$-antisymmetry—see Figure 4. It is also the line of symmetry between the original domain and the domain with M and N swapped. Both nm-antisymmetry and $\bar{n}\bar{m}$-symmetry give points in this "dual" domain—they are not necessarily in the original domain. For $\bar{m}\bar{n}$-antisymmetry (rotation) we have the inverted 180° rotation, as before, and the domain is unchanged. There is no symmetry along orthogonal diagonals.

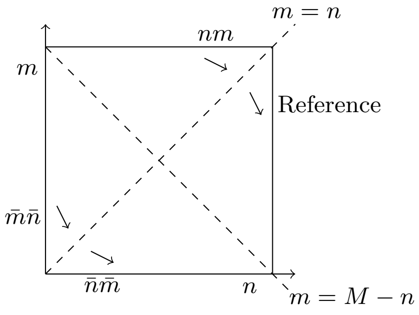

Fig. 3. Symmetries when $M = N$

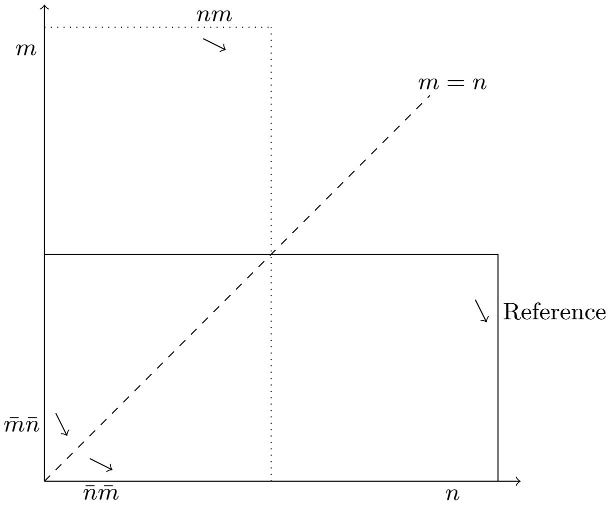

Fig. 4. Basic symmetries when $M \neq N$

For $M \neq N$, the scaled versions of symmetry correspond to a scaled version of Figure 3—see Figure 5. The diagonals are no longer at 45° and both nm-scaled-antisymmetry (correlation) and $\bar{n}\bar{m}$-scaled-symmetry (error) are no longer reflections in the diagonals, but they are skewed reflections. However, $\bar{m}\bar{n}$-antisymmetry (rotation) is still an inverted 180° rotation or double reflection.

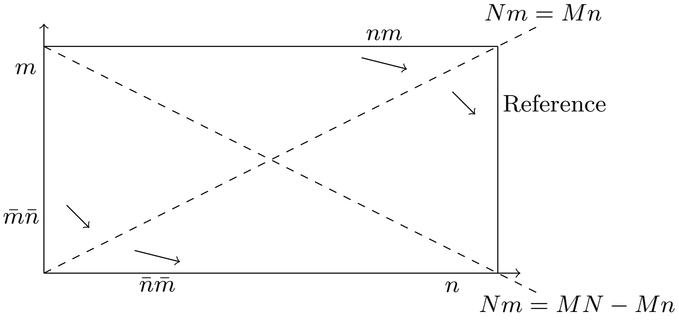

Fig. 5. Scaled symmetries when $M \neq N$

When $M \neq N$, both $\bar{n}\bar{m}$-symmetry and nm-antisymmetry impose no constraint on the ranking produced, since the duals use a different domain. For example, given a measure f, the following $\bar{n}\bar{m}$-symmetric measure $f2$ produces the same ranking when $M < N$ (and similar constructions yields measures with the same ranking when $M < N$ and nm-antisymmetric measures).

$$f2^M_N(m,n) = \begin{cases} f^M_N(m,n) & \text{if } M < N \\ f^M_N(m,n) + \mathcal{D}^{\bar{n}\bar{m}}(f)^M_N(m,n) & \text{if } M = N \\ \mathcal{D}^{\bar{n}\bar{m}}(f)^M_N(m,n) & \text{if } M > N \end{cases}$$

For the other forms of symmetry, or when $M = N$, arbitrary ranking for almost half the domain can be preserved. For example, we can obtain a correlation-antisymmetric measure $f3$ from an arbitrary measure f by using f for points above the line of symmetry and its dual for points below. By adding/subtracting a sufficiently large constant c, monotonicity is preserved.

$$f3^M_N(m,n) = \begin{cases} f^M_N(m,n) + c & \text{if } Mn < Nm \\ 0 & \text{if } Mn = Nm \\ \mathcal{D}^c(f)^M_N(m,n) - c & \text{if } Mn > Nm \end{cases}$$

Various statistical measures can be adapted in such a way to obtain monotone correlation-antisymmetric measures. For example, the Fisher exact test (which computes Bayesian probabilities) can be made into a similarity measure and may be preferable to its various approximations (ϕ, etc.) in some circumstances. Note that $\bar{m}\bar{n}$-antisymmetric measures have a distinct form of symmetry around the same line: rotation (or double reflection) rather than a scaled single reflection. For $M \neq N$, the Pearson measure is $\bar{m}\bar{n}$-antisymmetric but not correlation-antisymmetric, whereas reliability (confidence factor) is correlation-antisymmetric but not $\bar{m}\bar{n}$-antisymmetric.

Neither form of symmetry around the $Mn = Nm$ diagonal constrains the ranking of the portion of the domain that has a positive correlation with the base instance. In many STASS problems it is primarily the top parts of the ranking that are important (see [21], for example)—we often pay close attention to instances that are highly ranked and the rest are all but ignored. Of all the forms of symmetry and duals discussed, only error-symmetry and the relationship between points which are error-duals has a significant impact on the top-most part of the ranking. This is now investigated in more detail.

4 Relative weight of m and n

Different measures give different importance or weight to m and n. In many commonly used measures, m is given more weight than n (matches are considered more importance than non-matches and the false negative rate is generally more important than the false positive rate). For example, with the Jaccard measure, a 10% increase in m must always be accompanied by a greater than 10% increase in n to avoid the measure increasing. In contrast, measures which are error-symmetric give the same weight to m and n overall, and some measures give more weight to n overall, though this is less common in practice. The relative weight of m and n is one important way to distinguish between measures.

Monotonicity implies the surface slopes down along the line from $(M, 0)$ to $(0, N)$, or any line where m decreases and n increases. The relative weight of m and n gives an indication of the typical slope of orthogonal lines. A high m weight indicates that partial derivatives with respect to m are significantly greater than the absolute value of partial derivatives with respect to n, and $(0, 0)$ is lower than (M, N) on the surface. Equivalently, a high m weight indicates that contours of the surface are close to horizontal whereas a low m weight indicates the contours are close to vertical. For the Ample2 measure, all contours are parallel to $Nm = Mn$ and equal weight is given to m and n; this measure is error-symmetric. This paper gives two different, but related ways of defining the relative weight of m and n. One gives a partial order for monotone measures and the other gives a total order.

A related notion is given in [27], which suggests that for discriminant rules, *discrimination* $(1 - n/N)$ is more important than *completeness* (m/M) and for characteristic rules the reverse is the case. If we assume a STASS problem is ranking discriminant rules, two points such that the completeness of each point is the discrimination of the other point should thus be ordered according to their discrimination.

Definition 20 (discriminant-biased measure). *A measure f is* discriminant-biased *if for all points where f is defined, $m/M = 1-n'/N$ and $m'/M = 1-n/N$,*

$$C(f_N^M(m, n), f_N^M(m', n')) = C(m, m')$$

This implies that for two points which are error-duals of each other, measures should be higher for the point with a higher m value (whereas an error-symmetric measure would give both points the same value).

4.1 A partial order for monotone measures

Op of [2] is one extreme within the class of monotone measures—with maximal weight for m. The gradient of all contours is $\frac{1}{N+1}$ and any positive gradient less than $\frac{1}{N}$ results in the same ranking, which is optimal for SBFL of single bug programs. Its error-dual is another extreme, giving minimal weight to m and a gradient greater than M for all contours. This is optimal for another class of software debugging problems, where there may be several bugs but they are all "deterministic"—whenever a bug is executed the test case fails [5]. Note however that the unscaled version of the error-dual (the $\bar{n}\bar{m}$-dual) of *Op* does not always give minimal weight to m and we can have two measures that always produce the same ranking for fixed M and N but their $\bar{n}\bar{m}$-duals no not. The partial order and other properties discussed here critically depend on the scaling. For fixed M and N we can define a partial order over monotone measures. This gives rise to a complete lattice with these two measures as the top and bottom elements, respectively.

The partial order defined is based on the intuition that giving more weight to m increases the number of pairs of points where the relative measures of the two points is the same as the relative m values of the two points. Conversely, giving more weight to n increases agreement with the ordering on the negated n values. For some pairs of points, the ordering on m values is the same as the ordering on $-n$, and all monotonic measures agree with this ordering. For pairs of points not constrained by monotonicity, the ordering of *Op* is always the same as the ordering on m and the ordering of $\mathcal{D}^e(Op)$ is always the same as the ordering on $-n$. This ordering is referred to as the set m-weight because, assuming monotonicity and ignoring ties, it corresponds to the ordering of sets of pairs of points where the measure agrees with the ordering on m values.

Allowing arbitrary points where m or n are not integers results in an infinite number of points and a more complex structure overall. This paper therefore restrict attention to "integral" points, but to enable the use of error-duals of measures, our definition includes points where m or n are integers, and also their error-duals.

Definition 21 (integral point). *Given a domain (M, N), a point (m, n) is* integral *if m and n are integers, or Nm/M and Mn/N are integers.*

Definition 22 (greater set m-weight). *Given a domain (M, N), if f and g are measures then $f \sqsupseteq^M_N g$ (f has greater or equal set m-weight than g for M and N) if for all integral points (m, n) and (m', n') where f is defined,*

1. *if $m \geq m'$ then $C(f^M_N(m, n), f^M_N(m', n')) \geq C(g^M_N(m, n), g^M_N(m', n'))$ and*
2. *if $m < m'$ then $C(f^M_N(m, n), f^M_N(m', n')) \leq C(g^M_N(m, n), g^M_N(m', n'))$.*

If $f \sqsupseteq_N^M g$ and $g \sqsupseteq_N^M f$ we say f and g have equal set m-weight, $f =_N^M g$. If $f \sqsupseteq_N^M g$ and $g \neq_N^M f$ we say f has greater set m-weight than g.

Clearly $f \sqsupseteq_N^M f$, and if $f \sqsupseteq_N^M g$ and $g \sqsupseteq_N^M h$ then $f \sqsupseteq_N^M h$, so $\sqsupseteq_N^M$ is a partial order over measures. For a given natural numbers M and N, $=_N^M$ partitions the set of all measures into a set of equivalence classes. The number of equivalence classes is finite, since M and N are finite, thus the number of integral points and the number of rankings of those points is finite, and two measures are in the same equivalence class if and only if they always result in the same ranking:

Proposition 6. *Given a domain (M, N) and measures f and g, $f =_N^M g$ if and only if the ranking of all integral points using f is the same as that using g.*

Proof. The rankings are the same if and only if for all integral points where f and g are defined, $C(f_N^M(m,n), f_N^M(m',n')) = C(g_N^M(m,n), g_N^M(m',n'))$, which is clearly the case if and only if $f =_N^M g$.

Non-monotone measures can essentially give negative weight to m (and n), which obfuscates the relative weights of m and n, and also means there is no unique equivalence class with maximal or minimal set m-weight. However, by restricting attention to monotone measures we obtain a complete lattice where the top element is the equivalence class containing *Op* and the bottom element is the equivalence class containing $\mathcal{D}^e(Op)$.

Proposition 7. *For all M, N and monotone f, $Op \sqsupseteq_N^M f \sqsupseteq_N^M \mathcal{D}^e(Op)$.*

Proof. Suppose $m > m'$. Then $Op_N^M(m,n) > Op_N^M(m',n')$. Also, we can have $\mathcal{D}^e(Op)_N^M(m,n) > \mathcal{D}^e(Op)_N^M(m',n')$ only when $n > n'$, in which case we have $f_N^M(m,n) > f_N^M(m',n')$ because f is monotone.

Suppose $m < m'$. Then $Op_N^M(m,n) < Op_N^M(m',n')$, and $\mathcal{D}^e(Op)_N^M(m,n) < \mathcal{D}^e(Op)_N^M(m',n')$ only when $n < n'$, in which case $f_N^M(m,n) < f_N^M(m',n')$ because f is monotone. For $m = m'$, f, *Op* and $\mathcal{D}^e(Op)$ all give the same results of comparison due to monotonicity.

It is easy to show properties such as if $f \sqsupseteq_N^M g$ and $h \sqsupseteq_N^M i$ then $f+h \sqsupseteq_N^M g+i$. The lattice of monotone measures is symmetric, with the error-dual of a measure giving its reflection in the lattice.

Proposition 8. *Given a domain (M, N) and monotone measures f and g, $f \sqsupseteq_N^M g$ iff $\mathcal{D}^e(g) \sqsupseteq_N^M \mathcal{D}^e(f)$.*

Proof. Here the only if part is shown; the converse follows from $\mathcal{D}^e(\mathcal{D}^e(f)) = f$. From the definition of $\sqsupseteq_N^M$ we must deal with two cases: $m \geq m'$ and $m < m'$ (for all integral points (m,n) and (m',n') where f is defined). The error-dual points are $(m^d, n^d) = (M - Mn/N, N - Nm/M)$ and $(m'^d, n'^d) = (M - Mn'/N, N - Nm'/M)$. Thus $m^d \geq m'^d$ iff $n \leq n'$ and $m^d < m'^d$ iff $n > n'$ and there are four cases overall:

- $m \geq m' \wedge n > n'$:

$$\begin{aligned}
&C(\mathcal{D}^e(g)_N^M(m,n), \mathcal{D}^e(g)_N^M(m',n')) \\
&= C(g_N^M(m^d,n^d), g_N^M(m'^d,n'^d)) && \text{by } \mathcal{D}^e(g) \text{ definition} \\
&\geq C(f_N^M(m^d,n^d), f_N^M(m'^d,n'^d)) && \text{since } f \sqsupseteq_N^M g \text{ and } m^d < m'^d \\
&= C(\mathcal{D}^e(f)_N^M(m,n), \mathcal{D}^e(f)_N^M(m',n')) && \text{by } \mathcal{D}^e(f) \text{ definition}
\end{aligned}$$

- $m < m' \wedge n \leq n'$:

$$\begin{aligned}
&C(\mathcal{D}^e(g)_N^M(m,n), \mathcal{D}^e(g)_N^M(m',n')) \\
&= C(g_N^M(m^d,n^d), g_N^M(m'^d,n'^d)) && \text{by } \mathcal{D}^e(g) \text{ definition} \\
&\leq C(f_N^M(m^d,n^d), f_N^M(m'^d,n'^d)) && \text{since } f \sqsupseteq_N^M g \text{ and } m^d \geq m'^d \\
&= C(\mathcal{D}^e(f)_N^M(m,n), \mathcal{D}^e(f)_N^M(m',n')) && \text{by } \mathcal{D}^e(f) \text{ definition}
\end{aligned}$$

- $m \geq m' \wedge n < n'$: By monotonicity $C(\mathcal{D}^e(f)_N^M(m,n), \mathcal{D}^e(f)_N^M(m',n')) = 1$.
- $m < m' \wedge n \geq n'$: By monotonicity $C(\mathcal{D}^e(f)_N^M(m,n), \mathcal{D}^e(f)_N^M(m',n')) = 0$.

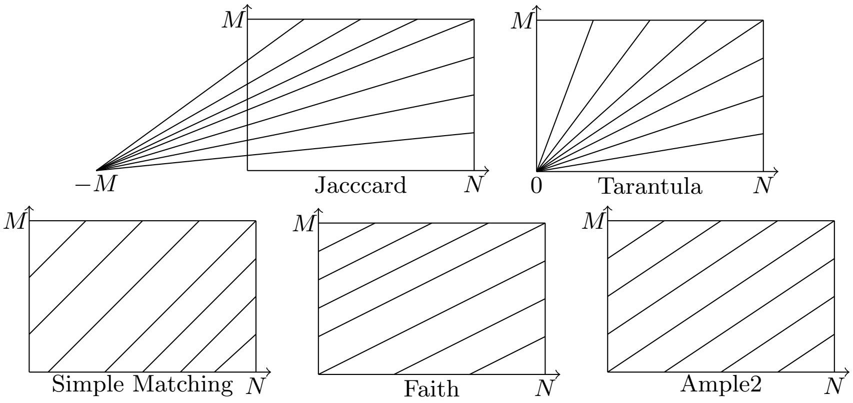

Fig. 6. Contour lines for several measures

Thus error-symmetric measures, which are equivalent to their own error-duals, are in the middle of the lattice. This implies the number of pairs of integral points in the domain where the measure gives the same ordering as m equals the number of pairs of integral points in the domain where the measure gives the same ordering as $-n$. The planar error-symmetric measure Ample2 has contours of gradient M/N and any monotone measure for which all contours have a lower gradient has a higher set m-weight than Ample2 (and vice versa).

For several previously proposed measures all contours are also linear—see Figure 6 ([5] and [21] have similar plots; the latter also discusses the same form of scaling used here for error-duals). For Jaccard (and equivalent measures), the contours converge where $m = 0$ and $n = -M$ so the maximum gradient is 1. The monotone version, Jaccard-m, has greater set m-weight than Ample2 for $M \geq N$. Thus if J is Jaccard-m we can conclude $Op \sqsupseteq^M_N J \sqsupseteq^M_N$ Ample2 $\sqsupseteq^M_N \mathcal{D}^e(J) \sqsupseteq^M_N \mathcal{D}^e(Op)$ for $M \geq N$. The inequalities are strict unless M and N are very small. For Tarantula, the contours converge where $m = n = 0$. If tweaked appropriately so it is defined for $m = n = 0$ and monotone for $m = 0$ it is correlation symmetric. Because the contour gradients are very high at some points and very low at other points, it is generally incomparable to planar measures with respect to the $\sqsupseteq^M_N$ ordering; it is in the middle of the lattice (see Proposition 9). Simple Matching, Faith, Ample2, Op and Russel and Rao are planar measures with contour gradients of 1, $\frac{1}{2}$, $\frac{M}{N}$, $\frac{1}{N+1}$ and 0, respectively. Thus Jaccard-m $\sqsupseteq^M_N$ Simple Matching and Faith $\sqsupseteq^M_N$ Simple Matching for all M and N. Simple Matching has a greater set m-weight than Ample2 if and only if $M > N$.

The class of Tversky ratio model measures of the form $\frac{m}{m+\alpha n+\beta(M-m)}$, where $\alpha, \beta \geq 0$, generalise several proposed measures [16]. For example, with $\alpha = \beta = 1$ we obtain the Jaccard measure. The contours of such measures are linear and all converge at the point $\frac{-\beta M}{\alpha}$ on the n axis. If $\frac{\beta}{\alpha}$ is sufficiently large the gradients of the contours are all close to zero and we obtain a meaure which is similar to Op (though it is not monotone for the $m = 0$ edge of the domain). However, at the other extreme we obtain a measure which is equivalent to Tarantula. Thus, if tweaked to ensure monotonicity, these Tversky ratio model measures range between the top of the lattice and the middle of the lattice. In terms of the ranking produced, the class of Tversky contrast model measures of the form $\theta m + \alpha n + \beta(M - m)$, where $\theta, \alpha, \beta \geq 0$, also generalise measures such as simple matching and Faith, where contours have a fixed gradient independent of M and N. These range between the top and bottom of the lattice.

No monotonic measure with maximal (or minimal) set m-weight is correlation-consistent. For example, (M, N) must be ranked above $(M - 1, 0)$, for maximal set m-weight but below $(M-1, 0)$ for correlation-consistency. In general, there is a tension between correlation-consistency and having a large (or small) m weight (which may be desirable for a particular domain). To be correlation consistent there must be a contour close to the line $Mn = Nm$, so not all contours can have a low (or high) gradient. Measures which are correlation-antisymmetric, like those which are error-symmetric, are in the middle of the lattice of measures.

Proposition 9. *If f is a correlation-antisymmetric measure and (M, N) a domain, the number of pairs of integral points in the domain where f gives the same ordering as m equals the number of pairs of integral points in the domain where f gives the same ordering as $-n$.*

Proof. Consider a pair of integral points (m, n) and (m', n'), and their correlation duals, (m^d, n^d) and (m'^d, n'^d). By the definition of correlation duals and antisymmetry, $C(m^d, m'^d) = -C(n, n')$ and $C(n^d, n'^d) = -C(m, m')$ so f gives the ordering as m for the pair of points iff $\mathcal{D}^c(f)$ gives the ordering as $-n$ for the dual pair of points.

4.2 Using a subset of the domain

It is straightforward to define a variants of set m-weight over a subset of the domain, by just considering pairs of integral points in this subset. We still obtain a partial order and Proposition 7 holds but Propositions 8 and 9 typically do not. Using a subset of the domain may be desirable because some parts of the domain, such as the positively correlated part, are generally more important than other parts.

The choice of whether to use a particular instance or its negation (or complement if viewed as a set) is often influenced by our desire to find positive correlations. If an instance has a negative correlation with the base instance it is always possible to use its negation instead and obtain a positive correlation (a point with negative correlation can be replaced by its rotation-dual). If this is done systematically no points have negative correlation so the contours for this part of the domain and whether a measure is correlation-antisymmetric is irrelevant. Similarly, rotational-antisymmetry is only relevant for the relative ranking of points with zero correlation. Thus the only form of symmetry that is important in this scenario (and assuming $M \neq N$) is error-symmetry and redefining set m-weight so it did not depend on points with negative correlation seems sensible. Note that by using rotation-duals in a different way we could make error-symmetry irrelevant, but there is no apparent good reason for doing so.

4.3 Quantifying the relative weight of m and n

Since we have a partial order rather than a total order, some measures are incomparable with respect to set m-weight—f may give more weight to m than g does for one part of the domain but less weight for another part. The following method can quantify the relative weight of m and n. Assuming monotonicity and ignoring ties, it corresponds to ordering on the cardinality of the set of pairs of integral points where the measure agrees with the ordering on m values.

The method considers the ranking produced by a measure f for all integral points and quantifies how much it differs from the ranking produced from Op and/or its error-dual. Let $p^M_N(f, m, n)$ be the position of point (m, n) in the ranking produced by f; where there are ties in the ranking, the mid point of the range of tied values is used. The difference between the ranking of f and Op is given by the total distance between positions in the rankings (also known as the number of *inversions*), for all integral points:

$$d^M_N(f) = \sum_{(m,n)} \left| p^M_N(f, m, n) - p^M_N(Op, m, n) \right|$$

It is convenient to scale this value as follows:

Definition 23 (Cardinality m-weight). *Given a domain* (M, N)*, the* cardinality m-weight *of a measure* f*,* $w^M_N(f)$ *is*

$$1 - \frac{d^M_N(f)}{d^M_N(\mathcal{D}^e(Op))}$$

For monotone measures, $w_N^M(f)$ varies from 0 (for $\mathcal{D}^e(Op)$) to 1 (for Op), with 0.5 for error-symmetric and correlation-antisymmetric measures. For non-monotone measures it can potentially have a value outside this range. As with set m-weight, the cardinality m-weight could be defined over just part of the domain (in which case, correlation-antisymmetric measures generally will not have a value of 0.5). For example, to analyse classes of measures such as that proposed for SBFL in [6] we have used a variant of cardinality m-weight that uses just the part of the domain above the line segments $(0,0)-(3M/4,N/4)$ and $(3M/4,N/4)-(M,N)$. This contains the top part of the ranking—statements executed in relatively large numbers of failed tests and/or relatively few passed tests. It is helpful to know how (this variant of) the cardinality m-weight varies with the proportion of failed tests. Intuitiively, a small proportion of failures suggests there may be a single bug and/or less consistent bugs (so a measure like Op is best) whereas a large proportion of failures suggests more bugs and/or more deterministic bugs (so a measure like $\mathcal{D}^e(Op)$ is best)—see Section 5. Note that for general scalable measures the m-weight cannot adapt to the proportion of failed tests.

5 Using domain knowledge to choose a measure

So far, this paper has described various properties of similarity measures. However, what we would ultimately like is a way to determine which measures are likely to work best in a given situation. A contribution to solving this difficult problem is presented next. Section 5.1 reviews a model-based approach to understanding SBFL that lead to the optimality results for Op and $\mathcal{D}^e(Op)$ and Section 5.2 proposes a method for interpolating between these two important boundary cases. Section 5.3 reports on an experiment that validates the approach.

5.1 Model-based software debugging

In [2] a very simple model program was used to investigate SBFL. It allowed the performance of different similarity measures to be assessed under "ideal" conditions where various parameters could be controlled precisely. It also allowed a more analytical approach to assessing different measures. The model used was a program of the form below with just four statements, one of which was a bug.

```
if (...)
    S1; /* BUG, two execution paths, one leads to failure */
else
    S2; /* two execution paths */
if (...)
    S3;
else
    S4;
```

Test cases for the program were simulated by choosing an execution path through the program, some of which lead to failure of the test. The program has eight possible execution paths, four of which include the bug and two of those lead to failure. For a given multiset of test cases, the spectra are used with a set similarity measure to rank the statements and a score is given to the ranking. Overall performance for a number of test cases T was assessed by averaging over all possible multisets of T execution paths (for larger T this was estimated by computing a large number of multisets pseudo-randomly with an appropriate distribution).

Experiments were conducted to determine the overall performance of numerous set similarity measures as various parameters were adjusted (such as the total number of tests, the number of failed tests and how "consistent" the bug was: the number of failed tests divided by the number of tests in which the bug was executed). This simple model was shown to be a good predictor of relative performance of set similaritity measures for real single-bug programs. Additionally, it was shown analytically that for the model program and any number of tests, *Op* (named O^p in [2]) performed at least as well as any other similarity measure overall. Other model programs have also been used, and by restricting attention to monotone measures, the optimality result for *Op* has been strengthened to all single-bug programs and all sets of test cases [3]. In [5] duals are discussed and (measures equivalent to) $\mathcal{D}^e(Op)$ are shown to be optimal for programs with only deterministic bug.

5.2 Interpolating between two boundary cases

The reason why *Op* is optimal for the software debugging problems analysed in [2, 3] is that it is assumed a single bug exists and the program fails a test only when the bug is executed. Because the simple model program of [2] has eight (equally probable) execution paths, if there are eight test cases the most likely (or "expected") outcome is that each of these paths is used once. This outcome can be reflected in a contingency table for each of the four instances (execution of the four program statements). It is helpful to concentrate on the contingency table for executing the bug, since this is the *cause* of failure of test cases. Recall that the bug is executed in four paths, two of which fail. In general, with perfect knowledge of the domain, we can determine an expected contingency table for the causes of the base instance:

Bug

2	2
0	4

Causes

m_c	n_c
o_c	p_c

For now, non-causal instances will be ignored, making the simplifying assumption that the expected outcomes for non-causal instances typically have no statistical correlation with the base instance. The important feature of the contingency table above is that the value of o for the bug, o_c, is zero (there are no false negatives or type II errors), whereas all other values are non-zero.

The ratio for the M column is 2:0 whereas the ratio for the N column is 2:4. Intuitively, maximal m-weight is optimal because the first ratio is infinitely more discriminating than the second.

For single-bug programs we always have $o_c = 0$, leading to optimality of Op in this case. In the natural sciences, boundary cases such as this rarely occur because causality is typically more complex and data is noisy. In SBFL there is another boundary case of interest, where all bugs are deterministic, so $n_c = 0$ (there are no false positives or type I errors). With $n_c = 0$ and other values non-zero, the ratio for the N column is infinitely more discriminating than that of the M column. For this class of debugging problems $\mathcal{D}^e(Op)$ is optimal, by similar reasoning to the proof of optimality of Op for the single bug case.

For single bug programs, $o_c = 0$ and the causal instance always has $m = M$, points at the top edge of the domain in our figures. For deterministic bug programs, $n_c = 0$ and the causal instances always have $n = 0$, points at the left edge of the domain. We can use a form of interpolation between these two boundary cases to obtain another measure of m-weight. Given a domain and an expected contingency table for causal instances, we can determine the expected value for m and n as a proportion of M and N, respectively: $m_c/(m_c{+}o_c)$ (which is the true positive rate or one minus the false negative rate) and $n_c/(n_c + p_c)$ (the false positive rate), respectively. The gradient of the line through this point and $(M, 0)$ gives an indication of what m-weight will lead to best performance. The *positive error rate (PER)*, defined below, ranges from zero, for a vertical line where minimal m-weight is optimal, to one, for a horizontal line where maximal m-weight is optimal:

Definition 24 (positive error rate (PER)). *Given a domain* (M, N)*, and contingency table (the expected values for causal instances),* (m, n, o, p)*, with* $FNR = 1 - m/M$ *and* $FPR = n/N$*. The* positive error rate

$$PER = \frac{FPR}{FNR + FPR}$$

unless $FNR = FPR = 0$*, in which case* $PER = 0.5$*.*

The PER gives information about the quality of the causal instance(s) as predictors of the base instance. It relates the number of false positives (type I errors) as a proportion of negative features, with the number of false negatives (type II errors) as a proportion of positive features. The former is larger precisely when the PER is greater than 0.5 and when points for causal instances are expected to be above the line of error-symmetry, $Mn = MN - Nm$. There are infinitely many ways of measuring the relative frequency of false positives and false negatives. It can be cast as an instance of the STASS problem (by another form of dual), so we can define monotonicity and a partial order. PER has desirable behaviour for the two boundary cases and has the line of error symmetry as a contour. It is a form of dual of the Tarantula measure (and precision).

We conjecture that as the positive error rate of domains increase, measures with higher m-weight will perform best in terms of ranking causal instances highly, on average. This conjecture is supported by experiments, one of which is described in detail in Section 5.3. Expert knowledge may be used to estimate the PER for a given domain. It is a single statistic which summarises a probability distribution and indicates the "best" we can do in terms of false positive and false negative rates. Typically there can be several "causal" instances with differing false positive and false negative rates. In estimating the PER we can consider the relative cost of false positives and false negatives in determining what is best, as is done in ROC analysis. This is discussed in Section 5.4.

5.3 A software debugging experiment

This experiment uses several debugging models in the style of [2] that span the range of possible PER values and for each one, determines the m-weight of the "best" measure (within a restricted class of measures—it is not known what the best possible measure is in general). There are six models, each with four statements, where execution of correct statements is statistically independent of test case failure but execution of buggy statements is correlated to varying degrees. They all have the following form:

```
if (...)
    S1; /* five execution paths */
if (...)
    S2; /* five execution paths */
if (...)
    S3;
if (...)
    S4;
```

The same models were used in [6] to assess learning of similarity measures for a range of data sets. In the first model, M1, only the first statement is a bug. In models M2 to M6 the first two statements are bugs, each of which cause failure in 20%, 40%, 60%, 80% and 100% of cases where they are executed, respectively. The first two statements are modelled using five execution paths and a number of those lead to failure, dependent on the model. The other two statements are each modelled using just one execution path. Also, each "if" has as many paths that do not execute the statement as those that do, so there are $10 \times 10 \times 2 \times 2 = 400$ paths in total.

The relative discrimination of m versus n (and thus the PER) drops as we go from model M1 to M6, and this affects what measure is best to use for each of these models. From previous results we know that Op is the best measure to use for M1 (because it has a single bug and the PER is 1) and its error-dual is the best measure to use for M6 (since it has only deterministic bugs and the PER is 0). For performance comparison, other measures that are planar were also used. They have the following form, with varying values of the parameter p:

$$f_N^M(m,n) = pm/M - (1-p)n/N$$

With p sufficiently close to 1 this measure is equivalent to Op, with p sufficiently close to 0 it is equivalent to $\mathcal{D}^e(Op)$ and for $p = 0.5$ it is equivalent to Ample2. The p value is thus an alternative way of quantifying the m-weight for this class of measures. For models M2 to M5 the p value that resulted in best performance was experimentally determined, using multiples of 0.01. Performance was measured by the rank of all the bugs, scaled so that if all bugs are at the top of the ranking the performance is 100 and if they are all at the bottom of the ranking the performance is 0. All reported figures are averages over 100 million multisets of 15 test cases (a relatively small number of tests is used because performance tends to converge for larger numbers of tests, making comparison of measures more difficult).

	M1	M2	M3	M4	M5	M6
m_c, n_c	40, 160	56, 144	104, 96	144, 56	176, 24	200, 0
o_c, p_c	0, 200	20, 180	40, 160	60, 140	80, 120	100, 100
FNR	0	0.26	0.28	0.29	0.31	0.33
FPR	0.44	0.44	0.38	0.29	0.17	0
PER	1	0.63	0.57	0.49	0.34	0
best p	1-ϵ	0.75	0.53	0.47	0.40	ϵ
cardinality w.	1	0.85	0.56	0.44	0.34	0
Op	**89.14**	86.19	89.17	90.62	91.22	91.25
$p = 0.75$	89.14	**86.19**	89.25	91.17	92.87	94.85
$p = 0.53$	88.94	86.08	**89.50**	92.17	94.65	97.44
$p = 0.47$	88.58	85.87	89.34	**92.20**	94.85	97.80
$p = 0.40$	88.00	85.59	89.00	92.02	**94.88**	97.93
$\mathcal{D}^e(Op)$	73.87	81.51	86.51	90.87	94.60	**98.02**

Fig. 7. Six debugging models and performance results

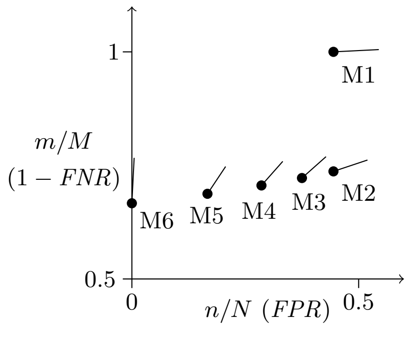

Fig. 8. Expected false positive/negative rates, best planar measures for the models

Figure 7 gives the results of the experiment. The first two rows give the contingency tables for the causes in each model. The next three rows give the expected false negative and false positive rates and the positive error rate computed from the contingency tables. The next row gives the best p value found empirically (except that 1-ϵ and ϵ are determined theoretically). Only multiples of 0.01 were considered for p. This appears to affect the results as performance typically does not increase or decrease smoothly as p changes. In particular, for model M2, the optimal p value may be rather less than 0.75. However, it is clear that the best p value decreases across the different models, supporting our conjecture. Figure 8 gives a graphical depiction of the same information. For each model it plots the expected false positive and negative rates (the expected point for the bugs in the "top left" quarter of the scaled domain). The gradient of the lines through the top left corner of the scaled domain ($m/M = 1$, $n/N = 0$) and each of these points drops from zero, for M1, to minus infinity, for M6, corresponding to the PER dropping from 1 to 0. The line segments for each model shown in Figure 8 give the contours of the best planar metric found, ranging from (almost) horizontal to (almost) vertical.

Row seven of Figure 7 gives the cardinality weight of the measures (with the best p value) for $M = N = 8$. Note that M and N vary over the different multisets of test cases, so these figures only give a general guide to this form of quantifying the m weight. The $\sqsupseteq^M_N$ relation holds between successive best measures for all M and N. The last six rows give the performance of each measure for each model. The maximum performance for each model is displayed in bold font. For each column, the performance peaks at the leading diagonal and decreases monotonically as we move away from the maximum (for the first two models and measures it is necessary to examine more significant figures than appear in the table). The experiments described in [6] use the same models but a different class of measures (the contours being hyperbolas with coefficients found using machine learning) and a different performance measure (the rank of the top-most bug rather than all bugs). They also show a trend of reducing m-weight across the models.

5.4 PER and ROC analysis

ROC analysis can be used to visualise and determine the relationships between a set similarity measure f, the performance of the associated binary classifiers f^α_c, the best threshold value α and the corresponding true positive and false positive rates. Consideration of PER essentially inverts this analysis. Figure 9 gives the ROC curve for f^α_c. Such curves can be constructed from "training" data sets, where correct classifications are known for all points, and used to estimate the best threshold for "real" data. Assuming there is a known constant cost for each false positive and a (possibly different) constant cost for each false negative, lines of fixed cost can be drawn. The top-most (lowest cost) such line which meets the ROC curve, and the point(s) at which it does so, gives the optimal threshold value(s). For f^α_c in Figure 9 this line is drawn in dashes. For this example it is assumed the cost of false positives divided by M is somewhat more that the cost of false negatives divided by N, so the gradient is somewhat less than one. The best pair of true and false positive rates (shown in Figure 9) are the coordinates of the point of intersection.

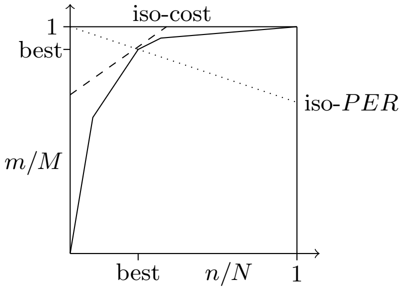

Fig. 9. ROC analysis of a set similarity classifier f_c^α

For the PER analysis suggested above, the starting point is (an estimate of) the best pair of true and false positive rates that can be achieved, using domain knowledge. For software debugging, the discussion of [5] can be adapted: at early stages of software development there are almost certain to be multiple bugs and deterministic bugs are relatively likely so a low PER (a relatively low false positive rate) estimate is reasonable, whereas late in development there are fewer bugs (perhaps just one) and they are typically less consistent, hence a high PER (a relatively low false negative rate) estimate is desirable. From this an iso-PER line can be drawn, shown as a dotted line in Figure 9. The PER can help with the choice of an appropriate measure f, and if it is used for a binary classifier f_c^α, the ROC curve should ideally intersect with the best iso-cost line and the iso-PER line at the same point.

Suppose that for a given classification problem with particular costs for false negatives and false positives, we (somehow) know the best possible set similarity measure, threshold and corresponding false negative and false positive rates. If the relative cost of false negatives is revised to be higher, ROC analysis can be used to determine the best threshold. Increasing the relative cost of false negatives decreases the gradient of the iso-cost lines, thus the point at which the lowest cost line intersects with the ROC curve is generally higher and further right. The revised best m and n values increase (and the false negative rate, $1 - m/M$, and the threshold decrease). ROC analysis says nothing about revising the set similarity measure—it is fixed in the analysis. However, the revised intersection point results in an increased PER which, according to our conjecture, suggests a measure with a higher m-weight would be best. This is consistent with the intuition that if the cost of false negatives is relatively high, a measure with relatively high m-weight performs best, since n is the number of false positives and m is related to the number of false negatives.

5.5 Further work

The PER gives at best a rough idea of a probability distribution, and even with perfect knowledge of the probability distribution we do not know how to construct an optimal measure in general. For some classes of probability distributions and methods of evaluating performance it may be possible to analytically determine the optimal set similarity measure, as has been done in the two boundary cases. The mathematics is likely to be rather complex but even if the assumptions are impractical, this theoretical approach may provide more insights into the problem.

A more empirical approach is to use machine learning techniques such as those in [6] to search for good sub-optimal measures for different scenarios. This approach seems to have great potential and can be guided by theoretical insights to restrict the class of measures considered. Without restriction the number of measures is so large that even machine learning techniques run into difficulties. The class of measures used in [6] was chosen in part because the measures are monotone and include measures which are optimal in the two boundary cases. However, most other properties discussed here played no role in the choice. By considering things such as forms of scalability and symmetry it may be possible to find a better class of measures and use machine learning to find good measures within that class.

6 Conclusion

Notions of similarity are pervasive in science. This paper explores in detail a particularly simple instance, which is refered to as similarity to a single set or STASS. The objects being compared are sets (or, equivalently, have just binary attributes that are all treated equally). All objects are compared to a "base" set using a "set similarity measure" (numeric function), resulting in a ranking of the objects from the most similar to the base set to the least similar. It is closely related to measuring similarity between any two sets or correlation in a two by two contingency table or confusion matrix. Even this very simple similarity problem has many important instances, from comparing diagnostic tests and other binary classifiers to locating bugs in computer programs, our primary application area. A large number of set similarity measures have been proposed in the literature but very little is known about how to choose the best one for a given application.

This paper gives a comprehensive discussion of various properties a set similarity measure may have in the context of STASS, refining previously identified properties, introducing new properties and discussing some relationships between properties. Several properties are refined so their definitions are weaker—dependent only on relative measures of similarity of pairs of sets (the ranking produced) rather than "absolute" measures of similarity. This results in important properties no longer being incompatible. New forms of symmetry are also considered, including "error-symmetry" which is related to the duality between the false positive and false negative rates. This form of symmetry leads to new ordering relationships on set similarity measures and a new statistic which can be useful in choosing a set similarity measure for a given application domain. At one extreme a similarity measure can minimize the false positive rate and at the other extreme it can minimize the false negative rate. Most commonly a compromise between these extremes is best. The insights described in this paper have helped in the design of effective set similarity measures for locating bugs and have the potential to be useful in many other areas.

Acknowledgements

I would like to thank the reviewers and Kotagiri Ramamohanarao for feedback on ealier versions of this paper.

References

1. Naish, L., Lee, H.J., Kotagiri, R.: Spectral debugging with weights and incremental ranking. In: 16th Asia-Pacific Software Engineering Conference, APSEC 2009, IEEE (December 2009) 168–175
2. Naish, L., Lee, H.J., Kotagiri, R.: A model for spectra-based software diagnosis. ACM Transactions on software engineering and methodology (TOSEM) **20**(3) (August 2011)
3. Naish, L., Lee, H.J., Kotagiri, R.: Spectral debugging: How much better can we do? In: 35th Australasian Computer Science Conference (ACSC 2012), CRPIT Vol. 122, CRPIT (2012)
4. Yoo, S.: Evolving human competitive spectra-based fault localisation techniques. In: Search Based Software Engineering. Springer (2012) 244–258
5. Naish, L., Lee, H.J.: Duals in spectral fault localization. In: Proceedings of ASWEC 2013, IEEE Press (2013)
6. Naish, L., Neelofar, Kotagiri, R.: Multiple bug spectral fault localization using genetic programming. In: Proceedings of ASWEC 2015, IEEE Press (2015)
7. Landsberg, D., Chockler, H., Kroening, D., Lewis, M.: Evaluation of measures for statistical fault localisation and an optimising scheme. In Egyed, A., Schaefer, I., eds.: Fundamental Approaches to Software Engineering. Lecture Notes in Computer Science. Springer (2015) 115–129
8. Terra, R., Brunet, J., Miranda, L.F., Valente, M.T., Serey, D., Castilho, D., da Silva Bigonha, R.: Measuring the structural similarity between source code entities (S). In: The 25th International Conference on Software Engineering and Knowledge Engineering, Boston, MA, USA, June 27-29, 2013., Knowledge Systems Institute Graduate School (2013) 753–758
9. Jaccard, P.: Étude comparative de la distribution florale dans une portion des Alpes et des Jura. Bulletin de la Société Vaudoise des Sciences Naturelles **37** (1901) 547–579
10. Ochiai, A.: Zoogeographic studies on the solenoid fishes found in Japan and its neighbouring regions. Bulletin of the Japanese Society of Scientific Fisheries **22** (1957) 526–530
11. Chao, A., Chazdon, R.L., Colwell, R.K., Shen, T.J.: A new statistical approach for assessing similarity of species composition with incidence and abundance data. Ecology Letters **8**(2) (2005) 148–159

12. Fligner, M.A., Verducci, J.S., Blower, P.E.: A modification of the Jaccard-Tanimoto similarity index for diverse selection of chemical compounds using binary strings. Technometrics **44**(2) (2002) 110–119
13. Sesli, M., Yegenoglu, E.: Comparison of similarity coefficients used for cluster analysis based on RAPD markers in wild olives. Genetics and Molecular Research **9**(4) (2010) 2248–2253
14. Patzkowsky, M.E., Holland, S.M.: Biofacies replacement in a sequence stratigraphic framework; Middle and Upper Ordovician of the Nashville Dome, Tennessee, USA. Palaios **14**(4) (August 1999) 301–317
15. Leicht, E.A., Holme, P., Newman, M.E.J.: Vertex similarity in networks. Physical Review E **73** (Feb 2006) 026120
16. Tversky, A.: Features of similarity. Psychological Review **84**(4) (July 1977) 327–352
17. Omhover, J.F., Rifqi, M., Detyniecki, M.: Ranking invariance based on similarity measures in document retrieval. In Detyniecki, M., Jose, J.M., Nürnberger, A., van Rijsbergen, C.J., eds.: Adaptive Multimedia Retrieval: User, Context, and Feedback: Third International Workshop, AMR 2005, Glasgow, UK, July 28-29, 20 05, Revised Selected Papers. Springer Berlin Heidelberg, Berlin, Heidelberg (2006) 55–64
18. Lesot, M.J., Rifqi, M., Benhadda, H.: Similarity measures for binary and numerical data: a survey. International Journal of Knowledge Engineering and Soft Data Paradigms **1**(1) (December 2009) 63–84
19. Baulieu, F.B.: A classification of presence/absence based dissimilarity coefficients. Journal of Classification **6**(1) (1989) 233–246
20. Batagelj, V., Bren, M.: Comparing resemblance measures. Journal of Classification **12**(1) (1995) 73–90
21. Fürnkranz, J., Flach, P.A.: ROC 'n' rule learning—Towards a better understanding of covering algorithms. Machine Learning **58**(1) (2005) 39–77
22. Powers, D.M.W.: Evaluation: From precision, recall and F-measure to ROC., informedness, markedness & correlation. Journal of Machine Learning Technologies **2**(1) (2011) 37–63
23. Levandowsky, M., Winter, D.: Distance between sets. Nature **234** (November 1971) 34–35
24. Santini, S., Jain, R.: Similarity measures. IEEE Transactions on pattern analysis and machine Intelligence **21**(9) (September 1999) 871–883
25. Rifqi, M., Berger, V., Bouchon-Meunier, B.: Discrimination power of measures of comparison. Fuzzy Sets and Systems **110**(2) (2000) 189 – 196
26. Bouchon-Meunier, B., Rifqi, M., Bothorel, S.: Towards general measures of comparison of objects. Fuzzy Sets and Systems **84**(2) (December 1996) 143–153
27. Kamber, M., Shinghal, R.: Evaluating the interestingness of characteristic rules. In: Proceedings of ACM SIGKDD international conference on knowledge discovery and data mining. (1996) 263–266
28. Eells, E., Fitelson, B.: Symmetries and asymmetries in evidential support. Philosophical Studies **107**(2) (2002) 129–142
29. Tentori, K., Crupi, V., Bonini, N., Osherson, D.: Comparison of confirmation measures. Cognition **103**(1) (2007) 107–119
30. Greco, S., Slowinski, R., Szczech, I.: Properties of rule interestingness measures and alternative approaches to normalization of measures. Information Sciences **216** (2012) 1–16

31. Tan, P.N., Kumar, V., Srivastava, J.: Selecting the right interestingness measure for association patterns. In: Proceedings of the eighth ACM SIGKDD international conference on knowledge discovery and data mining. KDD '02, New York, NY, USA, ACM (2002) 32–41
32. Lenca, P., Meyer, P., Vaillant, B., Lallich, S.: On selecting interestingness measures for association rules: User oriented description and multiple criteria decision aid. European Journal of Operational Research **184**(2) (2008) 610–626
33. Geng, L., Hamilton, H.J.: Interestingness measures for data mining: A survey. ACM Computing Surveys **38**(3) (September 2006)
34. Cha, S.H.: Comprehensive survey on distance/similarity measures between probability density functions. International Journal of Mathematical Models and Methods in Applied Sciences **1**(4) (2007) 300–307
35. Choi, S.S., Cha, S.H., Tappert, C.: A Survey of Binary Similarity and Distance Measures. Journal on Systemics, Cybernetics and Informatics **8**(1) (2010) 43–48
36. Piatetsky-Shapiro, G.: Discovery, analysis and presentation of strong rules. In Piatetsky-Shapiro, G., Frawley, W., eds.: Knowledge Discovery in Databases, Cambridge, MA, AAAI/MIT Press (1991) 229–248
37. Freitas, A.A.: On rule interestingness measures. Knowledge-Based Systems **12** (1999) 309–315
38. Major, J., Mangano, J.: Selecting among rules induced from a hurricane database. Journal of Intelligent Information Systems **4**(1) (1995) 39–52
39. Carnap, R.: Logical Foundations of Probability. University of Chicago Press (1962)