

# Finding Statistically Significant Interactions between Continuous Features

Mahito Sugiyama<sup>1,2\*</sup> and Karsten Borgwardt<sup>3,4</sup>

<sup>1</sup>National Institute of Informatics, Tokyo 101-8430, Japan

<sup>2</sup>JST PRESTO, Japan

<sup>3</sup>D-BSSE, ETH Zürich, Basel 4058, Switzerland

<sup>4</sup>SIB Swiss Institute of Bioinformatics, Switzerland  
mahito@nii.ac.jp, karsten.borgwardt@bsse.ethz.ch

## Abstract

The search for higher-order feature interactions that are statistically significantly associated with a class variable is of high relevance in fields such as Genetics or Healthcare, but the combinatorial explosion of the candidate space makes this problem extremely challenging in terms of computational efficiency and proper correction for multiple testing. While recent progress has been made regarding this challenge for binary features, we here present the *first solution for continuous features*. We propose an algorithm which overcomes the combinatorial explosion of the search space of higher-order interactions by deriving a lower bound on the  $p$ -value for each interaction, which enables us to massively prune interactions that can never reach significance and to thereby gain more statistical power. In our experiments, our approach efficiently detects all significant interactions in a variety of synthetic and real-world datasets.

## 1 Introduction

A big challenge in high-dimensional data analysis is the search for features that are statistically significantly associated with the class variable, while accounting for the inherent multiple testing problem. This problem is relevant in a broad range of applications including natural language processing, statistical genetics, and healthcare. To date, this problem of *feature selection* [Guyon and Elisseeff, 2003] has been extensively studied in statistics and machine learning, including the recent advances in selective inference [Taylor and Tibshirani, 2015], a technique that can assess the statistical significance of features selected by linear models such as the Lasso [Lee *et al.*, 2016].

However, current approaches have a crucial limitation: They can find only *single features or linear combinations of features*, but it is still an open problem to find *patterns*, that is, *multiplicative (potentially higher-order) interactions between features*. A relevant line of research towards this goal is *significant (discriminative) pattern mining* [Terada *et al.*, 2013a;

Llinares-López *et al.*, 2015; Papaxanthos *et al.*, 2016; Pellegrina and Vandin, 2018], which tries to find statistically associated feature interactions while controlling the *family-wise error rate (FWER)*, the probability to detect one or more false positive patterns. However, all existing methods for significant pattern mining only apply to combinations of *binary or discrete* features, and none of methods can handle real-valued data, although such data is common in many applications. If we binarize data beforehand to use existing significant pattern mining approaches, a binarization-based method may not be able to distinguish (un)correlated features (see Figure 1).

To date, there is no method that can find all higher-order interactions of *continuous* features that are significantly associated with an output variable and that accounts for the inherent multiple testing problem. To solve this problem, one has to address the following three challenges:

1. How to assess the significance for a multiplicative interaction of continuous features?
2. How to perform multiple testing correction? In particular, how to control the FWER (family-wise error rate), the probability to detect one or more false positives?
3. How to manage the combinatorial explosion of the candidate space, where the number of possible interactions is  $2^d$  for  $d$  features?

The second problem and the third problem are related with each other: If one can reduce the number of combinations by pruning unnecessary ones, one can gain statistical power and reduce false negative combinations while at the same time controlling the FWER. Although there is an extensive body of work in statistics on multiple testing correction including the FDR [Hochberg, 1988; Benjamini and Hochberg, 1995] and also several studies on approaches to interaction detection [Bogdan *et al.*, 2015; Su and Candès, 2016], none of these approaches addresses the problem of dealing with the combinatorial explosion of the  $2^d$ -dimensional search space when trying to finding higher-order significant multiplicative interactions.

*Our goal in this paper is to present the first method, called C-Tarone, that can find all higher-order interactions between continuous features that are statistically significantly associated with the class variable, while controlling the FWER.*

Our approach is to use the *rank order statistics* to directly estimate the probability of joint occurrence of each feature

\*Contact Author

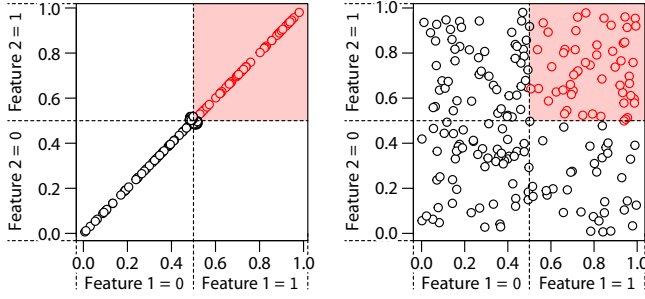


Figure 1: Although there is a clear correlation between Features 1 and 2 in the left panel and none in the right panel, after median-based binarization of Feature 1 and Feature 2, the estimated probability of the occurrence of the feature combination (the number of points for which Feature 1 = Feature 2 = 1, or the number of points in the red box) will be exactly the same in both examples. Hence if the left case is a significant interaction, the right uncorrelated case also becomes a significant interaction in binarization-based methods.

combination from continuous data, which is known as *copula support* [Tatti, 2013], and apply a *likelihood ratio test* to assess the significance of association between feature interactions and the class label, which solves the problem 1. We present the tight lower bound on  $p$ -values of association, which enables us to prune unnecessary interactions that can never be significant through the notion of *testability* proposed by Tarone [1990], and can solve both problems 2 and 3.

This paper is organized as follows: We introduce our method C-Tarone in Section 2. We introduce a likelihood ratio test as a statistical association test for interactions in Section 2.1, analyze multiple testing correction using the testability in Section 2.2, and present an algorithm in Section 2.3. We experimentally validate our method in Section 3 and summarize our findings in Section 4.

## 2 The Proposed Method: C-Tarone

Given a supervised dataset  $D = \{(v_1, y_1), (v_2, y_2), \dots, (v_N, y_N)\}$ , where each data point is a pair of an  $d$ -dimensional vector  $v_i = (v_i^1, v_i^2, \dots, v_i^d) \in \mathbb{R}^d$  and a binary class label  $y_i \in \{0, 1\}$ . We denote the set of features by  $V = \{1, 2, \dots, d\}$  and the power set of features by  $2^V$ . For each feature  $j \in \{1, 2, \dots, d\}$ , we write  $v^j = (v_1^j, v_2^j, \dots, v_N^j)$ , which is the  $N$ -dimensional vector composed of the  $j$ th feature of the dataset  $D$ .

Our goal is to find *every multiplicative feature interaction that is associated with class labels*. To tackle the problem, first we measure the joint occurrence probability  $\eta(J) \in \mathbb{R}$  of a feature combination  $J \in 2^V$  in a dataset. For each combination  $J \in 2^V$ , the size  $|J|$  corresponds to the order of an interaction, and we find arbitrary-order interactions in the form of combinations. If data is not real-valued but binary, that is,  $v_i^j \in \{0, 1\}$ , this problem is easily solved by measuring the *support* used in frequent itemset mining [Agrawal and Srikant, 1994; Aggarwal and Han, 2014]. Each feature combination  $J \in 2^V$  is called an *itemset*, and the support of  $J$  is defined as  $\eta(J) = (1/N) \sum_{i=1}^N \prod_{j \in J} v_i^j$ , which corresponds to joint probability of  $J$ . The support is always

from 0 to 1, and we can introduce a binary random variable  $X_J$  corresponding to the joint occurrence of  $J$ , where  $X_J = 1$  if features in  $J$  jointly occurs and  $X_J = 0$  otherwise,  $\eta(J)$  corresponds to the empirical estimate of the probability  $\Pr(X_J = 1)$ .

This approach can be generalized to continuous (real-valued) data using the *copula support*, which is the prominent result given by Tatti [2013] in the context of frequent pattern mining from continuous data. The copula support allows us to define the *binary* random variable  $X_J$  of joint occurrence of  $J$  and estimate  $\eta(J) = \Pr(X_J = 1)$  from *continuous* data. The key idea is to use the *normalized ranking* for each feature  $j \in V$ , which is defined as  $\pi(v_i^j) = (k - 1)/(N - 1)$  if  $v_i^j$  is the  $k$ th smallest value among  $N$  values  $v_1^j, v_2^j, \dots, v_N^j$ . Then we convert a dataset  $D$  with respect to features in a combination  $J \in 2^V$  into a single  $N$ -dimensional vector  $x_J = (x_{J1}, x_{J2}, \dots, x_{JN}) \in [0, 1]^N$  by

$$x_{Ji} = \prod_{j \in J} \pi(v_i^j) \quad (1)$$

for each data point  $i \in \{1, 2, \dots, N\}$ . Tatti [2013] showed that the empirical estimate of the probability  $\Pr(X_J = 1)$  of the joint occurrence of a combination  $J$  is obtained by the *copula support* defined as

$$\eta(J) = \frac{1}{N} \sum_{i=1}^N x_{Ji} = \frac{1}{N} \sum_{i=1}^N \prod_{j \in J} \pi(v_i^j). \quad (2)$$

Intuitively, joint occurrence of  $J$  means that rankings among features in  $J$  match with each other. The definition in Equation (2) is analogous to the support for binary data, where the only difference here is  $\pi(v_i^j)$  instead of binary value  $v_i^j$ .

The copula support always satisfies  $\eta(J) \leq 0.5$  by definition and has the *monotonicity* with respect to the inclusion relationship of combinations:  $\eta(J) \geq \eta(J \cup \{j\})$  for all  $J \in 2^V$  and  $j \in V \setminus J$ . Note that, although Tatti [2013] considered a statistical test for the copula support, it cannot be used in our setting due to the following two reasons: (1) his setting is unsupervised while ours is supervised; (2) multiple testing correction was not considered for the test and the FWER was not controlled.

### 2.1 Statistical Testing

Now we can formulate our problem of finding feature interactions, or itemset mining on continuous features, that are associated with class labels as follows: Let  $Y$  be an output binary variable of which class labels are realizations. The task is to find *all* feature combinations  $J \in 2^V$  such that the null hypothesis  $X_J \perp\!\!\!\perp Y$ , that is,  $X_J$  and  $Y$  are statistically independent, is rejected by a statistical association test while rigorously controlling the FWER, the probability of detecting one or more false positive associations, under a predetermined significance level  $\alpha$ .

As a statistical test, we propose to use a *likelihood ratio test* [Fisher, 1922], which is a generalized  $\chi^2$ -test and often called G-test [Woolf, 1957]. Although Fisher's exact test has been used as the standard statistical test in recent studies [Llinares-López *et al.*, 2015; Sugiyama *et al.*, 2015;

	$X_J = 1$	$X_J = 0$	Total
$Y = 1$	$\eta(J)r_1$	$r_1 - \eta(J)r_1$	$r_1$
$Y = 0$	$\eta(J)r_0$	$r_0 - \eta(J)r_0$	$r_0$
Total	$\eta(J)$	$1 - \eta(J)$	1

(a) Expected distribution.

	$X_J = 1$	$X_J = 0$	Total
$Y = 1$	$\eta_1(J)$	$r_1 - \eta_1(J)$	$r_1$
$Y = 0$	$\eta_0(J)$	$r_0 - \eta_0(J)$	$r_0$
Total	$\eta(J)$	$1 - \eta(J)$	1

(b) Observed distribution.

Table 1: Contingency tables.

Terada *et al.*, 2013a], it can be applied to only discrete test statistics and cannot be used in our setting.

Suppose that  $\Pr(Y = l) = r_l$  for each class  $l \in \{0, 1\}$ . From two binary variables  $X_J$  and  $Y$ , we obtain a  $2 \times 2$  contingency table, where each cell denotes the joint probability  $\Pr(X_J = l, Y = l')$  with  $l, l' \in \{0, 1\}$  and can be described as a four-dimensional probability vector  $\mathbf{p}$ :

$$\mathbf{p} = \left( \Pr(X_J = 1, Y = 1), \Pr(X_J = 1, Y = 0), \Pr(X_J = 0, Y = 1), \Pr(X_J = 0, Y = 0) \right).$$

Let  $\mathbf{p}_E$  be the probability vector under the null hypothesis  $X_J \perp\!\!\!\perp Y$  and  $\mathbf{p}_O$  be the empirical vector obtained from  $N$  observations. The difference between two distributions  $\mathbf{p}_O$  and  $\mathbf{p}_E$  can be measured by the Kullback–Leibler (KL) divergence  $D_{\text{KL}}(\mathbf{p}_O, \mathbf{p}_E) = \sum_i p_{O_i} \log(p_{O_i}/p_{E_i})$ , and the independence  $X_J \perp\!\!\!\perp Y$  is translated into the condition  $D_{\text{KL}}(\mathbf{p}_O, \mathbf{p}_E) = 0$ . In the G-test, which is a special case of likelihood ratio test, the test statistic is given as  $\lambda = 2ND_{\text{KL}}(\mathbf{p}_O, \mathbf{p}_E)$ , which follows the  $\chi^2$ -distribution with the degree of freedom 1.

In our case, for each combination  $J \in 2^V$ , the probability vector  $\mathbf{p}_E$  under the null is given as

$$\mathbf{p}_E = (\eta(J)r_1, \eta(J)r_0, r_1 - \eta(J)r_1, r_0 - \eta(J)r_0),$$

where  $\eta(J)$  is the copula support of a feature combination  $J$  and  $r_l$  is the ratio of the label  $l \in \{0, 1\}$  in the dataset. In contrast, the observed probability vector  $\mathbf{p}_O$  is given as

$$\mathbf{p}_O = (\eta_1(J), \eta_0(J), r_1 - \eta_1(J), r_0 - \eta_0(J)),$$

where  $\eta_1(J) = (1/N)\mathbf{x}_J \cdot \mathbf{y}$  and  $\eta_0(J) = (1/N)\mathbf{x}_J \cdot (1 - \mathbf{y})$  with  $\mathbf{x}_J$  defined in Equation (1). These two distributions are shown in Table 1. While it is known that the statistic  $\lambda$  does not exactly follow the  $\chi^2$ -distribution if one of components of  $\mathbf{p}_E$  is too small, such situation does not usually occur since  $\eta(J)$  is not too large as  $\eta(J) \leq 0.5$  by definition and not too small as such combinations are not testable, which will be shown in Section 2.3.

In the following, we write the set of all possible probability vectors by  $\mathcal{P} = \{\mathbf{p} \mid p_i \geq 0, \sum p_i = 1\}$ , and its subset given marginals  $a$  and  $b$  as  $\mathcal{P}(a, b) = \{\mathbf{p} \in \mathcal{P} \mid p_1 + p_2 = a, p_1 + p_3 = b\}$ .

## 2.2 Multiple Testing Correction

Since we have  $2^d$  hypotheses as each feature combination translated into a hypothesis, we need to perform *multiple testing correction* to control the FWER (family-wise error rate), otherwise we will find massive false positive combinations. The most popular multiple testing correction method is *Bonferroni correction* [Bonferroni, 1936]. In the method, the predetermined significance level  $\alpha$  (e.g.  $\alpha = 0.05$ ) is corrected as  $\delta_{\text{Bon}} = \alpha/m$ , where  $m$  is called a *correction factor* and is the number of hypotheses  $m = 2^d$  in our case. Each hypothesis  $J$  is declared as significant only if  $p\text{-value}(J) < \delta_{\text{Bon}} = \alpha/2^d$ . Then the resulting FWER  $< \alpha$  is guaranteed. However, it is well known that Bonferroni correction is too conservative. In particular in our case, the correction factor  $2^d$  is too massive due to the combinatorial effect and it is almost impossible to find significant combinations, which will generate massive *false negatives*.

Here we use the *testability* of hypotheses introduced by Tarone [1990] and widely used in the literature of significant pattern mining [Llinares-López *et al.*, 2015; Papaxanthos *et al.*, 2016; Sugiyama *et al.*, 2015; Terada *et al.*, 2013a], which allows us to prune unnecessary feature combinations that can never be significant while controlling the FWER at the same level. Hence Tarone’s testability always offers better results than Bonferroni correction if it is applicable. The only requirement of Tarone’s testability is the existence of the lower bound of the  $p$ -value given the marginals of the contingency table, which are  $\eta(J)$  and  $r_0$  (or  $r_1$ ) in our setting. In the following, we prove that we can analytically obtain the tight upper bound of the KL divergence, which immediately leads to the tight lower bound of the  $p$ -value.

**Theorem 1.** (TIGHT UPPER BOUND OF THE KL DIVERGENCE) *For  $a, b \in [0, 1]$  with  $a \leq b \leq 1/2$  and a probability vector  $\mathbf{p}_E = (ab, a(1-b), (1-a)b, (1-a)(1-b))$ ,*

$$D_{\text{KL}}(\mathbf{p}, \mathbf{p}_E) < a \log \frac{1}{b} + (b-a) \log \frac{b-a}{(1-a)b} + (1-b) \log \frac{1}{(1-a)} \quad (3)$$

for all  $\mathbf{p} \in \mathcal{P}(a, b)$  and this is tight.

Here we formally introduce how to prune unnecessary hypotheses by Tarone’s testability. Let  $\psi(J)$  be the lower bound of the  $p$ -value of  $J$  obtained from the upper bound of the KL-divergence proved in Theorem 1. Suppose that  $J_1, J_2, \dots, J_{2^d}$  be the sorted sequence of all combinations such that

$$\psi(J_1) \leq \psi(J_2) \leq \psi(J_3) \leq \dots \leq \psi(J_{2^d})$$

is satisfied. Let  $m$  be the threshold such that

$$m \cdot \psi(J_m) < \alpha \quad \text{and} \quad (m+1) \cdot \psi(J_{m+1}) \geq \alpha. \quad (4)$$

Tarone [1990] showed that the FWER is controlled under  $\alpha$  with the correction factor  $m$ . The set  $\mathcal{T} = \{J_1, J_2, \dots, J_m\}$  is called *testable* combinations, and each  $J \in \mathcal{T}$  is significant if  $p\text{-value}(J) < \delta_{\text{Tar}} = \alpha/m$ . On the other hand, combinations  $J_{m+1}, \dots, J_{2^d}$  are called *untestable* as they can never be significant. Since  $m \ll 2^d$  usually holds, we can expect to

obtain higher statistical power in Tarone’s method compared to Bonferroni method. Moreover, we present C-Tarone in the next subsection, which can enumerate such testable combinations without seeing untestable ones, hence we overcome combinatorial explosion of the search space of combinations.

Let us denote by  $B(a, b)$  the upper bound provided in Equation (3). We analyze the behavior of the bound  $B(a, b)$  as a function of  $a$  with fixed  $b$ . This is a typical situation in our setting as  $a$  corresponds to the copula support  $\eta(J)$ , which varied across combinations, while  $b$  corresponds to the class ratio  $r_1$ , which is fixed in each analysis. Assume that  $b \leq 1/2$ . When  $a < b$ , we have  $\partial B(a, b)/\partial a = \log(1 - a)/(b - a) > 0$ , hence it is monotonically increases as  $a$  increases. When  $b < a < 1/2$ , we have  $\partial B(a, b)/\partial a = \log(a - b)/a < 0$ , thereby it monotonically decreases as  $a$  increases. We illustrate the bound  $B(a, b)$  with  $b = 0.3$  and the corresponding minimum achievable  $p$ -value with the sample size  $N = 100$  in Figure 4 in Appendix.

### 2.3 Algorithm

We present the algorithm of C-Tarone that efficiently finds testable combinations  $J_1, J_2, \dots, J_m$  such that  $\psi(J_1) \leq \psi(J_2) \leq \dots \leq \psi(J_m)$ , where  $m$  satisfies the condition (4). We summarize C-Tarone in Algorithm 1, which performs depth-first search to find  $J_1, J_2, \dots, J_m$  such that  $\psi(J_1) \leq \psi(J_2) \leq \dots \leq \psi(J_m)$ , where  $m$  satisfies the condition (4). Suppose that  $r_1 \leq 1/2 \leq r_0$ . Since the lower bound of the  $p$ -value  $\psi(J)$  takes the minimum value when  $\eta(J) = r_1$  (see Figure 5 in Appendix by letting  $a = \eta(J)$  and  $b = r_1$ ) and is monotonically decreasing as  $\eta(J)$  decreases, for any  $\eta(J) \leq r_1$ ,  $\mathcal{C} \supseteq \mathcal{T}$  with  $\mathcal{C} = \{J \in 2^V \mid \eta(J) \geq \sigma\}$  and  $\mathcal{T} = \{J \in 2^V \mid \psi(J) \leq B(\sigma, r_1)\}$  is always guaranteed, where  $\sigma$  is a threshold for copula supports. Thus if the condition (4) is satisfied for some  $m \leq |\mathcal{T}|$ , the  $m$  smallest combinations in  $\mathcal{T}$  are the testable combinations.

Moreover, since  $\eta(J)$  has the monotonicity with respect to the inclusion relationship, that is,  $\eta(J) \geq \eta(J \cup \{j\})$  for all  $j \in V \setminus J$ , finding the set  $\mathcal{C}$  is simply achieved by DFS (depth-first search): Starting from the smallest combination  $\emptyset$  with assuming  $\eta(\emptyset) = 1$ , if  $\eta(J) \geq \sigma$  for a combination  $J$ , we update  $J$  by adding  $j \in V \setminus J$  and recursively check  $J$ . Otherwise if  $\eta(J) < \sigma$ , we prune all combinations  $K \supseteq J$  as  $\eta(K) < \sigma$  holds.

Our algorithm dynamically updates the threshold  $\sigma$  to enumerate testable combinations while pruning massive unnecessary combinations. First we set  $\sigma = 0$ , which means that all combinations are testable. Whenever we find a new combination  $J$ , we update  $\mathcal{C}$  and  $\mathcal{T}$  and check the condition (4). If  $|\mathcal{T}|B(\sigma, r_1) > \alpha$  holds,  $\sigma$  is too low and  $\mathcal{T}$  will become too large with the current  $\sigma$ , hence we update  $\sigma$  to  $\min_{J \in \mathcal{C}} \eta(J)$  and remove the combination  $\arg\min_{J \in \mathcal{C}} \eta(J)$ . Finally, when the algorithm stops, it is clear that the set  $\mathcal{T}$  coincides with the set of testable combinations, hence each combination  $J \in \mathcal{T}$  is significant if the  $p$ -value of  $J$  is smaller than  $\delta_{\text{Tar}} = \alpha/|\mathcal{T}|$ .

Since Algorithm 1 always outputs the set of testable combinations  $\mathcal{T}$ , which directly follows from  $\mathcal{C} \supseteq \mathcal{T}$  and the monotonicity of  $\eta(J)$ , the SIGNIFICANCETESTING function finds all significant combinations with the FWER  $\leq \alpha$ . Thus Algorithm 1 always finds all significant feature combinations with

---

#### Algorithm 1: C-Tarone.

---

```

1 C-TARONE( $D, \alpha$ )
2    $\sigma \leftarrow 0; \mathcal{C} \leftarrow \emptyset;$            //  $\sigma$  is a global variable
3   DFS( $\emptyset, 0, \mathcal{C}, D, \alpha$ );
4    $\mathcal{T} \leftarrow \{K \in \mathcal{C} \mid \psi(K) \leq B(\sigma, r_1)\};$ 
                                           // The set of testable combinations
5   SIGNIFICANCETESTING( $\mathcal{T}, \alpha$ );
6 DFS( $J, j_{\text{prev}}, \mathcal{C}, D, \alpha$ )
7   foreach  $j \in \{j_{\text{prev}} + 1, \dots, d\}$  do
8      $J \leftarrow J \cup \{j\};$ 
9     Compute  $\eta(J)$  by Equation (2);
10    if  $\eta(J) > \sigma$  then
11      Compute  $\psi(J); \mathcal{C} \leftarrow \mathcal{C} \cup \{J\};$ 
12       $\mathcal{T} \leftarrow \{K \in \mathcal{C} \mid \psi(K) \leq B(\sigma, r_1)\};$ 
13      while  $|\mathcal{T}|B(\sigma, r_1) \geq \alpha$  do
14         $J_{\min} \leftarrow \arg\min_{K \in \mathcal{C}} \eta(K);$ 
15         $\sigma \leftarrow \eta(J_{\min}); \mathcal{C} \leftarrow \mathcal{C} \setminus \{J_{\min}\};$ 
16         $\mathcal{T} \leftarrow \mathcal{T} \setminus \{J_{\min}\};$ 
17      DFS( $J, j, \mathcal{C}, D, \alpha$ );
18     $J \leftarrow J \setminus \{j\};$ 
19 SIGNIFICANCETESTING( $\mathcal{T}, \alpha$ )
20 foreach  $J \in \mathcal{T}$  do
21   if  $p\text{-value}(J) < \alpha/|\mathcal{T}|$  then output  $J$ ;
```

---

controlling the FWER under  $\alpha$ . Moreover, C-Tarone is independent of the feature ordering and the above completeness with respect to the set of significant combinations is always satisfied. The time complexity of C-Tarone is  $O(|\mathcal{T}|)$ .

### 3 Experiments

We examine the effectiveness and the efficiency of C-Tarone using synthetic and real-world datasets. We used Amazon Linux AMI release 2017.09 and ran all experiments on a single core of 2.3 GHz Intel Xeon CPU E7-8880 v3 and 2.0 TB of memory. All methods were implemented in C/C++ and compiled with gcc 4.8.5. The FWER level  $\alpha = 0.05$  throughout experiments.

There exists no existing method that can enumerate significant feature combinations from continuous data with multiple testing correction. Thus we compare C-Tarone to *significant itemset mining* method with prior *binarization* of a given dataset since significant itemset mining offers to enumerate all significant feature combinations while controlling the FWER from binary data. We employ median-based binarization as a preprocessing. For each feature  $j \in V$ , we pick up the median of  $v^j = (v_1^j, v_2^j, \dots, v_N^j)$ , denoted by  $\text{med}(j)$ , and binarize each value  $v_i^j$  as a pair  $(v_i^{\leq \text{med}(j)}, v_i^{> \text{med}(j)})$ , where  $v_i^{\leq \text{med}(j)} = 1$  if  $v_i^j \leq \text{med}(j)$  and 0 otherwise, and  $v_i^{> \text{med}(j)} = 1$  if  $v_i^j > \text{med}(j)$  and 0 otherwise. Thus a given dataset is converted to the binarized dataset with  $2d$  features. We use the state-of-the-art significant itemset mining algorithm LAMP ver.2 [Minato *et al.*, 2014] and employ implementation provided by [Llinares-López *et al.*, 2018],

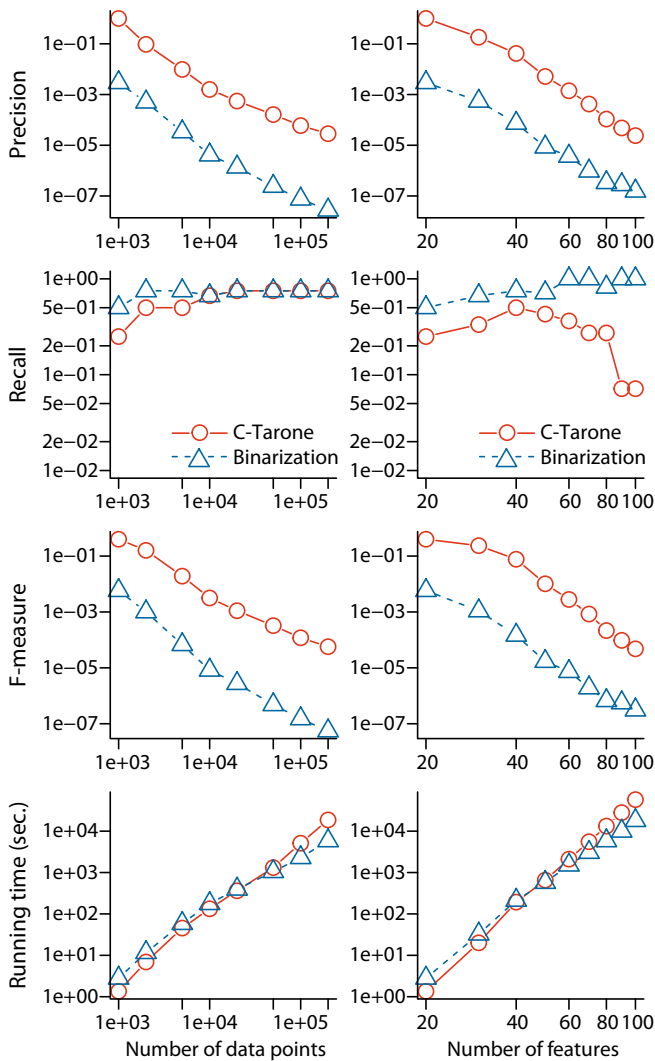


Figure 2: Results on synthetic data with the minor class ratio  $r_1 = 0.5$ . Regarding the scale of precision and F-measure, see comment at the last paragraph just before Section 3. The number of features is  $d = 20$  in the left column and the sample size is  $N = 1,000$  in the right column. Both x- and y-axes are in logarithmic scale. C-Tarone is shown in red circles, the binarization in blue triangles.

which incorporates the fastest frequent pattern mining algorithm LCM [Uno *et al.*, 2004] to enumerate testable feature combinations from binary data. Note that, in terms of runtime, comparison with the brute-force approach of testing all the  $2^d$  combinations with the Bonferroni correction is not valid as the resulting FWER is different between the brute-force and the proposed method.

We also tried other binarization approaches, *interordinal scaling* used in numerical pattern mining [Kaytoue *et al.*, 2011] and *interval binarization* in subgroup discovery [Grosskreutz and Rüping, 2009] as a preprocessing of significant itemset mining. However, both preprocessing generate too many binarized dense features, resulting in the lack of scalability in the itemset mining step in the enumeration of testable combinations. Hence we do not employ them as

comparison partners. Details of these binarization techniques are summarized in Section C in Appendix.

To evaluate the efficiency of methods, we measure the running time needed for enumerating all significant combinations. In the binarization method, we exclude the time used for binarization as this preprocessing step is efficient enough and negligible compared to the pattern mining step.

To examine the quality of detected combinations, we compute precision, recall, and the F-measure by comparing such combinations with those obtained by the standard decision tree method CART [Breiman *et al.*, 1984], which obtains multiplicative combinations of features in the form of binary trees. We used the `rpart` function in R with its default parameter setting, where the Gini index is used for splitting and the minimum number of data points that must exist in a node is 20. We apply the decision tree to each dataset and retrieve all the paths from the root to leaf nodes of the learned tree. In each path, we use the collection of features used in the path as a positive feature combination of the ground truth, that is, a feature combination found by C-Tarone (or binarization method) is deemed to be true positive if and only if it constitutes one of full paths from the root to a leaf of the learned decision tree. Note that the FWER is always controlled under  $\alpha$  in both of C-Tarone and binarization method, and our aim is to empirically examine the quality of the feature combinations compared to those selected by a standard decision tree. We used not forests but a single tree as the ground truth depends on the number of trees if we use forests, resulting in arbitrary results.

Please note that, due to the fact that there are up to  $2^d$  feature combinations and we only count an exact match between a retrieved pattern and a true pattern as a hit, precision and F-measure will be close to 0. Still, we compute them to allow for a comparison of the relative performance of the different approaches. Evaluation criteria that take overlaps between retrieved patterns and true patterns (partial matches) into account would lead to higher levels of precision, but they are a topic of ongoing research and not a focus of this work.

### Results on Synthetic Data

First we evaluate C-Tarone on synthetic data with varying the sample size  $N$  from 1,000 to 200,000, the number  $d$  of features from 20 to 100, and setting the class ratio to  $r_1 = 0.5$  or  $r_1 = 0.2$ , i.e., the number  $N_1$  of samples in the minor class is  $N/2$  or  $N/5$ . In each dataset, we generate 20% of features that are associated with the class labels. More precisely, first we generate the entire dataset from the uniform distribution from 0 to 1 and assign the class label 1 to the first  $N_1$  data point. Then, for the  $N_1$  data points in the class 1, we pick up one of the 20% of associated features and copy it to every associated feature with adding Gaussian noise with  $(\mu, \sigma^2) = (0, 0.1)$ . Hence there is no correlation in the class 0 across all features and there are positive correlations among such 20% of features in the class 1. The other 80% are uninformative features.

Results are plotted in Figure 2 for  $r_1 = 0.5$  (classes are balanced). See Figure 5 in Appendix for  $r_1 = 0.2$  (classes are imbalanced). In the figure, we plot results with varying  $N$  while fixing  $d = 20$  on the left column and those with varying

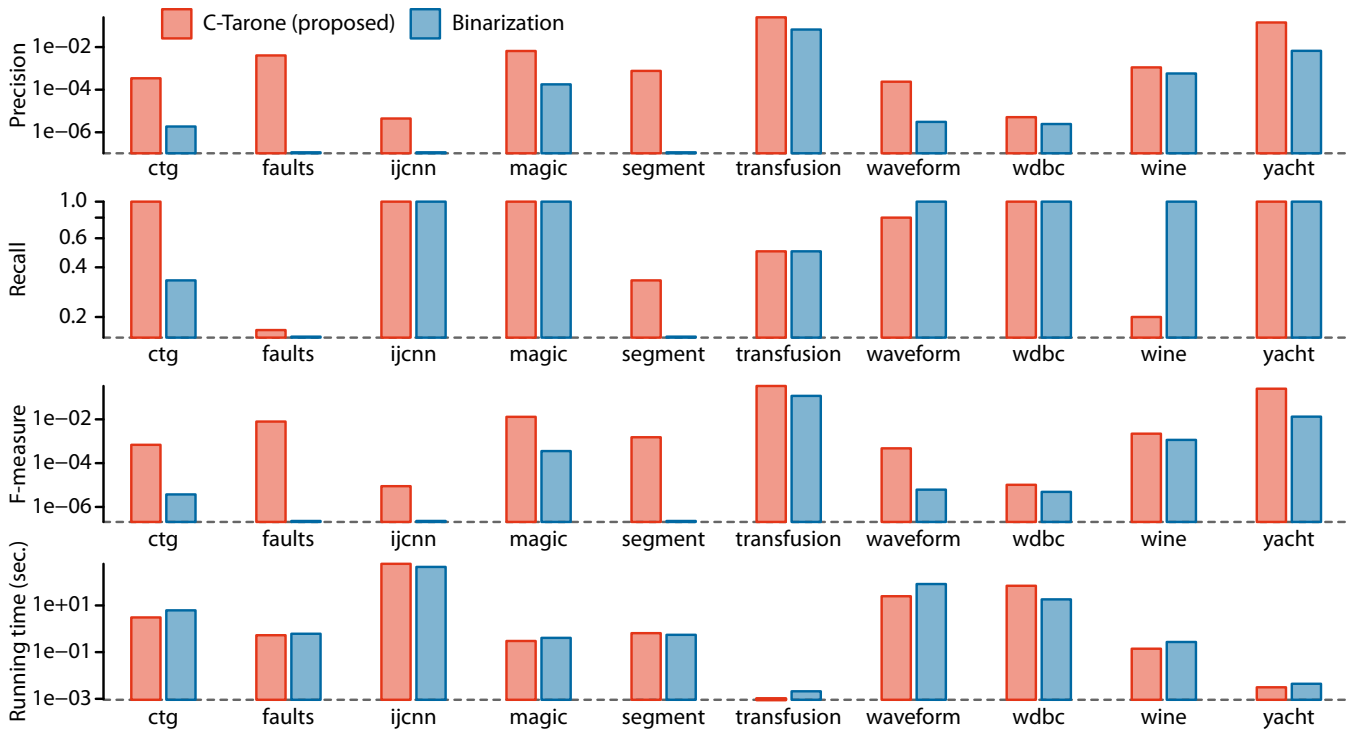


Figure 3: Results on real data. Regarding the scale of precision and F-measure, see the comment at the last paragraph just before Section 3. The y-axis is in logarithmic scale. C-Tarone is shown in red and the binarization approach is shown in blue. Higher (taller) is better in precision, recall, and F-measure, while lower is better in running time.

$d$  while fixing  $N = 1,000$  on the right column.

In comparison with the median-based binarization method (plotted in blue), C-Tarone (plotted in red) has a clear advantage regarding the precision, which is several orders of magnitude higher than the binarization method in every case. This is why binarization method cannot distinguish correlated and uncorrelated combinations as we discussed in Introduction and illustrated in Figure 1, resulting in including uncorrelated features into significant combinations. Although recall is competitive across various  $N$  and  $d$ , in all cases, the F-measure of C-Tarone are higher than those of the median-based binarization method. In both methods, precision drops when the sample size  $N$  becomes large: As we gain more and more statistical power for larger  $N$ , many feature combinations, even those with very small dependence to the class labels and not used in the decision tree, tend to reach statistical significance.

Although the algorithm LCM (itemset mining) used in the binarization approach is highly optimized with respect to the efficiency, C-Tarone is competitive with it on all datasets, as can be seen in Figure 2 (bottom row). To summarize, we observe that C-Tarone improves over the competing binarization approach in terms of the F-measure in detecting higher quality feature combinations in classification.

### Results on Real Data

We also evaluate C-Tarone on real-world datasets shown in Table 2 in Appendix, which are benchmark datasets for binary classification from the UCI repository [Lichman, 2013]. To

clarify the exponentially large search space, we also show the number  $2^d$  of candidate combinations for  $d$  features in the table. All datasets are balanced to maximize the statistical power for comparing detected significant combinations, i.e.  $r_1 = 0.5$ . If they are not balanced in the original dataset, we randomly subsample data points from the larger class.

We summarize results in Figure 3. Again, C-Tarone shows higher precision in all datasets than the binarization method and better or competitive recall, resulting in higher F-measure scores in all datasets. In addition, running time is competitive with the binarization method, which means that C-Tarone can successfully prune the massive candidate space for significant feature combinations. These results demonstrate the effectiveness of C-Tarone.

## 4 Conclusion

In this paper, we have proposed a solution to the open problem of finding all multiplicative interactions between *continuous* features that are significantly associated with an output variable after rigorously controlling for multiple testing. While interaction detection with multiple testing has been studied before, our approach, called *C-Tarone*, is the first to overcome the problem of detecting all higher-order interactions from the enormous search space  $2^d$  for  $d$  features.

Our work opens the door to many applications of searching significant feature combinations, in which the data is not adequately described by binary features, including large fields such as data analysis for high-throughput technologies

in biology and medicine. Our work here addresses the problem of finding continuous *features*. Finding significant combinations associated with *continuous output variables* is an equally challenging and practically relevant problem, that we will tackle in future work.

## Acknowledgments

This work was supported by JSPS KAKENHI Grant Numbers JP16K16115, JP16H02870, and JST, PRESTO Grant Number JPMJPR1855, Japan (M.S.). This work was funded in part by the SNSF Starting Grant ‘‘Significant Pattern Mining’’ (K.B.).

## A Proof of Theorem 1

Let  $f(x) = D_{\text{KL}}(\mathbf{p}, \mathbf{p}_E)$  with  $\mathbf{p} = (x, a - x, b - x, (1 - b) - (a - x))$  and  $a' = 1 - a, b' = 1 - b$ . We have

$$f(x) = x \log \frac{x}{ab} + (a - x) \log \frac{a - x}{ab'} + (b - x) \log \frac{b - x}{a'b} + (b' - a + x) \log \frac{b' - a + x}{a'b'}$$

The second derivative of  $f(x)$  is given as

$$\frac{\partial^2 f}{\partial x^2} = \frac{1}{1 - a - b + x} + \frac{1}{b - x} + \frac{1}{a - x} + \frac{1}{x},$$

which is always positive from the constraint  $0 < x < \min\{a, b\} = a$ . Thus  $f(x)$  is maximized when  $x$  goes to 0 or  $a$ . The limit of  $f(x)$  is obtained as

$$\begin{aligned} \lim_{x \rightarrow 0} f(x) &= a \log \frac{1}{b'} + b \log \frac{1}{a'} + (b' - a) \log \frac{b' - a}{a'b'}, \\ \lim_{x \rightarrow a} f(x) &= a \log \frac{1}{b} + (b - a) \log \frac{b - a}{a'b} + (1 - b) \log \frac{1}{a'}. \end{aligned}$$

To check which is larger, let  $\delta$  be the difference  $\lim_{x \rightarrow a} f(x) - \lim_{x \rightarrow 0} f(x)$ . Then it is obtained as  $\delta = -b \log b + b' \log b' + (b - a) \log(b - a) - (b' - a) \log(b' - a)$ . The partial derivative of  $\delta$  with respect to  $a$  is  $\partial \delta / \partial a = \log(1 - a - b) - \log(b - a)$ , which is always positive as  $(1 - a - b) - (b - a) = 1 - 2b \geq 0$  with the condition  $b \leq 1/2$ . Hence the difference  $\delta$  takes the minimum value at  $a = 0$  and we obtain

$$\delta \geq -b \log b + b' \log b' + b' \log b' - b' \log b' = 0.$$

Thus  $f(x)$  is the tight upper bound when  $x \rightarrow a$ .  $\square$

## B Related Work

Since the only line of work that tries to find multiplicative feature combinations while controlling the FWER is *significant pattern mining*, we provide an overview of this field in the following.

Significant pattern mining introduces statistical significance into the task of contrast (or *discriminative*) pattern mining [Dong and Bailey, 2013], where the objective is to find discriminative patterns with respect to class partitioning of a dataset. After early work on multiple testing correction in association rule mining [Hämäläinen, 2012;

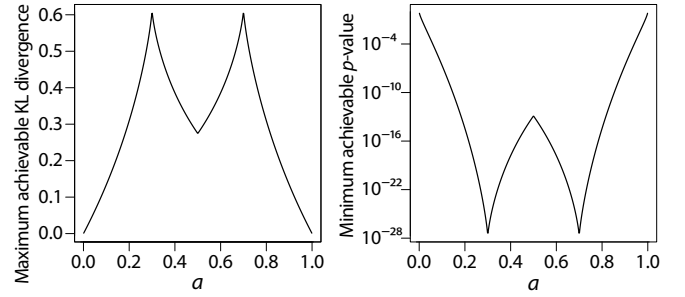


Figure 4: The upper bound  $B(a, b)$  of the KL divergence (left) and the corresponding  $p$ -value (right) with  $N = 100$  with respect to changes in  $a$  when  $b = 0.3$ .

Webb, 2007], [Terada *et al.*, 2013a] were the first to achieve control of the FWER in itemset mining, by successfully combining a pattern mining algorithm and Tarone’s trick [Tarone, 1990]. The enumeration algorithm has been improved in LAMP ver. 2 [Minato *et al.*, 2014] and significant subgraph mining [Sugiyama *et al.*, 2015]. To date, significant pattern mining has been extended to various types of tests and data, including a Westfall–Young permutation test to treat interdependencies among patterns [Llinares-López *et al.*, 2015; Terada *et al.*, 2013b], logistic regression [Terada *et al.*, 2016] or a Cochran–Mantel–Haenszel (CMH) test [Llinares-López *et al.*, 2017; Papaxanthos *et al.*, 2016] for categorical covariates, hypothesis streams [Webb and Petitjean, 2016], and top- $K$  significant patterns [Pellegrina and Vandin, 2018]. However, none of the above studies succeeded to directly perform significant pattern mining on continuous variables without prior binarization.

Although the field of *subgroup discovery* [Atzmueller, 2015; Novak *et al.*, 2009; Herrera *et al.*, 2011] also considers measures of statistical dependence for finding multiplicative feature combinations, e.g. [Grosskreutz and Rüping, 2009; Mampaey *et al.*, 2015; van Leeuwen and Ukkonen, 2016], none of these methods accounts for multiple testing by controlling the FWER.

## C Additional Binarization Methods

In interordinal scaling, each binarized feature is in the form of ‘‘ $\leq a$ ’’ or ‘‘ $\geq a$ ’’, where endpoints  $a$  are from a dataset, that is,  $a \in \{v_1^j, v_2^j, \dots, v_N^j\}$  for a feature  $j \in \{1, 2, \dots, d\}$ . Thus, for an  $d$ -dimensional real-valued vector  $\mathbf{v}_i \in D$ , each element  $v_i^j$  is expanded as the  $2N$ -dimensional binary vector such that

$$(v_i^{\leq v_1^j}, v_i^{\leq v_2^j}, \dots, v_i^{\leq v_N^j}, v_i^{\geq v_1^j}, v_i^{\geq v_2^j}, \dots, v_i^{\geq v_N^j}),$$

where each value for the binarized feature  $v_i^{\leq v_k^j} = 1$  if  $v_i^j \leq v_k^j$  and 0 otherwise. As a result, the dataset  $D$  is converted into the binary dataset with  $2dN$  features. In interval binarization, each binarized feature is in the form of ‘‘ $(a, b]$ ’’, where endpoints  $a, b$  are from data, and each element  $v_i^j$  of an  $d$ -dimensional vector  $\mathbf{v}_i$  is expanded as the  $(N(N - 1)/2)$ -dimensional binary vector such that

$$(v_i^{(v_1^j, v_2^j]}, v_i^{(v_1^j, v_3^j]}, \dots, v_i^{(v_1^j, v_N^j]}, v_i^{(v_2^j, v_3^j]}, \dots, v_i^{(v_{N-1}^j, v_N^j]}).$$

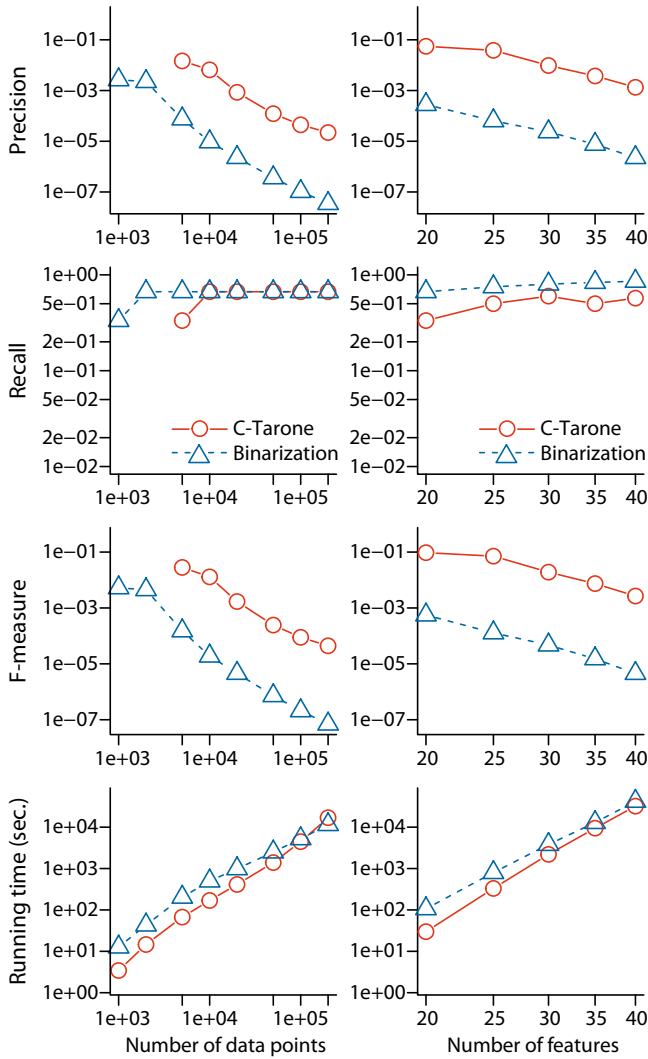


Figure 5: Results on synthetic data with the minor class ratio  $r_1 = 0.2$ . The number of features is  $d = 20$  in the left column and the sample size is  $N = 3,000$  in the right column. Both x- and y-axes are in logarithmic scale. C-Tarone is shown in red circles, the binarization approach in blue triangles. Missing points in (b) mean that no significant combination is detected.

Thus a dataset  $D$  is converted into the binary dataset with  $dN(N-1)/2$  features. Both interordinal scaling and interval binarization could finish their computation for a tiny dataset with  $(N, d) = (50, 5)$  in approximately 24 hours, but did not finish after 48 hours for  $(N, d) = (100, 10)$ , and they exceeded the memory limit (2.0 TB) for larger datasets.

## References

[Aggarwal and Han, 2014] C. C. Aggarwal and J. Han, editors. *Frequent Pattern Mining*. Springer, 2014.

[Agrawal and Srikant, 1994] R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In *Proceedings of the 20th International Conference on Very Large Data Bases*, pages 487–499, 1994.

Data	$N$	$d$	# candidate combinations (search space)
ctg	942	22	4,194,304
faults	316	27	134,217,728
ijcnn	9,706	22	4,194,304
magic	13,376	10	1,024
segment	660	19	524,288
transfusion	356	4	16
waveform	3,314	21	2,097,152
wdbc	424	30	1,073,741,824
wine	3,198	11	2,048
yacht	308	6	64

Table 2: Statistics of real data.

[Atzmueller, 2015] M. Atzmueller. Subgroup discovery. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 5(1):35–49, 2015.

[Benjamini and Hochberg, 1995] Y. Benjamini and Y. Hochberg. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society. Series B (Methodological)*, 57(1):289–300, 1995.

[Bogdan et al., 2015] M. Bogdan, E. Van Den Berg, C. Sabatti, W. Su, and E. J. Candès. SLOPE—adaptive variable selection via convex optimization. *The annals of applied statistics*, 9(3):1103–1140, 2015.

[Bonferroni, 1936] C. E. Bonferroni. Teoria statistica delle classi e calcolo delle probabilità. *Pubblicazioni del R Istituto Superiore di Scienze Economiche e Commerciali di Firenze*, 8:3–62, 1936.

[Breiman et al., 1984] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen. *Classification and Regression Trees*. CRC Press, 1984.

[Dong and Bailey, 2013] G. Dong and J. Bailey, editors. *Contrast Data Mining: Concepts, Algorithms, and Applications*. CRC Press, 2013.

[Fisher, 1922] R. A. Fisher. On the mathematical foundations of theoretical statistics. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 222(594-604):309–368, 1922.

[Grosskreutz and Rüping, 2009] H. Grosskreutz and S. Rüping. On subgroup discovery in numerical domains. *Data Mining and Knowledge Discovery*, 19(2):210–226, 2009.

[Guyon and Elisseeff, 2003] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003.

[Hämäläinen, 2012] W. Hämäläinen. Kingfisher: an efficient algorithm for searching for both positive and negative dependency rules with statistical significance measures. *Knowledge and Information Systems*, 32(2):383–414, 2012.

[Herrera et al., 2011] F. Herrera, C. J. Carmona, P. González, and M. J. del Jesus. An overview on subgroup discovery:



- foundations and applications. *Knowledge and Information Systems*, 29(3):495–525, 2011.
- [Hochberg, 1988] Y. Hochberg. A sharper bonferroni procedure for multiple tests of significance. *Biometrika*, 75(4):800–802, 1988.
- [Kaytoue *et al.*, 2011] M. Kaytoue, S. O. Kuznetsov, and A. Napoli. Revisiting numerical pattern mining with formal concept analysis. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence*, pages 1342–1347, 2011.
- [Lee *et al.*, 2016] J. D. Lee, D. L. Sun, Y. Sun, and J. E. Taylor. Exact post-selection inference, with application to the lasso. *Annals of Statistics*, 44(3):907–927, 06 2016.
- [Lichman, 2013] M. Lichman. UCI machine learning repository, 2013.
- [Llinares-López *et al.*, 2015] F. Llinares-López, M. Sugiyama, L. Papaxanthos, and K. M. Borgwardt. Fast and memory-efficient significant pattern mining via permutation testing. In *Proceedings of the 21st ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 725–734, 2015.
- [Llinares-López *et al.*, 2017] F. Llinares-López, L. Papaxanthos, D. Bodenham, D. Roqueiro, COPDGene Investigators, and K. Borgwardt. Genome-wide genetic heterogeneity discovery with categorical covariates. *Bioinformatics*, 33(12):1820–1828, 2017.
- [Llinares-López *et al.*, 2018] F. Llinares-López, L. Papaxanthos, D. Roqueiro, D. Bodenham, and K. Borgwardt. CASMAP: detection of statistically significant combinations of SNPs in association mapping. *Bioinformatics*, bty1020, 2018.
- [Mampaey *et al.*, 2015] M. Mampaey, S. Nijssen, A. Feelders, R. Konijn, and A. Knobbe. Efficient algorithms for finding optimal binary features in numeric and nominal labeled data. *Knowledge and Information Systems*, 42(2):465–492, 2015.
- [Minato *et al.*, 2014] S. Minato, T. Uno, K. Tsuda, A. Terada, and J. Sese. A fast method of statistical assessment for combinatorial hypotheses based on frequent itemset enumeration. In *Machine Learning and Knowledge Discovery in Databases*, volume 8725 of *LNCS*, pages 422–436. Springer, 2014.
- [Novak *et al.*, 2009] P. K. Novak, N. Lavrač, and G. I. Webb. Supervised descriptive rule discovery: A unifying survey of contrast set, emerging pattern and subgroup mining. *The Journal of Machine Learning Research*, 10:377–403, 2009.
- [Papaxanthos *et al.*, 2016] L. Papaxanthos, F. Llinares-Lopez, D. Bodenham, and K. M. Borgwardt. Finding significant combinations of features in the presence of categorical covariates. In *Advances in Neural Information Processing Systems*, volume 29, pages 2271–2279, 2016.
- [Pellegrina and Vandin, 2018] L. Pellegrina and F. Vandin. Efficient mining of the most significant patterns with permutation testing. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2070–2079, 2018.
- [Su and Candès, 2016] W. Su and E. J. Candès. SLOPE is adaptive to unknown sparsity and asymptotically minimax. *The Annals of Statistics*, 44(3):1038–1068, 2016.
- [Sugiyama *et al.*, 2015] M. Sugiyama, F. Llinares-López, N. Kasenburg, and K. M Borgwardt. Significant subgraph mining with multiple testing correction. In *Proceedings of 2015 SIAM International Conference on Data Mining*, pages 37–45, 2015.
- [Tarone, 1990] R. E. Tarone. A modified Bonferroni method for discrete data. *Biometrics*, 46(2):515–522, 1990.
- [Tatti, 2013] N. Tatti. Itemsets for real-valued datasets. In *2013 IEEE 13th International Conference on Data Mining*, pages 717–726, 2013.
- [Taylor and Tibshirani, 2015] J. Taylor and R. J. Tibshirani. Statistical learning and selective inference. *Proceedings of the National Academy of Sciences*, 112(25):7629–7634, 2015.
- [Terada *et al.*, 2013a] A. Terada, M. Okada-Hatakeyama, K. Tsuda, and J. Sese. Statistical significance of combinatorial regulations. *Proceedings of the National Academy of Sciences*, 110(32):12996–13001, 2013.
- [Terada *et al.*, 2013b] A. Terada, K. Tsuda, and J. Sese. Fast Westfall-Young permutation procedure for combinatorial regulation discovery. In *2013 IEEE International Conference on Bioinformatics and Biomedicine*, pages 153–158, 2013.
- [Terada *et al.*, 2016] A. Terada, D. duVerle, and K. Tsuda. Significant pattern mining with confounding variables. In *Advances in Knowledge Discovery and Data Mining (PAKDD 2016)*, volume 9651 of *LNCS*, pages 277–289, 2016.
- [Uno *et al.*, 2004] T. Uno, T. Asai, Y. Uchida, and H. Arimura. An efficient algorithm for enumerating closed patterns in transaction databases. In *Discovery Science*, volume 3245 of *LNCS*, pages 16–31, 2004.
- [van Leeuwen and Ukkonen, 2016] M. van Leeuwen and A. Ukkonen. Expect the unexpected – on the significance of subgroups. In *Discovery Science*, volume 9956 of *LNCS*, pages 51–66, 2016.
- [Webb and Petitjean, 2016] G. I. Webb and F. Petitjean. A multiple test correction for streams and cascades of statistical hypothesis tests. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1255–1264, 2016.
- [Webb, 2007] G. I. Webb. Discovering significant patterns. *Machine Learning*, 68(1):1–33, 2007.
- [Woolf, 1957] B. Woolf. The log likelihood ratio test (the G-test). *Annals of human genetics*, 21(4):397–409, 1957.