DESIGN AND ADJUSTMENT OF DEPENDENCY MEASURES

Simone Romano

Supervisors:
Prof. James Bailey         A/Prof. Karin Verspoor

Submitted in total fulfilment of the requirements of the degree of Doctor of Philosophy

Produced on archival quality paper

December 2015
Dependency measures are fundamental for a number of important applications in data mining and machine learning. They are ubiquitously used: for feature selection, for clustering comparisons and validation, as splitting criteria in random forest, and to infer biological networks, to list a few. More generally, there are three important applications of dependency measures: detection, quantification, and ranking of dependencies. Dependency measures are estimated on finite data sets and because of this the tasks above become challenging. This thesis proposes a series of contributions to improve performances on each of these three goals.

When differentiating between strong and weak relationships using information theoretic measures, the variance plays an important role: the higher the variance, the lower the chance to correctly rank the relationships. In this thesis, we discuss the design of a dependency measure based on the normalized mutual information whose estimation is based on many random discretization grids. This approach allows us to reduce its estimation variance. We show that a small estimation variance for the grid estimator of mutual information if beneficial to achieve higher power when the task is detection of dependencies between variables and when ranking different noisy dependencies.

Dependency measure estimates can be high because of chance when the sample size is small, e.g. because of missing values, or when the dependency is estimated between categorical variables with many categories. These biases cause problems when the dependency must have an interpretable quantification and when ranking dependencies for feature selection. In this thesis, we formalize a framework to adjust dependency measures in order to correct for these biases. We apply our adjustments to existing dependency measures between variables and show how to achieve better interpretability in quantification. For example, when a dependency measure is used to quantify the amount of noise on functional dependencies between variables, we experimentally demonstrate that adjusted measures have more interpretable range of variation. Moreover, we demonstrate
that our approach is also effective to rank attributes during the splitting procedure in random forests where a dependency measure between categorical variables is employed.

Finally, we apply our framework of adjustments to dependency measures between clusterings. In this scenario, we are able to analytically compute our adjustments. We propose a number of adjusted clustering comparison measures which reduce to well known adjusted measures as special cases. This allows us to propose guidelines for the best applications of our measures as well as for existing ones for which guidelines are missing in literature, e.g. for the Adjusted Rand Index (ARI).
DECLARATION

This is to certify that:

(a) The thesis comprises only my original work towards the degree of Doctor of Philosophy except where indicated in the Preface;

(b) Due acknowledgement has been made in the text to all other material used;

(c) The thesis is fewer than 80,000 words in length, exclusive of tables, maps, bibliographies and appendices.

Simone Romano
This thesis has been written at the Department of Computing and Information Systems, The University of Melbourne. Each chapter is based on manuscripts published or under review for publication. I declare that I am the primary author and have contributed to more than 50% of each of these papers.

Chapter 3 is based on the paper:


Chapter 4 is based on the paper:


Chapter 5 is based on the papers:


During the course of this thesis, several fruitful collaborations have also led to the following manuscripts. These are not discussed within this dissertation.

- “Extending Information Theoretic Validity Indices for Fuzzy Clusterings”, Yang Lei, James Bezdek, Nguyen Xuan Vinh, Jeffrey Chan, Simone Romano, and James Bailey. Under review in Transactions on Fuzzy Systems Journal (TFSJ)

- “Discovering Outlying Aspects in Large Datasets”, Nguyen Xuan Vinh, Jeffrey Chan, Simone Romano, James Bailey, Christopher Leckie, Kotagiri Ramamohanarao, and Jian Pei. To appear in Data Mining and Knowledge Discovery Journal (DAMI)

- “Generalized Information Theoretic Cluster Validity Indices for Soft Clusterings”, Yang Lei, James Bezdek, Nguyen Xuan Vinh, Jeffrey Chan, Simone Romano, and James Bailey. Published in Proceedings of Computational Intelligence and Data Mining 2014, pp. 24–31 (CIDM-14)

- “Effective Global Approaches for Mutual Information Based Feature Selection”, Nguyen Xuan Vinh, Jeffrey Chan, Simone Romano, and James Bailey. Published in Proceedings of the 20th International Conference on Knowledge Discovery and Data Mining 2014 (KDD-14), pp. 512–521
ACKNOWLEDGMENTS

Today I am submitting my PhD thesis, and this is one of the happiest days of my life. However, this would not have been possible without the support of the people I am mentioning below.

First of all, I would like to sincerely thank my supervisors: Professor James Bailey and Associate Professor Karin Verspoor. I consider myself very lucky of studying at the University of Melbourne with amazing supervisors. They have been always supportive, and their enthusiasm, as well as their deep knowledge in the field helped me to overcome obstacles I could not imagine to overcome. In particular, I would like to thank James who always believed in me and who always managed to keep me motivated with his positive attitude. Thank you for being a really great supervisor, and also for providing financial support through the whole PhD. I would like to thank Karin who showed me that one can achieve any professional accomplishment in both industry and academia while remaining a down to earth person. Thanks also for all the informal discussions we had which helped me being an open minded person, and for providing me with employment as a tutor for your subject. I would also like to mention and thank Doctor Nguyen Xuan Vinh. Even if he is not listed as a formal supervisor of mine, he acted as such. I can fairly say that without his expertise I would not have been able to obtain the results I got.

I would like to thank Professor Tim Baldwin for accepting to be a member of my Advisory Committee and for his invaluable insights and advice throughout my candidature. Moreover, I would like to acknowledge the University of Melbourne, the Computer and Information Systems (CIS) Department, and the Head of the Department Justin Zobel for the great opportunity I had.

I cannot forget to thank all the guys in the Machine Learning group. We have not just shared an office for the last few years, we shared the whole PhD experience through its highs and lows. I am honestly very happy to have shared all these moments with
you. In particular, I would like to say thanks especially to: Sergey Demyanov, Goce Ristanoski, Jiazhen He, and Florin Schimbinschi. Moreover, I would like to thank all the workmates from the CIS department: Andrey Kan, Shuo Zhou, Yun Zhou, Yang Lei, Yamuna Kankanige, Daniel, Alvin, Mohadesheh Ganji, Nikolay Grozev, Deepak Poola, Pallab Roy, Kian Ho, Marco Lui and the many others that I am forgetting to mention here.

During the years I spent in Melbourne I had the chance to meet tons of friends. I am not going to list all their names, as they are simply too many and I do not want to undergo the risk of forgetting someone. Nonetheless, I wish to particularly thank the closest friends of mine who shared my Melbournian experience: Tommy, Riccardo, Filippo, Andrea, Adriano, Ture, Miko, and Michael. Furthermore, I am grateful to all my Italian friends from my hometown Padova: our long-lasting friendship will never fade away. May there be more fun time together.

Of course, this would have not been possible without the love of my family. I thank you for the endless Skype chats listening to my rants and complaints, and for your invaluable advice. Even though you were not physically here during my PhD, I know you have always loved and supported me from Italy. Thanks Mum, Dad, Veronica, nonna Rosa, and nonna Emma.

Last but not least, I wish to sincerely thank you Sanna for putting up with me during the toughest last years of my PhD. Thanks for believing in me.

Thank you all,

Simone
## CONTENTS

1 INTRODUCTION
   1.1 Thesis Overview ................................................. 3

2 BACKGROUND
   2.1 Dependency Measures ............................................ 7
   2.2 Dependency Measures between Categorical Variables ............ 9
      2.2.1 Mutual Information ......................................... 10
      2.2.2 Pair-counting Measures ..................................... 13
      2.2.3 Accuracy and Agreement ..................................... 14
      2.2.4 Splitting Criteria in Classification Trees .................. 14
   2.3 Dependency Measures between Numerical Variables ............... 17
      2.3.1 Mutual Information ......................................... 17
      2.3.2 Correlation Based Measures .................................. 20
      2.3.3 Kernel Based Measures ...................................... 23
      2.3.4 New Measures Based on Information Theory and Discretization . 23
   2.4 Applications of Dependency Measures ............................. 26
      2.4.1 Feature Selection and Decision Tree Induction ............... 26
      2.4.2 Clustering Comparisons ...................................... 27
      2.4.3 Exploratory Analysis ........................................ 29

3 RANKING DEPENDENCIES IN NOISY DATA ............................. 31
   3.1 Introduction ..................................................... 31
   3.2 Related Work .................................................... 33
   3.3 The Randomized Information Coefficient .......................... 35
   3.4 Variance Analysis of RIC ........................................ 40
      3.4.1 Ensembles for Reducing the Variance .......................... 40
      3.4.2 Importance of Variance in Comparing Relationships using the
           Grid Estimator of Mutual Information .......................... 43
   3.5 Experiments on Dependency Between Two Variables ............... 47
3.5.1 Identification of Noisy Relationships ........................................ 49
3.5.2 Application to Network Inference ........................................... 59
3.5.3 Feature Filtering for Regression .............................................. 63
3.5.4 Run Time Comparison .......................................................... 66
3.6 Experiments on Dependency Between Two Sets of Variables .............. 67
  3.6.1 Identification of Multi-variable Noisy Relationships .................... 67
  3.6.2 Feature Selection for Regression .......................................... 72
3.7 Conclusions .................................................................................. 75

4 A FRAMEWORK TO ADJUST DEPENDENCY MEASURE ESTIMATES .......... 77
  4.1 Introduction ............................................................................... 77
  4.2 Related Work ........................................................................... 80
    4.2.1 Use of the Null for Quantification ...................................... 80
    4.2.2 Use of the Null for Ranking .............................................. 82
  4.3 Adjusting Estimates for Quantification ....................................... 83
    4.3.1 Experiments with Pearson Correlation and MIC ..................... 85
  4.4 Adjusting Estimates for Ranking ................................................ 87
    4.4.1 Ranking Biases of Raw and Standardized Measures (p-values) ... 91
    4.4.2 Experiments with Pearson Correlation and MIC ..................... 94
    4.4.3 Experiments with Gini gain in Random Forests ...................... 97
  4.5 Conclusion .................................................................................. 100

5 ADJUSTING CLUSTERING COMPARISONS MEASURES ......................... 103
  5.1 Introduction ............................................................................... 103
  5.2 Generalized Information Theoretic Measures ................................ 107
    5.2.1 Normalized Generalized IT Measures .................................. 110
  5.3 Baseline Adjustment ................................................................... 111
    5.3.1 Baseline Adjustment for Generalized IT measures .................. 113
    5.3.2 Computational complexity .................................................. 115
    5.3.3 Experiments on Measure Baseline ........................................ 116
    5.3.4 Large Number of Objects .................................................... 117
  5.4 Application scenarios for AMI_q ................................................. 121
5.4.1 Use AMI\(_q\) with small \(q\) such as AMI\(_{0.5}\) or AMI\(_1 = \text{AMI}\) when the reference clustering is unbalanced and there exist small clusters . 123

5.4.2 Use AMI\(_q\) with big \(q\) such as AMI\(_{2.5}\) or AMI\(_2 = \text{ARI}\) when the reference clustering has big equal sized clusters . . . . . . . . . . . 124

5.5 Standardization of Clustering Comparison Measures . . . . . . . . . . . 124

5.5.1 Standardization of Generalized IT Measures . . . . . . . . . . . 129

5.5.2 Computational complexity . . . . . . . . . . . . . . . . . . . . . 132

5.5.3 Large Number of Objects . . . . . . . . . . . . . . . . . . . . . 133

5.6 Experiments on Standardized Clustering Comparison Measures . . . . . 134

5.6.1 Interpretation of Standardized Clustering Comparison Measure . . 134

5.6.2 Bias Towards Clusterings with More Clusters . . . . . . . . . . . 135

5.6.3 Bias Towards Clusterings Estimated on Fewer Data Points . . . . . 137

5.6.4 SMI Running Time . . . . . . . . . . . . . . . . . . . . . . . . . . 137

5.6.5 Bias Correction Under Independent Clusterings with SMI\(_q\) . . . 139

5.6.6 Guidelines for the Use of Standardized Measures . . . . . . . . . . 142

5.7 Conclusion . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 143

6 CONCLUSIONS . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 145

6.1 Implications of the Presented Work . . . . . . . . . . . . . . . . . . . 147

6.2 Limitation of Current Research and Future Work . . . . . . . . . . . 149

6.2.1 Future Work About Randomized Estimations . . . . . . . . . . 149

6.2.2 Future Work About Adjustments of Dependency Measures . . . . 150
INTRODUCTION

Dependency measures are extensively used in data mining and machine learning to assess the amount of dependency between the variables in a data set. Variable selection for classification and regression in a supervised learning task is one of the main applications of dependency measures: they are used to identify and rank the most predictive variables to the target class [Guyon and Elisseeff, 2003]. They are also fundamental for decision tree induction [Criminisi et al., 2012]. During the decision tree induction process, the most predictive variable to the target class is identified according to a dependency measure in order to meaningfully split internal nodes in the tree. Even the prediction accuracy of a classification model can be seen as a dependency measure: it assesses the dependency between the predicted class and actual class [Witten et al., 2011].

Dependency measures are also extensively used in unsupervised learning in the clustering community [Aggarwal and Reddy, 2013]. In this case, their main application is external clustering validation [Hubert and Arabie, 1985]: clustering solutions obtained using different clustering algorithms are compared to a ground truth clustering of reference. Furthermore, dependency measures based on correlation are sometimes the core technique of clustering algorithms [Böhm et al., 2004] and they are used to compare clusterings in many other applications. For example, high dimensional data sets allow multiple valid clusterings. These different clusterings, also called different views, can be compared and analyzed with dependency measures [Müller et al., 2013]. Dependency measures can guide the generation of alternative clustering solutions and compare their diversity [Kontonasios and De Bie, 2013; Dang and Bailey, 2015], can be used to assess the similarity between clusterings in a cluster ensemble [Strehl and Ghosh, 2003], and can be used to explore the clustering space using results from the Meta-Clustering algorithm [Caruana et al., 2006; Lei et al., 2014b] when the task it to find similar/dissimilar clusterings from a query one.
Dependency measures between variables are also used for exploratory analysis in machine learning and data mining. There exist applications on inference of cellular, metabolic, gene regulatory, biological, and signaling networks [Schölkopf et al., 2004; Villaverde et al., 2013]. In this case, the dependency measure is used for reverse engineering the network of interaction between variables. Moreover, dependency measures are also extensively employed to compare neural time-series [Cohen, 2014] to explore the interaction between different areas of the brain.

There are three important applications of dependency measures between variables [Reimherr et al., 2013]:

**Detection:** Test for the presence of dependency. It is useful in case the dependency is not trivial or even its existence is questionable. For example, assess if there exists any dependency between bacterial species that colonize the gut of mammals [Reshef et al., 2011] or identify genes whose expression level oscillates [Sugiyama and Borgwardt, 2013];

**Quantification:** Summarization of the amount of dependency in an interpretable fashion. In this case, the presence of dependency has already been established and the target is to provide insights on the amount of dependency. The value of the dependency measure must be meaningful and must range in a predefined interval to aid intuition. For example, when the squared Pearson’s correlation coefficient is used to quantify the amount of linear dependency between two variables. This has a clear interpretation and the possible values range in $[0, 1]$;

**Ranking:** Sort the relationships of different variables based on the strength of their dependency. In this case, it is less important to obtain an interpretable number out of a dependency measure given that the user is interested only in the accurate ranking. For example, when a dependency measure is used to rank predictive variables to the target class during the decision tree induction phase.

However even though a dependency measure has nice theoretical properties, dependencies are estimated on a finite data set through an estimator and the tasks above become challenging. A dependency measure estimator is a random variable that can attain different possible values. When *detecting* dependencies, the estimates of a real dependency
have to be easily discriminated from the estimates under lack of dependency. If the range of possible values under real dependency and the range of possible values under lack of dependency overlap, a dependency measure has low detection power [Simon and Tibshirani, 2011]. When quantifying dependency the estimated values are dependent on the data sample used for the estimation. Dependency measure estimates can be high because of chance when the sample size is small, e.g., because of missing values, or when the dependency is estimated between categorical variables with many categories. Therefore, dependency estimates cannot be meaningfully compared between data sets and this is particularly detrimental if the dependency measure must have a clear interpretation. Moreover, this issue causes problems when ranking dependencies. Estimates take into account factors which are not dependent only on the strength of the dependency, and this causes biases in ranking variables with missing values or categorical variables with many categories.

The body of work presented in this thesis contributes to the development of statistical techniques to adjusting and designing dependency measure estimators to solve these challenges.

1.1 Thesis Overview

Chapter 2 defines fundamental concepts in machine learning and the different types of dependency measures available in literature as well as examples for their application scenarios.

In Chapter 3, we discuss the design of a dependency measure between continuous variables based on the estimation of the normalized mutual information [Cover and Thomas, 2012]. In order to estimate the normalized mutual information between numerical variables, we employ random discretization grids. We show that this helps in reducing the variance while minimizing the impact on bias. This is particularly useful when detecting and ranking noisy dependencies: the estimates obtained on different relationships have small variance, therefore the range of possible values under one relationship has small overlap with the range of estimates of a different relationship; this allows to better discriminate strong from weaker dependencies between variables. Moreover, we show that
a small estimation variance is likely to be more useful than a smaller bias if this bias is systematic: indeed systematic biases in estimation cancel each other out when ranking dependencies and when a measure is employed to detect the presence of a dependency. As a result of this theory, we define the Randomized Information Coefficient (RIC) and we demonstrate its effectiveness for detecting noisy dependencies and ranking dependencies for the applications of genetic network inference and feature selection for regression. Across these tasks, RIC shows to be competitive to other 15 state-of-the-art measures.

In Chapter 4, we introduce a framework to adjust dependency measure estimates for chance. Dependency measure estimates are estimated on a data set and they can be high because of chance when the sample size is small, or when the dependency is estimated between categorical variables with many categories. These problems do not allow to meaningfully compare dependency estimates across data sets. If the value of a dependency measure must have a clear interpretation, this causes problems in particular when quantifying dependencies. Moreover, these biases cause problems when ranking dependencies. Therefore, firstly we formally define the desiderata of dependency measure estimates such as unbiasedness for quantification and for ranking with regards to the data set used. Secondly, in order to achieve these properties, we propose a general framework to adjust dependency measures. Then, we apply our adjustments to dependency measures between numerical variables such as the Pearson’s correlation to detect linear dependencies and the Maximal Information Coefficient (MIC) [Reshef et al., 2011] and show how to achieve higher interpretability in quantification and higher accuracy in ranking. Ultimately, we demonstrate that our approach is effective to rank variables during the splitting procedure in random forests [Breiman, 2001] where a dependency measure between categorical variables is employed.

In Chapter 5, we discuss adjustments for chance of dependency measures between clusterings of the same data set, i.e., categorical variables. In order to adjust dependency measures, we compute summary statistics such as the expected value and the standard deviation of the distribution of a dependency measure estimator under the null hypothesis of independence between variables. In most of the cases the distribution can be approximated by the computation of the measure under independence via Monte Carlo permutations. Nonetheless, we show that when the dependency measure
is computed between partitions/clusterings of the same data set, we can analytically derive expected value and standard deviation for a class of measures we name $\mathcal{L}_\phi$. This allows us to generalize adjustments for clustering comparison measures. In particular, the Adjusted Rand Index (ARI) [Hubert and Arabie, 1985] based on pair-counting, and the Adjusted Mutual Information (AMI) [Vinh et al., 2009] based on Shannon information theory are very popular in the clustering community. Nonetheless it is an open problem as to what are the best application scenarios for each measure and guidelines in the literature for their usage are sparse, with the result that users often resort to using both. With our theory, we aim to bridge the gap between adjustment of measures based on pair-counting and measures based on information theory. Moreover, we show that expected value and standard deviation can also be computed analytically for a more general class of measures we name $\mathcal{N}_\phi$ under the assumption of large sample size.

Finally, in Chapter 6, we present the conclusions of our research and propose avenues for future work.
In this chapter, we introduce dependency measures between variables and their applications in machine learning. In Section 2.1 we give the definition of variable and discuss the different types of variables which can be found in a data set. Section 2.2 presents dependency measures between categorical variables and Section 2.3 discusses measures between numerical variables. Finally, in Section 2.4 we present a brief overview with examples of some applications of dependency measures in machine learning and data mining.

2.1 DEPENDENCY MEASURES

A data set $S_n = \{(x_1, x_2, \ldots, x_m)k\}_{k=0, \ldots, n-1}$ is a collection of $n$ objects described by the $m$ variables $(X_1, \ldots, X_m)$. Table 2.1 shows how the data set $S_n$ can be represented as a table: objects are also referred to as records, or instances, and each variable $X$ can

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$X_1$</td>
<td>$X_2$</td>
<td>$\ldots$</td>
<td>$X_m$</td>
</tr>
<tr>
<td>$(x_1 \ x_2 \ \ldots \ x_m)_0$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\vdots$</td>
<td>$\vdots$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$(x_1 \ x_2 \ \ldots \ x_m)_k$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\vdots$</td>
<td>$\vdots$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$(x_1 \ x_2 \ \ldots \ x_m)_{n-1}$</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 2.1: Data set $S_n$.

also be referred to as a feature or an attribute:

Definition 2.1.

A variable is a characteristic of an object which can take different values.
### Categorical Variables

**Nominal:** Limited number of values/categories/labels. These values cannot be meaningfully ordered.

**Mathematical operations:** \(=, \neq\)

**Examples:** zip codes, colors.

**Ordinal:** Limited number of values/categories/labels. These values can be ordered.

**Mathematical operations:** \(=, \neq, <, \leq, >, \geq\)

**Examples:** grades, street numbers.

### Numerical Variables

**Interval:** Discrete numerical domain

**Mathematical operations:** \(=, \neq, <, \leq, >, \geq, +, -\)

**Example:** calendar dates.

**Ratio:** Continuous numerical domain.

**Mathematical operations:** \(=, \neq, <, \leq, >, \geq, +, -, \times, :\)

**Example:** temperature, weight.

Table 2.2: Types of variables [Tan et al., 2005].

According to the different values it can take, a variable can be classified to one of the following two types: *categorical* or *numerical*. A variable is categorical if it takes a predefined number of values which assign each object to a particular category. Instead if a variable takes numbers as possible values, it is classified as numerical. These two categories can be further divided according the mathematical operations applicable to the variable. Table 2.2 shows an overview about type of variables.

Dependency measures are fundamental in machine learning and data mining:

**Definition 2.2.**

A dependency measure \(D\) assesses the amount of dependency between variables.

\(D(X, Y)\) can be computed between *two variables* \(X\) and \(Y\): e.g. if \(X = \text{weight}\) and \(Y = \text{height}\) then \(D(\text{weight}, \text{height})\). \(D(X, Y)\) can also compute the amount of dependency between *two sets of variables* \(X = \{X_1, \ldots, X_p\}\) with \(p\) variables and \(Y = \{Y_1, \ldots, Y_q\}\) with \(q\) variables: e.g. if \(X = \{\text{weight}, \text{height}\}\) and \(Y = Y = \text{Body Mass Index}\), then \(D(\{\text{weight}, \text{height}\}, \text{Body Mass Index})\).
The true value $D$ of a dependency measure can be computed when the true distribution of the variables compared is known. Nonetheless, on real tasks the true distribution of variables is unknown and dependency measures are estimated on data sets $S_n$. Let $\hat{D}(S_n|X, Y)$ be the estimator of the dependency measure $D$ between the two sets of variables $X$ and $Y$ on the data set $S_n$. In this thesis, we focus on the properties of the estimators $\hat{D}(S_n|X, Y)$ of different dependency measures $D$. If it is clear from the context, sometimes we omit the variables compared: i.e. we use $\hat{D}$ rather than the lengthy notation $\hat{D}(S_n|X, Y)$. In some sections where it is clear that the discussion is just about estimation of dependency measures, we also omit the symbol $\hat{.}$.

There exist a number of dependency measures in literature. These can be divided into dependency measures between categorical variables and dependency measures between numerical variables. We discuss dependency measures between categorical variables and in particular their estimators in Section 2.2. Dependency measures between continuous variables and their estimators are discussed in Section 2.3.

### 2.2 Dependency Measures between Categorical Variables

Let $X$ and $Y$ be two categorical variables on a data set consisting of $n$ objects. Let $U = \{u_1, \ldots, u_r\}$ and $V = \{v_1, \ldots, v_c\}$ be the the set of labels for $X$ and $Y$ respectively. $U$ consists of $r$ labels and $V$ of $c$ labels. Let $a_i$ be the number of objects in $S_n$ labeled as $u_i$. Similarly, let $b_j$ be the number of objects in $S_n$ labeled as $v_j$. Naturally, $\sum_{i=1}^{r} a_i = \sum_{j=1}^{c} b_j = n$ because both $U$ and $V$ induce a partition/clustering on the data set $S_n$. Indeed, each label $u_i$ and $v_j$ corresponds to a set/cluster of objects in $S_n$. Given that the labels $U$ and $V$ are applied to the same data set, it is possible to count the number of objects that are jointly labeled by $U$ and $V$. Let $n_{ij}$ denote the number of objects that are labeled as $u_i$ and as $v_j$. The co-occurrences of the different labels according to $U$ and to $V$ can be represented in matrix form by a $r \times c$ contingency table $M$ such as the one in Table 2.3. We refer to $a_i = \sum_j n_{ij}$ as the row marginals and $b_j = \sum_i n_{ij}$ as the column marginals. Figure 2.1 shows a toy example about a data set with $n = 8$ people and two variables.
Dependency measures between two categorical variables $X$ and $Y$ can be naturally estimated on the associated contingency table $\mathcal{M}$. Here we discuss a number of possible measures of dependence between categorical variables on a data set $S_n$.

2.2.1 Mutual Information

Information theory provides a well-established measure of dependence between two categorical variables, the mutual information between $X$ and $Y$ [Cover and Thomas, 2012]. Mutual information is based on the concept of Shannon’s entropy. The entropy for a categorical variable is defined as the expected value of its information content. Given
2.2 Dependency Measures between Categorical Variables

The probability distribution for $X$ and $Y$ over their labels $U$ and $V$, we can therefore define their entropy as follows

$$ H(X) \triangleq - \sum_{i=1}^{r} p_X(u_i) \log p_X(u_i) \quad (2.1) $$

$$ H(Y) \triangleq - \sum_{j=1}^{c} p_Y(y) \log p_Y(y) \quad (2.2) $$

Entropy is non-negative and it is equal to 0 when a categorical variable takes only one single label. On the other hand, entropy is maximized when the probability distribution over labels is uniform. More specifically, we have $0 \leq H(X) \leq \log r$ and $0 \leq H(Y) \leq \log c$. The joint entropy between the two variables $X$ and $Y$ is defined as follows:

$$ H(X,Y) \triangleq - \sum_{i=1}^{r} \sum_{j=1}^{c} p_{X,Y}(u_i, v_j) \log p_{X,Y}(u_i, v_j) \quad (2.3) $$

Moreover using conditional probabilities, it is possible to define the conditional entropy $H(X|Y)$ and $H(Y|X)$. The latter is defined as:

$$ H(Y|X) \triangleq \sum_{i=1}^{r} p_X(u_i) H(Y|u_i) = - \sum_{i=1}^{r} p_X(u_i) \sum_{j=1}^{c} p_{Y|X}(v_j|u_i) \log p_{Y|X}(v_j|u_i) \quad (2.4) $$

Mutual information $I(X,Y)$ quantifies the value of information shared between the two variables $X$ and $Y$ and can be defined using the entropy definitions:

$$ I(X,Y) \triangleq H(Y) - H(Y|X) = H(X) - H(X|Y) \quad (2.5) $$

$$ = H(X) + H(Y) - H(X,Y) = \sum_{i=1}^{r} \sum_{j=1}^{c} p_{X,Y}(u_i, v_j) \log \frac{p_{X,Y}(u_i, v_j)}{p_X(u_i)p_Y(v_j)} \quad (2.6) $$

$I(X,Y)$ is a non-negative measure of dependency.

Using the maximum likelihood estimation method, we estimate the empirical joint probability distribution of $X$ and $Y$ on the data set $S_n$. This can also be computed using the cells of the associated contingency table: $p_X(u_i) = \frac{a_i}{n}$, $p_Y(v_j) = \frac{b_j}{n}$, and

---

1 All logarithms are considered in base 2, $\log \equiv \log_2$
\( p_{X,Y}(u_i,v_j) = \frac{n_{ij}}{n} \). Therefore, we denote as \( \text{MI}(X,Y) \) the mutual information estimated between two categorical variables \( X \) and \( Y \) on the contingency table \( \mathcal{M} \):

\[
\text{MI}(X,Y) = \text{MI}(\mathcal{M}) = \sum_{i=1}^{r} \sum_{j=1}^{c} \frac{n_{ij}n}{a_i b_j} \log \frac{n_{ij}n}{a_i b_j} \tag{2.7}
\]

Entropy and conditional entropy can be estimated on the contingency table as well:

\[
H(X) = \sum_{i=1}^{r} \frac{a_j}{n} \log \frac{a_i}{n} \tag{2.8}
\]

\[
H(Y) = \sum_{j=1}^{c} \frac{b_i}{n} \log \frac{b_j}{n} \tag{2.9}
\]

\[
H(Y|X) = \sum_{i=1}^{r} \frac{a_i}{n} H(Y|u_i) = \sum_{i=1}^{r} \frac{a_i}{n} \sum_{j=1}^{c} \frac{n_{ij}}{a_i} \log \frac{n_{ij}}{a_i} \tag{2.10}
\]

The mutual information has many possible upper bounds that might be used to obtain the Normalized Mutual Information (NMI) which ranges in \([0,1]\):

\[
\text{NMI}(X,Y) = \frac{\text{MI}(X,Y)}{\max \text{MI}(X,Y)} \tag{2.11}
\]

where \( \max \text{MI}(X,Y) \) can be chosen among one of the following upper bounds to \( \text{MI} \):

\[
\text{MI}(X,Y) \leq \min \{H(X),H(Y)\} \leq \sqrt{H(X) \cdot H(Y)} \leq \ldots
\]

\[
\ldots \leq \frac{1}{2} (H(X) + H(Y)) \leq \max \{H(X),H(Y)\} \leq H(X,Y)
\]

Depending on the chosen upper bound, it is also possible to obtain information theoretic distance measures with metric properties [Vinh et al., 2010]. A distance measure with metric properties is indeed useful for designing efficient algorithms that exploit the nice geometric properties of metric spaces [Meilă, 2012]. An example of a true metric is the variation of information (VI), defined in [Meilă, 2007]:

\[
\text{VI}(X,Y) \triangleq 2H(X,Y) - H(X) - H(Y) = H(X|Y) + H(Y|X) \tag{2.12}
\]

\[
= H(X) + H(Y) - 2\text{MI}(X,Y)
\]
2.2 Dependency Measures between Categorical Variables

2.2.2 Pair-counting Measures

Pair-counting measures between categorical variables are very common in the clustering community [Albatineh et al., 2006; Meilă, 2007]. Indeed, each variable $X$ and $Y$ can define a clustering on the data set $S_n$ according to the labels $U$ and $V$ respectively. Pair-counting measures are computed on a contingency table $\mathcal{M}$ using the following quantities which counts pairs of objects in $S_n$:

- $N_{11}$, the pairs in the same cluster in both $U$ and $V$;
- $N_{00}$, the pairs not in the same cluster in $U$ and not in the same cluster in $V$;
- $N_{10}$, the pairs in the same cluster in $U$ and not in the same cluster in $V$;
- $N_{01}$, the pairs not in the same cluster in $U$ and in the same cluster in $V$.

All these quantities can be computed using the contingency table $\mathcal{M}$:

\[
N_{11} = \frac{1}{2} \sum_{i=1}^{r} \sum_{j=1}^{c} n_{ij} (n_{ij} - 1)
\]
\[
N_{00} = \frac{1}{2} \left( n^2 + \sum_{i=1}^{r} \sum_{j=1}^{c} n_{ij}^2 - \left( \sum_{i=1}^{r} a_i^2 + \sum_{j=1}^{c} b_j^2 \right) \right)
\]
\[
N_{10} = \frac{1}{2} \left( \sum_{j=1}^{c} b_j^2 - \sum_{i=1}^{r} \sum_{j=1}^{c} n_{ij}^2 \right)
\]
\[
N_{01} = \frac{1}{2} \left( \sum_{i=1}^{r} a_i^2 - \sum_{i=1}^{r} \sum_{j=1}^{c} n_{ij}^2 \right)
\]

It holds true that $N_{11} + N_{00} + N_{10} + N_{01} = \binom{n}{2}$. Using these quantities, it is possible to compute similarity measures, e.g. the Rand Index (RI) [Rand, 1971], or distance measures, e.g. the Mirkin index $\text{MK}(X, Y) \triangleq \sum_i a_i^2 + \sum_j b_j^2 - 2 \sum_{i,j} n_{ij}^2$, between partitions [Meilă, 2007]:

\[
\text{RI}(X, Y) \triangleq \frac{(N_{11} + N_{00})}{\binom{n}{2}}
\]

(2.13)
and
\[ MK(X,Y) = 2(N_{10} + N_{01}) = n(n-1)(1 - RI(X,Y)) \]  \hfill (2.14)

Another famous dependency measure which belongs in this category is the Jaccard similarity coefficient \( J \):
\[ J(X,Y) = \frac{N_{11}}{N_{11} + N_{10} + N_{01}} \]  \hfill (2.15)

### 2.2.3 Accuracy and Agreement

If \( X \) is the class predicted by a classification algorithm and \( Y \) is the actual class, the classification accuracy can be seen as a measure of dependence between the variables \( X \) and \( Y \) [Witten et al., 2011]. In this case, the two categorical variables share the same domain of labels: i.e. \( U = V \). Similarly, the amount of agreement between two annotators \( X \) and \( Y \) on a sample \( S_n \) of \( n \) items can be seen as a measure of dependency. Classification accuracy and agreement are computed on a contingency table \( M \) with the following formula:
\[ A(X,Y) = \sum_{i=1}^{r} \sum_{j=1}^{c} \frac{n_{ij}}{n} \]  \hfill (2.16)

### 2.2.4 Splitting Criteria in Classification Trees

In some decision tree induction algorithms such as C4.5 [Quinlan, 1993], a categorical variable \( X \) with \( r \) categories induces a split on internal nodes in \( r \) subsets. Splitting criteria evaluate the performance of a split and they can be considered as dependency measures between \( X \) and the class \( Y \). Most of the splitting criteria are based on impurity measures \( \text{Imp}(Y) \). These measures capture the amount of impurity for a given set of records \( S_n \) regarding the class attribute \( Y \). Their value is 0 if all records are of the same class and they attain their maximum value if the class distribution is uniform.
The average decrease of impurity of the class $Y$ due to a split on a categorical attribute $X$ is often used as a splitting criterion and referred to as gain $\Delta$:

$$\Delta(X, Y) = \text{Imp}(Y) - \sum_{i=1}^{r} \frac{a_i}{n} \text{Imp}(Y|u_i)$$

Various impurity measures have been used in literature. The most common impurity measures are:

(a) **Entropy**:

$$H(Y) = -\sum_{j=1}^{c} \frac{b_j}{n} \log \frac{b_j}{n}$$

(b) **Gini index**:

$$Gini(Y) = 1 - \sum_{j=1}^{c} \left(\frac{b_j}{n}\right)^2$$

If entropy is used as impurity measure the gain is usually called Information Gain (IG), and it is easy to show that it can be rewritten as the Mutual Information (MI) in Eq. (2.7):

$$\text{IG}(X, Y) = H(Y) - \sum_{i=1}^{r} \frac{a_i}{n} H(Y|u_i) = -\sum_{j=1}^{c} \frac{b_j}{n} \log \frac{b_j}{n} + \sum_{i=1}^{r} \frac{a_i}{n} \sum_{j=1}^{c} \frac{n_{ij}}{a_i} \log \frac{n_{ij}}{a_i}$$

$$= -\sum_{j=1}^{c} \sum_{i=1}^{r} \frac{n_{ij}}{n} \log \frac{b_j}{n} + \sum_{i=1}^{r} \sum_{j=1}^{c} \frac{n_{ij}}{n} \log \frac{n_{ij}}{a_i}$$

$$= \sum_{i=1}^{r} \sum_{j=1}^{c} \frac{n_{ij}}{n} \log \frac{n_{ij}n}{a_i b_j} = \text{MI}(X, Y)$$

The Gini Gain (GG) is instead defined as follows:

$$\text{GG}(X, Y) = Gini(Y) - \sum_{i=1}^{r} \frac{a_i}{n} Gini(Y|u_i) = 1 - \sum_{j=1}^{c} \left(\frac{b_j}{n}\right)^2 - \sum_{i=1}^{r} \frac{a_i}{n} \left(1 - \sum_{j=1}^{c} \left(\frac{n_{ij}}{a_i}\right)^2\right)$$

These two criteria are respectively implemented in C4.5 [Quinlan, 1993] and in CART [Breiman et al., 1984]. When splits are induced by categorical variables with 2 categories, IG has been found to be more biased than GG to variables where the 2 categories are
distribute uniformly [Breiman, 1996]. Recently the improved Grassberger’s estimator of entropy $H_G$ for discrete variables has been proposed and plugged in IG to improve classification accuracy of decision trees [Nowozin, 2012]. The Grassberger’s estimator of entropy $H_G$ is defined as follows:

$$H_G(Y) = \log n - \sum_{j=1}^{c} \frac{b_j}{n} G(b_j) \quad \text{where} \quad G(b_j) = \psi(b_j) + \frac{1}{2} (-1)^b \left( \psi \left( \frac{b_j + 1}{2} \right) - \psi \left( \frac{b_j}{2} \right) \right)$$

where $\psi$ is the digamma function. The $H_G$ estimator is claimed to be more effective when the variable $Y$ has many categories: i.e. $c \gg 1$. Another splitting criterion based on the entropy estimator is the Gain Ratio (GR) [Quinlan, 1993]. This was specifically designed to penalize variables $X$ with many categories and can be seen as a normalized mutual information:

$$\text{GR}(X, Y) = \frac{\text{IG}(X, Y)}{H(X)} = \frac{\text{MI}(X, Y)}{H(X)} = \frac{\text{MI}(X, Y)}{-\sum_{i=1}^{r} a_i \log \frac{a_i}{n}} \quad (2.20)$$

Other splitting criteria are based on statistics, such as the chi-square statistic [Kononenko, 1995]

$$\chi^2(X, Y) = \sum_{i=1}^{r} \sum_{j=1}^{c} \left( \frac{e_{ij} - n_{ij}}{e_{ij}} \right)^2 \quad \text{with} \quad e_{ij} = \frac{a_i \cdot b_j}{n}$$

and the $G$-statistic:

$$G(X, Y) = 2n \cdot \log_e 2 \cdot \text{MI}(X, Y)$$

Some other splitting criteria are based on the Minimum Description Length principle (MDL). The MDL principle roughly states that the problem of selecting the best split might be solved selecting that one that leads to the biggest compression [Kononenko, 1995]. Even if its formula is complicated, this is still a dependency measure between $X$ and $Y$ and can be computed on the associated contingency table $\mathcal{M}$:

$$\text{MDL}(X, Y) = \frac{1}{n} \left( \log \left( \binom{N}{b_1, \ldots, b_c} \right) - \sum_{i=1}^{r} \log \left( \binom{a_i}{n_{i1}, \ldots, n_i} \right) + \right)$$
\[
\log \left( \binom{n + c + 1}{c - 1} \right) - \sum_{i=1}^{r} \log \left( \binom{a_i + c - 1}{c - 1} \right)
\]

2.3 DEPENDENCY MEASURES BETWEEN NUMERICAL VARIABLES

Let \( X \) and \( Y \) be sets of \( p \) and \( q \) numerical variables respectively. There exist a number of ways to assess the amount of dependency between them. Here we present a review of possible ways to estimate the dependency between \( X \) and \( Y \) on a data set \( S_n = \{(x, y)_{k}\}_{k=0}^{n-1} = \{(x_1 \ldots x_p, y_1 \ldots y_q)_{k}\}_{k=0}^{n-1} \) of \( n \) records.

2.3.1 Mutual Information

We discussed the use of mutual information as a dependency measure between two categorical variables in Section 2.2.1. Mutual information is also a powerful tool to assess the dependency between numerical variables. Moreover, its definition can be extended to compare two sets of variables \( X \) and \( Y \). When the variables compared are numerical, mutual information is defined making use of the differential entropy which is defined as follows [Cover and Thomas, 2012]:

\[
h(X) \triangleq -\int_{\text{dom}(X)} f_X(x) \log f_X(x) \, dx
\]

where \( f_X(x) \) and \( \text{dom}(X) \) are the density function and the domain of \( X \) respectively. The mutual information between the two sets of variables \( X \) and \( Y \) is defined as:

\[
I(X, Y) \triangleq h(X) + h(Y) - h(X, Y) = \int_{\text{dom}(X)} \int_{\text{dom}(Y)} f_{X,Y}(x, y) \log \frac{f_{X,Y}(x, y)}{f_X(x) f_Y(y)} \, dx \, dy
\]

Unlike the case for categorical variables, there are many estimators of mutual information for numerical variables on a data set \( S_n \). The standard approach however consists of discretizing the space of possible values that \( X \) and \( Y \) can take using a grid \( G \), and then estimating the probability mass function using the frequency of occurrence. A grid
G for the sets of variables \( X \) of \( Y \) is the Cartesian product of the two partitions \( G_X \) and \( G_Y \). \( G_X \) is a partition of the domain of the variables in \( X \) in \( r \) disjoint sets \( u_i \). \( G_Y \) is a partition of the domain of the variables in \( Y \) in \( c \) disjoint sets \( v_j \). When a grid \( G \) is applied to a data set, we denote with \( (X,Y)|G \) the contingency table between \( X \) and \( Y \). An example of contingency table can be found in Table 2.3. Using the associated contingency table, the mutual information can be estimated with the estimator in Eq. (2.7). Nonetheless, there are many possible approaches to discretization of random variables. For example, a single random variable can be easily discretized according to equal-width or equal-frequency binning [Steuer et al., 2002]. We refer to \( I_{ew} \) and \( I_{ef} \) as the estimators of mutual information using equal width binning and equal frequency binning respectively. Figure 2.2 shows an example of equal width grid for the estimation of mutual information with contingency tables. Discretization of single random variables can also be performed according to more complex principles such as the minimum description length [Fayyad and Irani, 1993]. Mutual information estimators based on discretization in equal width intervals have been discussed in Steuer et al. [2002]. Particularly crucial is the choice of the number of bins used to discretize \( X \) and \( Y \): too big values lead to overestimation of mutual information due to a finite-sample effect. To mitigate this problem, adaptive partitioning of the discretization grid on the joint distribution \( (X,Y) \) has been proposed [Fraser and Swinney, 1986] and optimized for speed [Cellucci et al., 2005]. We note that there is no universally accepted optimal discretization technique. Moreover, even though for sets of variables \( X \) and \( Y \) few sensible discretization have been proposed [Dougherty et al., 1995; Garcia et al., 2013], to our knowledge, there is
no extensive survey about the estimation of mutual information with multiple variable discretization approaches.

Other competitive mutual information estimators used in practice are the kernel density estimator [Moon et al., 1995] and the Kraskov’s $k$ nearest neighbors estimator [Kraskov et al., 2004]. These estimators can be used to compare sets of variables $X$ and $Y$. An extensive comparison of these estimators can be found in Khan et al. [2007]. The kernel density estimator $I_{\text{KDE}}$ is computed using kernel functions $k(\cdot)$ to estimate the probability distributions $f_X(x)$, $f_Y(y)$, and $f_{X,Y}(x,y)$:

$$\hat{f}_X(x) = \frac{1}{nh} \sum_{i=0}^{n-1} k\left(\frac{x - x_i}{h}\right)$$

and

$$\hat{f}_{X,Y}(x,y) = \frac{1}{nh} \sum_{i=0}^{n-1} k\left(\frac{x - x_i}{h}, \frac{y - y_i}{h}\right)$$

where $h$ is a parameter which tunes the kernel width. The KDE estimator of the mutual information as theoretically defined in Eq. (2.21) is obtained by:

$$I_{\text{KDE}}(X,Y) \triangleq \frac{1}{n} \sum_{k=0}^{n-1} \log \frac{\hat{f}_{X,Y}(x_k,y_k)}{\hat{f}_X(x_k)\hat{f}_Y(y_k)} \quad (2.22)$$

The Kraskov’s estimator is instead defined using the $k$ nearest neighbors for each point of the data set. Let $\epsilon_i$ be the euclidean distance of a point to its $k$-th nearest neighbor, and let $n_{x_i}$ be the number of points at a distance less than or equal to $\epsilon_i/2$, then:

$$I_{\text{KNN}}(X,Y) \triangleq \psi(n) + \psi(k) - \frac{1}{k} - \frac{1}{n} \sum_{i=0}^{n-1} \psi(n_{x_i}) + \psi(n_{x_i}) \quad (2.23)$$

where $\psi$ is the digamma function.
The Kraskov’s estimator for mutual information is based on the derivation of the $k$NN estimator of Shannon’s entropy. The entropy for a set of variables $\mathbf{X} = (X_1, \ldots, X_p)$ has its own estimator on a data set $\mathcal{S}_n$:

$$H_{k\text{NN}}(\mathbf{X}) \triangleq \psi(n) - \psi(k) + \log(c_p) + \frac{p}{n} \sum_{i=0}^{n-1} \log(\epsilon_i)$$  \hspace{1cm} (2.24)

where $\psi$ is the digamma function, $c_p$ is the volume of a $p$-dimensional unit ball and $\epsilon_i$ is the euclidean distance from $x_i$ to its $k$-th nearest neighbor. Recently in Faivishevsky and Goldberger [2009] it was shown that if we take the average of the $k$NN estimator of entropy defined in Eq. (2.24) across all possible values $k$, we obtain the following smooth estimator of Shannon’s entropy:

$$H_{\text{mean}}(\mathbf{X}) = \text{const} + \frac{p}{n(n-1)} \sum_{i \neq j} \log ||x_i - x_j||$$  \hspace{1cm} (2.25)

where the constant $\text{const}$ depends on the sample size and the data dimensionality, and $|| \cdot ||$ is the Euclidean distance. This idea yields yet another interesting estimator of mutual information – the mean estimator of mutual information:

$$I_{\text{mean}}(\mathbf{X}, \mathbf{Y}) \triangleq H_{\text{mean}}(\mathbf{X}) + H_{\text{mean}}(\mathbf{Y}) - H_{\text{mean}}(\mathbf{X}, \mathbf{Y})$$  \hspace{1cm} (2.26)

### 2.3.2 Correlation Based Measures

When the user is only interested in linear dependencies between two variables $X$ and $Y$ on a data set $\mathcal{S}_n = \{(x,y)_k\}_{k=0\ldots n-1}$, the sample Pearson’s correlation coefficient $r$ is a powerful dependency measure:

$$r(X,Y) \triangleq \frac{\text{Cov}(X,Y)}{\sqrt{\text{Var}(X) \cdot \text{Var}(Y)}} = \frac{\sum_{k=0}^{n-1} (x_k - \bar{x})(y_k - \bar{y})}{\sqrt{\sum_{k=0}^{n-1} (x_k - \bar{x})^2 \sum_{k=0}^{n-1} (y_k - \bar{y})^2}}$$  \hspace{1cm} (2.27)

with $\bar{x} = \frac{1}{n} \sum_{k=0}^{n-1} x_k$ and $\bar{y} = \frac{1}{n} \sum_{k=0}^{n-1} y_k$. A value of 1 for $r$ corresponds to perfect positive linear correlation between $X$ and $Y$. Instead, the value -1 for $r$ corresponds
to perfect negative correlation. Therefore, the amount of linear dependency between \( X \) and \( Y \) can me measured with either \( |r| \) or \( r^2 \).

Nonetheless, the Pearson’s correlation coefficient is limited to the identification of linear dependency between two single variables \( X \) and \( Y \). In order to handle non-linear dependencies between two sets of variables, one can use the distance correlation (dCorr) proposed in Székely et al. [2009]. The dependency measure \( \text{dCorr}(X, Y) \) between the two sets of variables \( X \) and \( Y \) can be estimated on a data set \( S_n = \{ (x, y)_k \}_{k=0}^{n-1} \) making use of the Euclidean distance \( || \cdot || \) between pairs of points. Let \( a_{jk} \) and \( b_{jk} \) be the pairwise Euclidean distance between all points according to the variables in \( X \) and \( Y \) respectively:

\[
a_{jk} = ||x_j - x_k|| \quad \text{with} \quad j, k = 0, \ldots, n - 1
\]
\[
b_{jk} = ||y_j - y_k|| \quad \text{with} \quad j, k = 0, \ldots, n - 1
\]

Let \( A_{jk} \) and \( B_{jk} \) be the doubly centered distances defined as follows:

\[
A_{jk} = a_{jk} - \bar{a}_j \cdot - \bar{a}_k \cdot + \bar{a}_{\cdot \cdot}
\]
\[
B_{jk} = b_{jk} - \bar{b}_j \cdot - \bar{b}_k \cdot + \bar{b}_{\cdot \cdot}
\]

where \( \bar{a}_j = \frac{1}{n} \sum_{k=0}^{n-1} a_{jk} \), \( \bar{a}_k = \frac{1}{n} \sum_{j=0}^{n-1} a_{jk} \), and \( \bar{a}_{\cdot \cdot} = \frac{1}{n^2} \sum_{j=0}^{n-1} \sum_{k=0}^{n-1} a_{jk} \). The sample distance covariance (dCov) and the sample distance variance (dVar) on the data set \( S_n \) are defined as:

\[
\text{dCov}(X, Y) \triangleq \sqrt{\frac{1}{n^2} \sum_{j=0}^{n-1} \sum_{k=0}^{n-1} A_{jk} B_{jk}}
\]
\[
\text{dVar}(X) \triangleq \text{dCov}(X, X) = \sqrt{\frac{1}{n^2} \sum_{j=0}^{n-1} \sum_{k=0}^{n-1} A_{jk}^2}
\]
Finally, the distance correlation (dCorr) is defined similarly as Pearson’s correlation but making use of dCov and dVar:

\[
\text{dCorr}(X, Y) \triangleq \frac{\text{dCov}(X, Y)}{\sqrt{\text{dVar}(X) \cdot \text{dVar}(Y)}}
\]  

(2.28)

More recently, random projections have been employed to achieve speed improvements [Lopez-Paz et al., 2013], yielding the Randomized Dependency Coefficient (RDC). RDC might be seen as a randomized way to identify the maximal correlation between sets of variables. Its computation on a data set \( S_n \) is sketched as follows:

(a) **Copula transformation**: Each set of variables \( X \) and \( Y \) is transformed using the empirical copulas \( P(X) \) and \( P(Y) \) to obtain uniform marginal distributions respectively. This allows RDC to be invariant to the marginal distribution of \( X \) and \( Y \);

(b) **Random projections**: Each \( P(X) \) and \( P(Y) \) is independently projected on a space of \( k \) dimensions making using of non-linear functions \( \Phi \) parametrized with \( s \):

\[
\Phi(P(X); k, s), \quad \Phi(P(Y); k, s);
\]

(c) **Computation of Canonical Correlation**: RDC is obtained using Canonical Correlation Analysis (CCA) to identify the linear combination of the random projections obtained above that has maximal correlation. Let \( \alpha \) and \( \beta \) be basis vectors, then the Randomized Dependency Coefficient (RDC) is defined as follows:

\[
\text{RDC}(X, Y) \triangleq \max_{\alpha, \beta} \left( \alpha^T \Phi(P(X)), \beta^T \Phi(P(Y)) \right)
\]  

(2.29)

RDC can also be seen as an extension of the Alternative Conditional Expectation (ACE). Nonetheless, ACE identifies the best non-linear transformation of the input variables without making use of randomization. Moreover, ACE is only applicable to single variables \( X \) and \( Y \). More details about ACE can be found in Breiman and Friedman [1985].
2.3.3 Kernel Based Measures

The dependency between two sets of variables can also be measured employing the joint distribution of the studied variables under kernel embeddings. The Hilbert-Schmidt Independence Criterion (HSIC) [Gretton et al., 2005] is an example of such measures. Given two unique positive definite kernels \( k(\cdot) \) and \( l(\cdot) \) it is possible to estimate \( \text{HSIC}(X, Y) \) on a data set \( S_n \) as follows:

\[
\text{HSIC}(X, Y) \doteq \frac{1}{(n-1)^2} \text{tr}(KHLH) \tag{2.30}
\]

where \( K_{ij} = k(x_i, x_j) \), \( L_{ij} = l(y_i, y_j) \), and \( H_{ij} = \delta_{ij} - \frac{1}{n^2} \) with \( \delta \) Dirac function and \( i, j = 0, \ldots, n - 1 \).

A detailed analysis and discussion about HSIC is available in Gretton et al. [2012]. Moreover, HSIC was also extended recently to be invariant to the marginal distribution of \( X \) and \( Y \) employing copulas [Póczos et al., 2012].

2.3.4 New Measures Based on Information Theory and Discretization

More recently, new measures based on information theory and variable discretization, such as the Maximal Information Coefficient (MIC) presented in Reshef et al. [2011] and the Mutual Information Dimension (MID) [Sugiyama and Borgwardt, 2013], have been proposed. MIC was proposed as a dependency measure for functional relationships between \( X \) and \( Y \), performing discretization via grids over the joint distribution \( (X, Y) \). MIC satisfies a useful property called equitability, which allows it to act as a proxy for the coefficient of determination \( R^2 \) of a functional relationship [Reshef et al., 2015b]. Given a data sample \( S_n \) from the numerical variables \( X \) and \( Y \), MIC is the maximal normalized MI computed across all the possible \( r \times c \) grids to estimate the bivariate
frequency distribution of $X$ and $Y$. Each $r \times c$ discretizes the scatter plot of $X$ and $Y$ in $r \cdot c$ bins to compute their frequency distribution in a contingency table:

$$\text{MIC}(X, Y) \triangleq \max_{r \times c \text{ grids } G \text{ with } r \cdot c \leq n^\alpha} \frac{\text{MI}((X, Y)|G)}{\log_2 \min \{r, c\}}$$

where $\alpha$ is a parameter often set to 0.6 [Reshef et al., 2011]. Figure 2.3 shows the average value of MIC for different functional relationships and different percentages of white noise on data sets $S_n$ of $n = 60$ points. At a given percentage $p$ of white noise, $p \times n$ points are randomly scattered uniformly in the domain of $X$ and $Y$. MIC can be used as a proxy for the amount of noise: it returns the value 1 for functional relationships.

MID is instead based on information dimension theory and discretization. The information dimension of a numerical variable $X$ is defined as follows:

$$\dim X = \lim_{k \to +\infty} \frac{H(X|G(k))}{k}$$

where $G(k)$ is a grid which discretizes $X$ in $k$ equal-width bins. Similarly, the information dimension of the variables $X$ and $Y$ is defined as:

$$\dim XY = \lim_{k \to +\infty} \frac{H((X, Y)|G(k))}{k}$$

where $G(k)$ discretizes both $X$ and $Y$ with equal-width binning. MID between two variables $X$ and $Y$ is defined as follows:

$$\text{MID}(X, Y) \triangleq \dim X + \dim Y - \dim XY$$

Prominent features of MID include its efficiency and the ability to characterize multi-functional relationships with a score of 1.

Reshef et al. [2015b] also proposed two new statistics based on grids in this recent preprint: MIC$_e$ which is an improved estimator of the population value of MIC; and the Total Information Coefficient (TIC$_e$) to achieve high power when testing for independence between variables. In a thorough study, Reshef et al. [2015a] compared
many different dependency measures between variables and demonstrated that $\text{MIC}_e$ and $\text{TIC}_e$ are very competitive when targeting high equitability and high power respectively. $\text{MIC}_e$ optimizes the normalized mutual information over all grid cardinalities and grid cut-offs. $\text{TIC}_e$ still optimizes the possible cut-offs for a grid, but returns the average value over grid cardinalities instead. Independently, another statistic based on grids and normalized mutual information has been suggested in the attempt to maximize power: the Generalized Mean Information Coefficient (GMIC) [Luedtke and Tran, 2013].
2.4 APPLICATIONS OF DEPENDENCY MEASURES

The dependency measures discussed above are fundamental in machine learning and data mining. Hence there exist a large number of applications. In this section we describe in detail some common applications for dependency measures and provide examples to gain intuition.

2.4.1 Feature Selection and Decision Tree Induction

Dependency measures are often used for feature selection in order to make classification techniques more effective. Given a data set \( S_n = \{(x, y)_k\}_{k=0 \ldots n-1} \) of \( n \) records from \( X \) and \( Y \), classification techniques aim at labeling records of unknown class \( Y \) making use of their features \( X \). They basically map the features \( X \) into the class attribute \( Y \) (target variable), the object of the classification. The relationship between \( X \) and \( Y \) is learned using \( S_n \). Dependency measures between the variables \( X \) and the target variable \( Y \) are often used for feature selection in classification and regression tasks [Guyon and Elisseef, 2003].

Dependency measures are also fundamental in decision trees which are very interpretable classification techniques. Indeed, each internal node in a decision tree is associated to a specific test on a variable \( X \in X \) and each edge to child nodes is labeled with its different test results. Each leaf is labeled with a class from \( Y \). In Figure 2.4 shows a toy example of a data set where the target variable \( Y = \text{Risk} \). This toy example shows a data set that can be used to predict risk levels for a car insurance company. The figure also shows an example of decision tree induced on this data set.

A decision tree induction algorithm proceeds by hierarchically partitioning a training data set according to the variables \( X \) in order to obtain subsets which are as pure as possible to the given target class \( Y \). The best partitioning is defined as a split and it is selected according to a dependency measure computed on a contingency table. Given a categorical variable \( X \) with \( r \) values, some decision tree induction algorithms generate split of cardinality \( r \). C4.5 [Quinlan, 1993] is an example of such algorithms. Other
2.4 Applications of Dependency Measures

<table>
<thead>
<tr>
<th>ID</th>
<th>Age</th>
<th>Type of car</th>
<th>Risk</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>23</td>
<td>Sport car</td>
<td>A</td>
</tr>
<tr>
<td>2</td>
<td>18</td>
<td>Sport car</td>
<td>A</td>
</tr>
<tr>
<td>3</td>
<td>43</td>
<td>Sport car</td>
<td>A</td>
</tr>
<tr>
<td>4</td>
<td>68</td>
<td>Sedan</td>
<td>B</td>
</tr>
<tr>
<td>5</td>
<td>32</td>
<td>Pickup</td>
<td>B</td>
</tr>
<tr>
<td>6</td>
<td>20</td>
<td>Sedan</td>
<td>A</td>
</tr>
</tbody>
</table>

Figure 2.4: Example of decision tree induced on a data set $S_n$ with $n = 6$.

algorithms such as CART [Breiman et al., 1984] generate all the possible $2^r - 2$ binary splits for a variable with $r$ different categories. Instead, a numerical variable $X$ always induces binary splits based on cut-offs $x$ such that records with $X \leq x$ go on the left child node and records with $X > x$ go on the right one, as shown in Figure 2.5:

(a) Split induced by a categorical variable.  
(b) Split induced by a numerical variable.

Figure 2.5: Splits induced by categorical and numerical variables $X$ in C4.5 [Quinlan, 1993].

Each of the splits described above can be computed on a contingency table $M$ using a dependency measure $D$ for categorical variables as described in Section 2.2. C4.5 induces a unique split according a categorical variable but still identifies the best split for a numerical variable by trying all possible cut-offs. The approach used by C4.5 is sketched in Algorithm 2.1. FINDBESTSPLIT$(X, Y, D)$ identifies the best split given a set of variables $X$, the target variable $Y$, and the dependency measure $D$:

2.4.2 Clustering Comparisons

Dependency measures are extensively used in the clustering community. Indeed, two clusterings on the same data set $S_n$ can be seen as two categorical variables $X$ and $Y$. All the measures discussed in Section 2.2 can be used for comparing clusterings. In
Algorithm 2.1  Algorithm to find the best split according to a given dependency measure $D$.

\begin{algorithm}
\textproc{FindBestSplit}(\textbf{X}, \textbf{Y}, D) \\
\hspace{1em} \textit{CandidateTables} ← \emptyset \\
\hspace{1em} \textbf{for each} \, X ∈ \textbf{X} \, \textbf{do} \\
\hspace{2em} \textbf{if} \, X \, \textit{is categorical} \, \textbf{then} \\
\hspace{3em} \textit{CandidateTables} ← \textit{CandidateTables} \cup M \\
\hspace{2em} \textbf{if} \, X \, \textit{is numerical} \, \textbf{then} \\
\hspace{3em} \textbf{for each} \, M \, \textbf{with 2 rows associated to} \, X ≤ x \, \textbf{and} \, X > x \, \textbf{do} \\
\hspace{4em} \textbf{if} \, \textit{best} < D(M) \, \textbf{then} \\
\hspace{5em} M^* ← M \\
\hspace{3em} \textit{CandidateTables} ← \textit{CandidateTables} \cup M^* \\
\hspace{1em} \textbf{for each} \, M ∈ \textit{CandidateTables} \, \textbf{do} \\
\hspace{2em} \textbf{if} \, \textit{best} < D(M) \, \textbf{then} \\
\hspace{3em} M^* ← M \\
\hspace{1em} \textbf{return} \, M^*
\end{algorithm}

Figure 2.6: Example of clustering comparisons on a data set of $n = 15$ points. The contingency table provides graphical intuition about the overlapping between the two clusterings $U$ and $V$.

Figure 2.6 we show a data set $S_n$ with $n = 15$ data points in a two dimensional space. The clustering $U$ labels each data point with colors: red and blue. The clustering $V$ labels each data point with geometric figures: stars and circles. The overlapping between the two clusterings is shown by the associated contingency table $M$. Therefore, the amount of dependency between $U$ and $V$ can be assessed with a dependency measure $D$ between categorical variables.

Dependency measures are extensively used in the clustering community: for external clustering validation [Strehl and Ghosh, 2003], including soft clustering validation [Lei et al., 2014a]; for generating alternative or multi-view clusterings [Müller et al., 2013;
Dang and Bailey, 2015]; to explore the clustering space using results from the Meta-Clustering algorithm [Caruana et al., 2006; Lei et al., 2014b].

2.4.3 Exploratory Analysis

Dependency measures between variables are also used for exploratory analysis in machine learning and data mining. There exist applications on inference of biological networks [Reshef et al., 2011; Villaverde et al., 2013] and analysis of neural time-series data [Cohen, 2014]. Nonetheless, dependency measure can really be used to explore the dependency between any phenomena of interest for which data can be collected. For example, the correlation between Google search keywords can be identified using a dependency measure. In these scenarios, data on two phenomena of interested is collected in a data set $S_n$ and a dependency measure is employed to infer the dependency between them. For example, the amount of dependency between two genes $X$ and $Y$ can be inferred computing a dependency measure between the time series of value sampled at the same time from $X$ and from $Y$. Figure 2.7 shows a toy example about the data set $S_n$ related to the two time-series $X$ and $Y$. Being numerical variables, a dependency measure from Section 2.3 is better suited for this task.

2 https://www.google.com/trends/correlate
Figure 2.7: Example of application of dependency measures for exploratory analysis in machine learning. The dependency between two genes $X$ and $Y$ might be inferred by the amount of computed dependency between their time series. The amount of dependency can be computed using a dependency measure $D(X, Y)$. 

<table>
<thead>
<tr>
<th>time</th>
<th>$X$</th>
<th>$Y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_1$</td>
<td>20.4400</td>
<td>19.7450</td>
</tr>
<tr>
<td>$t_2$</td>
<td>19.0750</td>
<td>20.3300</td>
</tr>
<tr>
<td>$t_3$</td>
<td>20.0650</td>
<td>20.1700</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
RANKING DEPENDENCIES IN NOISY DATA

OUTLINE
In this chapter we introduce a dependency measure based on mutual information to detect and to rank dependencies in noisy data. We identify the gap in the literature of dependency measures based on estimation of mutual information with grinds in the Introduction and Related Work in Section 3.2. In Section 3.3, we introduce the dependency measure Randomized Information Coefficient (RIC) and we discuss its main property of low variance in detail in Section 3.4. Finally, we extensively test it on detection and ranking of dependency between variables in Section 3.5 and between sets of variables in Section 3.6.

3.1 INTRODUCTION

All the dependency measures $D(X, Y)$ between two sets of variables $X$ and $Y$ discussed in Chapter 2 are estimated on a data set $S_n$ through their estimator $\hat{D}(S_n|X, Y)$. Dependency measures summarize relationships between variables into a single number. Nonetheless, the true value $D$ cannot be accessed, this can only be inferred through $\hat{D}$. $\hat{D}(S_n|X, Y)$ is a random variable dependent on $S_n$ which has its own expected value and variance. When differentiating between strong and weak relationships between different variables, multiple realizations of $\hat{D}$ are compared in practice. In this scenario the variance of $\hat{D}$ plays an important role: the higher the variance, the lower the chance

In this chapter we present results from the following manuscript: “The Randomized Information Coefficient: Ranking Dependencies in Noisy Data”, Simone Romano, Nguyen Xuan Vinh, James Bailey, and Karin Verspoor. Under review in the Machine Learning Journal (MLJ)
to correctly rank different relationships in terms of strength. Indeed because of the variability of $\hat{D}$, sometimes weak relationships can obtain higher estimated dependency than stronger relationships. Moreover, the variance of $\hat{D}$ is also important when testing for independence between variables. More specifically, when testing for independence between $X$ and $Y$ on a data set $S_n$, the estimated $\hat{D}(S_n|X, Y)$ is compared to the estimates of $\hat{D}$ under the assumption of independent $X$ and $Y$ [Simon and Tibshirani, 2011]. An estimator $\hat{D}$ with small variance reduces the chances to obtaining similar estimates under real relationship and under independence.

In the case of mutual information discussed in Section 2.3.1, the importance of reducing variance while minimizing the impact on the bias is implied by the statements in Kraskov et al. [2004]; Margolin et al. [2006]; Schaffernicht et al. [2010], which can be summarized as: when comparing dependencies, systematic estimation biases cancel each other out. Therefore smaller variance for mutual information yields a more accurate ranking of relationships. This is particularly true for the estimators of mutual information based on grids which shows a systematic bias which depends to the number of bins used for the discretization. In this section, we investigate the role of bias and variance of the estimator of mutual information based on grids to compare relationships. Estimators based on grids are indeed very computationally efficient and easy to compute. Nonetheless, they have not had success when extended to compare sets of variables.

To quantify the dependency between two sets of numerical variables, we propose a low-variance measure based on information and ensemble theory that can capture many relationship types. Our measure, named the Randomized Information Coefficient (RIC), is computed by randomly generating $K$ discretization grids $G_k$ and averaging the Normalized Mutual Information (NMI) [Kvalseth, 1987] over all the grids as:

$$\text{RIC}(X, Y) \triangleq \frac{1}{K} \sum_{k=1}^{K} \text{NMI}\left((X, Y)|G_k\right)$$  \hspace{1cm} (3.1)

Normalization enables us to consider grids with different cardinalities. The normalized mutual information on a grid $G$ is defined as

$$\text{NMI}\left((X, Y)|G\right) \triangleq \frac{\text{MI}\left((X, Y)|G\right)}{\max\{H(X|G), H(Y|G)\}}$$  \hspace{1cm} (3.2)
where MI and $H$ are respectively the mutual information and the entropy function for discrete variables defined on contingency tables from Section 2.2.1. We choose to normalize by $\max \{H(X), H(Y)\}$ as it is the tightest upper bound that still preserves the metric properties of NMI [Vinh et al., 2010].

The intuition behind this measure is that on average a random grid can encapsulate the relationship between $X$ and $Y$. Both random discretization and ensembles of classifiers have been shown to be effective in machine learning, for example, in random forests [Breiman, 2001]. Substantial randomization has been shown to be even more effective in reducing the variance of predictions [Geurts et al., 2006]. Our aim is to exploit this powerful approach to develop an efficient, effective and easy-to-compute statistic for quantifying dependency between two set variables.

Our contributions in this section are three-fold:

- We propose a low-variance statistic (RIC) based on information and ensemble theory, which is efficient and easy to compute;

- Via theoretical analysis and extensive experimental evaluation, we link our measure’s strong performance on (i) discrimination between strong and weak noisy relationships, and (ii) ranking of relationships, to its low variance estimation of mutual information;

- We extensively demonstrate the competitive performance of RIC versus 15 state-of-the-art dependency measures using both simulated and real scenarios.

3.2 related work

RIC is a dependency measure to compare sets of variables based on normalized mutual information which is efficient and easy to compute. Table 3.1 shows a list of dependency measures currently available in literature. Not all of them is applicable to set of variables and some show high computational complexity with regards to the number of points $n$. Some complexities can be obtained with particular parameter choices or clever implementation techniques. We refer to the respective papers for a detailed analysis. Moreover,
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Mutual information estimators</td>
<td>$I_{ew}$</td>
<td>Mutual Information (Discretization equal width)</td>
<td>Steuer et al. [2002]</td>
<td>$\times$</td>
<td>$O(n^{1.5})$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$I_{ef}$</td>
<td>Mutual Information (Discretization equal frequency)</td>
<td>Steuer et al. [2002]</td>
<td>$\times$</td>
<td>$O(n^{1.5})$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$I_A$</td>
<td>Mutual Information (Adaptive Partitioning)</td>
<td>Cellucci et al. [2005]</td>
<td>$\times$</td>
<td>$O(n^{1.5})$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$I_{\text{mean}}$</td>
<td>Mutual Information (Mean Nearest Neighbors)</td>
<td>Faivishevsky and Goldberger [2009]</td>
<td>✔️</td>
<td>$O(n^2)$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$I_{KDE}$</td>
<td>Mutual Information (Kernel Density Estimation)</td>
<td>Moon et al. [1995]</td>
<td>✔️</td>
<td>$O(n^2)$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$I_{\text{kNN}}$</td>
<td>Mutual Information (Nearest Neighbors)</td>
<td>Kraskov et al. [2004]</td>
<td>✔️</td>
<td>$O(n^{1.5})$</td>
<td>$O(n^2)$</td>
</tr>
<tr>
<td>Correlation based</td>
<td>$r^2$</td>
<td>Squared Pearson’s Correlation</td>
<td>-</td>
<td>$\times$</td>
<td>$O(n)$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ACE</td>
<td>Alternative Conditional Expectation</td>
<td>Breiman and Friedman [1985]</td>
<td>$\times$</td>
<td>$O(n)$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>dCorr</td>
<td>Distance Correlation</td>
<td>Székely et al. [2009]</td>
<td>✔️</td>
<td>$O(n \log n)$</td>
<td>$O(n^2)$</td>
</tr>
<tr>
<td></td>
<td>RDC</td>
<td>Randomized Dependency Coefficient</td>
<td>Lopez-Paz et al. [2013]</td>
<td>✔️</td>
<td>$O(n \log n)$</td>
<td></td>
</tr>
<tr>
<td>Kernel based</td>
<td>HSIC</td>
<td>Hilbert-Schmidt Independence Criterion</td>
<td>Gretton et al. [2005]</td>
<td>✔️</td>
<td>$O(n^2)$</td>
<td></td>
</tr>
<tr>
<td>Information theory based</td>
<td>MIC</td>
<td>Maximal Information Coefficient</td>
<td>Reshef et al. [2011]</td>
<td>$\times$</td>
<td>$O(2^n)$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>GMIC</td>
<td>Generalized Mean Information Coefficient</td>
<td>Luedtke and Tran [2013]</td>
<td>$\times$</td>
<td>$O(2^n)$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MID</td>
<td>Mutual Information Dimension</td>
<td>Sugiyama and Borgwardt [2013]</td>
<td>$\times$</td>
<td>$O(n \log n)$</td>
<td>$O(n^2)$</td>
</tr>
<tr>
<td></td>
<td>MIC$_e$</td>
<td>Maximal Information Coefficient</td>
<td>Reshef et al. [2015b]</td>
<td>$\times$</td>
<td>$O(n)$</td>
<td>$O(n^{2.25})$</td>
</tr>
<tr>
<td></td>
<td>TIC$_e$</td>
<td>Total Information Coefficient</td>
<td>Reshef et al. [2015b]</td>
<td>$\times$</td>
<td>$O(n)$</td>
<td>$O(n^{2.25})$</td>
</tr>
<tr>
<td></td>
<td>RIC</td>
<td>Randomized Information Coefficient</td>
<td>-</td>
<td>✔️</td>
<td>$O(n^{1.5})$</td>
<td></td>
</tr>
</tbody>
</table>

Table 3.1: Dependency measures available in literature compared by their applicability to sets of variables and their best and worst case computational complexity.
recent advances in this area have delivered faster computational techniques for the most recently proposed measures of dependence. For example, the approximated estimator for the population value of MIC can be sped up [Tang et al., 2014; Zhang et al., 2014], and the new exact estimator MIC\textsubscript{e} provides very competitive computational complexity. Moreover, very recently a new technique for fast computation of distance correlation has been proposed [Huo and Szekely, 2014].

The estimators of mutual information based on grids exhibit very good computational complexity, are easy to implement, but are not straightforwardly applicable to sets of variables. RIC is based on the estimation of mutual information via grids of different cardinality and it is applicable to sets of variables. Ranking dependencies between variables with mutual information has been shown to be effective for a number of important applications, such as feature selection [Vinh et al., 2014] and network inference [Villaverde et al., 2013]. Moreover, despite the estimators of mutual information based on kernels, RIC is less sensitive to parameter tuning because it uses many random grids of different cardinality. In our work, the random discretization grids used in RIC can be seen as random projection. Another novel dependency measure based on correlation uses random projections too speed up computations: the Randomized Dependence Coefficient (RDC) discussed in Section 2.3.2. However, we do not use a linear measure of dependency, to assess the dependency between projections. This would require optimization across projections to return a meaningful result. Instead, we compute the normalized mutual information that quantifies non-linear dependencies for each possible projection (grid). This approach allows us to take into account every single grid and each of them contributes to the computation of the average value of NMI across grids. No optimization is required.

### 3.3 The Randomized Information Coefficient

The Randomized Information Coefficient (RIC), defined as in (3.1), is the average normalized mutual information across a set of $K$ randomized grids. RIC is computed on a data set $\{(x, y)\}_{i=0, \ldots, n-1} = \{(x_1 \ldots x_p, y_1 \ldots y_q)\}_{i=0, \ldots, n-1}$ of $n$ data points to capture the dependency between the set $X$ of $p$ variables and $Y$ of $q$ variables. Being based on
mutual information, the population value of RIC is always positive and it is equal to 0 under independence of $X$ and $Y$. This still holds true when either the variables in $X$ or the variables in $Y$ are dependent but the set $X$ is independent of the set $Y$.

Here we propose a few practical ways to obtain contingency tables $(X, Y)|G$ based on the random grid $G$. First of all, by performing $K_r$ random discretizations for both $X$ and $Y$, we can efficiently compute $K = K_r^2$ random grids obtained using all pairs of random discretizations. This allows us to generate fewer random discretizations than by independently generating each grid. The other required parameter is $D_{\text{max}}$, which determines the maximum number of random bins to discretize one variable. Once both variables are discretized, the NMIPROC procedure can be used to compute the normalized mutual information. Algorithm 3.1 presents the pseudo-code for RIC computation.

**Algorithm 3.1** RIC computation.

```
RIC(X, Y, K_r, D_{\text{max}})
1 for k ← 1 to K_r do
2 BinLabelX_k ← RANDOMDISCR(X, D_{\text{max}})
3 BinLabelY_k ← RANDOMDISCR(Y, D_{\text{max}})
4 for k ← 1 to K_r do
5 for k' ← 1 to K_r do
6 RIC+ = NMIPROC(BinLabelX_k, BinLabelY_{k'})
7 return RIC / K_r^2
```

**Discretization of Random Variables:** Next we present in Algorithm 3.2 the random discretization procedure for a single random variable $X$. A variable is discretized using a number of cut-offs $D$ chosen at random in $[1, D_{\text{max}} - 1]$. Each cut-off is chosen by sampling a random example of the variable with uniform distribution. The bin label for each data point can easily be encoded with integer values using $I(\text{cut-off} < x_i)$ with $D$ passes through the data points, where $I$ is the indicator function. The idea is inspired by random ferns [Özuysal et al., 2007; Kursa, 2012]: a type of random forest that achieves even higher speed. This can also be viewed as a random hash function [Wang et al., 2012] or a random projection on a finite set [Lopez-Paz et al., 2013]. This procedure can be easily implemented in any programming language, for example C++.
No sorting is required. The worst case computational complexity of this procedure is $O(D_{\text{max}} \cdot n)$. Figure 3.1 shows some examples of random grids obtained by random discretization.

**Algorithm 3.2** Random discretization of a random variable $X$.

\begin{verbatim}
RANDOMDISCR($X$, $D_{\text{max}}$)
  1. Choose the number of random cut-offs $D$ at random between $[1, D_{\text{max}} - 1]$
  2. for $d \leftarrow 1$ to $D$
  3.    cut-off $\leftarrow$ random data point
  4. for $i \leftarrow 0$ to $n - 1$
  5.    BinLabel$X(x_i)+ = I(\text{cut-off} < x_i)$
  6. return BinLabel$X$
\end{verbatim}

discretization with cut-off of both variables $X$ and $Y$ independently:

![Random grids obtained by applying Algorithm 3.2 to $X$ and $Y$ independently.](image)

Figure 3.1: Example of random grids obtained by applying Algorithm 3.2 to $X$ and $Y$ independently. The normalized mutual information can be computed then on the associated contingency tables.

**Discretization of Sets of Random Variables:** An efficient approach to randomly discretize a set of $p$ random variables $X$ consists not only in choosing cut-offs at random but also to randomly choose the variables to discretize: i.e. build a random fern [Kursa, 2012] on the set of features $X$. This is very computationally efficient: the worst case computational complexity is $O(D_{\text{max}} \cdot n)$ which is independent from the number of variables $p$. However, the straightforward implementation of a random fern presented in Algorithm 3.3 does not allow to have fine control on the number of generated bins $D_{\text{max}}$: the number of maximum bins $D_{\text{max}}$ is exponential in the number of cut-offs $D$, i.e. $D_{\text{max}} = 2^D$. Therefore $D$ cannot be greater than $\log_2 D_{\text{max}} - 1$. Moreover, many bins can be empty due to repeated choices of the same variable.
Algorithm 3.3 Random fern discretization of a set $X$ of random variables

**RandomDiscrFern**($X, D_{\text{max}}$)
1. Choose the number of random cut-offs $D$ at random between $[1, \log_2 D_{\text{max}} - 1]$
2. for $d \leftarrow 0$ to $D - 1$ do
3. $j \leftarrow$ random index of variable in $X$
4. cut-off $\leftarrow$ random data point for $X_j$
5. for $i \leftarrow 0$ to $n - 1$ do
6. $\text{BinLabel}_X(x_i) = 2^d \cdot \mathbb{I}(\text{cut-off} < x_{ij})$
7. return $\text{BinLabel}_X$

We therefore propose a novel randomized approach to discretize a set of $p$ variables $X$ in exactly $D$ bins, maintaining linear worst case complexity in the number of variables and records: $O(D_{\text{max}} \cdot n \cdot p)$. By choosing $D$ random data points as seeds, we can easily discretize a set of variables into $D$ non-empty bins by assigning each data point to its closest seed. We make use of the Euclidean norm to find the distances between points. For the ease of implementation, both random cut-offs in Algorithm 3.3 and random seeds in Algorithm 3.4 are chosen via sampling with replacement. Figure 3.2 shows an example of relationship between the set of variables $X$ and a single variable $Y$ as well as a possible discretization of $X$ using random seeds.

Algorithm 3.4 Random seeds discretization of a set $X$ of random variables

**RandomDiscrSeeds**($X, D_{\text{max}}$)
1. Choose the number of random seeds $D$ at random between $[2, D_{\text{max}}]$
2. Choose a set $S = \{s_1 \ldots s_j \ldots s_D\}$ of $D$ random seeds among the data points
3. for $i \leftarrow 0$ to $n - 1$ do
4. $\text{BinLabel}_X(x_i) = \arg \min_{j, s_j \in S} \text{Dist}(x_i, s_j)$
5. return $\text{BinLabel}_X$

The worst case computational complexity for Algorithm 3.1 to compute RIC between the set $X$ of $p$ variables and the set $Y$ of $q$ variables is thus determined by the discretization algorithm:

- $O\left(K_r \cdot D_{\text{max}} \cdot n + K_r^2(n + D_{\text{max}}^2)\right)$ if random ferns are used;
3.3 The Randomized Information Coefficient

Figure 3.2: Example of quadratic relationship between the set of variables $X = (X_1, X_2)$ and the variable $Y$. Fig. 3.2b shows an example of discretization with random seeds to $X$ by applying Algorithm 3.4.

- $O\left(K_r \cdot D_{\text{max}} \cdot n \cdot (p + q) + K_r^2 (n + D_{\text{max}}^2)\right)$ if random seeds are used.

$K_r$ controls the trade-off between accuracy and computational time. The more randomizations $K_r$ are used, the lower the variance, but the longer the computational time. Based on experimentation we consider $K_r = 20$ a reasonable value. The number of maximum bins $D_{\text{max}}$ should be chosen in order to avoid increasing the grid resolution towards the limit of $NI = 1$ where each point belongs to a single cell. In the worst case, for uniformly distributed variables and $n$ samples we would like to have at least one point per cell of the contingency table. This implies:

$$\frac{n}{D_{\text{max}}^2 D_{\text{max}}} \geq 1 \Rightarrow \frac{n}{D_{\text{max}}^2} \geq 1 \Rightarrow D_{\text{max}}^2 \leq n \Rightarrow D_{\text{max}} = \lceil \sqrt{n} \rceil$$

However, a larger value of $D_{\text{max}}$ might help to identify more complex relationships, at the cost of higher variance. $D_{\text{max}}$ can be tuned to obtain optimal performance. Given that in our analysis we used $D_{\text{max}} = O(\sqrt{n})$, RIC’s worst case computational complexity in the number of data samples is $O(n^{1.5})$. 
3.4 VARIANCE ANALYSIS OF RIC

In this section, we theoretically justify the use of random grids to obtain small variance with the RIC statistic. Then, we prove that a lower variance is beneficial when comparing dependencies and ranking relationships according to the grid estimator of mutual information.

3.4.1 Ensembles for Reducing the Variance

The main motivation for our use of random discretization grids is that averaging across independent random grids allows reduction of variance Geurts [2002]. By using random grids, it is possible to achieve small correlation between the different estimations of NMI. RIC variance tends to be a small value if the estimations are uncorrelated.

**Theorem 3.1.** Let $\text{NMI}_G = \text{NMI}(X, Y|G)$ be the normalized mutual information estimated on a random grid $G$ and RIC as per Eq. (3.1). If NMI estimations for RIC are uncorrelated then:

$$\lim_{K \to \infty} \text{Var}(\text{RIC}) = \text{Var}_G(E[\text{NMI}_G|G])$$

**Proof.** The variance of RIC can be decomposed using Eve’s law of total variance according the i.i.d. random variables grids $G_k$ with $k = 1 \ldots K$ as follows:

$$\text{Var}(\text{RIC}) = \text{Var}_{G_1 \ldots G_K} \left( E[\text{RIC}|G_1 \ldots G_K] \right) + E_{G_1 \ldots G_K} \left[ \text{Var}(\text{RIC}|G_1 \ldots G_K) \right]$$

$$= \text{Var}_{G_1 \ldots G_K} \left( E \left[ \frac{1}{K} \sum_{k=1}^{K} \text{NMI}_{G_k}|G_1 \ldots G_K \right] \right)$$

$$+ E_{G_1 \ldots G_K} \left[ \text{Var} \left( \frac{1}{K} \sum_{k=1}^{K} \text{NI}_{G_k}|G_1 \ldots G_K \right) \right]$$

$$= \text{Var}_{G_1 \ldots G_K} \left( \frac{1}{K} \sum_{k=1}^{K} E[\text{NMI}_{G_k}|G_k] \right)$$

$$+ E_{G_1 \ldots G_K} \left[ \frac{1}{K^2} \left( \sum_{k=1}^{K} \text{Var}(\text{NMI}_{G_k}|G_k) \right) \right]$$
3.4 Variance Analysis of RIC

\[ + \sum_{k \neq k'} \text{Cov}(\text{NMI}_{G_k}, \text{NMI}_{G_{k'}}|G_k, G_{k'}) \] 

\[ = \text{Var}_G(E[\text{NMI}_G|G]) + E_{G_1 \ldots G_K} \left[ \frac{1}{K^2} \left( \sum_{k=1}^{K} \text{Var}(\text{NMI}_{G_k}|G_k) \right) \right. \]

\[ + \sum_{k \neq k'} \text{Corr}(\text{NMI}_{G_k}, \text{NMI}_{G_{k'}}|G_k, G_{k'}) \]

\[ \times \text{Var}(\text{NMI}_{G_k}|G_k) \text{Var}(\text{NMI}_{G_{k'}}|G_{k'}) \left] \right) \]

If \( \text{Corr}(\text{NMI}_{G_k}, \text{NMI}_{G_{k'}}|G_k, G_{k'}) = 0 \) for all \( k \) and \( k' \), then:

\[ \text{Var}(\text{RIC}) = \text{Var}_G(E[\text{NMI}_G|G]) + E_{G_1 \ldots G_K} \left[ \frac{1}{K^2} \left( \sum_{k=1}^{K} \text{Var}(\text{NMI}_{G_k}|G_k) \right) \right. \]

\[ = \text{Var}_G(E[\text{NMI}_G|G]) + \frac{E_G[\text{Var}(\text{NMI}_G|G)]}{K} \]

that when \( K \to \infty \) is equal to \( \text{Var}_G(E[\text{NMI}_G|G]) \).

The expected value \( E[\text{NMI}_G|G] \) is less dependent on the data because of the random grid \( G \) and shows small variance across grids. Intuitively, this result suggests that some variance of the data can be captured with the random grids. We empirically validate this result in Figure 3.3. In practice, it is very difficult to obtain completely uncorrelated NMI estimations. Nonetheless, the use of random grids allows us to strongly decrease their correlation.

We aim to show that the decrease in variance is due to the random grid \( G \), by comparing the variance of \( \text{NMI}_F \) where \( F \) is a fixed grid with equal width bins for \( X \) and \( Y \). The number of bins for each variable is fixed to 9 for both \( G \) and \( F \), and cut-offs are generated in the range \([-2, 2]\) and \([-3, 3]\) for \( X \) and \( Y \), respectively. The chosen joint distribution \((X, Y)\) is induced on \( n = 100 \) points with \( X \sim \mathcal{N}(0, 1) \) and \( Y = X + \eta \) with \( \eta \sim \mathcal{N}(0, 1) \). The variance of RIC decreases as \( K \) increases because the random grids enable us to decorrelate the estimations of NMI. In general, if we allow grids of different cardinality (different number of cut-offs) and large \( K \), the variance can be decreased even further.
Figure 3.3: Variance of RIC compared to the variance of NMI\(_F\) on a fixed equal width grid \(F\). According to Theorem 3.1 if estimations are uncorrelated, the variance of RIC tends to the variance of \(E[\text{NMI}_G|G]\) which is less dependent on the data. In practice, estimations are always correlated. Nonetheless, the use of random grids helps in decreasing the correlation between them.

Using RIC in Algorithm 3.1 we can efficiently compute \(K = K_r^2\) grids. Increasing the number of random grids by increasing \(K_r\) is always beneficial. However, this is particularly important when the sample size \(n\) is small. In Figure 3.4 we show the behavior of RIC’s variance at the variation of \(K_r\) for different sample size \(n\) for the same relationship discussed above. The variance reaches the plateau already at \(K_r = 50\) when \(n = 500\). On the other hand, when the sample size is small, e.g. \(n = 50\), the variance is still decreasing at \(K_r = 100\). \(K_r\) might be chosen according to the sample size \(n\): i.e. larger if the sample size \(n\) is small and smaller if the sample size \(n\) is large. Nonetheless, having a large \(K_r\) is always beneficial in general, at the cost of computational time.

Figure 3.4: Variance of RIC in Algorithm 3.1 at the increase of the number of random grids for different sample size \(n\). Increasing \(K_r\) is always beneficial. However, it is particularly important when \(n\) is small. For example, the variance of RIC still decreases for \(K_r > 50\) for this particular relationship between \(X\) and \(Y\).
3.4.2 Importance of Variance in Comparing Relationships using the Grid Estimator of Mutual Information

When mutual information is used as a proxy for the strength of the relationship, a small estimation variance is likely to be more useful than a smaller bias when comparing relationships, as implied by some observations in Kraskov et al. [2004]; Margolin et al. [2006]; Schaffernicht et al. [2010]. The reason is that systematic biases cancel each other out. We formalize these observations as follows:

**Theorem 3.2.** Let \( \text{bias}(\hat{\phi}) = \phi - E[\hat{\phi}] \) be the bias of the estimator \( \hat{\phi} \). Let \( \hat{\phi}(s) = \hat{\phi}_s \) and \( \hat{\phi}(w) = \hat{\phi}_w \) be estimations of \( \phi \) on the strong relationship \( s \) and the weak relationship \( w \), where the true values are \( \phi_s > \phi_w \). The probability of making an error \( P(\hat{\phi}_s \leq \hat{\phi}_w) \) is bounded above:

\[
P(\hat{\phi}_s \leq \hat{\phi}_w) \leq \frac{\text{Var}(\hat{\phi}_s) + \text{Var}(\hat{\phi}_w)}{\text{Var}(\hat{\phi}_s) + \text{Var}(\hat{\phi}_w) + \left( \phi_s - \phi_w - (\text{bias}(\hat{\phi}_s) - \text{bias}(\hat{\phi}_w)) \right)^2}
\]

if \( E[\hat{\phi}_s] > E[\hat{\phi}_w] \) or equivalently if \( \phi_s - \phi_w > \text{bias}(\hat{\phi}_s) - \text{bias}(\hat{\phi}_w) \).

**Proof.** Let \( \hat{\Delta} = \hat{\phi}_w - \hat{\phi}_s \), if \( E[\hat{\Delta}] < 0 \) then:

\[
P(\hat{\Delta} \geq 0) = P(\hat{\Delta} - E[\hat{\Delta}] \geq -E[\hat{\Delta}]) \leq \frac{\text{Var}(\hat{\Delta})}{\text{Var}(\hat{\Delta}) + E[\hat{\Delta}]^2}
\]

according to the 1-sided Chebyshev inequality that is also known as the Cantelli’s inequality [Ross, 2012].

**Remark:** If there is a systematic bias component, the variance of a dependency measure is important also to identify if a relationship exists. The probability of making an error in determining if a relationship exists (testing for independence between \( X \) and \( Y \) with \( \hat{\phi} \)) is just a special case of Theorem 3.2 where \( \phi_w = 0 \).

Regarding the grid estimator \( \text{MI}_F = \text{MI}(\langle X, Y \rangle|F) \) on a fixed grid \( F \) with \( n_F \) bins of the differential mutual information between two variables \( I(X, Y) \) in Eq. (2.21), there
is always a systematic bias component which is a function of the number of samples \( n \) and the number of bins \( n_F \) [Moddemeijer, 1989]. This systematic bias component cancels out in \( \text{bias}(\text{MI}_{F,s}) - \text{bias}(\text{MI}_{F,w}) \). If the non-systematic estimation bias is small enough, then the denominator of the upper bound is dominated by \( I_s - I_w \). Therefore, the upper bound decreases because of the numerator, i.e., the sum of the variances. Of course variance is just part of the picture. It is worth to decrease the variance of an estimator if the estimand has some utility. Moreover, many estimators have a bias and variance trade-off. Deliberately reducing the variance at the expense of bias is not a good idea. Variance can be reduced if there is a strong systematic estimation bias component and if the effect on the non-systematic bias is minimal.

We empirically compare the probability of error as stated in Theorem 3.2 for the estimation the differential mutual information \( I \) with grids. RIC can be used to estimate mutual information if we average across grids of the same cardinality and do not normalize mutual information on the grids. Let \( s = (X, Y_s) \) and \( w = (X, Y_w) \) be the strong and the weak relationships where \( X \sim \mathcal{N}(0, 1) \), \( Y_s = X + \eta_s \) and \( Y_w = X + \eta_w \) with \( \eta_s \sim \mathcal{N}(0, 7) \) and \( \eta_w \sim \mathcal{N}(0, 1) \). Indeed, if \( X \sim \mathcal{N}(0, \sigma^2_X) \) and \( Y = X + \eta \) with \( \eta \sim \mathcal{N}(0, \sigma^2_\eta) \) it is possible to analytically compute the mutual information between \( X \) and \( Y \): \( I(X,Y) = 0.5 \log_2 (1 + \sigma^2_X / \sigma^2_\eta) \). In Figure 3.5 we compare the probability of error \( P(\text{RIC}_s < \text{RIC}_w) \) for RIC as an estimator of mutual information and the probability of error \( P(\text{MI}_{F,s} < \text{MI}_{F,w}) \) for the estimator \( I_F \) on a fixed equal width grid \( F \), with an increase of the number of random grids \( K \) for RIC. We generate 13 bins for \( X \) and \( Y \) for both \( F \) and RIC’s grids. The distributions are induced on \( n = 100 \) samples. The probability of error is smaller for RIC because of its small variance. Indeed, the probability of error decreases with the increase of \( K \), just as the variance decreases with bigger \( K \). The bias stays constant when \( K \) varies and it contributes less to a small probability of error. We aim to show also the contribution of bias and variance to the probability of error by computing its upper bound from Theorem 3.2. The upper bounds of the probability of error for RIC and \( \text{MI}_F \) are respectively:

\[
U(\text{RIC}) = \frac{\text{Var}(\text{RIC}_s) + \text{Var}(\text{RIC}_w)}{\text{Var}(\text{RIC}_s) + \text{Var}(\text{RIC}_w) + (I_s - I_w - (\text{bias}(\text{RIC}_s) - \text{bias}(\text{RIC}_w)))^2}
\]
3.4 VARIANCE ANALYSIS OF RIC

Figure 3.5: Probability of error in identifying the strong relationship. RIC’s probability is smaller due to its small variance.

\[ U(\text{MI}_F) = \frac{\text{Var}(\text{MI}_{F,s}) + \text{Var}(\text{MI}_{F,w})}{\text{Var}(\text{MI}_{F,s}) + \text{Var}(\text{MI}_{F,w}) + \left( I_s - I_w - (\text{bias}(\text{MI}_{F,s}) - \text{bias}(\text{MI}_{F,w})) \right)^2} \]

Figure 3.6 shows the behavior of the upper bound at the variation of \( K \) term by term. The bias difference for RIC as an estimator of the differential mutual information \( I \) is a bit bigger than the bias difference for MI\(_F\). Nonetheless, the probability of error decreases mainly because of the variance decrease of RIC.

Moreover, when the dependency measure with a systematic bias is used for ranking relationships, we can still show that reducing the estimator variance plays an important role.

**Corollary 3.3.** When ranking \( m \) relationships according to the true ranking \( \phi_1 > \phi_2 > \cdots > \phi_m \), the probability \( P(\hat{\phi}_1 > \hat{\phi}_2 > \cdots > \hat{\phi}_m) \) of accurately obtaining the correct ranking using the estimators \( \hat{\phi}_i, i = 1, \ldots, m \) is bounded below by:

\[
1 - \sum_{i=1}^{m-1} \frac{\text{Var}(\hat{\phi}_{i+1}) + \text{Var}(\hat{\phi}_i)}{\text{Var}(\hat{\phi}_{i+1}) + \text{Var}(\hat{\phi}_i) + \left( \phi_{i+1} - \phi_i - (\text{bias}(\hat{\phi}_{i+1}) - \text{bias}(\hat{\phi}_i)) \right)^2}
\]

if \( E[\hat{\phi}_{i+1}] > E[\hat{\phi}_i] \) or equivalently if \( \phi_{i+1} - \phi_i > \text{bias}(\hat{\phi}_{i+1}) - \text{bias}(\hat{\phi}_i) \) \( \forall i = 1 \ldots m - 1 \).

**Proof.** Let \( \mathcal{E}_i = \{ \hat{\phi}_{i+1} > \hat{\phi}_i \} \) be an event then:

\[
P(\hat{\phi}_1 > \hat{\phi}_2 > \cdots > \hat{\phi}_m) = P(\mathcal{E}_1 \cap \mathcal{E}_2 \cap \cdots \cap \mathcal{E}_{m-1}) = 1 - P(\mathcal{E}_1^c \cup \mathcal{E}_2^c \cup \cdots \cup \mathcal{E}_{m-1}^c)
\]
Probability of Error

Figure 3.6: Probability of error in identifying the strong relationship. RIC’s probability is smaller due to its small variance.

\[ \geq 1 - \sum_{i=1}^{m-1} P(\mathcal{E}^c_i) \]

where \( \mathcal{E}^c_i \) is the complementary event to \( \mathcal{E}_i \): \( \mathcal{E}^c_i = \{ \hat{\phi}_{i+1} \leq \hat{\phi}_i \} \). The corollary follows using the upper bound for \( P(\mathcal{E}^c_i) \) proved in Theorem 2:

\[ P(\mathcal{E}^c_i) \leq \frac{\text{Var}(\hat{\phi}_{i+1}) + \text{Var}(\hat{\phi}_i)}{\text{Var}(\hat{\phi}_{i+1}) + \text{Var}(\hat{\phi}_i) + (\phi_{i+1} - \phi_i - (\text{bias}(\hat{\phi}_{i+1}) - \text{bias}(\hat{\phi}_i))^2} \]

As we empirically demonstrated above for the grid estimator of mutual information, \( \text{bias}(\hat{\phi}_{i+1}) - \text{bias}(\hat{\phi}_i) \) tends to be small if there is some systematic bias component, and thus a small variance is the main contributor to the accuracy.
Remark about boostrapping: It is also natural to consider whether using bootstrapping improves the discrimination performance of a statistic by decreasing the variance. When bootstrapping, the statistic is actually estimated on around 63% of the samples and this decreases the discrimination ability of each measure. Similarly, sampling without replacement of a smaller number of points and averaging across different estimation of a measure is not expected to perform well. The best way to decrease the variance is thus to inject randomness in the estimator itself. This is the rationale for RIC. We achieve this goal by using a strong measure such as mutual information and injects randomness in its estimation in order to decrease the global variance.

3.5 experiments on dependency between two variables

In this section, we compare RIC\(^1\) with 15 other state-of-the-art statistics that quantify the dependency between two variables \(X\) and \(Y\). We focus on three tasks: identification of noisy relationships, inference of network of variables, and feature filtering for regression. Table 3.2 shows the list of competitor measures compared in this chapter and the parameters used in their analysis. The parameters used are the default parameters suggested by the authors of the measures in their respective papers. Indeed, only on the task of feature filtering for regression it is possible to tune parameters with cross-validation on a given data set. The tasks of inference of network of variables and identification of noisy relationships are unsupervised learning tasks and do not allow parameter tuning when applied to a new data set. Nonetheless, most of the default parameters are not tuned for hypothesis testing. Therefore, we decided to follow the approach used in Reshef et al. [2015a]. In this empirical study, leading measures of dependence are compared in terms of two important features: equitability and power against independence. We do not compare RIC in terms of equitability. The MIC and MIC\(_e\) statistics have been shown to be more equitable than mutual information and to be the state-of-the-art on this task. Rather, we focus on power against independence on different noise models. Given a particular noise model, we identify the best parameters for independence testing by maximizing the power on average on a set of relationships and different noise levels.

\(^1\) RIC implementation is available at https://sites.google.com/site/randinfocoeff/
## Table 3.2: Dependency measures compared in this paper and parameters used in the tasks of network inference, feature filtering for regression, and estimation of running times.

<table>
<thead>
<tr>
<th>Family</th>
<th>Acr.</th>
<th>Name</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mutual information</td>
<td>$I_{ew}$</td>
<td>Mutual Information (Discretization equal width)</td>
<td>$D = \lfloor \sqrt{n}/5 \rfloor$</td>
</tr>
<tr>
<td>estimators</td>
<td>$I_{ef}$</td>
<td>Mutual Information (Discretization equal frequency)</td>
<td>$D = \lfloor \sqrt{n}/5 \rfloor$</td>
</tr>
<tr>
<td></td>
<td>$I_A$</td>
<td>Mutual Information (Adaptive Partitioning)</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>$I_{mean}$</td>
<td>Mutual Information (Mean Nearest Neighbors)</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>$I_{KDE}$</td>
<td>Mutual Information (Kernel Density Estimation)</td>
<td>$h = n^{-1/6}$</td>
</tr>
<tr>
<td></td>
<td>$I_{kNN}$</td>
<td>Mutual Information (Nearest Neighbors)</td>
<td>$k = 6$</td>
</tr>
<tr>
<td>Correlation based</td>
<td>$r^2$</td>
<td>Squared Pearson’s Correlation</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>ACE</td>
<td>Alternative Conditional Expectation</td>
<td>$\epsilon = 10^{-12}$</td>
</tr>
<tr>
<td></td>
<td>dCorr</td>
<td>Distance Correlation</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>RDC</td>
<td>Randomized Dependency Coefficient</td>
<td>$k = 20, s = 1/6$</td>
</tr>
<tr>
<td>Kernel based</td>
<td>HSIC</td>
<td>Hilbert-Schmidt Independence Criterion</td>
<td>$\sigma_X, \sigma_Y = \text{med. dist.}$</td>
</tr>
<tr>
<td>Information theory</td>
<td>MIC</td>
<td>Maximal Information Coefficient</td>
<td>$\alpha = 0.6$</td>
</tr>
<tr>
<td>based</td>
<td>GMIC</td>
<td>Generalized Mean Information Coefficient</td>
<td>$\alpha = 0.6, p = -1$</td>
</tr>
<tr>
<td></td>
<td>MID</td>
<td>Mutual Information Dimension</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>TIC$_\epsilon$</td>
<td>Total Information Coefficient</td>
<td>$\alpha = 0.65$</td>
</tr>
<tr>
<td></td>
<td>RIC</td>
<td>Randomized Information Coefficient</td>
<td>$K_r = 20, D_{\text{max}} = \lfloor \sqrt{n} \rfloor$</td>
</tr>
</tbody>
</table>

All the measures compared in this section were thoroughly discussed in Section 2.3. The measures in the first group of Table 3.2 are mutual information estimators. $I_{ew}$ and $I_{ef}$ are respectively the equal-width and equal-frequency bin grid estimator of mutual information. $I_A^2$ is the adaptive grid estimator of mutual information that assures the number of points for each cell to be at least 5. We chose to fix the number of bins $D$ for $I_{ew}$ and $I_{ef}$ to $\lfloor \sqrt{n}/5 \rfloor$ because no universally accepted value was found in the literature.

---

2 From [http://www.iim.csic.es/~gingproc/mider.html](http://www.iim.csic.es/~gingproc/mider.html) [Villaverde et al., 2014].
Kraskov’s $k$ nearest neighbors estimator $I_{k\text{NN}}$ uses a fixed parameter $k = 6$ and the kernel density estimator $I_{\text{KDE}}$ uses the parameter $h = \frac{4}{p+q+2}^{1/(p+q+4)} n^{-1/(p+q+4)} = n^{-1/6}$ when comparing two variables given that the number of variables is $p + q = 2$. This is one possible kernel width and suggested as a default value in Steuer et al. [2002]. We also compare $I_{\text{mean}}$ from Faivishevsky and Goldberger [2009]. All other measures were used with the default parameters suggested in their respective papers as described in Table 3.2: dCorr, RDC, ACE, HSIC, MIC, GMIC, MID. As we discussed above, we tuned the parameters of each measure when testing for independence. Being the state-of-the-art in this task, we also introduced TIC in the analysis. Regarding RIC on computing dependency between two variables, we decided to generate discretizations for $X$ and $Y$ according Algorithm 3.2; we generate for each discretization a random number of cut-offs $D$ chosen at random uniformly in $[1, D_{\text{max}} - 1]$.

3.5.1 Identification of Noisy Relationships

We consider the task of discriminating between noise and a noisy relationship, i.e., determining whether a dependency exists by testing for independence between $X$ and $Y$, across a large number of dependency types. In Figure 3.7a, 12 different relationships between $X$ and $Y$ are induced on $n = 320$ data points. We use the same setting as in Simon and Tibshirani [2011]. In this study, the measure performance on a relationship is assessed by power at level $\alpha = 0.05$. For each test case, we generated 500 random data sets with $X$ and $Y$ being completely independent. These constitute the negative class or the complete noise class. Then, for each noise level between 1 and 30, we generate 500 other data sets to create the positive class or the noisy relationship class. We evaluate the ability of different measures to discriminate between complete noise and the noisy relationship classes by computing the power.

4 From http://tinyurl.com/ojlkrla [Margolin et al., 2006].
5 From http://tinyurl.com/ozadxzr.
6 From http://tinyurl.com/oja3k3v.
8 From http://mpba.fbk.eu/cmine.
9 From https://github.com/mahito-sugiyama/MID.
(sensitivity) for the positive class at level $\alpha = 0.05$. Experiments were carried out on two different noise models, namely additive noise and white noise. In the first scenario we add different levels of Gaussian noise by varying the noise standard deviation $\sigma_\eta$. In the second scenario we substitute some points of the relationship with uniform noise. Figures 3.7b and 3.7c show examples of noise levels for the linear relationship in the additive noise model and white noise model respectively. Given that all measures present good discrimination ability in the white noise model, level 1 (lowest noise) starts by assigning 40% of points to the relationship and 60% to uniformly distributed noise.

Figure 3.7: Relationships between two variables and example of additive and white noise.

Power at level $\alpha = 0.05$ of discrimination between complete noise and noisy relationship for each relationship type presented in the paper is shown in Figures 3.8 and 3.9.
These two figures show result for each measure with optimal parameters for independence testing. Indeed, given that the default parameters of each measure are not tuned for independence testing, we decided to follow the approach of Reshef et al. [2015a]: for a particular noise model we identify the parameters that maximize the average power for all level of noise and all relationship types. Parameter tuning is performed on Figure 3.10 and Figure 3.11. Regarding the discretization based estimators of mutual information $I_{ew}$ and $I_{ef}$, we varied the parameter $c$ in $D = \lfloor \sqrt{n/c} \rfloor$ to change the grid coarsity. Similarly, we varied the parameter $c$ in $D_{\text{max}} = \lfloor \sqrt{n/c} \rfloor$ for RIC. The parameter $s$ in RDC should be set around $1/6$ when comparing variables Lopez-Paz et al. [2013]. In order to identify the best parameter settings for power in RDC, we explored values around $1/6$ by varying $\tilde{p}$ in $s = \frac{1}{6} \cdot \tilde{p}$. Similarly, the kernel widths in HSIC are usually set to the median distance of the data points according $X$ and according $Y$. This is why we explored the values $\sigma_X = \text{med. dist. in } X \cdot \tilde{p}$ and $\sigma_Y = \text{med. dist. in } Y \cdot \tilde{p}$. The parameter $\tilde{p}$ can be seen as a percentage of the default parameters. According to our analysis, it is beneficial to tune parameters to target independence between $X$ and $Y$ for some measures. For example, a bigger number of nearest neighbors is beneficial to $I_{kNN}$ to achieve more power under the additive noise model. This was also discussed in the original paper: Kraskov et al. [2004]. Furthermore, measures which make use of kernels should employ kernels with larger width to maximize power under additive noise. Even though these parameters cannot be tuned on a new data set because the task is unsupervised, the analysis we provide here might guide the user when the particular noise model is known for a new data set to analyze.
Figure 3.8: Power in the additive noise scenario with optimal parameters (best viewed in color).
3.5 Experiments on dependency between two variables

Figure 3.9: Power in the white noise scenario with optimal parameters (best viewed in color).
Figure 3.10: Parameter tuning to maximize the power of each measure on average for the additive noise model when comparing variables. These plots show the average area under power curve and their average across relationship types.
Figure 3.11: Parameter tuning to maximize the power of each measure on average for the white noise model when comparing variables. These plots show the average area under power curve and their average across relationship types.
As discussed in Section 3.4.1, increasing $K_r$ for RIC helps to decrease its variance. This is particularly important in order to achieve power when testing against independence. Figure 3.12a and 3.12b show the area under power curve for each relationship tested in this paper and their average at the variation of $K_r$ for RIC. Increasing $K_r$ is very beneficial to increase power when the number of data points is small: $n = 100$. This is an interesting feature of RIC. Increasing $K_r$ gives more power but also increases the computational running time. Nonetheless, higher $K_r$ is needed only if the sample size is small.

![Graphs showing area under power curve for RIC at different $K_r$ values.](image)

(a) Power of RIC at small sample size: $n = 100$. (b) Power of RIC on larger samples: $n = 1,000$.

Figure 3.12: The power against independence of RIC always increases at the increase of $K_r$ because its variance decreases. This is particularly important when the number of data points is small: e.g., $n = 100$.

Figures 3.8 and 3.9 show the performance of each measures with optimal parameters; regarding RIC, $D_{\text{max}} = \lfloor \sqrt{n/4} \rfloor$ and $K_r = 200$. In order to compare the different measures on multiple data sets (relationships) we use the framework proposed in Demšar [2006]: we show the average rank across data sets for each measure. Moreover, in order to provide graphical intuition about their performance, we show their average rank sorted in ascending order using bar plots. Figure 3.13 present the performances on the additive noise model. RIC computed with $D_{\text{max}} = \lfloor \sqrt{n/4} \rfloor$ and $K_r = 200$ shows very competitive performance.
3.5 Experiments on Dependency Between Two Variables

RIC outperforms all mutual information estimators, in particular the discretization based $I_{ew}$, $I_{ef}$, $I_A$, and the $k$NN based $I_{kNN}$. The kernel based density estimator $I_{KDE}$ looks more competitive in noisy scenarios than all the other mutual information estimators, as also pointed out in Khan et al. [2007]. The performance of $I_{mean}$ is particularly surprising: even if $I_{mean}$ is a smooth estimator of mutual information, which guarantees low variance, it cannot discriminate very noisy relationships well. A careful look at its derivation reveals that $I_{mean}$ takes into account $k$NN with $k$ very large, e.g. $k = n - 1$. In fact, even $I_{kNN}$ in this case cannot discriminate between noise and noisy relationships. MIC with parameters optimized for independence testing shows to outperform distance correlation. MIC outperforms GMIC with parameter $p = -1$ when the parameter $\alpha$ is tuned independently for both of them. In particular, MIC obtains its optimum at $\alpha = 0.35$ and GMIC at $\alpha = 0.65$. The comparison carried out in Luedtke and Tran [2013] considered $\alpha = 0.6$ for both measures, concluding that GMIC was superior with this setting. The new information theoretic based measure MID also presents less competitive discrimination ability on this set of relationships because it is better suited for the white noise model. RDC achieves good results overall, in particular on the scenarios where it seems possible to linearize the relationship via a random projection: low frequency sinusoids and circle relationships. When the relationship is linear, $r^2$ is the

![Figure 3.13: Average rank of measures across relationships when the target is power maximization under the additive noise model. RIC with $D_{max} = \lfloor \sqrt{n}/4 \rfloor$ and $K_r = 200$ is very competitive in this scenario.](image-url)
best choice in terms of discrimination ability. This property reflects the motivation for
Dcorr, which was proposed as a distance based extension to the non-linear scenarios:
its performance is very competitive on the linear, and 4th root case. However, it fails in
the high frequency sinusoidal case and circle relationship. The best measure among the
competitors is the newly proposed TIC, which is explicitly designed for independence
testing. These results indicate that when the purpose is to identify an arbitrary relation-
ship in the additive noise scenario, RIC delivers extremely competitive performance on
average. If the user is interested in a specific relationship type, it will be best to choose
a particular dependency measure known to be specifically good for that scenario. The
results in Figures 3.8 and 3.9 may help guide the user in this choice.

RIC also shows good performance under the white noise model but it is outperformed
by HSIC. Average results with optimal parameters for each measure are shown in Figure
3.14. Regarding the grid estimators of mutual information, RIC, TIC, MIC, and MID,
a denser grid is better suited for the white noise scenario because points are uniformly
distributed on the joint domain \((X, Y)\). \(I_{kNN}\) presents competitive performance under
white noise when \(k\) is small. As in the additive noise model, TIC proved to be strong
competitor to RIC in this scenario. Instead, Dcorr seems to be little competitive under
the white noise model. HSIC with very small kernel width performs the best under white
noise.

![Figure 3.14: Average rank of measures across relationships when the target is power maximiza-
tion under the white noise model. RIC with \(D_{\text{max}} = \lfloor \sqrt{n} \times 10 \rfloor\) and \(K_r = 200\) is
competitive but yet it is outperformed by HSIC.](image)

---

**Ranking Dependencies in Noisy Data**

---
3.5.2 Application to Network Inference

We next employ the measures for biological network reverse engineering, which is a popular and successful application domain for dependency measures [Villaverde et al., 2013]. The applications include cellular, metabolic, gene regulatory, and signaling networks. Each of the $m$ variables is associated with a time series of length $n$. In order to identify the strongest relationships between variables (e.g., genes), a dependency measure $D$ is employed. Due to the natural delay of biochemical interactions in biological networks, the strongest dependency might occur only after some time [Xuan et al., 2012]. For this reason, we incorporate time delay into the dependency measures as $D_{\text{delayed}} = \max_{\tau \in [-\tau_m, +\tau_m]} D (X(t - \tau), Y(t))$, where $D$ is any measure from Table 3.2 and $\tau_m$ is the maximum time delay. We collected 10 data sets where the true interactions between the variables are known. A dependency measure is effective on this task if its output is high on real interactions (positive class) and low on non-interacting pairs of variables (negative class). We evaluate the performance of a measure with the average precision (AP), also known as the area under the precision-recall curve. In order to obtain meaningful comparisons and perform statistical hypothesis testing, we performed 50 bootstrap repetitions for each data set and computed the mean AP (mAP) across the repetitions.

We made use of the MIDER framework [Villaverde et al., 2014] for evaluating the performance of dependency measures. The first 7 data sets were retrieved from the MIDER framework. The last 3 data sets were generated using SynTren [Van den Bulcke et al., 2006], a generator of synthetic gene expression data. SynT1 and SynT1-s were generated starting from the Escherichia coli transcriptional regulatory network provided with the framework with default noise parameters where SynT1-s has shorter time series. SynT2 was generated starting from the synthetic direct acyclic graph provided with the framework. Based on the data sampling rate, we set $\tau_m = 3$ for these data sets, which cover most plausible time-delayed interactions. Table 3.3 shows a summary of the data sets used.

The small amount of data available and the high amount of noise in biological time series posed a very challenging task for all statistics. Mutual information estimators have
been extensively employed for this task [Villaverde et al., 2013]. Just recently, HSIC has been tested on network inference [Lippert et al., 2009] and even more recently dCorr has been shown to be competitive on this task [Guo et al., 2014]. To our knowledge, there is no prior comprehensive survey of the performance of RDC, $I_{\text{mean}}$, MIC, GMIC and MID on this task. We perform a comprehensive evaluation of RIC plus 14 other dependency measures on network inference. The results are shown in Table 3.4.

We use RIC with parameters $D_{\text{max}} = \lfloor \sqrt{n} \rfloor$ and $K_r = 20$ because on these tasks it is important to achieve high discrimination between strong relationships as well as weak relationships. Figure 3.15 presents the average rank of the measures across all tested networks. Overall, RIC performs consistently well across all data sets. It outperforms by far all the discretization based mutual information estimators as well as other information theoretic based measures including MIC, GMIC and MID. Among the mutual information estimators, $I_{\text{KDE}}$ and $I_{\text{NN}}$ show very good results. RIC’s main competitor was dCorr, which also shows very good performance mainly due to the crucial importance of the linear relationships between variables. Its results are very correlated with $r^2$ results, which in some cases provides the best result for a single data set. This is mainly due to its high ability to discriminate linear relationships well. We found RIC particularly competitive on short time series with a large number of variables.

| Name       | n   | m  | | Name       | n   | m  |
|------------|-----|----| |------------|-----|----|
| Glycolysis | 57  | 10 | | Pyrim      | 74  | 27 |
| Enzyme-cat | 250 | 8  | | Bodyfat    | 252 | 14 |
| Small-chain| 100 | 4  | | Triazines  | 186 | 60 |
| Irma-on-off| 125 | 5  | | Wisconsin  | 194 | 32 |
| Mapk       | 210 | 12 | | Crime      | 111 | 144|
| Dream4-10  | 105 | 10 | | Pole       | 1000| 48 |
| Dream4-100 | 210 | 100| | Qsar       | 384 | 482|
| SynT1      | 100 | 200| | Qsar2      | 384 | 186|
| SynT1-s    | 30  | 200| |             |     |    |
| SynT2      | 30  | 40 | |             |     |    |

Table 3.3: Summary of the data sets used for network inference (left) and regression (right): $n$ is the data points and $m$ is the number of variables.
3.5 Experiments on Dependency Between Two Variables

As well known within the machine learning community, there is no “free lunch”. In the context of this application, this wisdom is evident, observing in Table 3.4 that no method always performs the best or worst in every case. MID for example, is badly affected by additive noise commonly observed in biological time series and thus showed overall less competitive performance. Nonetheless, it achieved the best performance on Irma-on-off, an \textit{in vivo} yeast semi-synthetic network.
Table 3.4: Mean average precision (mAP) on 10 networks: \( n \) length of time series; \( m \) number of variables. Each cell shows mAP ± std and either (+), (=), or (−) means statistically greater, equal, or smaller according to the 1-sided paired \( t \)-test (\( \alpha = 0.05 \)) than RIC results.

<table>
<thead>
<tr>
<th></th>
<th>Glycolysis</th>
<th>Enzyme-cat</th>
<th>Small-chain</th>
<th>Irna-on-off</th>
<th>Mapk</th>
<th>Dream4-10</th>
<th>Dream4-100</th>
<th>SynT1</th>
<th>SynT1-s</th>
<th>SynT2</th>
</tr>
</thead>
<tbody>
<tr>
<td>((n, m))</td>
<td>(57, 10)</td>
<td>(250, 8)</td>
<td>(100, 4)</td>
<td>(125, 5)</td>
<td>(210, 12)</td>
<td>(105, 10)</td>
<td>(210, 100)</td>
<td>(100, 200)</td>
<td>(30, 200)</td>
<td>(30, 40)</td>
</tr>
<tr>
<td>RIC</td>
<td>67.5 ± 3.7</td>
<td>91.4 ± 2.0</td>
<td>91.4 ± 1.2</td>
<td>70.5 ± 3.5</td>
<td>57.6 ± 2.2</td>
<td>64.6 ± 6.1</td>
<td>10.3 ± 0.7</td>
<td>7.9 ± 0.4</td>
<td>6.6 ± 0.7</td>
<td>14.1 ± 3.8</td>
</tr>
<tr>
<td>dCot</td>
<td>67.8 ± 3.0 (=)</td>
<td>88.6 ± 2.3 (−)</td>
<td>91.7 ± 0.0 (+)</td>
<td>68.6 ± 2.6 (−)</td>
<td>50.0 ± 1.0 (−)</td>
<td>68.8 ± 5.9 (+)</td>
<td>12.6 ± 0.6 (+)</td>
<td>7.4 ± 0.3 (−)</td>
<td>6.7 ± 0.6 (=)</td>
<td>16.0 ± 2.5 (+)</td>
</tr>
<tr>
<td>( I_{KDE} )</td>
<td>67.9 ± 3.7 (=)</td>
<td>93.5 ± 2.1 (+)</td>
<td>88.1 ± 8.1 (−)</td>
<td>71.5 ± 5.6 (=)</td>
<td>59.0 ± 3.1 (+)</td>
<td>61.1 ± 7.0 (−)</td>
<td>9.7 ± 0.7 (−)</td>
<td>7.8 ± 0.3 (−)</td>
<td>6.4 ± 0.6 (=)</td>
<td>10.2 ± 1.8 (−)</td>
</tr>
<tr>
<td>( I_{KNN} )</td>
<td>65.5 ± 4.6 (−)</td>
<td>91.5 ± 3.8 (=)</td>
<td>90.0 ± 5.1 (−)</td>
<td>72.7 ± 6.6 (+)</td>
<td>68.8 ± 2.0 (+)</td>
<td>51.4 ± 6.1 (−)</td>
<td>8.5 ± 0.8 (−)</td>
<td>7.8 ± 0.4 (−)</td>
<td>7.0 ± 0.8 (+)</td>
<td>10.3 ± 1.8 (−)</td>
</tr>
<tr>
<td>( \tau^2 )</td>
<td>68.4 ± 2.7 (=)</td>
<td>86.0 ± 1.9 (−)</td>
<td>91.7 ± 0.0 (+)</td>
<td>69.0 ± 2.9 (−)</td>
<td>46.9 ± 1.3 (−)</td>
<td>56.7 ± 6.1 (−)</td>
<td>12.5 ± 0.6 (−)</td>
<td>6.7 ± 0.4 (−)</td>
<td>6.3 ± 0.6 (−)</td>
<td>14.5 ± 2.3 (=)</td>
</tr>
<tr>
<td>RDC</td>
<td>61.6 ± 7.6 (−)</td>
<td>89.2 ± 4.4 (−)</td>
<td>84.4 ± 9.9 (−)</td>
<td>68.3 ± 5.3 (−)</td>
<td>63.0 ± 2.7 (−)</td>
<td>57.8 ± 4.7 (−)</td>
<td>11.3 ± 0.9 (−)</td>
<td>6.8 ± 0.7 (−)</td>
<td>4.2 ± 0.5 (−)</td>
<td>10.4 ± 2.7 (−)</td>
</tr>
<tr>
<td>( I_A )</td>
<td>62.4 ± 4.6 (−)</td>
<td>89.3 ± 3.5 (−)</td>
<td>82.7 ± 10.8 (−)</td>
<td>70.9 ± 5.6 (−)</td>
<td>57.7 ± 3.5 (=)</td>
<td>61.1 ± 7.0 (−)</td>
<td>9.4 ± 0.6 (−)</td>
<td>7.4 ± 0.4 (−)</td>
<td>4.6 ± 0.6 (−)</td>
<td>10.3 ± 2.4 (−)</td>
</tr>
<tr>
<td>( I_{Cl} )</td>
<td>62.3 ± 4.3 (−)</td>
<td>91.7 ± 3.6 (−)</td>
<td>86.4 ± 7.5 (−)</td>
<td>73.0 ± 5.1 (−)</td>
<td>56.0 ± 3.0 (−)</td>
<td>58.4 ± 8.4 (−)</td>
<td>9.3 ± 0.6 (−)</td>
<td>7.2 ± 0.5 (−)</td>
<td>4.5 ± 0.7 (−)</td>
<td>10.2 ± 2.5 (−)</td>
</tr>
<tr>
<td>( I_{cov} )</td>
<td>63.5 ± 4.7 (−)</td>
<td>78.2 ± 7.6 (−)</td>
<td>90.9 ± 2.1 (−)</td>
<td>73.0 ± 6.3 (−)</td>
<td>50.6 ± 2.2 (−)</td>
<td>56.9 ± 6.1 (−)</td>
<td>10.1 ± 0.8 (−)</td>
<td>6.8 ± 0.5 (−)</td>
<td>4.0 ± 0.4 (−)</td>
<td>9.1 ± 1.4 (−)</td>
</tr>
<tr>
<td>( \text{MIC} )</td>
<td>64.4 ± 4.9 (−)</td>
<td>75.9 ± 9.6 (−)</td>
<td>84.9 ± 10.5 (−)</td>
<td>71.2 ± 5.5 (−)</td>
<td>45.1 ± 6.8 (−)</td>
<td>56.1 ± 8.8 (−)</td>
<td>8.8 ± 0.7 (−)</td>
<td>6.8 ± 0.5 (−)</td>
<td>5.5 ± 0.7 (−)</td>
<td>10.1 ± 1.9 (−)</td>
</tr>
<tr>
<td>( \text{GMIC} )</td>
<td>66.6 ± 4.2 (−)</td>
<td>89.0 ± 3.8 (−)</td>
<td>90.3 ± 3.0 (−)</td>
<td>68.8 ± 3.7 (−)</td>
<td>53.5 ± 6.6 (−)</td>
<td>57.2 ± 4.2 (−)</td>
<td>10.4 ± 0.7 (−)</td>
<td>7.3 ± 0.5 (−)</td>
<td>5.7 ± 0.7 (−)</td>
<td>12.5 ± 2.8 (−)</td>
</tr>
<tr>
<td>( \text{MD} )</td>
<td>35.7 ± 8.6 (−)</td>
<td>47.4 ± 1.7 (−)</td>
<td>75.2 ± 14.8 (−)</td>
<td>79.5 ± 11.2 (−)</td>
<td>37.5 ± 4.8 (−)</td>
<td>39.6 ± 6.2 (−)</td>
<td>3.9 ± 0.4 (−)</td>
<td>2.3 ± 0.4 (−)</td>
<td>1.6 ± 0.1 (−)</td>
<td>8.8 ± 1.5 (−)</td>
</tr>
<tr>
<td>( \text{ACE} )</td>
<td>67.4 ± 5.0 (−)</td>
<td>88.5 ± 6.2 (−)</td>
<td>84.4 ± 10.7 (−)</td>
<td>75.6 ± 7.0 (−)</td>
<td>62.0 ± 2.0 (+)</td>
<td>53.8 ± 7.0 (−)</td>
<td>9.9 ± 0.8 (−)</td>
<td>7.4 ± 0.4 (−)</td>
<td>6.1 ± 0.6 (−)</td>
<td>11.0 ± 2.4 (−)</td>
</tr>
<tr>
<td>( \text{HSIC} )</td>
<td>64.8 ± 3.7 (−)</td>
<td>87.7 ± 3.8 (−)</td>
<td>91.5 ± 12 (−)</td>
<td>68.2 ± 2.4 (−)</td>
<td>51.9 ± 2.3 (−)</td>
<td>64.5 ± 5.7 (−)</td>
<td>9.9 ± 0.9 (−)</td>
<td>7.5 ± 0.7 (−)</td>
<td>7.1 ± 1.1 (−)</td>
<td>11.7 ± 2.4 (−)</td>
</tr>
<tr>
<td>( I_{\text{mean}} )</td>
<td>46.8 ± 2.0 (−)</td>
<td>90.2 ± 0.0 (−)</td>
<td>91.6 ± 0.7 (−)</td>
<td>69.6 ± 3.4 (−)</td>
<td>33.0 ± 1.4 (−)</td>
<td>65.4 ± 5.6 (−)</td>
<td>8.1 ± 0.5 (−)</td>
<td>4.6 ± 0.2 (−)</td>
<td>2.7 ± 0.2 (−)</td>
<td>7.5 ± 0.9 (−)</td>
</tr>
</tbody>
</table>
3.5.3 Feature Filtering for Regression

In this section, we evaluate the performance of RIC and the other statistics as feature filtering techniques. A dependency measure $D$ can be used to rank the $m$ features $X_i$ on a regression task based on their prediction ability for the target variable $Y$. Only the top $m^*$ features according to $D$ are used to build a regressor for $Y$. Table 3.5 shows the average correlation coefficient between the predicted and the actual target value using the top $m^* \leq 10$ features using a $k$-NN regressor (with $k = 3$). Each value is obtained by averaging 3 random trials of 10-fold cross-validation for each $m^* \leq 10$.

The data sets collected have at least 10 features and in the case of $n > 1000$ records, we randomly sampled 1000 records to speed up the running time of dCorr, HSIC, $I_{KDE}$, $I_{mean}$, MIC and GMIC with default parameters. Records with missing values were deleted. We analyzed the performance on 8 data sets: 5 from the UCI machine learning repository\textsuperscript{10}, the Pole telecommunication data\textsuperscript{11}, and 2 data sets Qsar and Qsar2 from the website of the 3rd Strasbourg Summer School on Chemoinformatics\textsuperscript{12}. The list of data sets used is shown in Table 3.3.

![Figure 3.16: Average rank of measures when the task is maximizing the correlation coefficient between the predicted and the target value of a $k$-NN regressor. The $k$-NN regression is built on top of $m^*$ features. Results were averaged across $m^* \leq 10$ and all data sets.](http://ics.uci.edu/~mlearn)

\textsuperscript{10} http://ics.uci.edu/~mlearn
\textsuperscript{11} http://tunedit.org/
\textsuperscript{12} http://infochim.u-strasbg.fr/
As in section 3.5.2 we use RIC with parameters $D_{\text{max}} = \lfloor \sqrt{n} \rfloor$ to avoid low density grids that are better suited for testing of independence tasks. Overall, as can be observed from Figure 3.16, RIC performs consistently well on average. RIC is also particularly useful when the number of features $m$ is high and especially when their relationships to the target variable $Y$ are noisy. These represent the most challenging scenarios as can be justified by the low correlation coefficient achievable using the selected features, e.g., on the Pyrim and Triazines data sets. We also note the good performance of RIC on data sets where there are features that can take only a predefined number of values: e.g., discrete numerical features. Pole, Qsar, and Qsar2 include these type of features. For such features it is very difficult to either optimize a kernel or a grid size or find the optimal data transformation to obtain the maximal correlation with ACE, which explains the less competitive performance of HSIC, $I_{\text{KDE}}$, $I_{\text{A}}$, $I_{\text{mean}}$, and ACE. RIC is not affected by this problem as there is no optimization and grids are generated at random.
Table 3.5: Correlation coefficient between the predicted and actual target value on 8 data sets using kNN \((k = 3)\). The values are the mean correlation coefficient across the kNN regressors built on top of \(m^*\) features with \(m^* \leq 10\). \(n\) number of records; \(m\) number of features. Each cell shows mean correlation coefficient ± std and either (+), ( = ), or (−) means statistically greater, equal, or smaller according to the 1-sided paired \(t\)-test \((\alpha = 0.05)\) than RIC results.

<table>
<thead>
<tr>
<th>((n, m))</th>
<th>Pyrim</th>
<th>Bodyfat</th>
<th>Triazines</th>
<th>Wisconsin</th>
<th>Crime</th>
<th>Pole</th>
<th>Qsar</th>
<th>Qsar2</th>
</tr>
</thead>
<tbody>
<tr>
<td>RIC</td>
<td>(0.261\pm0.120)</td>
<td>(0.642\pm0.115)</td>
<td>\textbf{0.215}\pm0.120</td>
<td>(0.034\pm0.013)</td>
<td>(0.892\pm0.042)</td>
<td>(0.685\pm0.156)</td>
<td>(0.277\pm0.091)</td>
<td>\textbf{0.479}\pm0.053</td>
</tr>
<tr>
<td>dCort</td>
<td>(0.205\pm0.046(−))</td>
<td>(0.643\pm0.114(=))</td>
<td>(0.118\pm0.062(−))</td>
<td>(0.041\pm0.012(=))</td>
<td>(0.852\pm0.057(−))</td>
<td>(0.686\pm0.218(=))</td>
<td>\textbf{0.310}\pm0.025(+)</td>
<td>(0.382\pm0.130(−))</td>
</tr>
<tr>
<td>I_kDE</td>
<td>(0.231\pm0.068(−))</td>
<td>(0.635\pm0.117(−))</td>
<td>(0.148\pm0.095(−))</td>
<td>(0.039\pm0.012(=))</td>
<td>(0.614\pm0.047(−))</td>
<td>(0.686\pm0.221(=))</td>
<td>(0.291\pm0.029(=))</td>
<td>(0.424\pm0.151(−))</td>
</tr>
<tr>
<td>I_kNN</td>
<td>(0.216\pm0.051(−))</td>
<td>(0.639\pm0.116(−))</td>
<td>(0.098\pm0.035(−))</td>
<td>(0.038\pm0.011(=))</td>
<td>(0.893\pm0.051(−))</td>
<td>(0.621\pm0.134(−))</td>
<td>(0.300\pm0.094(+))</td>
<td>(0.423\pm0.025(−))</td>
</tr>
<tr>
<td>(r^2)</td>
<td>(0.264\pm0.064(=))</td>
<td>\textbf{0.644}\pm0.114(−)</td>
<td>(0.125\pm0.050(−))</td>
<td>(0.041\pm0.009(+))</td>
<td>(0.870\pm0.045(−))</td>
<td>(0.414\pm0.311(−))</td>
<td>(0.273\pm0.037(−))</td>
<td>(0.375\pm0.134(−))</td>
</tr>
<tr>
<td>RDC</td>
<td>(0.206\pm0.052(−))</td>
<td>(0.642\pm0.115(=))</td>
<td>(0.199\pm0.079(=))</td>
<td>(0.017\pm0.007(−))</td>
<td>(0.891\pm0.042(−))</td>
<td>(0.679\pm0.197(−))</td>
<td>(0.280\pm0.058(=))</td>
<td>(0.430\pm0.060(−))</td>
</tr>
<tr>
<td>(I_A)</td>
<td>(0.235\pm0.088(−))</td>
<td>(0.640\pm0.115(=))</td>
<td>(0.062\pm0.017(−))</td>
<td>(0.037\pm0.021(=))</td>
<td>(0.891\pm0.041(−))</td>
<td>(0.691\pm0.007(−))</td>
<td>(0.284\pm0.042(=))</td>
<td>(0.418\pm0.046(−))</td>
</tr>
<tr>
<td>(I_{el})</td>
<td>(0.190\pm0.068(−))</td>
<td>(0.640\pm0.116(=))</td>
<td>(0.171\pm0.053(−))</td>
<td>(0.036\pm0.015(=))</td>
<td>(0.889\pm0.041(−))</td>
<td>(0.693\pm0.156(−))</td>
<td>(0.278\pm0.104(−))</td>
<td>(0.429\pm0.028(−))</td>
</tr>
<tr>
<td>(I_{er})</td>
<td>(0.249\pm0.064(=))</td>
<td>(0.641\pm0.115(=))</td>
<td>(0.188\pm0.097(−))</td>
<td>(0.033\pm0.007(=))</td>
<td>(0.859\pm0.059(−))</td>
<td>(0.661\pm0.145(−))</td>
<td>(0.264\pm0.085(−))</td>
<td>(0.441\pm0.046(−))</td>
</tr>
<tr>
<td>MIC</td>
<td>(0.186\pm0.072(−))</td>
<td>(0.642\pm0.114(=))</td>
<td>(0.051\pm0.023(−))</td>
<td>(0.010\pm0.009(=))</td>
<td>(0.776\pm0.040(−))</td>
<td>(0.694\pm0.156(=))</td>
<td>(0.293\pm0.030(−))</td>
<td>(0.448\pm0.039(−))</td>
</tr>
<tr>
<td>GMIC</td>
<td>(0.206\pm0.069(−))</td>
<td>(0.634\pm0.118(−))</td>
<td>(0.141\pm0.056(−))</td>
<td>(0.026\pm0.005(−))</td>
<td>(0.803\pm0.055(−))</td>
<td>(0.734\pm0.179(−))</td>
<td>(0.292\pm0.058(=))</td>
<td>(0.468\pm0.054(−))</td>
</tr>
<tr>
<td>MID</td>
<td>(0.241\pm0.167(=))</td>
<td>(0.605\pm0.137(=))</td>
<td>(0.160\pm0.062(=))</td>
<td>\textbf{0.047}\pm0.030(+)</td>
<td>(0.178\pm0.047(−))</td>
<td>\textbf{0.808}\pm0.215(−)</td>
<td>(0.194\pm0.130(−))</td>
<td>(0.186\pm0.074(−))</td>
</tr>
<tr>
<td>ACE</td>
<td>(0.221\pm0.051(−))</td>
<td>(0.641\pm0.113(=))</td>
<td>(0.111\pm0.073(−))</td>
<td>(0.011\pm0.008(=))</td>
<td>\textbf{0.894}\pm0.042(−)</td>
<td>(0.000\pm0.000(−))</td>
<td>(0.270\pm0.056(=))</td>
<td>(0.439\pm0.023(−))</td>
</tr>
<tr>
<td>HSIC</td>
<td>(0.174\pm0.068(−))</td>
<td>(0.638\pm0.116(−))</td>
<td>(0.057\pm0.063(−))</td>
<td>(0.028\pm0.011(=))</td>
<td>(0.853\pm0.046(−))</td>
<td>(0.000\pm0.000(−))</td>
<td>(0.001\pm0.001(−))</td>
<td>(0.000\pm0.000(−))</td>
</tr>
<tr>
<td>(I_{mean})</td>
<td>(0.178\pm0.073(−))</td>
<td>(0.636\pm0.117(−))</td>
<td>(0.073\pm0.076(−))</td>
<td>(0.034\pm0.011(=))</td>
<td>(0.853\pm0.046(−))</td>
<td>(0.000\pm0.000(−))</td>
<td>(0.001\pm0.001(−))</td>
<td>(0.001\pm0.000(−))</td>
</tr>
</tbody>
</table>
3.5.4 Run Time Comparison

Here we compare the running times of each measure in Table 3.2 varying the amount of records $n$ on two independent variables $X$ and $Y$ uniformly distributed. The average run time on 30 simulations is shown in Figure 3.17a for each measure. RIC is very competitive in terms of speed and can be grouped with the fastest measures: $I_{ef}$, $I_{ew}$, $I_{kNN}$, $I_A$, $r^2$, MID, ACE, and RDC. On the other hand, dCorr, $I_{KDE}$, HSIC, $I_{mean}$, MIC, GMIC, and TIC$_e$ appear to be slower according to the implementations discussed at the beginning of Section 3.5 and the parameter setting from Table 3.2. As discussed in the related work section, different parameter setting yield more competitive running times for some measures. For example, TIC$_e$ can obtain close to linear complexity in the number of records if $\alpha = 0.2$. In our analysis, we chose to set $\alpha = 0.65$ because it is the choice that allows us to maximize power when testing for independence under additive noise.

![Figure 3.17: Running time in seconds (best viewed in color).](image)

Figure 3.17a shows the running time for RIC with default parameters $K_r = 20$ and $D_{max} = \lfloor \sqrt{n} \rfloor$. Similarly to other measures, the running time for RIC depends to its parameter setting. Figure 3.17b shows the different time taken by RIC on $n = 10^3$ records.
3.6 experiments on dependency between two sets of variables

In this section, we perform comparisons between the performance of measures which quantify the dependency between two sets of $p$ variables $X$ and $q$ variables $Y$. This is different from finding a subset of variables that are significantly correlated. In that case, new advances in that area yielded interesting measures to compare [Nguyen et al., 2014; Nguyen and Vreeken, 2015]. In our paper, we compare the measures discussed in Table 3.2. The Pearson’s correlation coefficient, ACE, $I_A$, MIC, GMIC, and MID are not applicable in these scenarios and there is no straightforward method to extend them to sets of variables available in literature.

3.6.1 Identification of Multi-variable Noisy Relationships

Here we extend the experiments of section 3.5.1 to sets of variables $X$ and $Y$. In particular, we test the power in identifying relationships between $X$ with $p = 3$ variables and a single variable $Y$ with the additive noise model. In order to use the same 12
relationships displayed in Figure 3.7a, we map the set of features $\mathbf{X}$ on a single feature $X' = \frac{X_1 + \ldots + X_p}{p}$ and obtain $Y$ according a given relationship plus additive noise. Figure 3.18 shows an example of a quadratic relationships between $Y$ and $\mathbf{X} = (X_1, X_2)$ ($p = 2$) with additive noise.

![Figure 3.18: Example of a quadratic relationship between $Y$ and $\mathbf{X} = (X_1, X_2)$ on the left plot. The plot on the right shows how $Y$ is obtained through the mapping of $\mathbf{X}$ into $X' = \frac{X_1 + X_2}{2}$.](image)

We fix the number of variables $p = 3$ for $\mathbf{X}$ because some measures require specific tuning in regards to the number of variables considered. For example, the most straightforward way to extend the discretization based estimators of mutual information $I_{\text{ew}}$ and $I_{\text{ef}}$ is to independently discretize all the variables in each set. This requires carefully choosing the number of discretization bins for each variable in $\mathbf{X}$ and each variable in $\mathbf{Y}$. If the same number of bins $D_X$ is chosen for all the variables in $\mathbf{X}$ and the same number of bins $D_Y$ is chosen for all the variables in $\mathbf{Y}$, it is possible to end up with as many as $D_X^p \cdot D_Y^q$ total bins. This issue makes it practically infeasible to use $I_{\text{ew}}$ and $I_{\text{ef}}$ in high $p, q$ scenarios. Given this limitation of the discretization based estimators of mutual information, we also made use of a multi-variable discretization approach of the set of variables $\mathbf{X}$ which allows a more sensible choice of the total number of bins. Even if methods for multi-variable discretization are available in literature [Garcia et al., 2013; Dougherty et al., 1995] to our knowledge there is no extensive survey about the performance of estimation of mutual information with multi-variable approaches. Therefore, we chose to discretize $\mathbf{X}$ and $\mathbf{Y}$ with the clustering algorithm $k$-means and then compute the mutual information. We name this measure $I_{k-\text{mean}}$. This allows us to choose the total number of bins (clusters) to be produced. In our case, where $p = 3$ and $q = 1$ we
chose compute $I_{ew}$ and $I_{ef}$ fixing $D_Y = 5$ and compute $D_X$ in order to limit the number of total bins in regards to the number $n$ of data points: $D_X^P \cdot D_Y \leq \frac{n}{5} \Rightarrow D_X = \left\lfloor \frac{\log n}{25} \right\rfloor$. When $n = 320$, $D_X = 2$.

Power at level $\alpha = 0.05$ of discrimination between complete noise and noisy relationship for each relationship type presented in the paper is shown in Figure 3.21. Parameter tuning for this task can be found in Figure 3.20. All the measures but $I_{KDE}$ find similar optimal parameters when comparing single variables. $I_{KDE}$ requires larger kernel width when comparing sets of variables. Furthermore, RDC seem to be little sensitive to the parameter $s$. In that case we optimized $s_X$ and $s_Y$ independently with $\tilde{p}$ where:

$$s_X = \frac{1}{6p} \cdot \tilde{p} = \frac{1}{6\frac{3}{2}} \cdot \tilde{p}$$

$\text{and}$

$$s_Y = \frac{1}{6q} \cdot \tilde{p} = \frac{1}{6\frac{1}{2}} \cdot \tilde{p}.$$  

Regarding RIC, in order to have full control on the number of bins produced, we compared the multi-variable discretization approach that uses random seeds as described in Algorithm 3.4. More specifically, we fixed the number of random seeds to $\left\lfloor \sqrt{n/c} \right\rfloor$ given that also choosing the number of random seeds at random might result in configurations with as few as 2 seeds, which strongly deteriorates the discrimination ability of mutual information on multiple variables. The parameter $c$ for RIC that maximizes the power on average is $c = 6$ which generates $\left\lfloor \sqrt{n/6} \right\rfloor$ seeds. This setting is very similar to the optimal parameter setting found for testing for independence between variables under additive noise in Section 3.5.1. Most of the measures obtain similar optimal parameters to the ones obtained when testing for independence between variables. Just $I_{KDE}$ seems to require even larger kernel width when comparing sets of variables.

Figures 3.19 show average rank of each measure across different relationships. RIC with $D_{max} = \left\lfloor \sqrt{n/6} \right\rfloor$ and $K_r = 200$ looks more competitive than all other measures but $I_{KDE}$. Therefore, the strongest competitor seems to be $I_{KDE}$ that with a careful choice of kernels achieves very good performance on simple relationships such as the linear, quadratic, and cubic. We also can see that the discretization based estimators of mutual information do not do a good job because they dramatically fail on some data set. Moreover, $I_{k-means}$ which produces the same number of bins as RIC has clearly lower performance than the latter. The superior performance of RIC is thus due to the randomization.
Figure 3.19: Average rank across relationships for the multi-variable additive noise model. RIC with $D_{\text{max}} = \lfloor \sqrt{n}/6 \rfloor$ and $K_r = 200$ shares the top position with $I_{KDE}$.

Figure 3.20: Parameter tuning to maximize the power of each measure on average for the additive noise model when comparing sets of variables. These plots show the average area under power curve and their average across relationship types.
3.6 Experiments on Dependency Between Two Sets of Variables

Figure 3.21: Power in the additive noise scenario when comparing sets of variables with optimal parameters (best viewed in color).
3.6.2 Feature Selection for Regression

We also tested multi-variable measures of dependency in the task of feature selection using a similar framework to section 3.5.3. Rather than filtering the features according to their individual importance to the target variable $Y$, we proceed by forward selection. The optimal set of $p$ features according to a dependency measure is identified by finding the best set of features $X = X^{p-1} \cup X_i$, with $X^{p-1}$ representing the set chosen at the previous iteration of forward selection and $X_i$ chosen among the possible $m - (p - 1)$ features of a data set. A multi-variable dependency measure can be fully employed in this case because we require to compute the dependency between $X$ features and the target variable $Y$ at each step of the iteration.

As in section 3.5.2 and 3.5.3 we use RIC with parameters $D_{\text{max}} = \lfloor \sqrt{n} \rfloor$ to avoid low density grids that are better suited for testing of independence tasks under additive noise. We use the random seeds discretization approach of Algorithm 3.4 with a fixed number of random seeds. We also choose to fix $D_X = 2$ and $D_Y = 5$ for the naive discretization based estimators of mutual information. Average results for all the measures are shown in Figure 3.22 and a table with detailed comparisons is presented in Table 3.6. We notice that the ranking by performance of classifier changes from the one obtained using the feature filtering approach, although RIC again shows competitive performance against the other approaches. All estimators of mutual information lose positions except for the $I_{\text{KDE}}$ kernel based estimator. It seems that on multiple variables kernels are more effective than in the univariate scenario. Indeed, HSIC also gains a few positions. RDC’s average performance stays the same and it still gets outperformed by dCorr. dCorr performs really well when computed on sets of variables. As previously noted, even in this case RIC outperforms $I_k\text{-means}$ and this result is due to the randomized approach.
3.6 Experiments on Dependency Between Two Sets of Variables

Figure 3.22: Average rank when the target is maximization of the correlation coefficient between the predicted and the target value for a $k$NN regressor. The $k$NN regression is built on top of $m^*$ features chosen by forward selection. Results are averaged across $m^* \leq 10$ and all data sets.
Table 3.6: Correlation coefficient between the predicted and actual target value on 8 data sets using kNN (k = 3). The values are the mean correlation coefficient across the kNN regressors built on top of m* features selected with forward selection with m* ≤ 10. n number of records; m number of features. Each cell shows mean correlation coefficient ± std and either (+), (=), or (−) means statistically greater, equal, or smaller according to the 1-sided paired t-test (α = 0.05) than RIC results.

<table>
<thead>
<tr>
<th>(n, m)</th>
<th>Pyrim</th>
<th>Bodyfat</th>
<th>Triazines</th>
<th>Wisconsin</th>
<th>Crime</th>
<th>Pole</th>
<th>Qsar</th>
<th>Qsar2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RIC</td>
<td>0.291±0.122</td>
<td>0.668±0.106</td>
<td>0.125±0.045</td>
<td>0.050±0.014</td>
<td><strong>0.931</strong>±0.056</td>
<td>0.778±0.235</td>
<td>0.347±0.126</td>
<td>0.401±0.026</td>
</tr>
<tr>
<td>Ione</td>
<td>0.288±0.100(=)</td>
<td>0.523±0.158(−)</td>
<td>0.104±0.064(−)</td>
<td>0.073±0.026(+ )</td>
<td>0.704±0.120(−)</td>
<td>0.743±0.204(−)</td>
<td>0.330±0.068(= )</td>
<td>0.394±0.014(= )</td>
</tr>
<tr>
<td>LD</td>
<td>0.248±0.089(−)</td>
<td>0.619±0.113(−)</td>
<td>0.063±0.034(−)</td>
<td>0.032±0.004(−)</td>
<td>0.675±0.120(−)</td>
<td>0.698±0.253(−)</td>
<td><strong>0.358</strong>±0.095(= )</td>
<td>0.345±0.050(−)</td>
</tr>
<tr>
<td>I mean</td>
<td><strong>0.373</strong>±0.100(+)</td>
<td>0.663±0.108(=)</td>
<td>0.093±0.047(−)</td>
<td>0.038±0.008(−)</td>
<td>0.781±0.016(−)</td>
<td>0.763±0.198(= )</td>
<td>0.22±0.083(−)</td>
<td>0.329±0.030(−)</td>
</tr>
<tr>
<td>I KDE</td>
<td>0.270±0.084(=)</td>
<td>0.628±0.121(−)</td>
<td>0.073±0.044(−)</td>
<td>0.042±0.013(= )</td>
<td>0.785±0.140(= )</td>
<td>0.764±0.260(= )</td>
<td>0.321±0.071(= )</td>
<td>0.411±0.146(= )</td>
</tr>
<tr>
<td>I kNN</td>
<td>0.261±0.073(=)</td>
<td>0.637±0.117(−)</td>
<td>0.049±0.021(−)</td>
<td>0.027±0.012(−)</td>
<td>0.510±0.238(−)</td>
<td><strong>0.804</strong>±0.217(+)</td>
<td>0.278±0.061(−)</td>
<td>0.391±0.030(−)</td>
</tr>
<tr>
<td>I mean</td>
<td>0.293±0.122(=)</td>
<td>0.626±0.019(−)</td>
<td>0.107±0.035(−)</td>
<td>0.032±0.015(−)</td>
<td>0.905±0.140(−)</td>
<td>0.425±0.102(−)</td>
<td>0.182±0.102(−)</td>
<td>0.232±0.053(−)</td>
</tr>
<tr>
<td>dCorr</td>
<td>0.275±0.063(=)</td>
<td><strong>0.672</strong>±0.104(=)</td>
<td>0.104±0.054(−)</td>
<td><strong>0.073</strong>±0.016(+ )</td>
<td>0.920±0.053(−)</td>
<td>0.772±0.279(= )</td>
<td>0.352±0.054(= )</td>
<td>0.362±0.129(−)</td>
</tr>
<tr>
<td>RDC</td>
<td>0.333±0.146(+)</td>
<td>0.550±0.172(−)</td>
<td><strong>0.173</strong>±0.046(+ )</td>
<td>0.026±0.013(−)</td>
<td>0.761±0.115(−)</td>
<td>0.715±0.203(−)</td>
<td>0.296±0.071(= )</td>
<td><strong>0.415</strong>±0.046(+ )</td>
</tr>
<tr>
<td>HSIC</td>
<td>0.327±0.104(=)</td>
<td>0.653±0.112(−)</td>
<td>0.084±0.027(−)</td>
<td>0.057±0.021(= )</td>
<td>0.775±0.000(−)</td>
<td>0.787±0.214(= )</td>
<td>0.361±0.046(= )</td>
<td>0.144±0.041(−)</td>
</tr>
</tbody>
</table>
3.7 CONCLUSIONS

In this chapter we presented the Randomized Mutual Information (RIC), an information theoretic measure of dependency between two sets random variables $X$ and $Y$, that makes use of an ensemble of random grids. Our theoretical analysis justifies the benefits of having a low-variance estimator of mutual information based on grids for the task of ranking relationships, where systematic biases cancel each other out. By reducing the estimation variance of mutual information with grids, RIC is extremely competitive for ranking different relationships. We experimentally demonstrated its strong performance on univariate $X$ and $Y$ on the task of discrimination of noisy relationships, network inference and feature filtering for regression. We have shown that RIC can be extended to multivariate $X$ and $Y$ with a subtle discretization scheme. We recommend RIC’s use with the default parameters: maximum number of random cut-offs $D_{\text{max}} = \lfloor \sqrt{n} \rfloor$ and number of random discretizations $K_r = 20$ for both $X$ and $Y$ in general applications. However, $D_{\text{max}}$ can be decreased when testing for independence under additive noise and $K_r$ can be increased to decrease the variance, at the cost of computational time.
A FRAMEWORK TO ADJUST DEPENDENCY MEASURE ESTIMATES FOR CHANCE

Outline
Quantification and ranking of dependencies estimated on finite samples are challenging. Dependency estimates are not equal to 0 when variables are independent, cannot be compared if computed on different sample size, and they are inflated by chance on variables with more categories. In this chapter, we introduce a framework to adjust dependency measures for chance to obtain higher interpretability in quantification tasks, and to obtain higher accuracy in ranking tasks. To achieve these goals, we make use of the distribution of the dependency measure estimator under the null hypothesis of independent variables. In Section 4.2, we discuss previous attempts of the use of this null for adjusting dependency measure estimates. In Section 4.3, we present the framework to adjust dependency estimates for quantification and in Section 4.4 we introduce the framework to adjust dependency estimates for ranking.

4.1 Introduction

Dependency measures $\mathcal{D}(X, Y)$ are employed in data mining and machine learning to assess the strength of the dependency between two continuous or categorical variables $X$ and $Y$. If the variables are continuous, we can use Pearson’s correlation to detect linear

dependencies, or use more sophisticated measures, such as the Maximal Information Coefficient (MIC) [Reshef et al., 2011] to detect non-linear dependencies as discussed in Section 2.3. If the variables are categorical we can use the well known mutual information (a.k.a. information gain) or the Gini gain [Kononenko, 1995]; see Section 2.2 for reference.

Nonetheless, there exist a number of problems when the dependency $D(X, Y)$ is estimated with $\hat{D}(S_n|X, Y)$ on a data sample $S_n$ of $n$ data points: a) even if the population value $D(X, Y) = 0$ when $X$ and $Y$ are statistically independent, estimates have a high chance to be bigger than 0 when $n$ is finite; b) when comparing pairs of variables which share the same fixed population value $D(X, Y)$, estimates are still dependent on the sample size $n$ and the number of categories of $X$ and $Y$. These issues diminish the utility of dependency measures on quantification tasks. For example, MIC was proposed in Reshef et al. [2011] as a proxy of the amount of noise on the functional dependence between $X$ and $Y$: it should “provide a score that roughly equals the coefficient of determination $R^2$ of the data relative to the regression function”, which is 0 under complete noise and 1 in noiseless scenarios. Nonetheless, MIC is not equal to 0 under complete noise, and MIC values are not comparable if computed on samples of different size $n$ because of the use of different data sets or in the case of variables with missing values:

**Example 4.1.** Given two uniform and independent variables $X$ and $Y$ in $[0, 1]$, the population value of MIC is 0 but the estimates $\text{MIC}(S_{20}|X, Y)$ on 20 data points are higher than $\text{MIC}(S_{80}|X, Y)$ on 80 data points. On average, they achieve the values of 0.36 and 0.25 respectively. The user expects this value to be 0. The following box plots show estimates for 10,000 simulations.

![Box plots showing estimates for MIC](image)

The example above shows that the estimated MIC does not have zero baseline for finite samples. The zero baseline property is well known in the clustering community [Vinh
et al., 2009], nonetheless this property does not hold for many dependency measures used in data mining.

Problems also arise when ranking dependencies on a finite data sample. For example, if Gini gain is used to rank the dependency between variables to the target class in random forest [Breiman, 2001], variables with more categories have more chances to be ranked higher:

**Example 4.2.**

Given a variable $X_1$ with two categories and a variable $X_2$ with one more category which are both independent of the target binary class $Y$, both the population value of Gini gain between $X_1$ and $Y$, and the population value between $X_2$ and $Y$ are equal to 0. However, when Gini gain is estimated on 100 data points, the probability of Gini($S_{100}|X_2, Y$) being greater than Gini($S_{100}|X_1, Y$) is equal to 0.7. The user expects 0.5 given that $X_1$ and $X_2$ are equally unpredictive to $Y$. This bias increases the chance to select $X_2$ to split a node in a decision tree.

It is common practice to use the $p$-value of Gini gain to correct this selection bias [Dobra and Gehrke, 2001]. Nonetheless, we will shortly see that $p$-values are effective only when the population value of a dependency measure is 0.

In this chapter, we identify that the issues discussed in Example 4.1 and 4.2 are due to inflated estimates arising from finite samples. Statistical properties of the distribution of the dependency measure estimator $\hat{D}(S_n|X, Y)$ under independence of $X$ and $Y$ can be used to adjust these estimates. The challenge is to formalize a general framework to adjust dependency measure estimates which also addresses the shortcomings of the use of $p$-values. We make the following contributions:

- We identify common biases of dependency measure estimates due to finite samples;
- We propose a framework to adjust estimates $\hat{D}(S_n|X, Y)$ which is simple, yet applicable to many dependency measures because it only requires to use the distribution of the estimator when $X$ and $Y$ are independent;
- We experimentally demonstrate that our adjustments improve interpretability when quantifying dependency (e.g., when using MIC as a proxy of the amount of noise) and accuracy when ranking dependencies (e.g., when using Gini gain in random forests).

4.2 RELATED WORK

We saw in Examples 4.1 and 4.2 that when it comes to estimating dependency on data samples via $\hat{D}(S_n|X,Y)$, the interpretability of quantification and accuracy of ranking become challenging. We claim that both tasks can take advantage of the distribution of $\hat{D}$ under the following null hypothesis:

**Definition 4.1.** $\hat{D}_0(S_n|X,Y)$ is the distribution of $\hat{D}(S_n|X,Y)$ on a sample $S_n$ under the null hypothesis that $X$ is statistically independent of $Y$.

This null hypothesis is commonly exploited only in detection tasks where the distribution $\hat{D}_0(S_n|X,Y)$ is computed under the null and a $p$-value is computed to filter out false discoveries [Reshef et al., 2011]. Furthermore, this null can be used also to aid quantification and ranking. The challenges are to identify the distribution under the null for a particular dependency measure, and to employ it in a framework to perform adjustments to the estimates. Here we discuss the use of this null hypothesis in previous research.

4.2.1 Use of the Null for Quantification

To our knowledge the first instance of a systematic approach using the null distribution $\hat{D}_0$ to achieve interpretability in quantification was proposed in the 1960 with the $\kappa$ coefficient of inter-annotator agreement [Witten et al., 2011]. The amount of agreement $A(S_n|X,Y)$ (dependency) between two annotators $X$ and $Y$ on a sample of $n$ items as defined in Section 2.2.3 can be adjusted for chance by subtracting its expected value $E[A_0(S_n|X,Y)]$ under the null hypothesis of independence between annotators. The $\kappa$
coefficient is obtained by normalization via division of its maximum value \( \max A = 1 \) to obtain an adjusted dependency measure in the range \([0, 1]\):

\[
\kappa(S_n|X, Y) = \frac{A(S_n|X, Y) - E[A_0(S_n|X, Y)]}{1 - E[A_0(S_n|X, Y)]}
\] (4.1)

Measures of agreement between two clusterings \( X \) and \( Y \) might also be unfairly inflated just because of statistical fluctuations when estimated on a data set \( S_n \). Commonly used measures for clustering comparisons computed on contingency tables, such as the Rand Index (RI) defined in Section 2.2.2, show an increasing trend when the number of clusters increases even if clusterings are random and independent. This is thoroughly discussed in detail in Chapter 5. In Hubert and Arabie [1985] it was proposed to adjust RI for chance as follows:

\[
\text{ARI}(S_n|X, Y) = \frac{RI(S_n|X, Y) - E[RI_0(S_n|X, Y)]}{\max RI(S_n|X, Y) - E[RI_0(S_n|X, Y)]}
\] (4.2)

An analytical formula for the expected value \( E[RI_0(S_n|X, Y)] \) of RI under the null of random and independent clusterings is used to remove the baseline component of the measure, and \( \max RI(S_n|X, Y) \) is an upper bound for RI that acts as normalization factor. The model of randomness adopted to compute the expected value is the permutation model, also called the hypergeometric model: fixing the number of points for each cluster, partitions are generated uniformly and randomly via permutations of records. Under this assumption, the distribution of the measure estimator \( \hat{D} \) can be computed and thus so can the expected value. It has been shown that this hypothesis behaves well in practical scenarios and it was recently employed to compute the expected value of MI in Vinh et al. [2009].

We argue that the approach of adjustment for chance to correct measure baseline should be applied to many other dependency measures estimators \( \hat{D} \) because it improves interpretability by guaranteeing a zero baseline to \( \hat{D} \). Moreover, we will shortly see that it helps in comparing estimates on different samples \( S_n \).
4.2.2 Use of the Null for Ranking

The problem of exaggerated dependency between two variables $X$ and $Y$ due to chance has been also extensively studied in the decision tree literature. For each internal node in a decision tree, the best partitioning according to a feature $X$, termed split, is selected in accordance with a splitting criterion $\hat{D}$ that quantifies its predictiveness towards the target classification $Y$. As discussed in Section 2.2.4, splitting criteria are in fact dependency measures between a feature $X$ and the class $Y$ estimated on a contingency table. Since the very first implementation of decision trees, ad hoc methods to reduce the bias toward selection of splits with many values have involved normalization [Quinlan, 1993]: gain ratio in Eq. (2.20) is an example of normalization for the information gain. However, even with normalization, partitions with higher cardinality are more preferred [White and Liu, 1994]. A key pitfall pointed out for splitting criteria is the lack of statistical significance concepts. This was formalized in Dobra and Gehrke [2001] where it has been proven that “a $p$-value of any split criterion is a nearly unbiased criterion”.

On the other hand, the use of the $p$-value is controversial. It has indeed been claimed that the $p$-value based on the chi-square distribution for splitting criteria such as the $G$-statistic “is not able to distinguish between more and less informative attributes” Kononenko [1995]. This behavior is mainly due to computer precision limitations when computing the $p$-value for informative features to the class.

In this chapter, we introduce a type of adjustment for chance by statistical standardization of $\hat{D}$. A standardized measure can discriminate between variables $X$ and $Y$ that show high dependency better than a $p$-value because it is less prone to computer precision errors. Moreover, we identify other shortcomings of $p$-values: they are unbiased only under the null hypothesis and not unbiased in general. Indeed in the next sections, we will see that their use actually yields bias towards variables induced on bigger $n$ or with fewer categories. This behavior has been overlooked in the decision tree community.

Unbiased ranking of proportion estimates on different amount of points $n$ is very important in websites such as Reddit¹, IMDB², and Amazon³. In this scenario, items

¹ https://www.reddit.com/
² http://www.imdb.com/
³ http://www.amazon.com/
4.3 Adjusting Estimates for Quantification

(forum posts, movie reviews, or items to sell) have to be ranked according to the scores given by users that can be either positive/negative or on a scale from 1 to 5. In Reddit, forum posts which receive just a few number of positive votes \( n \) have more chances to be ranked higher than others: e.g. post \( p_1 \) which has a 2 positive votes out of 2 has 100% proportion of positive votes and might be ranked higher than post \( p_2 \) which has 98 positive votes out of 100 (98% proportion); nonetheless, we might prefer to rank \( p_2 \) higher than \( p_1 \) because it is more popular. In order to penalize small \( n \), Reddit uses the lower bound of the \( (1 - \alpha/2) \) confidence interval based on the distribution of the proportion of positive votes out the total [Miller]. This task shares some similarities with the task of ranking dependency estimates \( \hat{D} \). Nonetheless, the estimation of the distribution of \( \hat{D} \) is a difficult task and it is achievable only for some dependency measures and when strong hypothesis on \( X \) and \( Y \) are verified (e.g. normality). On the other hand, the distribution of \( \hat{D}_0 \) under the null of \( X \) and \( Y \) independent is much easier to estimate and this can be employed similarly to adjust biases on ranking tasks. In this chapter, we will introduce another adjustment for chance to improve accuracy on ranking dependency measure estimates which can be parametrized on a statistical significance parameter \( \alpha \) which can be tuned.

4.3 Adjusting Estimates for Quantification

To guarantee good interpretability in quantification tasks, dependency measure estimates should be equal to 0 on average when \( X \) and \( Y \) are independent, and their values should be comparable on average across different data samples of different size. This can be formalized as follows:

**Property 4.1** (Zero Baseline).

*If \( X \) and \( Y \) are independent then \( E[\hat{D}(S_n|X,Y)] = 0 \) for all \( n \).*

**Property 4.2** (Quantification Unbiasedness).

*If \( D(X_1,Y_1) = D(X_2,Y_2) \) then \( E[\hat{D}(S_n|X_1,Y_1)] = E[\hat{D}(S_m|X_2,Y_2)] \) for all \( n \) and \( m \).*
We saw in Example 4.1 that MIC does not satisfy either property. Therefore, we propose an adjustment that can be applied to MIC and in general to any dependency estimator $\hat{D}$.

**Definition 4.2** (Adjustment for Quantification).

$$A\hat{D}(S_n|X,Y) \triangleq \frac{\hat{D}(S_n|X,Y) - E[\hat{D}_0(S_n|X,Y)]}{\max \hat{D}(S_n|X,Y) - E[\hat{D}_0(S_n|X,Y)]}$$

is the adjustment of $\hat{D}(S_n|X,Y)$, where $\max \hat{D}(S_n|X,Y)$ and $E[\hat{D}_0(S_n|X,Y)]$ are respectively the maximum of $\hat{D}$, and its expected value under the null.

$A\hat{D}(S_n|X,Y)$ has always zero baseline (Property 4.1) being 0 on average when $X$ and $Y$ are independent, and attains 1 as maximum value. This adjustment can be applied to $r^2$ and MIC to increase their interpretability when they are used as proxies of the amount of noise in a linear relationship and a functional relationship respectively. We just have to identify their distributions on the sample $S_n$ under the null:

- $r^2_0(S_n|X,Y)$: follows a Beta distribution with parameters $\frac{1}{2}$ and $\frac{n-2}{2}$ [Giles];

- $\text{MIC}_0(S_n|X,Y)$: this distribution can be computed using $s = 1, \ldots, S$ Monte Carlo permutations $\text{MIC}_0^{(s)}$ of MIC computed on the sample $S_n^0 = \{(x_{\sigma_x(k)}, y_{\sigma_y(k)})\}$ obtained by permutations of the original sample $S_n$; $\sigma_x(k)$ and $\sigma_y(k)$ are the permuted indexes of the points $x_k$ and $y_k$ respectively. Then, the expected value of MIC under the null can be estimated with:

$$E\text{MIC}_0 = \frac{1}{S} \sum_{s=1}^{S} \text{MIC}_0^{(s)}.$$  \hspace{1cm} (4.3)

Therefore the adjusted Pearson’s correlation squared $r^2$ and the adjusted MIC are:

$$Ar^2(S_n|X,Y) = \frac{r^2(S_n|X,Y) - \frac{1}{n-1}}{1 - \frac{1}{n-1}},$$  \hspace{1cm} (4.4)

and

$$\text{AMIC}(S_n|X,Y) = \frac{\text{MIC}(S_n|X,Y) - \text{EMIC}_0}{1 - \text{EMIC}_0},$$  \hspace{1cm} (4.5)
where $E[r_0^2(S_n|X,Y)] = \frac{1}{n-1}$ and $\text{EMIC}_0 = \frac{1}{n} \sum_{s=1}^{S} \text{MIC}_0^{(s)}$. $\text{EMIC}_0$ converges to $E[\text{MIC}_0(S_n|X,Y)]$ at the limit of infinite permutations. However, good estimation accuracy can be obtained even with few permutations because of the law of large numbers [Good, 2005]. Moreover, our $\text{Ar}^2$ turns out to be equal to the adjusted coefficient of determination $\bar{R}^2$ [Giles] in linear regression taking into account different sample sizes $n$ and more explanatory variables. This is interesting because $\text{Ar}^2$ and $\bar{R}^2$ are obtained using different techniques. In the next section, we will see how our adjustments satisfy Property 4.1 and Property 4.2.

4.3.1 Experiments with Pearson Correlation and MIC

We aim to experimentally verify the zero baseline Property 4.1 and that our adjustment in Definition 4.2 enables better interpretability. We generate a linear relationship between a uniformly distributed $X$ in $[0,1]$ and $Y$ on $n = 30$ points adding different percentages of white noise. We compare $r^2$ and $\text{Ar}^2$. Each white noise level is obtained by substituting a given percentage of points from the relationship and assigning to the $Y$ coordinate a random value in $[0,1]$. Figure 4.1 shows the average $r^2$ and $\text{Ar}^2$ for 2,000 simulated relationships with a given percentage of white noise: $r^2$ is not zero on average when the amount of noise is 100% (last plot on the right). On the other hand, $\text{Ar}^2$ is very close to zero when there is complete noise and it fully exploits its range of values from one to zero, mapping the domain from 0% to 100% noise. This yields more interpretability and enables $\text{Ar}^2$ to be used as a proxy to quantify the amount of noise in linear relationships.

Similarly, we generated a linear, a quadratic, a cubic, and a 4th root relationship between $X$ and $Y$ in $[0,1] \times [0,1]$ on $n = 60$ points with different levels of noise to compare MIC and AMIC. Figure 4.2 shows that the value of MIC computed with default parameters [Reshef et al., 2011], is about 0.26 on average for complete noise. AMIC computed with $S = 30$ permutations, is instead very close to zero and it exploits better its range of values $[0,1]$. AMIC is more interpretable than MIC and can be used more intuitively as a proxy for the amount of noise in a functional relationship.
Figure 4.1: Average value of \( r^2 \) and \( \Delta r^2 \) for different percentages of white noise. Linear relationship between \( X \) and \( Y \) induced on \( n = 30 \) points in \([0, 1] \times [0, 1] \). \( \Delta r^2 \) becomes zero on average on 100% noise enabling a more interpretable range of variation.

Figure 4.2: Average value of MIC and AMIC for different percentages of white noise. Linear, quadratic, cubic, and 4th root relationship between \( X \) and \( Y \) induced on \( n = 60 \) points in \([0, 1] \times [0, 1] \). AMIC becomes zero on average on 100% noise enabling a more interpretable range of variation.
The average value of a dependency estimator should not be biased with regards to the sample \( S_n \) as stated in Property 4.2. In Figure 4.3, we show that \( r^2 \) and MIC suffer from this problem: their estimates are higher on average when \( n \) is smaller. Figure 4.3 shows the average value of raw and adjusted measures on 2,000 simulations for different levels of noise and sample size \( n \): \( r^2 \) and \( Ar^2 \) are compared on linear relationships; MIC and AMIC are compared on linear, quadratic, cubic, and 4th root relationships. Neither the zero baseline Property 4.1 nor the quantification unbiasedness Property 4.2 is verified for the raw measures \( r^2 \) and MIC, shown respectively in Figure 4.3a and 4.3c. Instead, \( Ar^2 \) and AMIC in Figure 4.3b and 4.3d, satisfy both properties: they have zero baseline and their average value is not biased with regards to the sample size \( n \). We claim that these properties improve interpretability when quantifying dependency and enhance equitability for MIC [Reshef et al., 2011].

4.4 Adjusting Estimates for Ranking

When the task is ranking dependencies according to their strength, dependencies induced on smaller sample size \( n \) or on variables with more categories have more chances to be ranked higher as shown in Example 4.2 for Gini gain. This issue is due to inflated estimates due to finite samples. Indeed, \( r^2 \) and MIC suffer from the same problem.

Consider this experiment: we generate five samples \( S_n \) with \( n = [20, 40, 60, 80, 100] \) to simulate different amount of missing values for a joint distribution \((X, Y)\) where \( X \) and \( Y \) are independent. For each sample, we compute \( r^2(S_n | X, Y) \), we select \( S_n \) that achieves the highest value, and iterate this process 10,000 times. Given that the population value \( \rho^2(X, Y) = 0 \) for all samples, all samples should have equal chances to maximize the \( r^2 \). However, Figure 4.4 shows that \( S_{20} \) has higher chances to maximize \( r^2 \). This implies that dependencies estimated on samples with missing values have higher chances to be ranked higher in terms of strength.

We would like that dependencies which share the same population value for \( D \) had the same chances to maximize the dependency estimate \( \hat{D} \) even if estimated on different samples. More formally:
Figure 4.3: Average value of $r^2$, $A_r^2$, MIC, and AMIC on different amount of noise and different sample size $n$. $r^2$ and $A_r^2$ are compared on linear relationships. Raw measures show higher values for smaller $n$ on average. Instead, Property 4.2 of unbiasedness with regards to $n$ is empirically verified for adjusted measures.
Figure 4.4: Probability to select the sample $S_n$ with $n = [20, 40, 60, 80, 100]$ according to $r^2(S_n|X,Y)$ by fixing the population value $\rho^2(X,Y) = 0$. The relationship with $n = 20$ has more chances to be ranked higher.

**Property 4.3** (Ranking Unbiasedness).

*If $D(X_1,Y_1) = D(X_2,Y_2) = \ldots = D(X_K,Y_K)$ then the probability of $\hat{D}(S_{ni}|X_i,Y_i)$ being equal or greater than any $\hat{D}(S_{nj}|X_j,Y_j)$ is $\frac{1}{K}$ for all $n_i, n_j$, $1 \leq i \neq j \leq K$.*

For example in Figure 4.4 we would like constant probability of selection equal to $\frac{1}{5} = 0.20$. Property 4.3 is useful to achieve higher accuracy when the task is ranking the pair of variables that show the stronger relationship.

Biases in ranking are well known in the decision tree community [Dobra and Gehrke, 2001] as shown in Example 4.2. Distributional properties of the raw dependency measure have to be employed to adjust for biases in ranking. For example, ranking according to $p$-values is a possible solution [Dobra and Gehrke, 2001]. Nonetheless, they are susceptible to precision errors when $X$ and $Y$ are very dependent [Kononenko, 1995]. Therefore, here we propose to adjust measures by statistical standardization. Standardized measures still determine if the estimate $\hat{D}$ is statistically significant and are less prone to precision errors. We define the standardization technique to any dependency measure estimate $\hat{D}$ to employ it for unbiased ranking:

**Definition 4.3** (Standardization for Ranking).

$$S\hat{D}(S_n|X,Y) \triangleq \frac{\hat{D}(S_n|X,Y) - E[\hat{D}_0(S_n|X,Y)]}{\sqrt{\text{Var}(\hat{D}_0(S_n|X,Y))}}$$

is the standardized $\hat{D}(S_n|X,Y)$, where $E[\hat{D}_0(S_n|X,Y)]$ and $\text{Var}(\hat{D}_0(S_n|X,Y))$ are, respectively, the expected value and the variance of $\hat{D}$ under the null.

There is a subtle connection between standardized measures $S\hat{D}$ and $p$-values:
Proposition 4.1. The \( p \)-value associated to the estimate \( \hat{D}(S_n | X, Y) \) is bounded above by a function inversely proportional to its standardized measure \( S\hat{D}(S_n | X, Y) \):

\[
p\text{-value} < \frac{1}{1 + \left(S\hat{D}(S_n | X, Y)\right)^2}
\]

Proof. Let \( \hat{D}_0(S_n | X, Y) \) be the random variable under the null hypothesis of independence between two variables \( X \) and \( Y \) associated to the statistic \( \hat{D}(S_n | X, Y) \). The \( p \)-value is defined as:

\[
p\text{-value} = P\left( \hat{D}_0(S_n | X, Y) \geq \hat{D}(S_n | X, Y) \right)
\]

\[
= P\left( \hat{D}_0(S_n | X, Y) - E[\hat{D}_0(S_n | X, Y)] \geq \hat{D}(S_n | X, Y) - E[\hat{D}_0(S_n | X, Y)] \right)
\]

\[
= P\left( \frac{\hat{D}_0(S_n | X, Y) - E[\hat{D}_0(S_n | X, Y)]}{\sqrt{\text{Var}(\hat{D}_0(S_n | X, Y))}} \geq \frac{\hat{D}(S_n | X, Y) - E[\hat{D}_0(S_n | X, Y)]}{\sqrt{\text{Var}(\hat{D}_0(S_n | X, Y))}} \right)
\]

\[
= P\left( \frac{\hat{D}_0(S_n | X, Y) - E[\hat{D}_0(S_n | X, Y)]}{\sqrt{\text{Var}(\hat{D}_0(S_n | X, Y))}} \geq S\hat{D}(S_n | X, Y) \right).
\]

Let \( Z \) be the standardized random variable \( \frac{\hat{D}_0(S_n | X, Y) - E[\hat{D}_0(S_n | X, Y)]}{\sqrt{\text{Var}(\hat{D}_0(S_n | X, Y))}} \), then using the one side Chebyshev’s inequality also known as the Cantelli’s inequality [Ross, 2012]:

\[
p\text{-value} = P(Z \geq S\hat{D}(S_n | X, Y)) < \frac{1}{1 + \left(S\hat{D}(S_n | X, Y)\right)^2}
\]

For example, if \( S\hat{D}(S_n | X, Y) \) is equal to 4.46 the associated \( p \)-value is smaller than 0.05.

Even though \( p \)-values have been extensively employed to correct for selection bias in the decision tree community, it is very difficult to satisfy the ranking unbiasedness Property 4.3 with either \( p \)-values or standardized measures. Therefore we also define an adjustment to dependency measures whose bias can be tuned according to a parameter
adjusting estimates for ranking

\[ \hat{D}(S_n|X,Y)(\alpha) \triangleq \hat{D}(S_n|X,Y) - q_0(1-\alpha) \]

is the adjustment at level \( \alpha \in (0,1] \) of \( \hat{D}(S_n|X,Y) \), where \( q_0(1-\alpha) \) is the \((1-\alpha)\)-quantile of \( \hat{D}_0(S_n|X,Y) \) under the null: i.e.,

\[ P(\hat{D}(S_n|X,Y) \leq q_0(1-\alpha)) = 1-\alpha. \]

At a fixed significance level \( \alpha \), the quantile \( q_0(1-\alpha) \) induces more penalization when the estimate is not statistically significant. With regards to Example 4.2, fixing \( \alpha = 0.05 \) we penalize the variable \( X_1 \) and the variable \( X_2 \) by \( q_0(0.95) \) equal to 0.036 and 0.053 respectively. The latter variable gets penalized more because it is less statistically significant having more categories. In contrast, \( S\hat{D} \) fixes the amount of penalization based on statistical significance and does not allow to tune the bias during ranking. In the next section, we aim to show the shortcomings of raw measures and standardized measures for ranking tasks.

4.4.1 Ranking Biases of Raw and Standardized Measures (p-values)

We use \( r^2 \) and its adjusted versions in a case study: \( Ar^2 \) is defined as per Eq. (4.4), \( Ar^2(\alpha) = r^2 - q_0(1-\alpha) \) where \( q_0(1-\alpha) \) is computed with the Beta distribution (see Section 4.3), and the standardized \( r^2 \) is defined as:

\[ Sr^2(S_n|X,Y) = \frac{r^2(S_n|X,Y) - \frac{1}{n-1}}{\sqrt{\frac{2(n-2)}{(n-1)^2(n+1)}}} \quad (4.6) \]

We do not evaluate p-values because their use is equivalent to the use of standardized measures which are also much easier to compute and less prone to precision errors.

We perform similar experiments as in the previous section: we fix the population value for a dependency and compute estimates on different samples \( S_n \) to compute their probability of selection. We select samples according \( r^2 \), \( Ar^2 \), and \( Sr^2 \). Figure 4.6a shows
the probability of selection of different samples at fixed population value $\rho^2 = 0$. We can clearly see that the ranking unbiasedness Property 4.3 is satisfied if we use $Sr^2$ (top plot). On the other hand, the sole adjustment for quantification $Ar^2$ is not enough to remove $r^2$ bias towards small $n$. Nonetheless, Figure 4.6b shows that if we generate a linear relationship between $X$ and $Y$ with 10% white noise (i.e., $\rho^2$ is fixed to a value greater than 0), $Sr^2$ is biased towards big $n$. This is because we prefer statistically significant relationships. This phenomena might have been overlooked in the decision tree community [Strobl et al., 2007a; Frank and Witten, 1998].

Given that it is difficult to satisfy the ranking unbiasedness Property 4.3 in general, we show how $\alpha$ in our adjustment $Ar^2(\alpha)$ might be used to tune the bias when it is possible. Figure 4.5 shows that with big $\alpha$ ($\alpha \approx 0.4$) relationships on small $n$ have higher probability to be selected. On the other hand, small $\alpha$ ($\alpha \approx 0.05$) tunes the bias towards higher sample size $n$.

![Figure 4.5: Probability of selection of a sample $S_n$ when $X$ is linearly related to $Y$ with 10% white noise using $Ar^2(\alpha)$: $\alpha$ tunes the bias towards small $n$ with a big $\alpha$ (bottom plot) or big $n$ with a small $\alpha$ (top plot).](image)

On a real ranking task, it is reasonable to rank according to $Ar^2(\alpha)$ and see how the rank changes with changes of $\alpha$ rather than relying on a single ranking based on biased
measures such as $r^2$, $Ar^2$, or $Sr^2$. The best value for $\alpha$ can be chosen by cross-validation when it is possible. Similar conclusions can be drawn for MIC and its adjusted versions.

Figure 4.6: Probability to select the sample $S_n$ induced on $n = [20, 40, 60, 80, 100]$ according to adjusted measures: $Sr^2$ satisfies the ranking unbiasedness Property 4.3 when $\rho^2 = 0$ but not when $\rho^2 > 0$. All measures show to be biased in the latter case: it is difficult to satisfy Property 4.3 in general.
4.4.2 Experiments with Pearson Correlation and MIC

MIC and $r^2$ have been used in Reshef et al. [2011] to identify the strongest related pair of socio-economic variables using the WHO data set. This data set is a collection of $m = 357$ variables for $n = 201$ countries. Some of the variables have a high percentage of missing values and they are available on much fewer than $n = 201$ samples. In this section, we aim to alert the users of MIC and $r^2$ about ranking biases for relationships induced on different sample size $n$. We conduct an experiment: we choose a reference socio-economic variable $Y$ and select the top related variable according to $r^2$ and its adjusted versions. Then, we estimate the dependency between two variables based on the data points available for both $X$ and $Y$. We only consider dependencies estimated on at least $n \geq 10$ data points. Figure 4.7 shows the top-most dependent variable $X$ to $Y =$“Breast cancer number of female deaths” using $r^2$, $A^2$, $S^2$, and $A^2(\alpha = 0.1)$. The top-most dependent variable according to $r^2$ and $A^2$ is $X =$“Aid given” which quantifies the amount of aid given to poor countries in million US$. Instead, $S^2$ and $A^2(\alpha = 0.1)$ identify $X =$“Breast cancer number of new female cases” which seems a more reasonable choice given that the number of deaths might be correlated with new cancer cases. Indeed as seen in the previous Section, $r^2$ and $A^2$ favour variables induced on small $n$. Moreover from the plot in Figure 4.7 we see that they are very sensitive to extreme values or outliers: i.e. the United States show a very high number of deaths due to breast cancer $\approx 43,000$ in a year and a very high amount of aid given $\approx 20$ Billion US$; this increases the chances for a high $r^2$ or $A^2$.

MIC is even more inclined to select variables induced on small $n$. For example we see in Figure 4.8 that if we target $Y =$“Maternal mortality” which quantifies the number of female deaths during pregnancy (out of 100,000 live births), and we choose MIC or AMIC to identify the top dependent variable, we get $X =$“Oil consumption per person” (tonnes per year). There seems to exist an inversely proportional relationship between $X$ and $Y$, possibly due to the common cause of overall economic development but it is difficult to argue in favor of the amount of oil/energy consumption per person as the most dependent variable to maternal mortality. We also identified the top variables according
Table 4.1: Average sample size $n$ for the chosen top variables with different adjustments. The user of dependency measures should be aware of the bias of raw dependency estimators $\hat{D}$ towards small $n$ and try to explore results from their adjusted versions $S\hat{D}$ and $A\hat{D}(\alpha)$ when ranking. Ultimately, the latter can be chosen to tune the bias towards smaller $n$ (big $\alpha$) or big $n$ (small $\alpha$).

<table>
<thead>
<tr>
<th>Measure</th>
<th>$r^2$</th>
<th>MIC</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\hat{D}$</td>
<td>114.6 (min)</td>
<td>103.1 (min)</td>
</tr>
<tr>
<td>$A\hat{D}$</td>
<td>115.1</td>
<td>106.9</td>
</tr>
<tr>
<td>$S\hat{D}$</td>
<td>133.7 (max)</td>
<td>131.8 (max)</td>
</tr>
<tr>
<td>$A\hat{D}(\alpha = 0.4)$</td>
<td>116.2 (min)</td>
<td>111.7 (min)</td>
</tr>
<tr>
<td>$A\hat{D}(\alpha = 0.05)$</td>
<td>121.1</td>
<td>119.4</td>
</tr>
<tr>
<td>$A\hat{D}(\alpha = 0.1)$</td>
<td>120.2 (max)</td>
<td>117.4 (max)</td>
</tr>
</tbody>
</table>

to SMIC and AMIC($\alpha = 0.01$) computed with 10,000 Monte Carlo permutations. More specifically:

$$\text{SMIC}(\hat{D}|X,Y) = \frac{\text{MIC}(\hat{D}|X,Y) - \text{EMIC}_0}{\text{SDMIC}_0},$$

where $\text{SDMIC}_0$ is the unbiased estimator of the standard deviation of MIC permutations $\text{SDMIC}_0 = \sqrt{\frac{1}{S-1} \sum_{s=1}^{S} (\text{MIC}_0^{(s)} - \text{EMIC}_0)^2}$; and

$$\text{AMIC}(\hat{D}|X,Y)(\alpha) = \text{MIC}(\hat{D}|X,Y) - q_0(1 - \alpha),$$

where $q_0(1 - \alpha)$ is the $\lceil (1 - \alpha) \cdot S \rceil$-th MIC value from the sorted list of $S$ MIC permutations in ascending order. The top variables according to SMIC and AMIC($\alpha = 0.01$) are instead variables related to communicable/non-communicable (infectious/non-infectious) diseases which is more intuitively related to mortality.

Table 4.1 shows the average sample size $n$ for the chosen top variables with different adjustments. The user of dependency measures should be aware of the bias of raw dependency estimators $\hat{D}$ towards small $n$ and try to explore results from their adjusted versions $S\hat{D}$ and $A\hat{D}(\alpha)$ when ranking. Ultimately, the latter can be chosen to tune the bias towards smaller $n$ (big $\alpha$) or big $n$ (small $\alpha$).
**Figure 4.7:** Plot of the top-most dependent variable $X$ to $Y = \text{Breast cancer number of new female cases}$ according different adjustments for $r^2$, $Ar^2$ favor relationships on small $n$. $Sr^2$, and $Ar^2(\alpha = 0.1)$ penalize relationships on small $n$ and select more reasonably $X = \text{Breast cancer number of new female cases}$.

**Figure 4.8:** Plot of the top-most dependent variable $X$ to $Y = \text{Maternal Mortality}$ according different adjustments for MIC. MIC and AMIC are biased towards small $n$. SMIC and AMIC($\alpha = 0.1$) select more reasonably either $X = \text{Years of life lost to communicable diseases}$ or $X = \text{Years of life lost to non-communicable diseases}$. 
4.4.3 Experiments with Gini gain in Random Forests

Splitting criteria are known to be biased towards variables induced on small \( n \) or categorical with many categories. Standardized measures and \( p \)-values are the state-of-the-art strategy to solve this problem [Dobra and Gehrke, 2001; Strobl et al., 2007a; Frank and Witten, 1998]. However, we saw that standardized measures such as the standardized \( r^2 \) are unbiased in ranking only when the population value \( D(X,Y) = 0 \), and the user might better tune the bias using the parameter \( \alpha \). This is particularly important when the bias can be tuned with cross-validation, for example when Gini gain is used as splitting criterion in random forests. This is true also for adjustments of dependency measures between categorical variables, such as the Gini Gain (GG) defined in Eq. (2.19). Here we use the expected value \( E_0[\text{Gini}] \) and the variance \( \text{Var}_0(\text{Gini}) \) of Gini proposed in Dobra and Gehrke [2001] to standardize Gini gain as per Definition 4.3. According to Dobra and Gehrke [2001], the expected value of Gini gain under the null using the multinomial model is:

\[
E[\text{GG}_0(S_n|X,Y)] = \frac{r-1}{n} \left( 1 - \sum_{j=1}^{c} \left( \frac{b_j}{n} \right)^2 \right) \quad (4.7)
\]

and its variance is:

\[
\text{Var}[\text{GG}_0(S_n|X,Y)] = \frac{1}{n^2} \left[ (r-1) \left( 2 \sum_{j=1}^{c} \left( \frac{b_j}{n} \right)^2 + 2 \left( \sum_{j=1}^{c} \left( \frac{b_j}{n} \right) \right)^2 - 4 \sum_{j=1}^{c} \left( \frac{b_j}{n} \right)^3 \right) - 2 \sum_{i=1}^{r} \left( \frac{1}{a_i} - \frac{r}{n} + \frac{1}{n} \right) \times \left( -2 \sum_{j=1}^{c} \left( \frac{b_j}{n} \right)^2 - 6 \left( \sum_{j=1}^{c} \left( \frac{b_j}{n} \right)^2 \right)^2 + 8 \sum_{j=1}^{c} \left( \frac{b_j}{n} \right)^3 \right) \right] \quad (4.8)
\]

We employ these statistics to compute:

\[
\text{SGG}(S_n|X,Y) = \frac{\text{GG}(S_n|X,Y) - E[\text{GG}_0(S_n|X,Y)]}{\text{Var}(\text{GG}_0(S_n|X,Y))}
\]
Moreover, we employ them to compute the adjusted Gini gain $\text{AGG}(\alpha)$ as follows:

**Proposition 4.2.** The adjustment for ranking at level $\alpha \in (0, 1]$ for Gini gain is:

$$\text{AGG}(S_n|X,Y)(\alpha) = \text{GG}(S_n|X,Y) - \tilde{q}_0(1 - \alpha)$$

where $\tilde{q}_0(1 - \alpha)$ is an upper bound for the $(1 - \alpha)$-quantile of Gini gain equal to:

$$E[\text{GG}_0(S_n|X,Y)] + \sqrt{\frac{1 - \alpha}{\alpha} \text{Var}(\text{GG}_0(S_n|X,Y))}.$$

**Proof.** Let $\mu$ and $\sigma$ be the expected value and standard deviation of $\text{GG}$ under the null respectively. We apply the Cantelli’s inequality to find an upper bound for $q_0(1 - \alpha)$:

$$P(\text{GG} \leq \mu + \lambda\sigma) \geq \lambda^2 \frac{\mu^2}{1 + \lambda^2}$$

for $\lambda \geq 0$. If we set $\frac{\lambda^2}{1 + \lambda^2} = \alpha$ then $P(\text{GG} \leq \mu + \sqrt{\frac{1 - \alpha}{\alpha}}\sigma) \geq \alpha$. This implies $q_0(1 - \alpha) \leq \mu + \sqrt{\frac{1 - \alpha}{\alpha}}\sigma$.

We tested the standardized Gini Gain (SGG) on selection bias correction on the same data in Example 4.2. SGG in Figure 4.9a shows close to uniform probability of selection for the two identical variables $X_1$ and $X_2$ and this is consistent with the literature on the use of $p$-values in decision trees. Nonetheless, if we fix the dependency between $Y$ and $X_1$, and between $Y$ and $X_2$, SGG becomes biased towards the variable with fewer number of categories because it is more statistically significant. Indeed, SGG is unbiased as per Property 4.3 only under the null hypothesis. In Figure 4.9b, we fix the dependency between $Y$ and the variables by fixing the true probability distribution in order that the first label $u_1$ is predictive towards the label $v_1$ of $Y$: $p_{Y|X}(v_1|u_1) = 0.65$ and $p_{Y|X}(v_2|u_1) = 0.35$. Then, we generate the other labels independently to the class: $X_2$ has two categories and $X_3$ has three categories. In this case, even if the probability of selection of the two variables should be equal, SGG selects more often the variable with fewer number of categories. This might have been overlooked in the decision tree community.

We compared WEKA random forests with GG, SGG, and AGG($\alpha$) as splitting criteria. To our knowledge this is the first time SGG and AGG($\alpha$) are tested in random forest. The forest is built on 1,000 trees taking care of sampling data with no replacement (50% training set records for each tree) to not introduce further biases towards categorical
Variables with many categories [Strobl et al., 2007b]. We employed 17 UCI data sets and 2 data sets with many categorical variables studied in Altmann et al. [2010]. The latter data sets are related to biological classification problems and some of the variables can take as many categories as the number of amino acids at a given site in a viral protein: e.g. in the HIV data set, there exist variables which can take 21 possible values and induce splits of 21-cardinality in the trees. Table 4.2 shows the AUC performance of random forest computed with 50 bootstrap 2-fold cross-validation using different splitting criteria. We fixed $\alpha$ in AGG($\alpha$) to show that using a value of 0.05 or 0.1 on average increases the random forest’s AUC: see Figure 4.10. Indeed, the performance of random forests with AGG($\alpha = 0.05$) is statistically better than the one built with Gini according to the 1-sided Sign test: $p$-value = 0.0106. Although the observed effect size is small, it was consistent, and there is no extra computational effort. Furthermore, the user could tune $\alpha$ on each data set with cross-validation to either achieve penalization of high number of categories (small $\alpha$) simulating the behavior of SGG, or penalization

Figure 4.9: Probability to select $X_1$ with 2 categories and $X_3$ with three categories when $Y$ is equally dependent on them with Standardized Gini Gain (SGG). SGG satisfies the ranking unbiasedness Property 4.3 under the null when both variable are independent to $Y$. Nonetheless, if $X_1$ and $X_2$ are equally predictive to $Y$, SGG selects more often the variable induced on fewer categories.

(a) Under the null.  
(b) Not under the null, and $X_1$ and $X_2$ have equal predictiveness.
100 A Framework to Adjust Dependency Measure Estimates for Chance

Figure 4.10: AUC of random forest varying $\alpha$ in AGG($\alpha$): with $\alpha = \{0.01, 0.05\}$ it achieves the best results on average.

of small number of categories (big $\alpha$) simulating Gini for the best performance. We strongly believe that adjusted splitting criteria are beneficial given that i) they can be plugged in random forests where Gini gain is currently used, ii) they exhibit the same computational complexity as the Gini gain, and iii) they are easy to implement, in particular much easier than the estimation of their confidence interval with a possibilistic loss function proposed recently [Serrurier and Prade, 2015].

4.5 Conclusion

In this chapter we discussed how to adjust dependency measure estimates between two variables $X$ and $Y$ using the null hypothesis of their independence. This is particularly important to achieve interpretable quantification of the amount of dependency. For this task, we proposed the quantification adjusted measures $Ar^2$ and AMIC. However, quantification adjustment is not enough to achieve accurate ranking of dependencies. In particular, it is very difficult to achieve ranking unbiasedness. In this task, the user should explore the possible rankings obtained with standardized and ranking adjusted measures, varying the parameter $\alpha$. We demonstrated that our $Sr^2$, $Ar^2(\alpha)$, SMIC, and AMIC($\alpha$) can be used to obtain more meaningful rankings, and that AGG($\alpha$) yields
Table 4.2: Random forest AUC using different splitting criteria. Either (+), (=), or (−) means statistically greater, equal, or smaller according to the 1-sided paired t-test at level 0.05 than random forest AUC with Gini gain.

<table>
<thead>
<tr>
<th>Data set</th>
<th>Variable with max number of categories</th>
<th>Number of classes</th>
<th>$m_{\text{categorical}} + m_{\text{numerical}} = m$</th>
<th>n</th>
<th>GG</th>
<th>SGG</th>
<th>AGG ($\alpha = 0.05$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Credit-g</td>
<td>11</td>
<td>2</td>
<td>13 + 7 = 20</td>
<td>1000</td>
<td>77.72</td>
<td>78.43 (+)</td>
<td>77.93 (+)</td>
</tr>
<tr>
<td>australian</td>
<td>14</td>
<td>2</td>
<td>8 + 6 = 14</td>
<td>690</td>
<td>92.6</td>
<td>93.15 (+)</td>
<td>93.09 (+)</td>
</tr>
<tr>
<td>bio-promoters</td>
<td>4</td>
<td>2</td>
<td>57 + 0 = 57</td>
<td>106</td>
<td>96.9</td>
<td>97.19 (+)</td>
<td>97.26 (+)</td>
</tr>
<tr>
<td>flags</td>
<td>14</td>
<td>8</td>
<td>26 + 2 = 28</td>
<td>194</td>
<td>90.54</td>
<td>91.77 (+)</td>
<td>91.83 (+)</td>
</tr>
<tr>
<td>kr-vs-kp</td>
<td>3</td>
<td>2</td>
<td>36 + 0 = 36</td>
<td>3196</td>
<td>99.87</td>
<td>99.87 (=)</td>
<td>99.87 (=)</td>
</tr>
<tr>
<td>led7</td>
<td>2</td>
<td>10</td>
<td>7 + 0 = 7</td>
<td>3200</td>
<td>94.17</td>
<td>94.17 (=)</td>
<td>94.17 (=)</td>
</tr>
<tr>
<td>lymph</td>
<td>8</td>
<td>4</td>
<td>15 + 3 = 18</td>
<td>148</td>
<td>92.68</td>
<td>93.06 (+)</td>
<td>93.03 (+)</td>
</tr>
<tr>
<td>mfeat-pixel</td>
<td>7</td>
<td>10</td>
<td>240 + 0 = 240</td>
<td>2000</td>
<td>99.58</td>
<td>99.63 (+)</td>
<td>99.64 (+)</td>
</tr>
<tr>
<td>mito</td>
<td>21</td>
<td>2</td>
<td>23 + 0 = 23</td>
<td>175</td>
<td>78.89</td>
<td>78.8 (=)</td>
<td>78.63 (=)</td>
</tr>
<tr>
<td>monks1</td>
<td>4</td>
<td>2</td>
<td>6 + 0 = 6</td>
<td>556</td>
<td>99.97</td>
<td>99.86 (−)</td>
<td>97.36 (−)</td>
</tr>
<tr>
<td>monks2</td>
<td>4</td>
<td>2</td>
<td>6 + 0 = 6</td>
<td>601</td>
<td>65.33</td>
<td>71.3 (+)</td>
<td>78.08 (+)</td>
</tr>
<tr>
<td>monks3</td>
<td>4</td>
<td>2</td>
<td>6 + 0 = 6</td>
<td>554</td>
<td>98.69</td>
<td>98.7 (=)</td>
<td>98.72 (=)</td>
</tr>
<tr>
<td>solar-flare</td>
<td>6</td>
<td>6</td>
<td>11 + 0 = 11</td>
<td>323</td>
<td>89.12</td>
<td>89.22 (+)</td>
<td>89.17 (+)</td>
</tr>
<tr>
<td>splice</td>
<td>6</td>
<td>3</td>
<td>60 + 0 = 60</td>
<td>3190</td>
<td>99.52</td>
<td>99.52 (=)</td>
<td>99.53 (=)</td>
</tr>
<tr>
<td>steel</td>
<td>2</td>
<td>2</td>
<td>6 + 27 = 33</td>
<td>1941</td>
<td>99.94</td>
<td>99.94 (−)</td>
<td>99.94 (−)</td>
</tr>
<tr>
<td>tae</td>
<td>2</td>
<td>3</td>
<td>2 + 3 = 5</td>
<td>151</td>
<td>72.64</td>
<td>72.63 (−)</td>
<td>72.79 (+)</td>
</tr>
<tr>
<td>tic-tac-toe</td>
<td>3</td>
<td>2</td>
<td>9 + 0 = 9</td>
<td>958</td>
<td>97.78</td>
<td>97.91 (+)</td>
<td>97.93 (+)</td>
</tr>
<tr>
<td>c-to-u</td>
<td>5</td>
<td>2</td>
<td>42 + 3 = 45</td>
<td>2694</td>
<td>89.81</td>
<td>89.48 (−)</td>
<td>89.25 (−)</td>
</tr>
<tr>
<td>HIV</td>
<td>21</td>
<td>2</td>
<td>1030 + 0 = 1030</td>
<td>355</td>
<td>84.06</td>
<td>89.22 (+)</td>
<td>89.53 (+)</td>
</tr>
</tbody>
</table>

1-tailed sign test p-value (* means statistically significant at level 0.05) 0.0592 0.0106*
higher accuracy in random forests. The code for our measures and experiments has been made available online\footnote{https://sites.google.com/site/adjdep/}.
ADJUSTING CLUSTERING COMPARISONS MEASURES

Outline
In this chapter we discuss how to analytically obtain the adjustments for quantification and ranking discussed in Chapter 4 for clustering comparison measures. We generalize adjustments for the information theoretic measures discussed in Section 2.2.1 and the pair-counting measures discussed in Section 2.2.2 through the adjustment of generalized Information Theoretic (IT) measures based on the Tsallis $q$-entropy. In Section 5.2 we introduce generalized IT measures. In Section 5.3 we discuss how to adjust them for quantification by guaranteeing constant baseline. In Section 5.4 we identify the application scenarios of adjusted information theoretic measures and adjusted pair-counting measures. Moreover, we propose to adjust generalized IT measure with standardization for ranking purposed in Section 5.5 and conduce experimental analysis with them in Section 5.6.

5.1 Introduction

Clustering comparison measures are used to compare partitions/clusterings of the same data set. In the clustering community [Aggarwal and Reddy, 2013], they are extensively used for external validation when the ground truth clustering is available. Clustering
comparisons measures are dependency measures between two clusterings. In Section 2.2.2, we discussed the popular family of clustering comparison measures based on pair-counting. This category comprises the well known similarity measures Rand Index (RI) and the Jaccard similarity coefficient (J). Recently, information theoretic (IT) measures have been also extensively used to compare partitions [Strehl and Ghosh, 2003]. In particular, the normalized Mutual Information (NMI) discussed in Section 2.2.1 is often employed in practice. Given the variety of different possible measures, it is very challenging to identify the best choice for a particular application scenario [Wu et al., 2009].

When quantifying the amount of dependency between two clusterings, a desirable property of clustering comparison measures is to have a constant baseline in the case of random independent partitions. Adopting a probabilistic interpretation of the partition problem, an expected value can be computed under the assumption of random and independent clusterings and then subtracted from the measure. We formalized this approach of adjusting for quantification purposes in Chapter 4 in Section 4.3. Notable examples of adjusted measures for clustering comparisons are the Adjusted Rand Index (ARI) [Hubert and Arabie, 1985]:

\[
ARI = \frac{RI - E[RI]}{\max RI - E[RI]},
\]

and the Adjusted Mutual Information (AMI) [Vinh et al., 2009]:

\[
AMI = \frac{MI - E[MI]}{\max MI - E[MI]}.\]

They take a 0 expectation value when partitions are independent, and are bounded above by 1 via the use of a normalization factor (an upper bound of the measure). It is common to see published research that validates clustering solutions against a reference ground truth clustering with the ARI or the AMI. Nonetheless there are still open problems: there are no guidelines for their best application scenarios shown in the literature to date and authors often resort to employing them both and leaving the reader to interpret.
Moreover, some clustering comparisons measures are susceptible to selection bias: when selecting the most similar partition to a given ground truth partition, clustering comparison measures are more likely to select partitions with many clusters. Our first key observation is that employing a baseline adjustment to a clustering comparison measure does not guarantee that it is bias free. In fact, it is still susceptible to selection bias, the tendency to choose inappropriate clustering solutions i) with more clusters, or ii) induced on fewer data points, when compared to a reference one. To illustrate, consider the following experiment on a data set of \( n = 100 \) records: a reference clustering of 4 equal-size clusters is compared in turn with 9 clustering solutions. Each solution is randomly generated with equal-size clusters and the number of clusters in each is 2, 3, 4, 5, 6, 7, 8, 9, 10 respectively. We then use the AMI measure to select the most similar clustering solution to the reference and then repeat the whole procedure 5,000 times. Figure 5.1 shows the probability of selecting a clustering solution with \( r \) clusters. We see that a clustering with \( r = 10 \) clusters will be selected more often than one with 2 clusters, even though we expect the former solution should be no more similar to the reference than the latter solution, due to the randomness of generation. So, although the AMI may have a constant baseline, it is still biased towards selecting clusterings containing more clusters. Indeed, clustering comparison measures might also need to be adjusted for ranking using the framework of Chapter 4 in Section 4.4.

![Figure 5.1: Probability AMI measure selects a random clustering solution with \( r \) clusters as the most similar solution to the reference clustering which has 4 clusters. AMI is biased towards selecting clustering solutions with a higher number of clusters.](image-url)
When adjusting clustering comparison measures, the key technical challenge is the analytical computation of the expected value and variance under the null hypothesis of independent clusterings. Regarding RI and MI, the model of randomness adopted to compute the expected value is the permutation model, also called the hypergeometric model: fixing the number of points for each cluster, partitions are generated uniformly and randomly via permutations of records. Under this assumption, the distribution of clustering comparison measure is known and thus so is the expected value. It has been shown that this hypothesis behaves well in practical scenarios. In this chapter, we aim at using this assumption to compute the variance of clustering comparison measures under independent clustering. Moreover, we also aim to bridge the gap between the adjustment of pair-counting measures and the adjustment of Information Theoretic (IT) measures. In Furuichi [2006]; Simovici [2007] it has been shown that generalized IT measures based on the Tsallis $q$-entropy [Tsallis et al., 2009] are a further generalization of IT measures in Section 2.2.1 and some pair-counting measures such as RI from Section 2.2.2. We will exploit this useful idea to connect ARI and AMI. In order to adjust for quantification and for ranking clustering comparison measures based on information theory and pair-counting, we analytically compute the expected value and variance for generalized IT measures. To solve this problem, we propose a technique applicable to a broader class of measures we name $L_\phi$, which includes generalized IT measures as a special case. This generalizes previous work which provided analytical adjustments for narrower classes: measures based on pair-counting from the family $L$ [Albatineh et al., 2006], and measures based on the Shannon’s mutual information [Vinh et al., 2009, 2010]. Moreover, we define a family of measures $N_\phi$ which generalizes many clustering comparison measures. For measures that belong in this family, the expected value can be analytically approximated when the number of objects is large. Table 5.1 summarizes the development of this line of work over the past 30 years and positions our contribution.

In summary, we make the following contributions:

- We define families of measures for which the expected value and variance can be computed analytically when the clusterings are random and independent;
5.2 Generalized Information Theoretic Measures

In this chapter, we discuss clustering similarity measures $S(U, V)$ between two partitions (clusterings) $U$ and $V$ of the same data set of $n$ objects. $U$ is a clustering of $r$ clusters and $V$ is a clustering of $c$ clusters. Clustering comparison measures are dependency
measures estimated on a contingency table such as the one in Table 2.3. Therefore, we make use of the same notation introduced in the Background Chapter 2.

Generalized Information Theoretic (IT) measures based on the generalized Tsallis $q$-entropy [Tsallis, 1988] can be defined for random variables [Furuichi, 2006] and also be applied to the task of comparing partitions [Simovici, 2007] when they are estimated on a contingency table. Indeed, these measures have also seen recent application in the machine learning community. More specifically, it has been shown that they can act as proper kernels [Martins et al., 2009]. Furthermore, empirical studies demonstrated that careful choice of $q$ yields successful results when comparing the similarity between documents [Vila et al., 2011], decision tree induction [Maszczyk and Duch, 2008; Wang et al., 2015], and reverse engineering of biological networks [Lopes et al., 2011]. It is important to note that the Tsallis $q$-entropy is equivalent to the Havrda-Charvat-Daróczy generalized entropy proposed in Havrda and Charvát [1967]; Daróczy [1970]. Results available in literature about these generalized entropies are equivalently valid for all the proposed versions.

Given $q \in \mathbb{R}^+ - \{1\}$, the generalized Tsallis $q$-entropy for a partition $V$ is defined as:

$$H_q(V) \triangleq \frac{1}{q-1} \left(1 - \sum_{j=1}^{c} \left(\frac{b_j}{n}\right)^q\right). \quad (5.3)$$

Similarly to the case of Shannon entropy, we have the joint $q$-entropy $H_q(U,V)$ and the conditional $q$-entropies $H_q(U|V)$ and $H_q(V|U)$. Conditional $q$-entropy is computed according to a weighted average parametrized in $q$. More specifically the formula for $H_q(V|U)$ is:

$$H_q(V|U) \triangleq \sum_{i=1}^{r} \left(\frac{a_i}{n}\right)^q H_q(V|u_i) = \sum_{i=1}^{r} \left(\frac{a_i}{n}\right)^q \frac{1}{q-1} \left(1 - \sum_{j=1}^{c} \left(\frac{n_{ij}}{a_i}\right)^q\right) \quad (5.4)$$

The $q$-entropy reduces to the Shannon entropy computed with natural logarithms for $q \to 1$:

$$\lim_{q \to 1} H_q(V) = H(V) = \sum_{j=1}^{c} \frac{b_j}{n} \log \frac{b_j}{n}$$
In Furuichi [2006], using the fact that \( q > 1 \) implies \( H_q(U) \geq H_q(U|V) \), it is shown that non-negative MI can be naturally generalized with \( q \)-entropy when \( q > 1 \):

\[
\text{MI}_q(U, V) \triangleq H_q(U) - H_q(U|V) = H_q(V) - H_q(V|U) = H_q(U) + H_q(V) - H_q(U, V)
\]  
(5.5)

However, \( q \) values smaller than 1 are allowed if the assumption that \( \text{MI}_q(U, V) \) is always positive can be dropped. In addition, generalized IT measures can be used to define the generalized variation of information distance (\( \text{VI}_q \)) which tends to VI in Eq. (2.12) when \( q \to 1 \):

\[
\text{VI}_q(U, V) \triangleq H_q(U|V) + H_q(V|U) = 2H_q(U, V) - H_q(U) - H_q(V)
\]  
(5.6)

\[
= H_q(U) + H_q(V) - 2\text{MI}_q(U, V)
\]

In Simovici [2007] it was shown that \( \text{VI}_q \) is a proper metric and interesting links were identified between measures for comparing partitions \( U \) and \( V \). We state these links in Proposition 5.1 given that they set the fundamental motivation of our paper:

**Proposition 5.1.** [Simovici, 2007] When \( q = 2 \) the generalized variation of information, the Mirkin index, and the Rand index are linearly related:

\[
\text{VI}_2(U, V) = \frac{1}{n^2} \text{MK}(U, V) = \frac{n - 1}{n} (1 - \text{RI}(U, V)).
\]

**Proof.**

\[
\text{VI}_q(U, V) = 2H_q(U, V) - H_q(U) - H_q(V)
\]

\[
= \frac{2}{q - 1} \left( 1 - \sum_{i=1}^{r} \sum_{j=1}^{c} \left( \frac{n_{ij}}{n} \right)^q \right)
\]

\[
- \frac{1}{q - 1} \left( 1 - \sum_{i=1}^{r} \left( \frac{a_i}{n} \right)^q \right)
\]

\[
- \frac{1}{q - 1} \left( 1 - \sum_{j=1}^{c} \left( \frac{b_j}{n} \right)^q \right)
\]
adjusting clustering comparisons measures

\[= \frac{1}{(q-1)n^q} \left( \sum_{i=1}^{r} a_i^q + \sum_{j=1}^{c} b_j^q - 2 \sum_{i=1}^{r} \sum_{j=1}^{c} n_{ij}^q \right).\]

When \( q = 2 \),

\[\text{VI}_2(U, V) = \frac{1}{n^2} (\sum_i a_i^2 + \sum_j b_j^2 - 2 \sum_{i,j} n_{ij}^2) = \frac{1}{n^2} \text{MK}(U, V) = \frac{n-1}{n} (1 - \text{RI}(U, V)).\]

Generalized IT measures are not only a generalization of IT measures in the Shannon sense but also a generalization of pair-counting measures for particular values of \( q \).

5.2.1 Normalized Generalized IT Measures

To allow a more interpretable range of variation, a clustering similarity measure should be normalized: it should achieve its maximum at 1 when \( U = V \). An upper bound to the generalized mutual information \( \text{MI}_q \) is used to obtained a normalized measure. \( \text{MI}_q \) can take different possible upper bounds [Furuichi, 2006]. Here, we choose to derive another possible upper bound using Eq. (5.6) when we use the minimum value of \( \text{VI}_q = 0 \):

\[\max \text{MI}_q = \frac{1}{2} (H_q(U) + H_q(V)).\]

This upper bound is valid for any \( q \in \mathbb{R}^+ - \{1\} \) and allows us to link different existing measures as we will show in the next sections of the paper. The normalized Mutual Information with \( q \)-entropy (\( \text{NMI}_q \)) is defined as follows:

\[
\text{NMI}_q(U, V) \triangleq \frac{\text{MI}_q(U, V)}{\max \text{MI}_q(U, V)} = \frac{\text{MI}_q(U, V)}{\frac{1}{2} (H_q(U) + H_q(V))} = \frac{H_q(U) + H_q(V) - H_q(U, V)}{\frac{1}{2} (H_q(U) + H_q(V))}.
\]

(5.7)

Even if \( \text{NMI}_q(U, V) \) achieves its maximum 1 when the partitions \( U \) and \( V \) are identical, \( \text{NMI}_q(U, V) \) is not a suitable clustering comparison measure. Indeed, it does not show constant baseline value equal to 0 when partitions are random. We explore this through an experiment. Given a dataset of \( n = 100 \) objects, we randomly generate uniform partitions \( U \) with \( r = 2, 4, 6, 8, 10 \) sets and \( V \) with \( c = 6 \) sets independently from each others. The average value of \( \text{NMI}_q \) over 1,000 simulations for different values of \( q \) is shown in Figure 5.3. It is reasonable to expect that when the partitions are independent, the average value of \( \text{NMI}_q \) is constant irrespectively of the number of sets
of the partition $U$. This is not the case. This behaviour is unintuitive and misleading when comparing partitions. Computing the analytical expected value of generalized IT measures under the null hypothesis of random and independent $U$ and $V$ is important; it can be subtracted from the measure itself to adjust its baseline for chance such that their value is 0 when $U$ and $V$ are random. Given Proposition 5.1, this strategy also allows us to generalize adjusted for chance pair-counting and Shannon IT measures.

### 5.3 baseline adjustment

In order to adjust the baseline of a similarity measure $S(U, V)$, we have to compute its expected value under the null hypothesis of independent partitions $U$ and $V$. We adopt the assumption used for RI [Hubert and Arabie, 1985] and the Shannon MI [Vinh et al., 2009]: partitions $U$ and $V$ are generated independently with fixed number of points $n$ and fixed marginals $a_i$ and $b_j$; this is also denoted as the permutation or the hypergeometric model of randomness. We are able to compute the exact expected value for a similarity measure in the family $\mathcal{L}_\phi$:

**Definition 5.1.** Let $\mathcal{L}_\phi$ be the family of similarity measures $S(U, V) = \alpha + \beta \sum_{ij} \phi_{ij}(n_{ij})$ where $\alpha$ and $\beta$ do not depend on the entries $n_{ij}$ of the contingency table $M$ and $\phi_{ij}(\cdot)$ are bounded real functions.
Intuitively, $\mathcal{L}_\phi$ represents the class of measures that can be written as a linear combination of $\phi_{ij}(n_{ij})$. A measure between partitions uniquely determines $\alpha$, $\beta$, and $\phi_{ij}$. However, not every choice of $\alpha$, $\beta$, and $\phi_{ij}$ yields a meaningful similarity measure. $\mathcal{L}_\phi$ is a superset of the set of $\mathcal{L}$ defined in Albatineh et al. [2006] as the family of measures $S(U, V) = \alpha + \beta \sum_{ij} n_{ij}^2$, i.e. $S \in \mathcal{L}$ are special cases of measures in $\mathcal{L}_\phi$ with $\phi_{ij}(\cdot) = (\cdot)^2$. Figure 5.2 shows a diagram of the similarity measures discussed in Section 2.2 and their relationships.

**Lemma 5.2.** If $S(U, V) \in \mathcal{L}_\phi$, when partitions $U$ and $V$ are random:

$$E[S(U, V)] = \alpha + \beta \sum_{ij} E[\phi_{ij}(n_{ij})] \text{ where } E[\phi_{ij}(n_{ij})]$$

is

$$\min\{a_i, b_j\} \sum_{n_{ij} = \max\{0, a_i + b_j - n\}} \phi_{ij}(n_{ij}) \frac{a_i! b_j! (n - a_i)! (n - b_j)!}{n! n_{ij}! (a_i - n_{ij})! (b_j - n_{ij})! (n - a_i - b_j + n_{ij})!}$$

(5.9)

**Proof.** The expected value of $S(U, V)$ according to the hypergeometric model of randomness is $E[S(U, V)] = \sum_{\mathcal{M}} S(\mathcal{M}) P(\mathcal{M})$ where $\mathcal{M}$ is a contingency table generated via permutations. This is reduced to

$$E[S(U, V)] = \sum_{\mathcal{M}} (\alpha + \beta \sum_{ij} \phi_{ij}(n_{ij})) P(\mathcal{M}) = \alpha + \beta \sum_{\mathcal{M}} \sum_{ij} \phi_{ij}(n_{ij}) P(\mathcal{M}).$$

Because of linearity of the expected value, it is possible to swap the summation over $\mathcal{M}$ and the one over cells obtaining

$$\alpha + \beta \sum_{ij} \sum_{n_{ij}} \phi_{ij}(n_{ij}) P(n_{ij}) = \alpha + \beta \sum_{ij} E[\phi_{ij}(n_{ij})]$$

where $n_{ij}$ is a hypergeometric distribution with the marginals $a_i, b_j, \text{ and } n$ as parameters, i.e. $n_{ij} \sim \text{Hyp}(a_i, b_j, n)$ where

$$P(n_{ij}) = \frac{\binom{b_j}{n_{ij}} \binom{n - b_j}{a_i - n_{ij}}}{\binom{n}{a_i}}.$$
with \( n_{ij} \in [\max \{0, a_i + b_j - n\}, \min \{a_i, b_j\}] \).

The expect value should be computed over all possible contingency tables \( \mathcal{M} \) with fixed number of points and fixed marginals and this is extremely time expensive. It has also been shown that the mere counting of such contingency tables with fixed marginals is \#P-complete [Dyer et al., 1997]. Nonetheless, the complexity of the problem has been dramatically reduced by reordering the sums in the expected value. Lemma 5.2 extends the results in Albatineh and Niewiadomska-Bugaj [2011] showing exact computation of the expected value of measures in the family \( \mathcal{L} \). Given that generalized IT measures belong in \( \mathcal{L}_\phi \) we can employ this result to adjust them.

5.3.1 Baseline Adjustment for Generalized IT measures

Using Lemma 5.2 it is possible to compute the exact expected value of \( H_q(U, V) \), \( VI_q(U, V) \) and \( MI_q(U, V) \):

**Theorem 5.3.** When the partitions \( U \) and \( V \) are random:

\[
\begin{align*}
(a) & \quad E[H_q(U, V)] = \frac{1}{q-1} \left( 1 - \frac{1}{n^q} \sum_{ij} E[n^q_{ij}] \right) \\
& \quad \text{with } E[n^q_{ij}] \text{ from Eq. (5.9) with } \phi_{ij}(n_{ij}) = n^q_{ij}; \\
(b) & \quad E[MI_q(U, V)] = H_q(U) + H_q(V) - E[H_q(U, V)]; \\
(c) & \quad E[VI_q(U, V)] = 2E[H_q(U, V)] - H_q(U) - H_q(V).
\end{align*}
\]

**Proof.** The results easily follow from Lemma 5.2 and the hypothesis of fixed marginals.

It is worth noting that this approach is valid for any \( q \in \mathbb{R}^+ - \{1\} \). We can use these expected values to adjust for baseline generalized IT measures. We use the method proposed in Hubert and Arabie [1985] to adjust similarity measures, such as \( MI_q \), and distance measures, such as \( VI_q \):

\[
\begin{align*}
\text{AMI}_q & \triangleq \frac{MI_q - E[MI_q]}{\max MI_q - E[MI_q]} \\
\text{AVI}_q & \triangleq \frac{E[VI_q] - VI_q}{E[VI_q] - \min VI_q}
\end{align*}
\] (5.10)
VI\textsubscript{q} is a distance measure, thus \( \min VI\textsubscript{q} = 0 \). For MI\textsubscript{q} we use the upper bound \( \max MI\textsubscript{q} = \frac{1}{2}(H_q(U) + H_q(V)) \) as for NMI\textsubscript{q} in Eq. (5.7). An exhaustive list of adjusted versions of Shannon MI can be found in Vinh et al. [2010], when the upper bound \( \frac{1}{2}(H_q(U) + H_q(V)) \) is used the authors named the adjusted MI as AMI\textsubscript{sum}.

It is important to note that this type of adjustment turns distance measures into similarity measures, i.e., AVI\textsubscript{q} is a similarity measure. It is also possible to maintain both the distance properties and the baseline adjustment using NVI\textsubscript{q} \( \equiv \) \( VI\textsubscript{q} /E[VI\textsubscript{q}] \) which can be seen as a normalization of VI\textsubscript{q} with the stochastic upper bound \( E[VI\textsubscript{q}] \) [Vinh et al., 2009]. It is also easy to see that AVI\textsubscript{q} = 1 − NVI\textsubscript{q}. The adjustments in Eq. (5.10) also enable the measures to be normalized. AMI\textsubscript{q} and AVI\textsubscript{q} achieve their maximum at 1 when \( U = V \) and their minimum is 0 when \( U \) and \( V \) are random partitions.

According to the chosen upper bound for MI\textsubscript{q}, we obtain the nice analytical form shown in Theorem 5.4. Our adjusted measures quantify the discrepancy between the values of the actual contingency table and their expected value in relation to the maximum discrepancy possible, i.e. the denominator in Eq. (5.11). It is also easy to see that all measures in \( L_\phi \) resemble this form when adjusted.

**Theorem 5.4.** Using \( E[n_{ij}^q] \) in Eq. (5.9) with \( \phi_{ij}(n_{ij}) = n_{ij}^q \), the adjustments for chance for MI\textsubscript{q}(\( U, V \)) and VI\textsubscript{q}(\( U, V \)) are:

\[
AMI\textsubscript{q}(U, V) = AVI\textsubscript{q}(U, V) = \frac{\sum_{ij} n_{ij}^q - \sum_{ij} E[n_{ij}^q]}{\frac{1}{2}(\sum_i a_i^q + \sum_j b_j^q) - \sum_{ij} E[n_{ij}^q]} \tag{5.11}
\]

**Proof.** The using the upper bound \( \frac{1}{2}(H_q(U) + H_q(V)) \) to MI\textsubscript{q}, AMI\textsubscript{q} and AVI\textsubscript{q} are equivalent. Therefore we compute AVI\textsubscript{q}. The denominator is equal to:

\[
E[VI\textsubscript{q}] = \frac{2}{(q-1)n^q} \left( \frac{1}{2} \left( \sum_i a_i^q + \sum_j b_j^q \right) - \sum_{i,j} E[n_{ij}^q] \right).
\]

The numerator is instead \( \frac{2}{(q-1)n^q} \left( \sum_{ij} n_{ij}^q - \sum_{i,j} E[n_{ij}^q] \right). \)

From now on we only discuss AMI\textsubscript{q}, given that it is identical to AVI\textsubscript{q}. There are notable special cases for our proposed adjusted generalized IT measures. In particular, the Adjusted Rand Index (ARI) [Hubert and Arabie, 1985] is just equal to AMI\textsubscript{2}. ARI is
a classic measure, heavily used for validation in social sciences and the most popular clustering validity index.

**Corollary 5.5.** It holds true that:

1. \( \lim_{q \to 1} \text{AMI}_q = \lim_{q \to 1} \text{AVI}_q = \text{AMI} = \text{AVI} \) with Shannon entropy;
2. \( \text{AMI}_2 = \text{AVI}_2 = \text{ARI} \).

**Proof.** Point (a) follows from the limit of the \( q \)-entropy when \( q \to 1 \). Point (b) follows from:

\[
\text{AVI}_2 = \frac{E[\text{VI}_2] - \text{VI}_2}{E[\text{VI}_2] - \min \text{VI}_2} = \frac{n-1}{n} \left( \frac{RI - E[RI]}{RI - E[RI]} \right) = \text{ARI}
\]

Therefore, using the permutation model we can perform baseline adjustment to generalized IT measures. Our generalized adjusted IT measures are a further generalization of particular well known adjusted measures such as AMI and ARI. It is worth noting, that ARI is equivalent to other well known measures for comparing partitions [Albatineh et al., 2006]. Furthermore, there is also a strong connection between ARI and Cohen’s \( \kappa \) statistics used to quantify inter-rater agreement [Warrens, 2008].

### 5.3.2 Computational complexity

Significant computational speedups might be obtained in computing the expected value of \( \text{AMI}_q \) in Eq. (5.11) if the probabilities are computed iteratively as:

\[
P(n_{ij} + 1) = P(n_{ij}) \frac{(a_i - n_{ij})(b_j - n_{ij})}{(n_{ij} + 1)(N - a_i - b_j + n_{ij} + 1)}.
\]

Therefore, the computational complexity of computing the expected value become linear in the number of records and symmetric in \( c \) and \( r \):

**Proposition 5.6.** The computational complexity of \( \text{AMI}_q \) is \( \mathcal{O}(n \cdot \max \{r, c\}) \).
Proof. The computation of $P(n_{ij})$ where $n_{ij}$ is a hypergeometric distribution Hyp($a_i, b_j, n$) is linear in $n$. However, the computation of the expected value $E[n^q_{ij}] = \sum_{n_{ij}} n^q_{ij} P(n_{ij})$ can exploit the fact that $P(n_{ij})$ are computed iteratively. We compute $P(n_{ij})$ only for $\max \{0, a_i + b_j - n\}$. In both cases $P(n_{ij})$ can be computed in $\mathcal{O}(\max \{a_i, b_j\})$. We can compute all other probabilities iteratively as shown above in constant time. Therefore:

$$
\sum_{i=1}^{r} \sum_{j=1}^{c} \left( \mathcal{O}(\max \{a_i, b_j\}) + \sum_{n_{ij}=0}^{\min \{a_i, b_j\}} \mathcal{O}(1) \right) = \sum_{i=1}^{r} \sum_{j=1}^{c} \mathcal{O}(\max \{a_i, b_j\})
$$

$$
= \sum_{i=1}^{r} \mathcal{O}(\max \{ca_i, n\}) = \mathcal{O}(\max \{cn, rn\}) = \mathcal{O}(n \cdot \max \{c, r\})
$$

If all the possible contingency tables $\mathcal{M}$ obtained by permutations were generated, the computational complexity of the exact expected value would be $\mathcal{O}(n!)$. However, this can be dramatically reduced using properties of the expected value.

5.3.3 Experiments on Measure Baseline

Here we show that our adjusted generalized IT measures have a baseline value of 0 when comparing random partitions $U$ and $V$. In Figure 5.4 we show the behaviour of $\text{AMI}_q$, ARI, and AMI on the same experiment proposed in Section 5.2.1. They are all close to 0 with negligible variation when the partitions are random and independent. Moreover, it is interesting to see the equivalence of $\text{AMI}_2$ and ARI. On the other hand, the equivalence of $\text{AMI}_q$ and AMI with Shannon’s entropy is obtained only at the limit $q \to 1$.

We also point out that $\text{NMI}_q$ does not show constant baseline when the relative size of the sets in $U$ varies when $U$ and $V$ are random. In Figure 5.5, we generate random partitions $V$ with $c = 6$ sets on $n = 100$ points, and random binary partitions $U$ independently. $\text{NMI}_q(U, V)$ shows different behavior at the variation of the relative size of the biggest set in $U$. This is unintuitive given that the partitions $U$ and $V$ are random and independent. We obtain the desired property of a baseline value of 0 with $\text{AMI}_q$. 
5.3.4 Large Number of Objects

In this section, we introduce a very general family of measures which includes $\mathcal{L}_\phi$. For measures belonging to this family, it is possible to find an approximation of their expected value when the number of objects $n$ is large. This allows us to identify approximations for the expected value of measures in $\mathcal{L}_\phi$, as well as for measures not in $\mathcal{L}_\phi$, such as the Jaccard coefficient as shown in Figure 5.2.

Let $\mathcal{N}_\phi$ be the family of measures which are non-linear combinations of $\phi_{ij}(n_{ij})$:

**Definition 5.2.** Let $\mathcal{N}_\phi$ be the family of similarity measures

$$S(U, V) = \phi\left(\frac{n_{11}}{n}, \ldots, \frac{n_{ij}}{n}, \ldots, \frac{n_{rc}}{n}\right)$$

where $\phi$ is a bounded real function as $n$ reaches infinity.
Note that $\mathcal{N}_\phi$ is a generalization of $\mathcal{L}_\phi$. At the limit of large number of objects $n$, it is possible to compute the expected value of measures in $\mathcal{N}_\phi$ under random partitions $U$ and $V$ using only the marginals of the contingency table $\mathcal{M}$.

**Lemma 5.7.** If $S(U, V) \in \mathcal{N}_\phi$, then

$$
\lim_{n \to +\infty} E[S(U, V)] = \phi\left(\frac{a_1 b_1}{n}, \ldots, \frac{a_i b_j}{n}, \ldots, \frac{a_r b_c}{n}\right).
$$

**Proof.** $S(U, V)$ can be written as $\phi(\frac{n_{i1}}{n}, \ldots, \frac{n_{ij}}{n}, \ldots, \frac{n_{rc}}{n})$. Let $X = (X_1, \ldots, X_{rc}) = (\frac{n_{i1}}{n}, \ldots, \frac{n_{ij}}{n}, \ldots, \frac{n_{rc}}{n})$ be a vector of $rc$ random variables where $n_{ij}$ is a hypergeometric distribution with the marginals as parameters: $a_i$, $b_j$ and $n$. The expected value of $\frac{n_{ij}}{n}$ is $E[\frac{n_{ij}}{n}] = \frac{1}{n} \frac{a_i b_j}{n}$. Let $\mu = (\mu_1, \ldots, \mu_{rc}) = (E[X_1], \ldots, E[X_{rc}]) = (\frac{a_1 b_1}{n}, \ldots, \frac{a_i b_j}{n}, \ldots, \frac{a_r b_c}{n})$ be the vector of the expected values. The Taylor approximation of $S(U, V) = \phi(X)$ around $\mu$ is:

$$
\phi(X) \simeq \phi(\mu) + \sum_{t=1}^{rc} (X_t - \mu_t) \frac{\partial \phi}{\partial X_t} + \frac{1}{2} \sum_{t=1}^{rc} \sum_{s=1}^{rc} (X_t - \mu_t)(X_s - \mu_s) \frac{\partial^2 \phi}{\partial X_t \partial X_s} + \ldots
$$

Its expected value is (see Section 4.3 of [Ang and Tang, 2006]):

$$
E[\phi(X)] \simeq \phi(\mu) + \frac{1}{2} \sum_{t=1}^{rc} \sum_{s=1}^{rc} \text{Cov}(X_t, X_s) \frac{\partial^2 \phi}{\partial X_t \partial X_s} + \ldots
$$

We just analyse the second order remainder given that it dominates the higher order ones. Using the Cauchy-Schwartz inequality we have that $|\text{Cov}(X_t, X_s)| \leq \sqrt{\text{Var}(X_t)\text{Var}(X_s)}$. Each $X_t$ and $X_s$ is equal to $\frac{n_{ij}}{n}$ for some indexes $i$ and $j$. The variance of each $X_t$ and $X_s$ is therefore equal to $\text{Var}(\frac{n_{ij}}{n}) = \frac{1}{n^2} \frac{a_i b_j n-a_i n-b_j}{n-1}$. When the number of records is large also the marginals increase: $n \to +\infty \Rightarrow a_i \to +\infty$, and $b_j \to +\infty \forall i, j$. However, because of the permutation model, all the fractions $\frac{a_i}{n}$ and $\frac{b_j}{n}$ stay constant $\forall i, j$. Therefore, also $\mu$ is constant. However, at the limit of large $n$, the variance of $\frac{n_{ij}}{n}$ tends to 0: $\text{Var}(\frac{n_{ij}}{n}) = \frac{1}{n^2} \frac{a_i b_j}{n} \left(1 - \frac{a_i}{n}\right) \left(1 + \frac{1}{n-1} - \frac{b_j}{n}\right) \to 0$. Therefore, at large $n$:

$$
E[\phi(X)] \simeq \phi(\mu) = \phi\left(\frac{a_1 b_1}{n}, \ldots, \frac{a_i b_j}{n}, \ldots, \frac{a_r b_c}{n}\right).
$$
In Morey and Agresti [1984] the expected value of the RI was computed using an approximated value based on the multinomial distribution. It turns out this approximated value is equal to what we obtain for RI using Lemma 5.7. The authors of [Albatineh et al., 2006] noticed that the difference between the approximation and the expected value obtained with the hypergeometric model is small on empirical experiments when \( n \) is large. We point out that this is a natural consequence of Lemma 5.7 given that \( \text{RI} \in L_\phi \subseteq N_\phi \). Moreover, the multinomial distribution was also used to compute the expected value of the Jaccard coefficient (\( J \)) in Albatineh and Niewiadomska-Bugaj [2011], obtaining good results on empirical experiments with many objects. Again, this is a natural consequence of Lemma 5.7 given that \( J \not\in L_\phi \).

Generalized IT measures belong in \( L_\phi \subseteq N_\phi \). Therefore we can employ Lemma 5.7. When the number of objects is large, the expected value under random partitions \( U \) and \( V \) of \( H_q(U, V) \), \( MI_q(U, V) \), and \( VI_q(U, V) \) in Theorem 5.3 depends only on the entropy of the partitions \( U \) and \( V \):

**Theorem 5.8.** It holds true that:

(a) \( \lim_{n \to +\infty} E[H_q(U, V)] = H_q(U) + H_q(V) - (q - 1)H_q(U)H_q(V) \);

(b) \( \lim_{n \to +\infty} E[MI_q(U, V)] = (q - 1)H_q(U)H_q(V) \);

(c) \( \lim_{n \to +\infty} E[VI_q(U, V)] = H_q(U) + H_q(V) - 2(q - 1)H_q(U)H_q(V) \).

**Proof.** \( E[H_q(U, V)] = \frac{1}{q-1} \left(1 - \sum_{ij} E\left[\left(\frac{a_i}{n}\right)^q \left(\frac{b_j}{n}\right)^q\right]\right) \) and according to Lemma 5.7 for large \( n \): \( E[H_q(U, V)] \simeq \frac{1}{q-1} \left(1 - \sum_{ij} \left(\frac{a_i}{n}\right)^q \left(\frac{b_j}{n}\right)^q\right) = \frac{1}{q-1} \left(1 - \sum_i \left(\frac{a_i}{n}\right)^q \sum_j \left(\frac{b_j}{n}\right)^q\right) \). If we add an subtract \( 1 - \sum_i \left(\frac{a_i}{n}\right)^q \) and \( \sum_j \left(\frac{b_j}{n}\right)^q \) in the parenthesis above:

\[
E[H_q(U, V)] \simeq \frac{1}{q-1} \left(1 - \sum_i \left(\frac{a_i}{n}\right)^q \sum_j \left(\frac{b_j}{n}\right)^q\right) \]

\[
+ 1 - \sum_i \left(\frac{a_i}{n}\right)^q - \sum_j \left(\frac{b_j}{n}\right)^q \]

\[
- 1 + \sum_i \left(\frac{a_i}{n}\right)^q + \sum_j \left(\frac{b_j}{n}\right)^q \]

\[
= \frac{1}{q-1} \left(1 - \sum_i \left(\frac{a_i}{n}\right)^q \sum_j \left(\frac{b_j}{n}\right)^q\right) \]
\[
\begin{align*}
&= \frac{1}{q-1} \left( 1 - \sum_i \left( \frac{a_i}{n} \right)^q \right) + \frac{1}{q-1} \left( 1 - \sum_j \left( \frac{b_j}{n} \right)^q \right) \\
&+ \frac{1}{q-1} \left( -1 - \sum_i \left( \frac{a_i}{n} \right)^q \sum_j \left( \frac{b_j}{n} \right)^q + \sum_i \left( \frac{a_i}{n} \right)^q + \sum_j \left( \frac{b_j}{n} \right)^q \right) \\
&= H_q(U) + H_q(V) + \frac{1}{q-1} \left( \left( 1 - \sum_i \left( \frac{a_i}{n} \right)^q \right) \left( \sum_j \left( \frac{b_j}{n} \right)^q \right) \right) \\
&= H_q(U) + H_q(V) - (q-1)H_q(U)H_q(V)
\end{align*}
\]

Point (b) and (c) follow from Equations (5.5) and (5.6).

Result (a) recalls the property of non-additivity that holds true for random variables [Furuichi, 2006]. Figure 5.6 shows the behaviour of \(E[H_q(U, V)]\) when the partitions \(U\) and \(V\) are generated uniformly at random. \(V\) has \(c = 6\) sets and \(U\) has \(r\) sets. On this case, \(H_q(U) + H_q(V) - (q-1)H_q(U)H_q(V)\) appears to be a good approximation already for \(n = 1000\). In particular, the approximation is good when the number of records \(n\) is big with regards to the number of cells of the contingency table in Table 2.3: i.e., when \(\frac{n}{rc}\) is large enough.

Figure 5.6: \(E[H_q(U, V)]\) (solid) and their limit value \(H_q(U) + H_q(V) - (q-1)H_q(U)H_q(V)\) (dashed). The solid line coincides approximately with the dashed one in 5.6b when \(n = 1000\). The limit value is a good approximation for \(E[H_q(U, V)]\) when \(\frac{n}{rc}\) is large enough.
5.4 application scenarios for AMI\textsubscript{$q$}

In this section we aim to answer to the question: Given a reference ground truth clustering $V$, which is the best choice for $q$ in AMI\textsubscript{$q$}(U, V) to validate the clustering solution $U$? By answering this question, we implicitly identify the application scenarios for ARI and AMI given the results in Corollary 5.5. This is particularly important for external clustering validation. Nonetheless, there are a number of other applications where the task is to find the most similar partition to a reference ground truth partition: e.g., categorical feature selection [Vinh et al., 2014], decision tree induction [Criminisi et al., 2012], generation of alternative or multi-view clusterings [Müller et al., 2013], or the exploration of the clustering space with the Meta-Clustering algorithm [Caruana et al., 2006; Lei et al., 2014b] to list a few.

Different values for $q$ in AMI\textsubscript{$q$} yield to different biases. The source of these biases can be identified by analyzing the properties of the $q$-entropy. In Figure 5.7 we show the $q$-entropy for a binary partition at the variation of the relative size $p$ of one cluster. This can be analytically computed: $H_q(p) = \frac{1}{q-1}(1 - p^q - (1 - p)^q)$. The range of variation for $H_q(p)$ is much bigger if $q$ is small. More specifically when $q$ is small, the difference in entropy between an unbalanced partition and a balanced partition is big.

Let us focus on an example. Let $V$ be a reference clustering with 3 clusters of size 50 each, and let $U_1$ and $U_2$ be two clustering solutions with the same number of clusters and same cluster sizes. The contingency tables for $U_1$ and $U_2$ are shown on Figure 5.8. Given that both contingency tables have the same marginals, the only difference between AMI\textsubscript{$q$}($U_1, V$) and AMI\textsubscript{$q$}($U_2, V$) according to Eq. (5.10) lies in MI\textsubscript{$q$}. Given

![Figure 5.7: Tsallis $q$-entropy $H_q(p)$ for a binary clustering where $p$ is the relative size of one cluster. When $q$ is small, the $q$-entropy varies in bigger range.](image-url)
that both solutions $U_1$ and $U_2$ are compared against $V$, the only term that varies in $\text{MI}_q(U, V) = H_q(V) - H_q(V|U)$ is $H_q(V|U)$. In order to identify the clustering solution that maximizes $\text{AMI}_q$ we have to analyze the solution that decreases $H_q(V|U)$ the most. $H_q(V|U)$ is a weighted average of the entropies $H_q(V|u_i)$ computed on the rows of the contingency table as shown in Eq. (5.4), and this is sensitive to values equal to 0. Given the bigger range of variation of $H_q$ for small $q$, small $q$ implies higher sensitivity to row entropies of 0. Therefore, small values of $q$ tends to decrease $H_q(V|U)$ much more if the clusters in the solution $V$ are pure: i.e., clusters contain elements from only one cluster in the reference clustering $V$. In other words, $\text{AMI}_q$ with small $q$ prefers pure clusters in the clustering solution.

When the marginals in the contingency tables for two solutions are different, another important factor in the computation of $\text{AMI}_q$ is the normalization coefficient \( \frac{1}{2}(H_q(U) + H_q(V)) \). Balanced solutions $V$ will be penalized more by $\text{AMI}_q$ when $q$ is small. Therefore, $\text{AMI}_q$ with small $q$ prefers unbalanced clustering solutions. To summarize, $\text{AMI}_q$ with small $q$ such as $\text{AMI}_{0.5}$ or $\text{AMI}_1 = \text{AMI}$ with Shannon’s entropy:

- Is biased towards pure clusters in the clustering solutions;
- Prefers unbalanced clustering solutions.

By contrary, $\text{AMI}_q$ with bigger $q$ such as $\text{AMI}_{2.5}$ or $\text{AMI}_2 = \text{ARI}$:

- Is less biased towards pure clusters in the clustering solution;
- Prefers balanced clustering solutions.

Given a reference clustering $V$, these biases can guide the choice of $q$ in AMI$_q$ to identify more suitable clustering solutions.

5.4.1 *Use AMI$_q$ with small $q$ such as AMI$_{0.5}$ or AMI$_1 = AMI$ when the reference clustering is unbalanced and there exist small clusters*

If the reference cluster $V$ is unbalanced and presents small clusters, AMI$_q$ with small $q$ might prefer more appropriate clustering solutions $U$. For example, in Figure 5.9 we show two contingency tables associated to two clustering solutions $U_1$ and $U_2$ for the reference clustering $V$ with 4 clusters of size $[10, 10, 10, 70]$ respectively. When there exist small clusters in the reference $V$ their identification has to be *precise* in the clustering solution. The solution $U_1$ looks arguably better than $U_2$ because it shows many pure clusters. In this scenario we advise the use of AMI$_{0.5}$ or AMI$_1 = AMI$ with Shannon’s entropy because it gives more weight to the clustering solution $U_1$.

![Contingency Tables](image)

Figure 5.9: AMI$_q$ with small $q$ prefers the solution $U_1$ because its clusters are pure. When the reference clustering has small clusters their identification in the solution has to be *precise*. In this scenario we advise the use of AMI$_{0.5}$ or AMI$_1 = AMI$. 

5.4.2 Use $\text{AMI}_q$ with big $q$ such as $\text{AMI}_{2.5}$ or $\text{AMI}_2 = \text{ARI}$ when the reference clustering has big equal sized clusters

If $V$ is a reference clustering with big equal size clusters it is less crucial to have precise clusters in the solution. Indeed, precise clusters in the solution penalize the recall of clusters in the reference. In this case, $\text{AMI}_q$ with bigger $q$ might prefer more appropriate solutions. In Figure 5.10 we show two clustering solutions $U_1$ and $U_2$ for the reference clustering $V$ with 4 equal size clusters of size 25. The solution $U_2$ looks better than $U_1$ because each of its clusters identifies more elements from particular clusters in the reference. Moreover, $U_2$ has to be preferred to $U_1$ because it consists in 4 equal sized clusters like for the reference clustering $V$. In this scenario we advise the use of $\text{AMI}_{2.5}$ or $\text{AMI}_2 = \text{ARI}$ because it gives more importance to the solution $U_2$.

5.5 Standardization of clustering comparison measures

Selection of the most similar partition $U$ to a reference partition $V$ is biased according to the chosen similarity measure, the number of sets $r$ in $U$, and their relative size. This phenomena is known as selection bias and it has been extensively studied in decision trees [White and Liu, 1994]. Researchers in this area agree that in order to achieve unbiased selection of partitions, distribution properties of similarity measures have to
be taken into account [Dobra and Gehrke, 2001; Shih, 2004; Hothorn et al., 2006]. Using
the permutation model to compute expected value and variance of measures in \( L \), we
now aim to analytically standardize various clustering comparison measures using the
approach discussed in Chapter 4 Section 4.4.

To standardize measures \( S(U, V) \) we must analytically compute their variance:

**Lemma 5.9.** If \( S(U, V) \in L_{\phi} \), when partitions \( U \) and \( V \) are random:

\[
\text{Var}(S(U, V)) = \beta^2 \left( E\left[ \left( \sum_{ij} \phi_{ij}(n_{ij}) \right)^2 \right] - \left( \sum_{ij} E[\phi_{ij}(n_{ij})] \right)^2 \right)
\]

where \( E\left[ \left( \sum_{ij} \phi_{ij}(n_{ij}) \right)^2 \right] \) is equal to

\[
\sum_{ij} \sum_{n_{ij}} \phi(n_{ij})P(n_{ij}) \cdot \left[ \phi_{ij}(n_{ij}) + \sum_{i' \neq i} \sum_{n_{i'j}} \phi_{i'j}(n_{i'j})P(n_{i'j}|n_{ij}) + \right.
\]

\[
+ \sum_{j' \neq j} \sum_{n_{ij'}} P(n_{ij'}|n_{ij}) \left( \phi_{ij'}(n_{ij'}) + \right.
\]

\[
\left. \left. + \sum_{i' \neq i} \sum_{n_{i'j'}} \phi_{i'j'}(n_{i'j'})P(n_{i'j'}|n_{ij}, n_{ij}) \right] \right) \\
\text{(5.12)}
\]

where

\[
n_{ij} \sim \text{Hyp}(a_i, b_j, n)
\]

\[
n_{i'j'}|n_{ij} \sim \text{Hyp}(b_j - n_{ij}, a_{i'}, n - a_i)
\]

\[
n_{ij'}|n_{ij} \sim \text{Hyp}(a_i - n_{ij}, b_{j'}, n - b_j)
\]

\[
n_{i'j'}|n_{ij'}, n_{ij} \sim \text{Hyp}(a_{i'}, b_{j'} - n_{ij'}, n - a_i)
\]

are hypergeometric random variables.

**Proof.** Using the properties of the variance we can show that

\[
\text{Var}(S(U, V)) = \beta^2 \text{Var}(\sum_{ij} \phi_{ij}(n_{ij})) = \beta^2 \left( E[(\sum_{ij} \phi_{ij}(n_{ij}))^2] - \left( \sum_{ij} E[\phi_{ij}(n_{ij})] \right)^2 \right).
\]
\((E[\sum_{ij} \phi_{ij}(n_{ij})])^2 = (\sum_{ij} E[\phi_{ij}(n_{ij})])^2\) can be computed using Eq. (5.9). The first term in the sum is instead:

\[
E[(\sum_{ij} \phi_{ij}(n_{ij}))^2] = \sum_{ij} \sum_{i'j'} E[\phi_{ij}(n_{ij})\phi_{i'j'}(n_{i'j'})]
\]

\[
= \sum_{ij} \sum_{i'j'} \sum_{n_{ij}} \sum_{n_{i'j'}} \phi_{ij}(n_{ij})\phi_{i'j'}(n_{i'j'}) P(n_{ij}, n_{i'j'})
\]

We cannot find the exact form of the joint probability \(P(n_{ij}, n_{i'j'})\) thus we rewrite it as \(P(n_{ij})P(n_{i'j'}|n_{ij})\). The random variable \(n_{ij}\) is an hypergeometric distribution that simulates the experiment of sampling without replacement the \(a_i\) objects in the set \(u_i\) from a total of \(n\) objects. Sampling one of the \(b_j\) objects from \(v_j\) is defined as success: \(n_{ij} \sim \text{Hyp}(a_i, b_j, n)\). The random variable \(n_{i'j'}|n_{ij}\) has a different distribution depending on the possible combinations of indexes \(i, i', j, j'\). Thus

\[
E[(\sum_{ij} \phi_{ij}(n_{ij}))^2] = \sum_{ij} \sum_{n_{ij}} \phi_{ij}(n_{ij}) \sum_{i'j'} \phi_{i'j'}(n_{i'j'}) P(n_{ij}, n_{i'j'})
\]

\[
= \sum_{ij} \phi_{ij}(n_{ij}) P(n_{ij}) \sum_{i'j'} \phi_{i'j'}(n_{i'j'}) P(n_{i'j'}|n_{ij})
\]

which, by taking care of all possible combinations of \(i, i', j, j'\), is equal to:

\[
\sum_{ij} \sum_{n_{ij}} \phi_{ij}(n_{ij}) P(n_{ij}) \cdot \left[ \sum_{i'=i, j'=j} \sum_{n_{i'j'}} \phi_{i'j'}(n_{i'j'}) P(n_{i'j'}|n_{ij}) + \sum_{i' \neq i, j'=j} \sum_{n_{i'j'}} \phi_{i'j'}(n_{i'j'}) P(n_{i'j'}|n_{ij}) + \sum_{i'=i, j' \neq j} \sum_{n_{i'j'}} \phi_{i'j'}(n_{i'j'}) P(n_{i'j'}|n_{ij}) + \sum_{i' \neq i, j' \neq j} \sum_{n_{i'j'}} \phi_{i'j'}(n_{i'j'}) P(n_{i'j'}|n_{ij}) \right]
\]

(5.13)

In order to make the derivation of the distribution \(n_{i'j'}|n_{ij}\) easier to follow, we employ an urn model with red, blue, and white marbles as illustrative example. Figure 5.11 represents such urn containing \(n\) total marbles among which \(b_j\) are red, \(b_{j'}\) are blue, and \(n - b_j - b_{j'}\) are white. This urn is used to simulate the sampling experiment without
replacement modeled by the hypergeometric distribution. For example, it is easy to see that the random variable $n_{ij} \sim \text{Hyp}(a_i, b_j, n)$ models the probability of obtaining $n_{ij}$ red marbles among $a_i$ drawn from an urn of $n$ marbles among which $b_j$ are red.

The distribution of $n_{i'j'}$ can be found according to the following cases:

**Case 1:** $i' = i \land j' = j$

$P(n_{i'j'}|n_{ij}) = 1$ if and only if $n_{ij} = n_{ij}$ and $0$ otherwise. This case produces the first term $\phi_{ij}(n_{ij})$ enclosed in square brackets.

**Case 2:** $i' = i \land j' \neq j$

In this case, the possible successes are the $b_{j'}$ blue marbles. We have already sampled $n_{ij}$ red marbles and we are not interested in red marbles any more, thus the total ones available are $n - b_j$. Thus, $n_{ij'}|n_{ij} \sim \text{Hyp}(a_i - n_{ij}, b_{j'}, n - b_j)$.

**Case 3:** $i' \neq i \land j' = j$

This case is symmetric to the previous one where $a_{i'}$ is now the possible number of successes. Therefore $n_{i'j}|n_{ij} \sim \text{Hyp}(b_j - n_{ij}, a_{i'}, n - a_i)$.

**Case 4:** $i' \neq i \land j' \neq j$
This is the most complicated case. When all indexes are different we cannot write \( n_{ij'}|n_{ij} \) as a single hypergeometric distribution. In order compute \( P(n_{i'j'}|n_{ij}) \), we have to impose a further condition:

\[
P(n_{i'j'}|n_{ij}) = \sum_{n_{ij'}} P(n_{i'j'}|n_{ij'}, n_{ij}) P(n_{ij'}|n_{ij})
\]

The distribution of \( n_{i'j'}|n_{ij}, n_{ij} \) can be computed. We might think about this scenario as the second draw from the urn in Figure 5.11. We have already sampled \( a_i \) marbles focusing on the red ones as successes. We are now going to sample other \( a_{i'} \) marbles but focusing on blue ones as successes. Just knowing that \( n_{ij} \) red ones have already been sampled does not allow us to know how many blue ones remain in the urn. Indeed, only with that information we can obtain the hypergeometric distribution. If we know that \( n_{ij'} \) blue marbles have already been sampled we know there are \( b_{j'} - n_{ij'} \) possible successes and thus \( n_{i'j'}|n_{ij', n_{ij}} \sim \text{Hyp}(a_{i'}, b_{j'} - n_{ij'}, n - a_i) \). So the last two terms in Eq. (5.13) can be put together:

\[
\sum_{i'\neq i, j'\neq j} \phi_{ij'}(n_{ij'}) P(n_{ij'}|n_{ij}) + \sum_{i'\neq i, j'\neq j} \phi_{i'j'}(n_{i'j'}) P(n_{i'j'}|n_{ij})
\]

\[
= \sum_{j'\neq j} \sum_{n_{ij'}} P(n_{ij'}|n_{ij}) \phi_{ij'}(n_{ij'}) + \sum_{j'\neq j} \sum_{n_{ij'}} P(n_{i'j'}|n_{ij'}) \sum_{i'\neq i} \phi_{i'j'}(n_{i'j'}) P(n_{i'j'}|n_{ij'})
\]

\[
= \sum_{j'\neq j} \sum_{n_{ij'}} P(n_{ij'}|n_{ij}) \left( \phi_{ij'}(n_{ij'}) + \sum_{i'\neq i} \phi_{i'j'}(n_{i'j'}) P(n_{i'j'}|n_{ij'}) \right)
\]

By putting everything together we get that \( E[(\sum_{ij} \phi_{ij}(n_{ij}))^2] \) is equal to:

\[
\sum_{ij} \phi(n_{ij}) P(n_{ij}) \cdot \left[ \phi_{ij}(n_{ij}) + \sum_{i'\neq i} \phi_{i'j}(n_{i'j}) P(n_{i'j}|n_{ij}) + \sum_{j'\neq j} \phi_{ij'}(n_{ij'}) P(n_{ij'}|n_{ij}) \right]
\]
We can use the expected value to standardize $S \in L_\phi$, such as generalized IT measures.

### 5.5.1 Standardization of Generalized IT Measures

The variance under the permutation model of generalized IT measures is:

**Theorem 5.10.** Using Eqs. (5.9) and (5.12) with $\phi_{ij}(\cdot) = (\cdot)^q$, when the partitions $U$ and $V$ are random:

(a) $\text{Var}(H_q(U, V)) = \frac{1}{(q-1)n^q} \left( E[(\sum_{ij} n_{ij}^q)^2] - (\sum_{ij} E[n_{ij}^q])^2 \right)$;

(b) $\text{Var}(\text{MI}_q(U, V)) = \text{Var}(H_q(U, V))$

(c) $\text{Var}(\text{VI}_q(U, V)) = 4\text{Var}(H_q(U, V))$

**Proof.** The results follow from Lemma 5.9, the hypothesis of fixed marginals and properties of the variance.

We define the standardized version of the similarity measure $\text{MI}_q$ ($\text{SMI}_q$), and the standardized version of the distance measure $\text{VI}_q$ ($\text{SVI}_q$) as follows:

$$\text{SMI}_q \triangleq \frac{\text{MI}_q - E[\text{MI}_q]}{\sqrt{\text{Var}(\text{MI}_q)}}, \quad \text{SVI}_q \triangleq \frac{E[\text{VI}_q] - \text{VI}_q}{\sqrt{\text{Var}(\text{VI}_q)}},$$  \hspace{1cm} (5.14)

As for the case of $\text{AMI}_q$ and $\text{AVI}_q$, it turns out that $\text{SMI}_q$ is equal to $\text{SVI}_q$:

**Theorem 5.11.** Using Eqs. (5.9) and (5.12) with $\phi_{ij}(\cdot) = (\cdot)^q$, the standardized $\text{MI}_q(U, V)$ and the standardized $\text{VI}_q(U, V)$ are:

$$\text{SMI}_q(U, V) = \text{SVI}_q(U, V) = \frac{\sum_{ij} n_{ij}^q - \sum_{ij} E[n_{ij}^q]}{\sqrt{E[(\sum_{ij} n_{ij}^q)^2] - (\sum_{ij} E[n_{ij}^q])^2}}$$  \hspace{1cm} (5.15)
Proof.

For $\text{SMI}_q$: the numerator is equal to $H_q(U, V) - E[H_q(U, V)] = \frac{1}{(q-1)n^q} \left( \sum_{ij} n_{ij}^q - \sum_{i,j} E[n_{ij}^q] \right)$. According to Theorem 5.10, the denominator is instead:

$$\sqrt{\text{Var}(\text{MI}_q(U, V))} = \sqrt{\text{Var}(H_q(U, V))} = \frac{1}{(q-1)n^q} \sqrt{E[(\sum_{ij} n_{ij}^q)^2] - (E[\sum_{ij} n_{ij}^q])^2}.$$ 

For $\text{SVI}_q$: the numerator is equal to $2H_q(U, V) - 2E[H_q(U, V)] = \frac{2}{(q-1)n^q} \left( \sum_{ij} n_{ij}^q - \sum_{i,j} E[n_{ij}^q] \right)$. According to Theorem 5.10, the denominator is instead:

$$\sqrt{\text{Var}(\text{VI}_q(U, V))} = \sqrt{4\text{Var}(H_q(U, V))} = \frac{2}{(q-1)n^q} \sqrt{E[(\sum_{ij} n_{ij}^q)^2] - (E[\sum_{ij} n_{ij}^q])^2}.$$ 

Therefore, $\text{SMI}_q$ and $\text{SVI}_q$ are equivalent.

This formula shows that we are interested in maximizing the difference between the sum of the cells of the actual contingency table and the sum of the expected cells under randomness. However, standardized measures differs from their adjusted counterpart because of the denominator, i.e. the standard deviation of the sums of the cells. Indeed, $\text{SMI}_q$ and $\text{SVI}_q$ measure the number of standard deviations $\text{MI}_q$ and $\text{VI}_q$ are from their mean.

An interesting point to note is that, standardization unifies several existing information theoretic measures for clustering. To see this, let us define the Standardized Mutual Information, the Standardized Variation of Information (SVI), and Standardized $G$-statistic (SG) using Shannon’s entropy:

$$\text{SMI} \triangleq \frac{\text{MI} - E[\text{MI}]}{\sqrt{\text{Var}(\text{MI})}}, \quad (5.16)$$

$$\text{SVI} \triangleq \frac{E[\text{VI}] - \text{VI}}{\sqrt{\text{Var}(\text{VI})}}. \quad (5.17)$$
and
\[ SG \triangleq \frac{G - E[G]}{\sqrt{\text{Var}(G)}} \quad (5.18) \]

**Theorem 5.12.** Standardization unifies the mutual information MI, variation of information VI and the G-statistic: \( \text{SMI} = \text{SVI} = SG \).

**Proof.** \( H(U) \) and \( H(V) \) are constant under the fixed marginal assumption, thus the VI is a linear function of the MI under the hypergeometric hypothesis. The G-statistic is equal to a linear scaling of the MI \(( G = 2n \cdot \log_e (2) \cdot \text{MI}) \). The standardized version of a linear function of MI is equal to SMI because of the properties of expected value and variance. \( \square \)

This ‘unification’ property is useful, recalling that for the normalized mutual information NMI and adjusted mutual information AMI, depending on the upperbound used, there can be as many as 5 different variants for each measure Vinh et al. [2010].

Our generalized standardization of IT measures allows us to generalize the standardized information theoretic measures defined above and also the standardization of pair-counting measures such as the Rand Index. To see this, let us define the Standardized Rand Index (SRI):

\[ \text{SRI} \triangleq \frac{RI - E[RI]}{\sqrt{\text{Var}(RI)}} \quad (5.19) \]

**Corollary 5.13.** It holds true that:

(a) \( \lim_{q \to 1} \text{SMI}_q = \lim_{q \to 1} \text{SVI}_q = \text{SMI} = \text{SVI} = SG \) with Shannon entropy;

(b) \( \text{SMI}_2 = \text{SVI}_2 = \text{SRI} \).

**Proof.** Point (a) follows from the limit of the \( q \)-entropy when \( q \to 1 \) and the linear relation of G-statistic to MI: \( G = 2n \log_e \text{eMI} \). Point (b) follows from:

\[ \text{SVI}_2 = \frac{E[\text{VI}_2] - \text{VI}_2}{\sqrt{\text{Var}(\text{VI}_2)}} = \frac{N-1}{N} \left( \frac{\text{RI} - E[\text{RI}]}{\sqrt{\text{Var}(\text{RI})}} \right) = \text{SRI} \]

\( \square \)
5.5.2 Computational complexity

The computational complexity of $\text{SMI}_q$ is dominated by computation of the second moment of the sum of the cells defined in Eq. (5.12):

**Proposition 5.14.** The computational complexity of $\text{SVI}_q$ is $O(n^3c \cdot \max\{c, r\})$.

**Proof.** Each summation in Eq. (5.12) can be bounded above by the maximum value of the cell marginals and each sum can be done in constant time. The last summation in Eq. (5.12) is:

$$
\sum_{j' = 1}^{c} \sum_{n_{ij'} = 0}^{\max\{a_i, b_{j'}\}} \sum_{i' = 1}^{r} \sum_{n_{i'j'} = 0}^{\max\{a_{i'}, b_{j'}\}} O(1) = \sum_{j' = 1}^{c} \sum_{n_{ij'} = 0}^{\max\{a_i, b_{j'}\}} O(\max\{n, rb_{j'}\})
$$

$$
= \sum_{j' = 1}^{c} O(\max\{a_in, a_irb_{j'}, b_{j'n}, rb_{j'}^2\})
$$

$$
= O(\max\{ca_in, a_inr, rbn^2\})
$$

The above term is thus the computational complexity of the inner loop. Using the same machinery one can prove that:

$$
\sum_{j = 1}^{c} \sum_{i = 1}^{r} \sum_{n_{ij} = 0}^{\max\{a_i, b_{j}\}} O(\max\{ca_in, a_irn, rbn^2\}) = O(\max\{c^2n^3, rcn^3\}) = O(n^3c \cdot \max\{c, r\})
$$

Note that the complexity is quadratic in $c$ and linear in $r$. This happens because of the way we decided to condition the probabilities in Eq. (5.12) in the proof of Lemma 5.9. With different conditions, it is possible to obtain a formula symmetric to Eq. (5.12) with complexity $O(n^3r \cdot \max\{r, c\})$. Therefore, if the number of columns $c$ in the contingency table is greater than the number of rows $r$, a longer computational time is incurred. For example, if we fix the number of records $n$, the computation time for the variance of MI for a contingency table with $r = 6$ rows and $c = 2$ columns is bounded above by
5.5 Standardization of Clustering Comparison Measures

12n^3. Yet, for the same but transposed table (with \( r = 2 \) rows and \( c = 6 \) columns), the time is bounded by 36n^3. Given that we can transpose a contingency table and obtain identical variance results for MIq, we can always transpose to tables where the number of rows \( r \) is higher than the number of columns \( c \), thus making the computation faster.

5.5.3 Large Number of Objects

It is likely to expect that the variance of generalized IT measures decreases when partitions are generated on a large number of objects \( n \). Here we prove a general result about measures of the family \( N_\phi \).

**Lemma 5.15.** If \( S(U, V) \in N_\phi \), then \( \lim_{n \to +\infty} \text{Var}(S(U, V)) = 0 \).

**Proof.** Let \( X = (X_1, \ldots, X_{rc}) = (n_{i1}/n, \ldots, n_{ij}/n, \ldots, n_{rc}/n) \) be a vector of \( rc \) random variables where \( n_{ij} \) is a hypergeometric distribution with the marginals as parameters: \( a_i \), \( b_j \) and \( n \). Using the Taylor approximation [Ang and Tang, 2006] of \( S(U, V) = \phi(X) \), it is possible to show that:

\[
\text{Var}(\phi(X)) \approx \sum_{t=1}^{rc} \sum_{s=1}^{rc} \text{Cov}(X_t, X_s) \frac{\partial \phi}{\partial X_t} \frac{\partial \phi}{\partial X_s} + \ldots
\]

Using the Cauchy-Schwarz inequality we have that \( |\text{Cov}(X_t, X_s)| \leq \sqrt{\text{Var}(X_t) \text{Var}(X_s)} \).

Each \( X_t \) and \( X_s \) is equal to \( n_{ij}/n \) for some indexes \( i \) and \( j \). The variance of each \( X_t \) and \( X_s \) is therefore equal to \( \text{Var}(n_{ij}/n) = \frac{1}{n^2} \frac{a_i b_j n - a_i n - b_j n}{n-1} \).

When the number of records is large also the marginals increase: \( n \to +\infty \Rightarrow a_i \to +\infty \), and \( b_j \to +\infty \) \( \forall i, j \). However, because of the permutation model, all the fractions \( \frac{a_i}{n} \) and \( \frac{b_j}{n} \) stay constant \( \forall i, j \). Therefore, at the limit of large \( n \), the variance of \( n_{ij}/n \) tends to 0: \( \text{Var}(n_{ij}/n) = \frac{1}{n} \frac{a_i b_j}{n} \left( 1 - \frac{a_i}{n} \right) \left( 1 + \frac{1}{n-1} - \frac{b_j}{n} \right) \to 0 \) and thus \( \text{Var}(\phi(X)) \) tends to 0.

Given that generalized IT measures belong in the family \( N_\phi \), we can prove the following:

**Theorem 5.16.** It holds true that:

\[
\lim_{n \to +\infty} \text{Var}(H_q(U, V)) = \lim_{n \to +\infty} \text{Var}(\text{MI}_q(U, V)) = \lim_{n \to +\infty} \text{Var}(\text{VI}_q(U, V)) = 0 \quad (5.20)
\]
Proof. Trivially follows from Lemma 5.15.

Therefore, \( \text{SMI}_q \) attains very large values when \( n \) is large. In practice of course, \( n \) is finite, so the use of \( \text{SMI}_q \) is beneficial.

5.6 Experiments on Standardized Clustering Comparison Measures

In this section, we first carry out some experiments with the Standardized Mutual Information (SMI). Then, we conduct experiments with \( \text{SMI}_q \), and we provide guidelines for their application.

5.6.1 Interpretation of Standardized Clustering Comparison Measure

We provide an example that highlights how SMI can be interpreted. We are aiming to determine whether a clustering solution (Clustering \( U \)) is significant compared to a random solution when performing external validation against a reference clustering (Clustering \( V \)). Both clusterings consist of 2 clusters of equal size 50 and their cluster overlap is represented in the contingency table in Table 5.2. The agreement between

\[
\begin{array}{c|cc}
& V & \\hline \\
U & 50 & 50 \\
50 & 47 & 3 \\
50 & 3 & 47 \\
\end{array}
\]

Table 5.2: Contingency table related to clusterings \( U \) and \( V \) that show high agreement. SMI value is 64.22 which highlights that the clustering solution \( U \) is significantly better than a random clustering solution.

the two clusterings is very high, indeed cluster \( u_1 \) in \( U \) shares 47 elements with cluster \( v_1 \) in \( V \), and 47 elements are also shared between cluster \( u_2 \) in \( U \) and \( v_2 \) in \( V \). This characteristic is highlighted by SMI, which has a value of 64.22. It means that the MI between \( U \) and \( V \) is 64 standard deviations away from the mean value under the hypothesis of random and independent clusterings and therefore highly significant.
5.6 Experiments on Standardized Clustering Comparison Measures

In order to achieve even more interpretability, a $p$-value for mutual information might be obtained by fitting a distribution parametrized on the mean and the standard deviation. Good candidates might be the Gamma and the Normal distributions [Dobra and Gehrke, 2001; Vinh et al., 2009]. However, there are no theoretical proofs about the quality of these approximations available in literature. A conservative approach we can take is to use Cantelli’s inequality as for Proposition 4.1 from Chapter 4. This holds for all distributions:

$$p\text{-value} < \frac{1}{1 + (\text{SMI}(U, V))^2}$$

This inequality states that if SMI is greater than 4.36, then the upper bound for the $p$-value under the hypergeometric null hypothesis is 0.05. In the above example, we get a $p$-value of $\sim 0.0002$, which again is highly significant.

5.6.2 Bias Towards Clusterings with More Clusters

Consider the scenario where the user has to compare some clustering solutions to a reference one and select the one that agrees with it the most using information theoretic measures. Each solution might have been obtained using different clustering algorithms, or setting different parameters for a single algorithm of choice, e.g. varying $k$ in $k$-Means. Using MI and AMI, clustering solutions with more clusters have more chances to be selected. We have observed that although the AMI has a constant baseline of 0, its variance increases as the number of clusters increases, thus creating a bias towards clustering with more clusters.

Let $V$ be a reference clustering of $n = 500$ records with 4 equal size clusters. If we randomly generate clusterings $U$ with different number of clusters $c$ independently from the reference one, we do not expect any clustering solution to outperform the others in terms of agreement with $V$. We carry out an experiment as follows: we generate a pool of 9 random clusterings $U$ with different numbers of clusters $r = 2, 3, 4, 5, 6, 7, 8, 9, 10$ and give a win to the solution that obtains the higher value for respectively SMI, AMI, and MI against the reference clustering $V$. If a measure is unbiased, we expect that each
clustering is selected as often as the others, that is \( \frac{1}{9} = 11.1\% \) of the time. Figure 5.12 shows the estimated ‘winning’ frequencies obtained from 5,000 trials. We can see that random clusterings \( U \) with 10 clusters are selected more than 50% of the time if we use the MI. Even if we use the adjusted-for-chance AMI, such clusterings are selected 15% of the time versus the 7% for the random ones with 2 clusters. As observed, SMI helps to decrease this bias significantly. SMI shows close to constant probability of selection across different solutions but negligible differences might still exists because we are not using the full distribution of MI.
5.6.3 Bias Towards Clusterings Estimated on Fewer Data Points

Clustering solutions might also be induced on different numbers of data points. This is the application scenario commonly encountered in modern data processing tasks, such as streaming or distributed data. In streaming, the initial snapshots of the data often contain fewer data. Similarly, in distributed data processing, each node might have limited access to a small part of the whole data set, due to scale or privacy requirements. On the same data, one can still encounter this situation, as in the following scenario: recall that a discrete feature can be interpreted as a clustering, in which each cluster contains data points having the same feature value. Suppose we have a number of features (clusterings) and wish to compare the similarity of each against a reference clustering (class label), then choosing the feature with highest similarity to the class. If the features have missing values, then the respective clustering solutions will contain varying numbers of data points.

In these situations, there is selection bias if one uses MI and AMI as the clustering comparison measure. To demonstrate this point, we generate a random reference clustering with 4 clusters and 100 data points and then generate 5 random clustering solutions with 4 clusters, each induced using a different number of data points (20, 40, 60, 80 and 100). Each of the 5 clusterings is compared against the reference clustering (discarding from the reference any points not present in the candidate clustering solution). Even though each solution is random and independent from the reference clustering, MI and AMI select the one with 20 records significantly more often than the one with 100. Figure 5.13 shows the ‘winning’ probabilities estimated from 10,000 trials. As observed, SMI helps to decrease the bias significantly.

5.6.4 SMI Running Time

We compare the execution time of SMI implemented in Matlab\(^1\) and the Fisher’s exact test available in R implemented as discussed in Mehta and Patel [1983]. We make

\(^1\) The code is available at https://sites.google.com/site/icml2014smi
Figure 5.13: Estimated selection probability of a random clustering solution varying the number of points used to induce it. MI and AMI show strong bias toward selection of solutions with fewer data points. SMI shows close to constant selection probability across the solutions.

this comparison, since Fisher’s test is a very popular, yet expensive exact method and makes a good benchmark for assessing the relative runtime performance of SMI given that its computational effort is the same as for an exact $p$-value for MI. On a quadcore Intel Core-i7 2.9GHz PC with 16Gb of RAM, the average running time for 10 random clusterings is provided in Table 5.3. Each two compared clusterings were generated by assigning randomly each record to one cluster with equal probability and independently from the others. Even with a carefully-tuned implementation, Fisher’s exact test is extremely computationally expensive: it becomes intractable for a fairly small number of records $n$ and when either the number of rows $r$ or columns $c$ increases. We note that
computing the Fisher’s exact test with the network algorithm implemented in R also requires significant memory, i.e. \( \sim 1 \)Gb of RAM for the data used herein.

It is worth noting that computing the SMI is highly amenable to parallelization and it is easy to implement a parallel version using modern programming languages. For example, Matlab provides the \texttt{parfor} construct that splits the load of a \texttt{for} loop on different CPUs. We can choose to parallelize the outer loops in \( i \) and \( j \) to exploits better parallelism even on \( 2 \times 2 \) tables. It is achievable by iterating on another variable \( u \) from 1 to \( r \times c \) and using \( i \) and \( j \) as follows: \( i \leftarrow \lfloor u/c \rfloor, \ j \leftarrow (u - 1) \mod c + 1 \). We indeed obtain almost linear speedup when \( r > 2 \) or \( c > 2 \). For example, for \( r = 5 \) and \( c = 5 \), the speedups for two and four CPU cores are 1.96 and 3.64 folds on average.

\[
\begin{array}{ccccccc}
\text{SMI} & 0.30 & 0.64 & 1.12 & 1.72 & 2.47 & 3.30 \\
\text{SMI}_{\text{parallel}} & 0.15 & 0.27 & 0.40 & 0.55 & 0.80 & 1.01 \\
\text{Fisher’s} & 0.01 & 0.61 & 67.06 & 857.11 & \text{N/A} & \text{N/A} \\
\end{array}
\]

\[
\begin{array}{cccccccc}
\text{SMI} & 0.65 & 1.53 & 2.94 & 5.00 & 7.59 & 11.00 \\
\text{SMI}_{\text{parallel}} & 0.30 & 0.51 & 0.97 & 1.52 & 2.33 & 3.35 \\
\text{Fisher’s} & 0.65 & 11.32 & 242.67 & 844.62 & \text{N/A} & \text{N/A} \\
\end{array}
\]

Table 5.3: Running times in seconds for SMI and Fisher’s exact test. Fisher’s exact test becomes intractable when the number of records \( n \) is large or the number of rows \( r \) or columns \( c \) is large.

5.6.5 \textit{Bias Correction Under Independent Clusterings with} SMI_q

As in the previous sections, here we evaluate the performance of standardized measures SMI_q based on the generalized \( q \)-entropy on selection bias correction when partitions \( U \) are generated at random and independently from the reference partition \( V \). This hypothesis has been employed in previous published research to study selection bias [White
and Liu, 1994; Frank and Witten, 1998; Dobra and Gehrke, 2001; Shih, 2004; Hothorn et al., 2006]. In this section, we experimentally demonstrate that also the normalized mutual information $\text{NMI}_q$ without adjustment for chance is biased towards the selection of partitions $U$ with more clusters at any $q$. Furthermore, $\text{AMI}_q$ shows different biases according to different $q$. Therefore, it is beneficial to perform standardization to correct for selection bias. Although the choice of whether performing standardization or not is application dependent. In particular, we discussed in Chapter 4 that standardized measures are a way to correct for selection bias under the null hypothesis of independence between clusterings, and they are not unbiased in general. Nonetheless, the use of standardized measures is still beneficial in decision trees. In other venues, it has also been argued that in some cases the selection of clustering solutions should be biased towards clusterings with the same number of clusters as in the reference [Amelio and Pizzuti, 2015]. In this section we aim to show the effects of selection bias when clusterings are independent and that standardization helps in reducing it.

Given a reference partition $V$ on $n = 100$ objects with $c = 4$ sets, we generate a pool of random partitions $U$ with $r$ ranging from 2 to 10 sets. Then, we use $\text{NMI}_q(U, V)$ to select the closest partition to the reference $V$. The plot at the bottom of Figure 5.14 shows the probability of selection of a partition $U$ with $r$ sets using $\text{NMI}_q$ computed on 5000 simulations.
Figure 5.14: Selection bias towards partitions with different \( r \) when compared to a reference \( V \). The probability of selection should be uniform when partitions are random. Using SMI\(_{q}\) we achieve close to uniform probability of selection for \( q \) equal to 1.001, 2 and 3 respectively.
We do not expect any partition to be the best given that they are all generated at random: *i.e., the plot is expected to be flat if a measure is unbiased*. Nonetheless, we see that there is a clear bias towards partitions with 10 sets if we use NMI \(_q\) with \(q\) respectively equal to 1.001, 2, or 3. We can see that the use of the adjusted measures such as AMI \(_q\) helps in decreasing this bias, in particular when \(q = 2\). On this experiment when \(q = 2\), baseline adjustment seems to be effective in decreasing the selection bias because the variance AMI\(_2 = \text{ARI}\) is almost constant. However for all \(q\), using SMI\(_q\) we obtain close to uniform probability of selection of each random partition \(U\).

### 5.6.6 Guidelines for the Use of Standardized Measures

To conclude, we provide a radar chart to highlight the relative utility information theoretic measures in Figure 5.15. Each axis assesses the capability with respect to a particular clustering comparison scenario. In some situations, the user might be interested to know how far a solution is from the maximum agreement achievable with the reference clustering for *quantification* purposes. In this case, VI\(_q\), NMI\(_q\) and AMI\(_q\) are good choices. On the other hand, SMI\(_q\) is particularly useful when the task is the selection of a clustering based on multiple clustering comparisons against a reference for *ranking* purposes. However, it is less important to correct for selection bias if the number of objects \(n\) is big with regards to the number of cells in the contingency table in Table 2.3: i.e., when \(\frac{n}{r \cdot c}\) is large. When the number of objects is large AMI\(_q\) might be sufficient to avoid selection bias. In this scenario, SMI\(_q\) is not needed and AMI\(_q\) might be preferred as it can be computed more efficiently. Furthermore for large \(n\) the variance under independent clusterings goes to 0 according to Theorem 5.16, and this produces large SMI\(_q\) and small \(p\)-values. Moreover, the user should also be aware of results from Chapter 4: standardized measures and \(p\)-values are useful to correct for selection bias under the null of independent clusterings, i.e., when there is small dependency between the clustering solutions \(U\) and the reference \(V\).
5.7 Conclusion

In this chapter, we computed the exact expected value and variance of measures of the family $\mathcal{L}_\phi$, which contains generalized IT measures. We also showed how the expected value for measures $S \in \mathcal{N}_\phi$ can be computed for large $n$. Using these statistics, we proposed $\text{AMI}_q$ and $\text{SMI}_q$ to adjust generalized IT measures both for baseline and for selection bias. $\text{AMI}_q$ is a further generalization of well known measures for clustering comparisons such as ARI and AMI. This analysis allowed us to provide guidelines for their best application in different scenarios. In particular ARI might be used as external validation index when the reference clustering shows big equal sized clusters. AMI can be used when the reference clustering is unbalanced and there exist small clusters. The standardized $\text{SMI}_q$ can instead be used to correct for selection bias among many possible candidate clustering solutions when the number of objects is small. Furthermore, it can
also be used to test the independence between two partitions. All code has been made available online\(^2\).

\(^2\) https://sites.google.com/site/adjgenit/
In this thesis, we have described several contributions to improve dependency measures estimates on different goals. We summarize the contributions of the thesis and outline the future directions of research as follows.

In Chapter 3, we presented the Randomized Information Coefficient (RIC) to rank dependencies between sets of variables in noisy data. RIC is based on the estimation of the normalized mutual information with many random grids. This allows us to achieve low estimation variance which is particularly important for the grid estimator of mutual information, given that its bias is mainly systematic. We proved that systematic biases cancel each other out when testing for independence between sets of variables (detecting dependency) and when ranking dependencies. Indeed, we showed that the detection power of RIC increases at the increase of the number of random grids used for its estimation. Detecting dependencies on small sample size $n$ is more challenging, hence the use of many random grids is particularly important when $n$ is small. Furthermore, we experimentally demonstrated that RIC is competitive in terms of detection power against 15 other state-of-the-art measures under additive and white noise. Another successful application domain for RIC is biological network reverse engineering. In this task, a dependency measure must be able to rank the dependency between two variables that are interacting higher than the dependencies between variables that are not interacting. The dependency is computed on short (small $n$) time series available for every variable (e.g., a gene). Also in this scenario, RIC positively compares to other measures. Other applications of RIC include feature filtering and feature selection for regression tasks.

In Chapter 4, we formalized a framework to adjust dependency measure estimates for chance. Dependency measures are estimated on finite data sets and interpretability of their quantification and accuracy when used for ranking become challenging. Dependency measure estimates are dependent on the sample size $n$ and therefore cannot be
compared among different data sets or on the same data set if there are missing values. Moreover, even if under independent variables a dependency measure is supposed to be equal to zero, the estimated dependency is often greater than zero on average. This causes problems when the Maximal Information Coefficient (MIC) is used to quantify the amount of noise on functional relationships between two variables. We proposed the Adjusted MIC (AMIC) which shows zero baseline when variables are independent for any $n$ and better exploits the range of values $[0, 1]$: 1 when there exists functional noiseless relationship between variables, and 0 on average when variables are independent. Different sample size $n$ causes problems also when ranking relationships: dependency measures estimated on smaller $n$ have more chances to be ranked higher than relationships estimated on larger $n$. We formalized two types of adjustments for dependency measures to use for ranking purposes: adjustment by standardization and adjustment by penalization on statistical significance based on a parameter $\alpha$. Regarding MIC, we discussed and compared with experimental analysis its adjusted versions for ranking in order to correct biases towards variables with missing values.

Because of finite samples, dependency measure estimates are high because of chance also when computed between categorical variables with many categories. We proposed to implement adjusted for ranking splitting criteria in random forest in order to increase its classification accuracy. In particular, we experimentally demonstrated that our Adjusted Gini Gain (AGG) with $\alpha = 0.05$ significantly improves over the standard Gini Gain (GG).

In Chapter 5, we presented how to analytically perform adjustment for quantification and for ranking for clustering comparison measures. We solved the key technical challenge of analytically computing expected value and variance of the class $\mathcal{L}_\phi$ of dependency measures computed on a contingency table. This class includes as a special case the clustering comparison measure Mutual Information based on the Tsallis $q$-entropy ($\text{MI}_q$). By computing the analytical expected value for $\text{MI}_q$ we were able to generalize well known adjusted clustering comparison measures used in the clustering community such as the Adjusted Mutual Information (AMI) and Adjusted Rand Index (ARI). This allowed us to provide guidelines for their best application in different scenarios. Moreover, we also proposed an adjustment based on standardization for $\text{MI}_q$. The
Standardized Mutual Information based on the Tsallis $q$-entropy (SMI$_q$) might help in reducing the bias towards clusterings with more clusters when the number of clustered objects is small.

6.1 IMPLICATIONS OF THE PRESENTED WORK

The work presented in this thesis has several implications. We showed that the choice of a particular dependency measure is very much task dependent. The user has to clearly identify if he/she is interested in either detection, quantification, or ranking of dependencies.

If the user is interested in detection, we showed that the estimation variance of a dependency measure estimator is very important. We addressed this problem in Chapter 3 proposing RIC by introducing randomization into the grid estimator of mutual information. To our knowledge this was the first attempt to design a dependency measure based on randomization and ensembles. Both randomization and ensembles have been shown to be effective in machine learning. It might be beneficial to introduce randomization in other dependency measure estimators.

In Chapter 4, we alerted the users of dependency measures about new possible biases in quantification and ranking. When MIC is used as a proxy to quantify the amount of noise in a functional relationship, the user should be careful at not comparing MIC on data sets of different sample size. In contrast, the Adjusted MIC (AMIC) enables more interpretable quantification across different data sets. Moreover if a data set includes variables available in a different number of samples because of missing values, there exists biases in ranking using MIC. In this case, other types of adjustments to MIC might be beneficial: the Standardized MIC (SMIC) and the Adjusted MIC at level $\alpha$ (AMIC($\alpha$)).

Because of finite sample estimation, dependency measure estimators between categorical variables are also inflated when computed between variables with many categories. We discussed how this happens for the Gini Gain (GG) used in random forest in Section 4.4.3. This bias can be corrected with the Standardized Gini Gain (SGG) or the Adjusted Gini Gain at level $\alpha$ (AGG($\alpha$)). The latter can also be tuned in random forests
with cross-validation varying $\alpha$ to achieve the best accuracy. Similar adjustments might be applied to other dependency measures between categorical variables used as splitting criteria.

In Chapter 5, we bridged the gap between adjusted pair-counting measures and information theoretic measures by proposing the Adjusted Mutual Information with Tsallis’ $q$-entropy ($\text{AMI}_q$). This allowed us to propose guidelines for the best application scenarios of the Adjusted Rand Index (ARI) and the Adjusted Mutual Information with Shannon’s entropy (AMI): ARI might be used as external validation index when the reference clustering shows big equal sized clusters. AMI can be used when the reference clustering is unbalanced and there exist small clusters. We also showed how to analytically compute expected value and variance under independent clusterings for the class of measures $L_\phi$. Researchers in this area could exploit this useful theory to analytically compute expected value and variance of clustering comparison measures which fall in this category.

To conclude, we provide guidelines on how the different machine learning tools produced in this thesis can be used to improve on the three main different tasks for dependency measures. The results discussed in this thesis can aid detection, quantification, and ranking of dependencies between variables as follows:

**Detection:** The Randomized Information Coefficient (RIC) can be used to detect dependencies between numerical variables because it has high power when testing for independence;

**Quantification:** The adjustment for quantification can be used to obtain a more interpretable range of values. For example, the Adjusted Maximal Information Coefficient (AMIC) is more interpretable as a proxy for the amount of noise for functional relationships: 1 for a noiseless functional relationship and 0 on average when variables are independent. When comparing clusterings, the Adjusted Mutual Information based on the Tsallis $q$-entropy ($\text{AMI}_q$) is equal to 1 when the clusterings are identical and equal to 0 when their agreement is due to chance;

**Ranking:** Adjustment for ranking can be used to correct for biases towards variables with missing values or variables with many categories. We proposed to adjust
dependency measures by standardization and by penalization on statistical significance based on the parameter $\alpha$. The Standardized MIC (SMIC) and the Adjusted MIC at level $\alpha$ (AMIC($\alpha$)) can be used to correct for biases towards variables with missing values. Regarding dependencies between categorical variables, we demonstrated that the Standardized Gini Gain (SGG) and the Adjusted Gini Gain at level $\alpha$ (AGG($\alpha$)) can be successfully plugged in random forests as splitting criteria to improve their accuracy.

All code about the dependency measure estimators and the experiments carried out in this thesis can be found in GitHub at: https://github.com/ialuronico.

6.2 LIMITATION OF CURRENT RESEARCH AND FUTURE WORK

Our results open several directions for future work, and in this final section of the thesis, we first point out the limitations of the presented work, then highlight some potential avenues of future research.

6.2.1 Future Work About Randomized Estimations

The random seeds discretization technique for RIC discussed to discretize sets of variables might have problems with large sets of variables: it is indeed based on the Euclidean distance between points which is affected by the curse of dimensionality. As discussed in Section 3.3, preliminary results with the discretization technique based on random ferns were not too successful because this technique does not allow to fine tune the number of discretization bins. Nonetheless, it is still possible to randomly discretize a subspace using a random tree [Geurts et al., 2006]. Using a random tree it is possible to set a lower limit for the number of instances defined by each leaf. However, this would require careful implementation.

In Chapter 3, the kernel density estimator of mutual information ($I_{\text{KDE}}$) showed to be really competitive for the tasks of testing for independence between variables and genetic network inference. As well as the estimator of mutual information based on
grids, even this estimator is highly amenable to randomization: a different random kernel width might be chosen for its estimation for every point of the data sample used. This approach would allow to reduce its estimation variance and increase its performance on the tasks above. Randomization could also be introduced in the \( k \)NN estimator of mutual information \((I_{k\text{NN}})\): for example we could choose a different random \( k \) for each data point. It would be interesting to see the performance of this approach on synthetic and real data. Moreover, it would also be interesting to apply this approach to novel proposed estimators of mutual information [Gao et al., 2014, 2015] or estimators of other measures of dependence.

### 6.2.2 Future Work About Adjustments of Dependency Measures

Even complex dependency measures such as the Maximal Information Coefficient (MIC) allow to be adjusted if their expected value and variance are computed with Monte Carlo simulations. Nonetheless, this is computationally demanding given that a dependency measure must be estimated many times. Analytical approaches are preferable because they allow shorter computational times. Even if MIC is based on mutual information, it is not enough to plug-in the Adjusted Mutual Information (AMI) in the MIC algorithm in order to adjust MIC. MIC is based on the maximization of mutual information across different grids and this has to be taken into account. A promising future research direction is the identification of analytical adjustments for complex measures such as MIC.

In Chapter 4, we showed that it is very difficult to obtain an unbiased ranking of relationships between variables. In particular, a perfect unbiased ranking can only be obtained under the null hypothesis of independence between variables using either \( p \)-values or standardized dependency measures. When we are not under this hypothesis, we proposed to adjust a dependency estimator for ranking according to the parameter \( \alpha \) based on statistical significance. Nonetheless, this is an ad hoc solution. In supervised learning problems, \( \alpha \) can be successfully tuned with cross-validation. However, it is an open problem as to what is the best \( \alpha \) for an unsupervised learning problem.
Dependency measure estimates can obtain high values because of chance also when they are computed on sets of more variables. For example when a dependency measure is used for feature selection for regression, a larger set of variables might induce higher estimated dependency to the target class than a smaller set. This phenomenon should be studied in detail in particular for dependency measure between many variables such as RIC. Moreover, it would be beneficial for a number of newly defined dependency measures [Nguyen et al., 2014; Nguyen and Vreeken, 2015]. Once it is possible to compute the expected value and the variance of a dependency measure under the assumption of independent variables, it is also possible to define adjustments for quantification and ranking to decrease biases towards different dimensionality.


M. Sugiyama and K. M. Borgwardt. Measuring statistical dependence via the mutual information dimension. In *Proceedings of the Twenty-Third international joint confer-


Author/s:
Romano, Simone

Title:
Design and adjustment of dependency measures

Date:
2015

Persistent Link:
http://hdl.handle.net/11343/91431