Comparing sets of patterns with the Jaccard index

Sam Fletcher
School of Computing and Mathematics,
Charles Sturt University, Bathurst, Australia
sam.pt.fletcher@gmail.com

Md Zahidul Islam
School of Computing and Mathematics,
Charles Sturt University, Bathurst, Australia

Abstract

The ability to extract knowledge from data has been the driving force of Data Mining since its inception, and of statistical modelling long before even that. Actionable knowledge often takes the form of patterns, where a set of antecedents can be used to infer a consequent. In this paper we offer a solution to the problem of comparing different sets of patterns. Our solution allows comparisons between sets of patterns that were derived from different techniques (such as different classification algorithms), or made from different samples of data (such as temporal data or data perturbed for privacy reasons). We propose using the Jaccard index to measure the similarity between sets of patterns by converting each pattern into a single element within the set. Our measure focuses on providing conceptual simplicity, computational simplicity, interpretability, and wide applicability. The results of this measure are compared to prediction accuracy in the context of a real-world data mining scenario.

Keywords: Machine Learning; Metrics; Data Mining; Patterns; Rules; Utility Measures; Quality Evaluation.

1 Introduction

The discovery of patterns in data is the cornerstone of data mining; sometimes for predicting the future, and other times for extracting meaning. It is the latter case that this paper will concern itself with. By extracting meaning, statisticians and data analysts are able to elevate otherwise meaningless data into usable information; information that can be acted on or used to discover knowledge (Han et al. 2006). The extraction of meaning from data is most often explicated as mining patterns from data, preferably in a form that is interpretable by humans. Many branches of data mining have concerned themselves over the years with this endeavour, not limited to frequent pattern (itemset) mining (Han et al. 2007), decision trees (Quinlan 1993) and forests (Breiman 2001a), and association rule mining (Ordonez & Zhao 2011). Constructing models from data in order to make accurate predictions is useful, but it is important to remember that the accuracy of a model does not guarantee the truthfulness of the patterns contained in the model. The history of science is littered with untruthful models that accurately predicted observations, from Bohr's model of the atom to the geocentric model of the Solar System. Assessing the patterns discovered by data mining algorithms is often overlooked in the current zeitgeist, instead favouring a dogged pursuit of higher prediction accuracy (Hand 2006, Wagstaff 2012). We argue that this is to the detriment of the discipline and provide a simple but powerful method for comparing sets of patterns, aiding a renewed focus in pattern assessment seen in recent years (Aodha et al. 2014, Letham et al. 2013).

Two terms warrant further explanation: data and patterns. When we talk about data, we are talking about the contents of a dataset x, where a dataset is a two-dimensional matrix of rows and columns, with each cell in the dataset containing a single datum (i.e., piece of data, or value). Each row r represents the record of a unique participant in the dataset, and each column A represents an attribute or feature of the participants. The value v that a record r has for attribute A is the value that corresponds to some real-life fact about the participant. An attribute A can use either discrete (i.e., categorical or unordered) values, or continuous (i.e., numerical or ordered) values. We write r_A when referring to the value that r has for attribute A. We use n to denote the number of records in the whole dataset, and m to denote the number of attributes.

By "pattern", we refer to a set of criteria, where if a record meets all of the criteria, allow an analyst to infer additional information about the record that was not previously known. More formally, patterns take the form $\psi \rightarrow c \in C$. That is to say, if criteria ψ is met, attribute C is predicted to equal c. C is known as the consequent in this context, but is functionally the same as a class attribute. Each record r has a class attribute, referred to as r_C. Functioning as the antecedent, ψ is a set of conditions for a subset of the m attributes used in the data, where each attribute A has values $v \in A$. Conditions take the form $A=v$, or can use other operators such as $A>v$ if A is ordered. An example would be

$$\{Education = PhD, Age > 45\} \rightarrow \; Income = High \, .$$

Another example could be

$$\{milk, eggs\} \rightarrow \{bread\} \, ,$$

where *milk* and *eggs* can be thought of as Boolean (binary) attributes. Other examples can be seen in Table 1. Patterns in any of these forms are sometimes referred to as decision rules or association rules, and when collected together can be called a decision list or rule list (Letham et al. 2013).[1]

$\boldsymbol{i}$	$\boldsymbol{\psi_i}$	$\boldsymbol{c}$
0	$Clump\ Thickness \leq 5.5\ AND$ $Uniformity\ of\ Cell\ Size \leq 2.5\ AND\ Bare\ Nuclei \leq 4.5$	Benign
1	$Uniformity\ of\ Cell\ Size > 2.5\ AND$ $Uniformity\ of\ Cell\ Shape \leq 2.5$	Benign
2	$Clump\ Thickness > 6.5\ AND\ Uniformity\ of\ Cell\ Size\ >\ 2.5$ $AND\ Uniformity\ of\ Cell\ Size \leq 4.5\ AND$ $Uniformity\ of\ Cell\ Shape > 2.5\ AND\ Bare\ Nuclei\ >\ 2.5$ $AND\ Mitoses \leq 1.5$	Malignant
3	$Uniformity\ of\ Cell\ Size > 4.5\ AND$ $Uniformity\ of\ Cell\ Shape > 2.5\ AND\ Bland\ Chromatin > 4.5$	Malignant

Table 1 – Some examples of patterns discovered by CART in the WBC dataset.

[1] For example, a decision tree might be "flattened" so as to no longer have roots or leaves, and it merely becomes an unordered set (a rule list) of unordered sets of attribute conditions (rules). The decision tree would then be indistinguishable from a rule list.

1.1 Problem statement

Consider the following question:

Given two sets of patterns Ψ_1 and Ψ_2, how similar are they?

To the best of our knowledge, this question has yet to be answered in the literature, and is answered in this paper. The ability to answer this question has clear benefits in many areas of Data Mining and Knowledge Discovery, such as:

- Comparing different classifiers, or classifiers with different parameters (Islam & Giggins 2011, Shotton et al. 2013);
- Comparing patterns discovered manually by statisticians, to patterns discovered with machine learning techniques (Breiman 2001b);
- Comparing the quality of the patterns discovered in data before and after applying privacy-preserving techniques to the data (Fletcher & Islam 2014, 2015a, Friedman & Schuster 2010); and
- Finding differences in different samples of data, including temporal scenarios with time series (Baron et al. 2003).

Answering the above question requires a measurement of some kind, and it is a measure (more specifically, a metric) that we propose in this paper. We convert the patterns in Ψ_1 and Ψ_2 into discrete elements, and take advantage of the Jaccard index to measure the similarity between sets made up of these elements. This is discussed in full in Section 3. In order to be a useful measure, we consider several external factors to be part of the problem statement. These are factors that often determine whether researchers and data miners wilfully choose to use a measure:

- conceptual simplicity;
- computational simplicity;
- interpretability; and
- wide applicability.

We discuss how our proposed measure fulfils these external factors in Section 3. Section 2 provides additional background information and related work. In Section 4, we demonstrate our measure in action with two real-world scenarios: one in which a user wishes to compare two classifiers; and one where a user wants to compare to privacy-preserving data mining algorithms. We conclude with Section 5.

2 Related work

Patterns play a key role in the knowledge discovery and decision-making process. Several fields of machine learning – notably frequent pattern mining (Han et al. 2007, 2000) – focus specifically on finding patterns. Fields such as classification can also find patterns by using decision forests (Breiman 2001a) or other classifiers (Han et al. 2006). Patterns can be assessed for their usefulness (Geng & Hamilton 2006) and their interpretability (Letham et al. 2013), or monitored for any changes in temporal scenarios (Baron et al. 2003). While differing in methodology, approaches such as these agree on the importance of patterns and attest to the value of patterns for gaining knowledge.

It is important to note that assessing the quality of patterns in these ways is not the same as measuring the performance of a model at achieving a goal (Caruana & Niculescu-Mizil 2004, Sokolova & Lapalme 2009). Prediction accuracy is often used to measure a model's performance (Cheng et al. 2007, Letham et al. 2013), but is disconnected from any reliable assessments of pattern quality (Fletcher & Islam 2014, Islam et al. 2003). A user should be aware of the specific goals they wish their model to achieve and how important the truthfulness (Kifer & Gehrke 2006) or interpretability of the patterns in the model are, and then use multiple measures to assess if their needs are met.

Our proposed application of the Jaccard index assesses if two sets of patterns are similar. Attempts to measure the similarity between trusted patterns and newly discovered patterns have been made in the past (Islam & Brankovic 2011), but the measure used was designed for a specific problem, lacking any applicability in a wider context. We discuss the value of widely applicable measures in Section 3.3.

The field of privacy-preserving data mining (Dwork 2008, Fung et al. 2010) is known for struggling with the inherent trade-off that must be made between the amount of privacy provided and the quality of the perturbed data (Fung et al. 2005, Nergiz & Clifton 2007). A naive approach would be to use measures like RMSE to compare the original data to the perturbed data directly (Willmott 1982, Willmott et al. 2009), however this ignores the correlations in the data necessary for information discovery (Agrawal & Aggarwal 2001, Fletcher & Islam 2015b). Just like how frequent pattern mining and other fields use prediction accuracy heavily, so too does privacy-preserving data mining (Chaudhuri et al. 2011, Friedman & Schuster 2010, Fung et al. 2005, Mohammed et al. 2011), but the same problems encountered by the former when assessing patterns also plague the latter (Fletcher & Islam 2014, Islam et al. 2003). Attempts at directly measuring the loss of pattern retention as privacy needs are increased have been made (Fletcher & Islam 2014), but the measure is not applicable in any wider context outside of privacy preservation.

There is therefore a need for insightful ways to compare between patterns gained by different means, regardless of what data mining algorithms or data are used. We do so in this paper, taking both practical and mathematical considerations into account (Meila 2007).

3 Comparing sets of patterns with the Jaccard index

The Jaccard index (Jaccard 1901) is a well-known measurement of the similarity between two sets S and T, defined as the size of the intersection divided by the size of the union of the two sets:

$$J(S,T) = \frac{|\mathrm{S} \cap \mathrm{T}|}{|\mathrm{S} \cup \mathrm{T}|},$$

where we say J(S,T)=1 if |S∪T|=0. By reinterpreting a set of patterns as a set of elements, we are able to use the Jaccard index to measure the similarity between two sets of patterns. In Section 3.1 we describe how we convert patterns in the sets Ψ_1 and Ψ_2 into elements for sets S and T. In Sections 3.3 and 3.4 we outline the practical and mathematical benefits of using the Jaccard index to measure the difference between two sets of patterns.

3.1 Converting patterns into elements in a set

Our aim is to compare two sets of patterns and describe their similarities with an intuitive, quantitative number. To do so with the Jaccard index, we must first translate each pattern $\psi \rightarrow c$

(such as those seen in Table 1) into element *s* of set *S*. Each antecedent ψ is made up of attributes – continuous, discrete or binary attributes – that specify the conditions that must be met in order for the consequent *C* to be predicted to equal *c*. To condense the set ψ as well as *c* into a single element *s*, we use the following equation:

$$s = \mathbf{1}_{\psi}(A_1), \mathbf{1}_{\psi}(A_2), \dots, \mathbf{1}_{\psi}(A_m), c^{\psi}$$

where we use c^{ψ} to simply refer to the class value (consequent) of pattern ψ, and $\mathbf{1}_{\psi}(A)$ is the indicator function:

$$\mathbf{1}_{\psi}(A) := \begin{cases} 1 & A \in \psi \\ 0 & A \notin \psi \end{cases}.$$

Essentially, Equation 2 is recording the presence or absence of each attribute in pattern ψ, as well as recording the consequent *c*. Table 2 illustrates this with converted versions of the patterns seen in Table 1. The three 1's in ψ_0 refer to the positions in the WBC dataset of the three attributes used by ψ_0 in Table 1, and similarly for the other patterns. We include the consequent in *s* because of the role it plays in the definition of a pattern – without a consequent, an antecedent hardly means anything at all.

The conversion described in Equation 2 is lossy when used on non-binary attributes, but is lossless for binary attributes. While there are certainly disadvantages to being lossy with some attributes, there are also several advantages, such as being able to treat categorical and numerical attributes equally, without bias. For example, *Weather=Sunny* is clearly a different pattern to *Weather=Cloudy*, but what about *Weather*=27.85° compared to *Weather*=27.84°? Heuristics could be developed to handle these cases, but ultimately it still relies on heuristics. Instead, our approach asks the question "Does *Weather* make up part of the pattern?" – if it is, then it is more similar to another pattern that also cares about *Weather* than it is to a pattern that does not care about *Weather*. Experimentally, it is found in Section 4 that making these less granular comparisons such as "Do the two patterns both care about *Weather*?" does a good job of distinguishing between two sets of patterns. Simultaneously, the proposed approach maintains a lossless conversion for patterns with binary attributes such as *{milk, eggs}→bread*.

i	s_i
0	110001000a
1	011000000a
2	111001001b
3	011000100b

Table 2 - The encoded versions of the patterns shown in Table 1.

3.2 Steps

The steps for calculating the similarity between two sets of patterns are as follows:

1. Take Ψ_1 and Ψ_2 as input.
2. Convert Ψ_1 and Ψ_2 into sets *S* and *T* respectively, using the process in Section 3.1.
3. For each element in *S*, scan *T* to see if that element exists in *T*. The number of pairs found equals $|S \cap T|$.

4. Deduce the number of unpaired elements in S and T using the results from Step 3. This number plus $|S \cap T|$ equals $|S \cup T|$.
5. Calculate $J(S,T)$.

3.3 Practical considerations

As discussed in Section 1.1, a good measure should not be difficult for data analysts to harness effectively. We assess our proposed application of the Jaccard index with four factors that influence an analyst's decision to use a measure:

3.3.1 Conceptual simplicity

A good measure is one that can be easily and intuitively understood. If a measure has too many moving parts or variables, it can quickly become a "black box" of sorts, where analysts can no longer conceptualize all the possible outputs that the measure could produce. Our implementation of the Jaccard index avoids these risks by being very straight-forward – it is the number of patterns two sets of patterns share, divided by the number of unique patterns across both sets. There are no variables beyond the two sets of patterns, and there are no parameters that need expert knowledge to properly adjust. Our encoding of the patterns into single elements is simple and intuitive, making the conceptualization of the processes involved in the measure no more difficult than in a standard application of the Jaccard Index. Incorporating attribute value conditions into the encoding process would require a robust definition of "similarity" or "distance" for both continuous and discrete attribute values that did not induce any biases in the calculation, and would undoubtedly increase the conceptual complexity of the measure. This is one avenue for future work, but currently appears to be infeasible due to the fundamental differences between continuous and discrete attributes.

3.3.2 Computational simplicity

A measure can become infeasible if it does not scale well as the variables become large. The computation time of our measure is very satisfactory, mostly due to the fact that it does not use the underlying raw data in any way, or even a classifier – it just needs the sets of patterns. Encoding all the patterns in Ψ_1 and Ψ_2 as the sets S and T requires each pattern in both sets to be read and converted once, with the presence of each attribute being checked once in each pattern. Therefore the computational complexity is $O(m(|\Psi_1| + |\Psi_2|))$. Once S and T have been generated, the intersection and union in the Jaccard index can both be calculated by comparing each element in S to every element in T, where each element is m digits long. This gives a computational complexity of $O(m \cdot |S| \cdot |T|)$. Since the length of the sets of patterns and the respective sets of encoded elements will always be equal, the total computation time of our measure is $O(m(|\Psi_1| + |\Psi_2| + |\Psi_1| \cdot |\Psi_2|))$. As a rough example, classification algorithms such as Random Forest (Breiman 2001a) might generate several hundred patterns, and the kinds of datasets that Random Forest might be applied to generally have less than 1000 attributes.

3.3.3 Interpretability

In order for analysts to judge the result of a measure and act on it in a meaningful way, the result needs to be interpretable. This can be as simple as being able to parse the result into a sentence. In our case, our measure can be interpreted as "out of all the patterns that appear in sets of patterns Ψ_1 and Ψ_2, ($J(S,T)$×100)% can be found in both sets". An addendum could be added about how the patterns are compared at a level of granularity that ignores differences in attribute value conditions, without making the measure any more difficult to understand and interpret.

3.3.4 Wide applicability

Measures that accurately encapsulate specific scenarios are undeniably useful, but measures capable of crossing academic discipline boundaries are far more appealing. They provide ways for researchers and professionals to interface with work outside of their personal scope and build connections between otherwise isolated fields of science. Historically, the statistics and machine learning disciplines have learned this lesson the hard way, with the relatively new field of machine learning often recreating mathematics and methods invented previously by statisticians (Breiman 2001b). The ubiquity of prediction accuracy as a measure of performance and the role it plays in creating dialogue among researchers demonstrates the advantages of measures with wide applicability. Our proposed encoding method and application of the Jaccard index is general enough to be viable in any situation involving two groups of patterns discovered from data with the same attributes. Our measure is completely independent of how the patterns were discovered, built, or connected. It can handle non-binary consequents – a common constraint for other measures (Felkin 2007). It can handle continuous consequents as well, but only if the range of values are first discretised into "buckets" (Kotsiantis & Kanellopoulos 2006). It can even work in scenarios without a C component, where ψ might merely represent a collection of attributes that appear together frequently. In this situation, s would intuitively drop the c component and become equal to $\mathbf{1}_{\psi}(A_1), \mathbf{1}_{\psi}(A_2), \ldots, \mathbf{1}_{\psi}(A_m)$.

Our measure is conceptually and computationally simple, it produces easily interpretable results, and it is an appropriate measure in a wide variety of scenarios. Not constraining ourselves by requiring definitions of similarity or distance for continuous and discrete attribute value conditions allows these qualities to be improved even further.

3.4 Mathematical properties

A good measure should be capable of distinguishing between *slightly* different sets of patterns and *very* different sets of patterns. The Jaccard index possess several properties that make this possible, as well as possessing properties that strengthen the versatility of the measure. Our encoding method is designed in a way that does not interfere with these properties.

3.4.1 Bounds

The Jaccard index has the bounds $0 \leq J(S,T) \leq 1$. We can narrow down the upper bound further by using the difference in size between the sets. In situations where $|S|>|T|$, the maximum similarity is when $T \subseteq S$, in which case $J(S,T)=|T|/|S|$. The larger the size difference between the sets, the smaller the upper bound is. To put it formally:

$$J(S,T) \leq \frac{|T|}{|S|}, where\ |S| \geq |T|\,.$$

It is reasonable to assume that the user will know the size of sets, making this upper bound easy to incorporate when interpreting the measure's result, and possibly being quite informative. A similar situation exists for the popular prediction accuracy measure, where the lower bound is equal to the relative frequency of the most common class value.[2]

If the flexibility of the Jaccard index's upper bound is undesirable, a user is free to shorten S until both sets have equal length. The frequent pattern mining discipline does this often, only

[2] Theoretically it can go lower, but then the model is providing no benefit to the user and is actually causing harm – it is worse than a random guess.

selecting patterns with high support (Han et al. 2007, 2000). Many other measures could be used to remove the least valuable patterns depending on how the user defines "valuable" (Geng & Hamilton 2006). A user could also divide the Jaccard index result by the upper bound and reinterpret the result as "the ratio between the actual result and the best possible result", but this approach could easily misguide the user about how similar the sets of patterns actually are and is not recommended.

The lower bound can also be narrowed in very specific scenarios: when the number of patterns in S and T is more than the number of unique patterns that could exist. This indicates that $|S \cap T|>0$; some overlap between the sets must exist. This can occur when the number of unique patterns that could exist given the number attributes and class values in the dataset (i.e. the number of ways you could write s, $|C|(2^m - 1)$ is less than $|S|+|T|$.[3] [4]

It is also possible that the pattern creation process puts a constraint k on the number of attributes that could be present in any one pattern (Webb & Brain 2002). It has been demonstrated that increased pattern length can actually decrease the information gained due to the decreased generality of the patterns (Cheng et al. 2007), as well as decreasing the interpretability of the patterns (Freitas 2013, Huysmans et al. 2011, Letham et al. 2013, Vellido et al. 2012). Note that the data mining processes used to generate Ψ_1 and Ψ_2 might have different k's, denoted k_{Ψ_1} and k_{Ψ_2}, but the combinations possible with a smaller k are a subset of a larger k's combinations. We can write the number of combinations possible in a scenario with constraint k as $\sum_{i=1}^{k}\binom{m}{i}$, where $\binom{m}{i}$ is the binomial coefficient and equals $m!/i!\,(m-i)!$.

Strictly speaking, the lower bound of our measure is

$$J(S,T) \geq max\left(\frac{|S|+|T|-|C|\cdot\sum_{i=1}^{k}\binom{m}{i}}{|C|\cdot\sum_{i=1}^{k}\binom{m}{i}}, 0\right),$$

where $k = max(k_{\Psi_1}, k_{\Psi_1}) \leq m$. If k=m (i.e. No constraint was put on the length of the patterns),

$$|C|\cdot\sum_{i=1}^{m}\binom{m}{i} = |C|(2^m - 1)\,.$$

To use some example numbers, if $|S|+|T|$=100 and there are five attributes and two class values, there are $2(2^5\text{-}1)$=62 possible combinations of attributes and class values. The lower bound then becomes 0.61, ruling out over half the original range. We reiterate that in most situations, the lower bound will simply equal 0. These simple upper and lower bounds allow the user to interpret the measure's results with far more insight than would be possible if the bounds were ignored.

3.4.2 Metric properties

Metrics are a subset of measures, defined by four mathematical properties they possess: non-negativity; identity of indiscernibles; symmetry; and triangle inequality. We can describe each of these, respectively, with the Jaccard index:

- $J(S,T)\geq 0$
- $J(S,T)=1 \Leftrightarrow S=T$

[3] We subtract one from 2^m because an empty ψ is not a legal antecedent.

[4] Note that $|S \cup T| \leq |S|+|T|$.

- $J(S,T)$=$J(T,S)$
- $J(S,U) \geq J(S,T) + J(T,U)$. Note that we use "≥" since the Jaccard index is traditionally a metric of similarity, not distance. If a distance is preferred, the Jaccard Distance can be very easily used: $d_j(S,T) = 1 - J(S,T)$.

It is straight-forward to see how the Jaccard index satisfies the first three properties, since its maximum bounds are $0 \leq J(S,T) \leq 1$ and Equation 1 does not change if S and T are swapped. A proof of the Jaccard Index satisfying the triangle inequality has been previously constructed (Levandowsky & Winter 1971, Lipkus 1999). Because the Jaccard index has these properties, mathematicians can use them when constructing proofs and can use more assumptions that are guaranteed to hold true. The triangle inequality especially is well-known as a strong mathematical property when analysing metrics (Chawla et al. 2005) and designing efficient data structures and algorithms (Meila 2007). Measures that are not metrics often output results that require unintuitive interpretations (Meila 2007, Willmott et al. 2009).

4 Experiments

Below, we explore two example scenarios of how our proposed measure can be used as a complementary measure to prediction accuracy, discovering new information that prediction accuracy cannot tell us. In Section 4.1 we compare two variations of the CART algorithm in terms of pattern similarity and prediction accuracy. In Section 4.2 we use the same two measures, but this time we compare two privacy-preserving decision tree algorithms.

4.1 Scenario 1: comparing classifiers

In this section we perform a short case study on one of the example scenarios given in Section 1.1: comparing classifiers with different parameters. Specifically, we use the implementation of the CART classifier (Breiman et al. 1984) found in the scikit-learn software (Pedregosa et al. 2011), and we compare two different objective functions for the splitting criteria: the Gini index (Breiman et al. 1984) and information gain (Quinlan 1996). This comparison represents a reasonably straight-forward scenario where a common question is being asked: "How different would our results be if we changed classifier?" By using single decision trees (rather than decision forests or other classifiers), we generate a manageable number of patterns directly from the classifier, rather than needing to filter a larger number of patterns down using additional processes. By using a simple experimental set-up, our results are easily reproducible.

Dataset	Jaccard Index	Jaccard Upper Bound	Prediction Accuracy Difference
WBC	0.077	6/8=0.750	0.007
Vehicle	0.184	21/24=0.875	0.011
Banknotes	0.750	7/7=1.000	0.027
RedWine	0.125	21/24=0.875	0.013
Spambase	0.100	21/23=0.913	0.008
PageBlocks	0.546	14/16=0.889	0.004
OptDigits	0.019	26/29=0.897	0.022
PenWritten	0.000	21/24=0.875	0.020
GammaTele	0.310	18/20=0.900	0.005
Shuttle	0.714	6/6=1.000	0.000
Credit	0.136	12/13=0.920	0.014
Yeast	0.364	14/16=0.875	0.004
Cardio	0.143	7/9=0.778	0.007
Adult	0.400	17/18=0.944	0.001
Bank	0.172	17/17=1.000	0.008
TicTacToe	0.161	17/19=0.895	0.004
Car	0.400	6/8=0.750	0.028
Nursery	0.615	10/11=0.909	0.000
Chess	0.476	14/17=0.824	0.005

Table 3 - The Jaccard index and prediction accuracy difference between CART-G and CART-I for 19 datasets.

For this experiment, patterns in the form $\psi \rightarrow c$ are extracted from the decision trees by "flattening" each root-to-leaf path. That is, a chain of nodes from the root node to a leaf node (where the most common class value in the leaf node is the predicted class value) forms a pattern. Each leaf node is part of one pattern. Root-to-leaf paths are an ordered list, and the same attribute may appear multiple times, but these properties disappear when a root-to-leaf path is flattened into a list of conditions ψ.

We apply the CART classifier on 19 datasets from the UCI Machine Learning Repository (Bache & Lichman 2013) using five-fold cross-validation with stratified folds[5], and measure the Jaccard index between CART with the Gini index (referred to as CART-G) and CART with information gain (referred to as CART-I). We also measure the prediction accuracy of both CART-G and CART-I on the test data from the unused fold. Our aim is to see whether the user could learn anything new by going beyond a simple comparison of prediction accuracies and also comparing the classifiers with the Jaccard index.

For each dataset in our experiments, the minimum support threshold of each leaf in the decision trees is 2% of the records. The lower bound of the Jaccard index is 0 for all datasets. The lower bound for prediction accuracy depends on the relative frequency of the majority class label in each dataset. The upper bound for the Jaccard index depends on the number of patterns found with each classifier and is reported along with the Jaccard index results in Table 3. Since our question is "How different would our results be if we changed classifier?", we

[5] Stratified folds have the same distribution of class values as the dataset as a whole.

report the absolute difference between the prediction accuracy of CART-G and CART-I, also in Table 3. We display the same information as a scatter plot in Figure 1, where we graph the Jaccard index and prediction accuracy difference for each dataset as a point in 2D space.

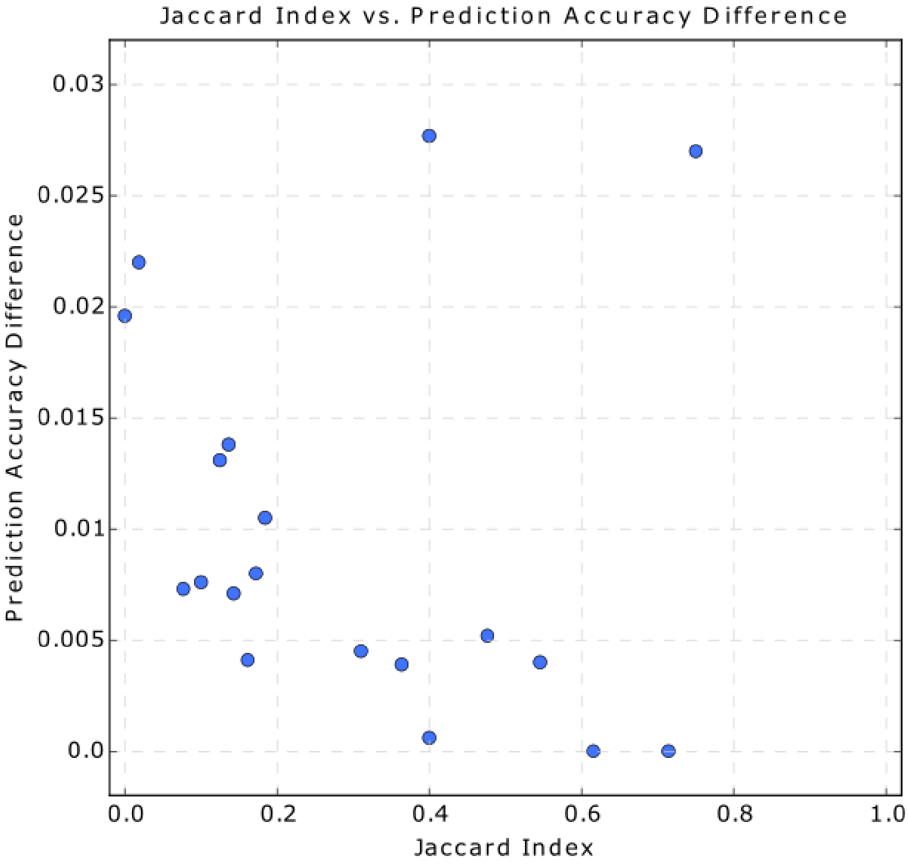

Figure 1 - The Jaccard index compared to the prediction accuracy difference for each dataset. Note the scope of the axes – the maximum prediction accuracy difference is small, while the Jaccard index results make use of the full range between the bounds.

We can see that the detected differences in prediction accuracy are small – too small to trust that the detected difference is solely due to the classifiers (Hand 2006), or that the result is precise enough to remain consistent after repeated runs of the experiment (Fletcher & Islam 2014). If a user found themselves in the scenario described above, the only thing they could reasonably conclude from the prediction accuracy results is "the two classifiers are very similar". However the Jaccard index results inform the user that this is not true at all – the patterns uncovered by the two classifiers are similar for some datasets, but on others they can have very few patterns in common. It turns out that while there is almost no difference in prediction accuracy when choosing either the Gini index or information gain, the structure of the trees can be very different for some datasets! The Jaccard index allows the user to learn information such as "only 40% of the patterns in the Adult dataset found by either classifier were found by both classifiers", while prediction accuracy tells the user "for the Adult dataset, the two classifiers differ by only a tenth of a percent when predicting the consequent of unseen records". Our proposed measure does not replace prediction accuracy, but instead provides information that prediction accuracy (or any other measure, to the best of our knowledge) is incapable of providing.

4.2 Scenario 2: comparing anonymization techniques

In this section we perform another case study on one of the example scenarios given in Section 1.1; this time we compare two privacy-preserving data mining algorithms. Both are decision tree algorithms that use differential privacy to ensure that the trees do not leak personal information, but they do so in different ways. We refer the reader to the respective papers for details of the algorithms if they are so inclined: FI (Fletcher & Islam 2015a) and FS (Friedman & Schuster 2010).

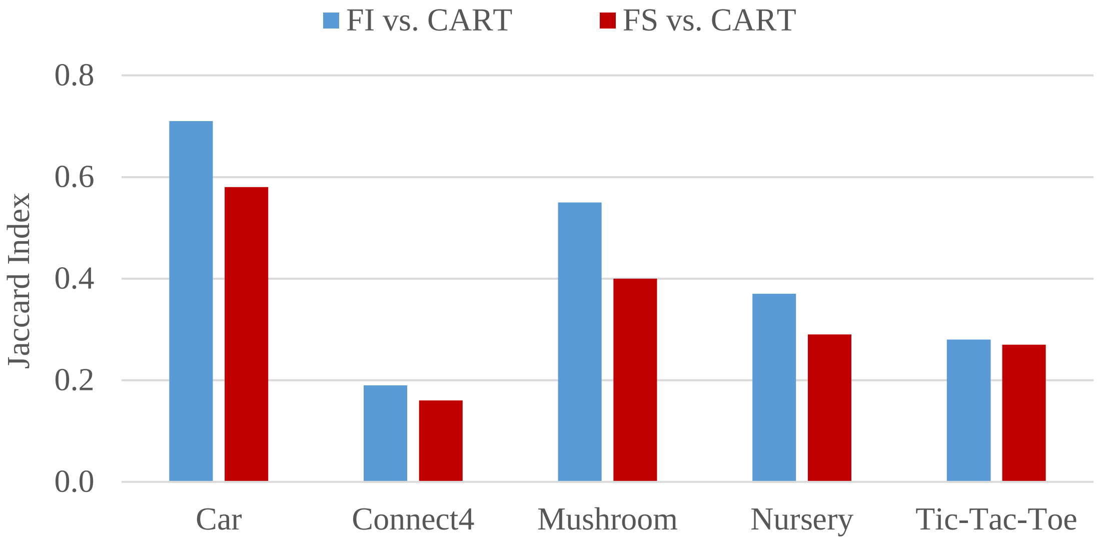

Figure 2 - The similarity between the sets of patterns produced by FI and CART, compared to the similarity between the patterns produced by FS and CART.

We compare FI to FS using the Jaccard index in Figure 2. Here, we are measuring how many patterns discovered by FI are also discovered by the (non-private) CART-G algorithm (Breiman et al. 1984), and then doing the same with FS. We build one tree with each algorithm[6], extract all the root-to-leaf paths in each tree, convert the paths into elements of a set as described in Section 3.1, and measure their similarity to each other. This process was performed on five datasets and repeated with five-fold cross-validation with stratified folds, and Figure 2 presents the aggregated results. By using the Jaccard index, we learn that FI finds more of the patterns discovered by CART (where privacy requirements are not perturbing the model) than FS does. Since the patterns discovered by CART are not perturbed, we can trust that they are more representative of real-world phenomena than patterns discovered under privacy restrictions.

[6] The same parameters are used for each tree: maximum depth of δ=5; privacy budget of ε=2; forest size of τ=1.

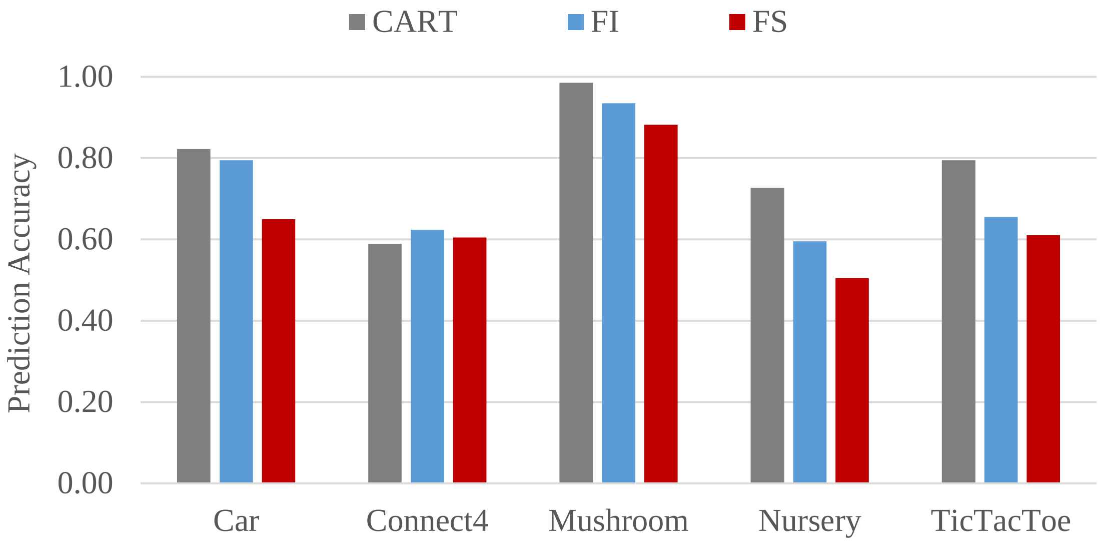

Figure 3 - The prediction accuracy of a tree built using FI (Fletcher & Islam 2015a), FS (Friedman & Schuster 2010) and CART (Breiman et al. 1984).

Accompanying Figure 2, we also provide the prediction accuracy of the three algorithms in Figure 3. Here we confirm that CART achieves higher accuracy, as expected for a non-private algorithm, and discover that FI outperforms FS in terms of prediction accuracy. By using two different measures, and learning different things from each of them, the usefulness of having workload-specific measures becomes clear. No measure can claim to be a "one size fits all" solution for quantifying what the user is interested in, and our Jaccard index measure is no exception. It does, however, perform well as a "one size fits some" solution (as is true of any good measure), and informs us about the quality of the patterns discovered by different algorithms.

5 Conclusion

In this paper we propose a method for measuring the similarity between two sets of patterns. The method was designed with a focus on conceptual simplicity, computational simplicity, interpretable results, and applicability in a wide variety of scenarios. One such scenario was explored in detail to demonstrate the information a user could learn from our proposed measure. Strong mathematical properties are provided to aid users in interpreting their results and making comparisons between results. For example if a third classifier was added to the scenario portrayed in Section 4, the triangle inequality allows users to use their intuition that the similarity between classifier A and C cannot be lower than the sum of the similarities between classifiers A and B and classifiers B and C.

Our use of the Jaccard index successfully measures an aspect of data mining results that no pre-existing measure is able to do. As the data mining community continues to put more focus on the discovery of interpretable patterns in data, the ability to distinguish between different sets of patterns will be highly useful.

References

Agrawal, D. & Aggarwal, C. (2001), "On the design and quantification of privacy preserving data mining algorithms", *20th ACM SIGMODSIGACT-SIGART Symposium on Principles of Database Systems*, ACM, pp. 247–255.

Aodha, O. M., Stathopoulos, V., Terry, M., Jones, K. E., Brostow, G. J. & Girolami, M. (2014), "Putting the Scientist in the Loop – Accelerating Scientific Progress with Interactive Machine Learning", *22nd International Conference on Pattern Recognition*, IEEE, Stockholm, Sweden, pp. 9–17.

Bache, K. & Lichman, M. (2013), UCI Machine Learning Repository, URL: *http://archive.ics.uci.edu/ml/*

Baron, S., Spiliopoulou, M. & Gunther, O. (2003), "Efficient monitoring of 21 patterns in data mining environments", *7th East-European Conference on Advances in Databases and Informations Systems*, Springer, Dresden, Germany, pp. 253–265.

Breiman, L. (2001a), "Random forests", *Machine Learning,* **45**(1): 5–32.

Breiman, L. (2001b), "Statistical Modeling: The Two Cultures", *Statistical Science,* **16**(3): 199–231.

Breiman, L., Friedman, J., Stone, C. & Olshen, R. (1984), *Classification and Regression Trees*, Chapman & Hall/CRC.

Caruana, R. & Niculescu-Mizil, A. (2004), "Data Mining in Metric Space: An Empirical Analysis of Supervised Learning Performance Criteria", *10th SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, Seattle, Washington, pp. 69–78.

Chaudhuri, K., Monteleoni, C. & Sarwate, A. (2011), "Differentially private empirical risk minimization", *The Journal of Machine Learning Research,* **12**(1): 1069–1109.

Chawla, S., Dwork, C. & McSherry, F. (2005), "Toward privacy in public databases", *Theory of Cryptography*, pp. 363–385.

Cheng, H., Yan, X., Han, J. & Hsu, C.W. (2007), "Discriminative Frequent Pattern Analysis for Effective Classification", *23rd International Conference on Data Engineering*, IEEE, pp. 716–725.

Dwork, C. (2008), "Differential Privacy: A survey of results", *Theory and Applications of Models of Computation*, Springer, Xi'an, China, pp. 1–19.

Felkin, M. (2007), "Comparing classification results between n-ary and binary problems", *Quality Measures in Data Mining*, Springer, pp. 277–301.

Fletcher, S. & Islam, M. Z. (2014), "Quality evaluation of an anonymized dataset", *22nd International Conference on Pattern Recognition*, IEEE, Stockholm, Sweden, pp. 3594–3599.

Fletcher, S. & Islam, M. Z. (2015a), "A Differentially Private Decision Forest", *13th Australasian Data Mining Conference*, Sydney, Australia, pp. 1–10.

Fletcher, S. & Islam, M. Z. (2015b), "Measuring Information Quality for Privacy Preserving Data Mining", *International Journal of Computer Theory and Engineering,* **7**(1): 21–28.

Freitas, A. (2013), "Comprehensible classification models: A position paper", *ACM SIGKDD Explorations Newsletter,* **15**(1): 1–10.

Friedman, A. & Schuster, A. (2010), "Data Mining with Differential Privacy", *16th SIGKDD Conference on Knowledge Discovery and Data Mining,* ACM, Washington, DC, USA, pp. 493–502.

Fung, B., Wang, K., Chen, R. & Yu, P. (2010), "Privacy-preserving data publishing: A survey of recent developments", *ACM Computing Surveys***, 42**(4): 1–53.

Fung, B., Wang, K. & Yu, P. (2005), "Top-down specialization for information and privacy preservation", *21st International Conference on Data Engineering*, IEEE, pp. 205–216.

Geng, L. & Hamilton, H. J. (2006), "Interestingness measures for data mining: a survey", *ACM Computing Surveys,* **38**(3): 1–32.

Han, J., Cheng, H., Xin, D. & Yan, X. (2007), "Frequent pattern mining: current status and future directions", *Data Mining and Knowledge Discovery,* **15**(1): 55–86.

Han, J., Kamber, M. & Pei, J. (2006), *Data Mining: Concepts and Techniques*, Morgan Kaufmann Publishers, San Diego.

Han, J., Pei, J. & Yin, Y. (2000), "Mining frequent patterns without candidate generation", *ACM SIGMOD Record*, ACM, Dallas, Texas, **12**(1): 1–12.

Hand, D. J. (2006), "Classifier Technology and the Illusion of Progress", *Statistical Science*, **21**(1): 1–14.

Huysmans, J., Dejaeger, K., Mues, C., Vanthienen, J. & Baesens, B. (2011), "An empirical evaluation of the comprehensibility of decision table, tree and rule based predictive models", *Decision Support Systems,* **51**(1): 141–154.

Islam, M. Z., Barnaghi, P. & Brankovic, L. (2003), "Measuring Data Quality: Predictive Accuracy vs. Similarity of Decision Trees", *6th International Conference on Computer & Information Technology*, Dhaka, Bangladesh, pp. 457–462.

Islam, M. Z. & Brankovic, L. (2011), "Privacy preserving data mining: A noise addition framework using a novel clustering technique", *Knowledge-Based Systems,* **24**(8): 1214–1223.

Islam, M. Z. & Giggins, H. (2011), "Knowledge discovery through SysFor: a systematically developed forest of multiple decision trees", *9th Australasian Data Mining Conference,* Australian Computer Society, Inc., Ballarat, Australia, pp. 195–204.

Jaccard, P. (1901), "Etude comparative de la distribution florale dans une portion des Alpes et du Jura", *Bulletin de la Soci´et´e Vaudoise des Sciences Naturelles* **37**(1): 547–579.

Kifer, D. & Gehrke, J. (2006), "Injecting utility into anonymized datasets", *SIGMOD International Conference on Management of Data,* ACM, New York, New York, USA, pp. 217–228.

Kotsiantis, S. & Kanellopoulos, D. (2006), "Discretization techniques: A recent survey", *GESTS International Transactions on Computer Science and Engineering* **32**(1): 47–58.

Letham, B., Rudin, C., Mccormick, T. H. & Madigan, D. (2013), "Interpretable classifiers using rules and Bayesian analysis: Building a better stroke prediction model", Technical Report 609, University of Washington.

Levandowsky, M. & Winter, D. (1971), "Distance between sets", *Nature,* **234**(5323): 34–35.

Lipkus, A. (1999), "A proof of the triangle inequality for the Tanimoto distance", *Journal of Mathematical Chemistry,* **26**(1-3): 263–265.

Meila, M. (2007), "Comparing clusterings – an information based distance", *Journal of Multivariate Analysis,* **98**(1): 873–895.

Mohammed, N., Chen, R., Fung, B. C. & Yu, P. S. (2011), "Differentially private data release for data mining", *7th SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, New York, New York, USA, pp. 493–501.

Nergiz, M. E. & Clifton, C. (2007), "Thoughts on k-anonymization", *Data & Knowledge Engineering,* **63**(3): 622–645.

Ordonez, C. & Zhao, K. (2011), "Evaluating association rules and decision trees to predict multiple target attributes", *Intelligent Data Analysis,* **15**(1): 173–192.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. & Duchesnay, E. (2011), "Scikit-learn: Machine Learning in Python", *Journal of Machine Learning Research* **12**(1): 2825–2830.

Quinlan, J. R. (1993), *C4.5: Programs for Machine Learning*, 1st edn, Morgan kaufmann.

Quinlan, J. R. (1996), "Improved Use of Continuous Attributes in C4.5", *Journal of Artificial Intelligence Research* **4**(1): 77–90.

Shotton, J., Sharp, T., Kohli, P., Nowozin, S., Winn, J. & Criminisi, A. (2013), "Decision Jungles: Compact and Rich Models for Classification", *Advances in Neural Information Processing Systems*, pp. 234–242.

Sokolova, M. & Lapalme, G. (2009), "A systematic analysis of performance measures for classification tasks", *Information Processing & Management* **45**(4): 427–437.

Vellido, A., Martin-Guerroro, J. D. & Lisboa, P. J. (2012), "Making machine learning models interpretable", *European Symposium on Artificial Neural Networks*, Bruges, Belgium, pp. 163–172.

Wagstaff, K. L. (2012), "Machine Learning that Matters", *29th International Conference on Machine Learning*, ACM, Edinburgh, Scotland, pp. 1–6.

Webb, G. & Brain, D. (2002), "Generality is predictive of prediction accuracy", *Pacific Rim Knowledge Acquisition Workshop*, pp. 117–130.

Willmott, C. J. (1982), "Some comments on the evaluation of model performance", *Bulletin of the American Meteorological Society,* **63**(11): 1309–1313.

Willmott, C. J., Matsuura, K. & Robeson, S. M. (2009), "Ambiguities inherent in sums-of-squares-based error statistics", *Atmospheric Environment,* **43**(3): 749–752.