MOX-Report No. 59/2020

High-Order Interaction Learning via Targeted Pattern Search

Massi, M.C.; Franco, N.R; Ieva, F.; Manzoni, A.; Paganoni, A.M.; Zunino, P.

MOX, Dipartimento di Matematica
Politecnico di Milano, Via Bonardi 9 - 20133 Milano (Italy)

mox-dmat@polimi.it http://mox.polimi.it

High-Order Interaction Learning via Targeted Pattern Search

Michela C. Massi[1,2], Nicola R. Franco[1], Francesca Ieva[1,2,3], Andrea Manzoni[1], Anna Maria Paganoni[1,2], Paolo Zunino[1]

[1] *MOX Laboratory, Math Department, Politecnico di Milano, Milan, Italy*
[2] *CADS-Center for Analysis, Decisions and Society, Human Technopole, Milan, Italy*
[3] *CHRP-National Center for Healthcare Research and Pharmacoepidemiology, University of Milano-Bicocca, Milan, Italy*

Summary. Logistic Regression (LR) is a widely used statistical method in empirical studies in many research fields. However, these real-life scenarios oftentimes share complexities that would hinder the application of the as-is model. First and foremost, the need to include high-order interactions to capture the variability of their data. Moreover, these studies are seldom developed in imbalanced settings, with datasets growing wider, sample size from very large to extremely small and a strong need for model and results interpretability. In this paper we present a novel algorithm, `High-Order Interaction Learning` via targeted `Pattern` search (`HOILP`), to select interaction terms of varying order to include in a LR for an imbalanced binary classification task when input data is categorical. `HOILP`'s rationale is built on the duality between item sets and categorical interactions, and is composed of (i) an interaction learning step based on a well-known frequent item set mining algorithm and (ii) a novel dissimilarity-based interaction selection step, that allows the user to control for the number of interactions to include in the LR model. Besides `HOILP` we present here two variants (`Scores HOILP` and `Clusters HOILP`), that can suit even more specific needs. Through a set of experiments we validate our algorithm and prove its wide applicability to real-life research scenarios, surpassing the performance of a benchmark state-of-the-art algorithm.

1. Introduction

Despite the evolution of statistical learning theory, Logistic Regression (LR) is still a commonly used statistical method in empirical studies in many research fields (Dreiseitl and Ohno-Machado, 2002). It is widely regarded as the model of choice for situations where the occurrence of a binary (dichotomous) outcome is to be predicted from one or more predicting variables (Boateng and Abaye, 2019). Examples can be found in medical research (Boateng and Abaye, 2019), educational research (Niu, 2020), public health research (Lemon et al., 2003), political sciences (Nicolau, 2007), economics (Sloane and Theodossiou, 1994; Zaghdoudi, 2013) and many others.

Regardless of the context of application, it is oftentimes the case that the logit of the expected value of the dichotomous response variable cannot be explained solely by additive functions of the predicting variables. In other terms, when the function $f(x, y)$ cannot be expressed as $g(x) + h(y)$ for some function g and f, we say that there's an interaction in f between x and y.

Many of the aforementioned field of research share the need for introducing these interaction effects between their predictors to better infer on - and characterize - the outcome. For instance, in Genome-Wide Association Studies (GWAS) there is increasing awareness that *epistasis*, or gene-gene interaction, plays a role in susceptibility to common human diseases (Moore, 2003; Onay et al., 2006). It has been argued (Moore, 2003) that epistasis is a ubiquitous component of the genetic architecture of common human diseases and that complex interactions are more important than the independent main effects of any susceptibility gene.

This calls for novel statistical approaches to identify the complex non-additive relationships between multiple variables to include into a predictive model that would fully capture the underlying relationship with the target. However, when dealing with real-world applications of LR and interaction search such as the one described above, several additional considerations are to be kept in mind.

First of all, nowadays datasets are growing *wider*, but the sample size may vary from extremely large to very limited. Linear models scale relatively well when handled via standard software, up to thousand of features together (Lim and Hastie, 2015). However, scalability becomes an issue when dealing with interactions. Indeed, in a model with p numerical predictors that includes a constant, every predictor and every possible interaction, there is a number of terms equal to

$$\sum_{k=0}^{p} \binom{p}{k} = 2^p$$

This exponential growth gets even steeper in the presence of categorical covariates, as each one of the features' levels has to be considered. Nevertheless, dealing with a huge moltitude of categorical covariates is getting more frequent for instance in the aforementioned GWAS studies, where interactions between millions of Single Nucleotide Polimorphisms (SNPs) with triplets of levels affect even extremely rare mutations in the patients' phenotype. This makes the inclusion of higher order interactions technically intractable, and causes the problem of finding interactions to fall in the framework of "large p small n" problems, where the number of features significantly surpasses the number of observations and the *curse of dimensionality* affects undeniably the reliability of statistical modeling.

In particular, when fitting LR models with no prior knowledge regarding the structure of the parameters, the traditional choice is Maximum Likelihood Estimation (MLE). However, a recent contribution from Sur and Candès (2019) shows how as p grows w.r.t. n, estimates seem systematically biased, in the sense that effect magnitudes are overrestimated, they are far more variable than classically predicted, and inference measures, e.g. p-values, are unreliable especially at small values. This makes both prediction and interpretation of results extremely questionable and hinders the validity of the analysis.

Moreover, oftentimes researchers are forced not only to deal with complex dynamics in the intrinsic structure of data, high dimensionalities and small sample sizes, but they're required to cluster observations that show a significant disproportion when it comes to the number of observed cases in one, or more, of the classes. The extremely common issue of class imbalance requires specific tailoring of classification models that

would otherwise incur in the risk of losing predictive power on the underrepresented class. This effect would be particularly concerning in some application areas, such as healthcare or insurance fraud, where correctly identifying the underrepresented class is the most - if not the only - relevant matter.
It is for all these reasons that newly developed statistical methods need to be tailored to address the aforementioned complexities, in order for LR to be significantly and effectively applied in most research domains. Our present work goes in this direction and focuses on developing a novel approach to Interaction Learning specifically designed to: (i) identify a restricted number of most salient interactions of arbitrarily high order among *categorical* features; (ii) optimize the selection of interactions for the separation of the classes in cases of strong imbalance; (iii) allow the user to keep control over the number of interactions to include in the model, to foster significance and interpretability of the resulting LR. All these aspects were considered in the design of the proposed algorithm while guaranteeing (iv) scalability, tractable computational times, and robustness to varying dimensionality, sample size and imbalance ratio.

1.1. Interaction Learning Approaches and Our Contribution

Discovering interactions is an active area of research (Lim and Hastie, 2015). However, when discussing related approaches in the multifaceted scenario our proposal deals with, it is important to make a distinction between two fairly independent lines of research that address different aspects of the subject matter.

Note that, as previously mentioned, our proposition focuses on identifying interactions between categorical features, where the risk of including an intractable number of covariates when considering interaction terms is extremely high. Indeed, the first relevant line of research deals with a similar setting - even though some studies try to expand their applicability to continuous features - and it is devoted to finding *lists* of high-order feature interactions associated with a target variable. One example can be found in Shah and Meinshausen (2014). The group of researchers devoted to *significant (discriminative) pattern mining* expands this concept taking into account the statistical significance of the association, controlling the *Family-Wise Error Rate (FWER)*, or the probability to detect false positive patterns (Llinares-López et al., 2015; Papaxanthos et al., 2016; Pellegrina and Vandin, 2018; Sugiyama and Borgwardt, 2019). These works found a broad range of applications in some of the domains we mentioned beforehand, such as statistical genetics or healthcare (Llinares-López et al., 2019; Ceddia et al., 2020). However, their final objective is that of identifying all significantly associated interactions, without specific requirements on the number of selected patterns or their predictive power and tractability once fed to a classification model as LR.

The other related research line focuses more on the identification of interactions to use within non-linear models and Generalized Linear Models (GLM) like LR. Most of these methods deal with different regularization strategies to shrink models to the most useful primary effects and interaction terms (Radchenko and James, 2010; Rosasco et al., 2010; Bien et al., 2013; Lim and Hastie, 2015), while Chen et al. (2011) propose a stochastic search algorithm for variable selection in Bayesian LR.
One drawback of using penalized models when selecting among very large numbers of

features, is the complexity of directly controlling for the number of terms introduced in the final classification model - despite their undeniable potential in producing powerful classification models. Especially when dealing with high-order categorical interaction terms, even in the presence of a very strong penalization, the actual number of terms in a LR may easily explode. As previously mentioned, the growing number of predictors may lead to an inflation of the estimates, hindering the attempt to make significant inference out of the resulting model, besides reducing the results interpretability.

Our contribution lies at the crossing of these two research lines. Indeed, we first develop a *targeted* (as explained later) high-order interaction search algorithm to produce a list of useful interactions associated with the target variable and ranked on the basis of their Odds Ratio (OR). Then, we introduce a novel feature selection method designed to pick from the list only a predefined number of salient interaction terms to include in a LR model that would discriminate well between classes (even in cases of small sample size and imbalanced setting) and allow for interpretable results and significant statistical inference.
The algorithm was developed in Python 3 (Van Rossum and Drake, 2009), and the package is available upon request.

The paper is structured as follows. In Section 2 we detail our methodology, starting from the main algorithm, `HOILP`, and then discussing a few possible alternatives (respectively in 2.3.1 and 2.3.2). In Section 3 we validate our design choices for the algorithms and provide empirical evidence that both `HOILP` and its variants can perform very well in binary classification tasks with categorical covariates, even under conditions of high imbalance and low sample size. In Section 4, by running experiments on both simulated and real data, we compare our approaches with `glinternet` (Lim and Hastie, 2015), a state-of-the-art algorithm for interaction learning in contexts of Logistic Regression. There we show that `HOILP` and its variants can perform comparably well, with the additional advantage of returning models that are much more interpretable. In Section 5 we draw the conclusions and discuss possible future developments. Finally, more technical algorithmic details and mathematical insights can be found in the Appendix, respectively in subsections A and B.

2. High-Order Interaction Learning via Targeted Pattern Search

In this section we formally introduce `HOILP`, the proposed `H`igh-`O`rder `I`nteraction `L`earning via targeted `P`attern search algorithm.

Note that the method was developed to handle interactions among categorical covariates. Therefore, we will firstly provide some context and notation to describe the specific setting we are dealing with and its intrinsic peculiarities. Then, we will detail the two main steps of the algorithm, that together represent the Interaction Learning core of `HOILP`, i.e. the preliminary *targeted* pattern search and the subsequent novel dissimilarity-based interaction selection method.

Because of our interest in developing a methodology broadly applicable to most real-life research domains, we include in this work two variants of the main algorithm - namely

`Scores HOILP` and `Clusters HOILP`. As further detailed in the last part of this section, these two alternative strategies are meant to satisfy specific user and data requirements, while also improving interpretability of the model.

2.1. Notation

Let us first provide some background to set the scene and introduce the needed notation to follow the details of our proposed methodology.
We are given p categorical random variables $X_1, .., X_p$, where each X_j takes values in some set of labels $\{l^{(j)}_{m_1}, .., l^{(j)}_{m_j}\}$. Notice that the number of levels m_j depends on j: in particular, the categorical variables may have different supports.

First of all, we define the so-called dummy variables

$$X_j^{(i)} = \mathbb{1}_{\{l^{(j)}_{m_i}\}}(X_j) \qquad j = 1, .., p, \quad i = 1, .., m_j$$

so that $X_j^{(i)}$ is $\{0, 1\}$-valued and equals 1 if and only if X_j attains the ith level. Next, we introduce the collection of all possible interaction terms among the covariates $X_1, .., X_p$ as

$$\mathcal{I} = \left\{ \prod_{j \in J} X_j^{(i_j)} \quad \middle| \quad J \subseteq \{1, .., p\},\ i_j \in \{1, .., m_j\} \right\}$$

so that each interaction is a binary r.v. that encodes the co-presence of certain level combinations. We adopt the standard convention by which the *order* of an interaction corresponds to the number of terms within it (equivalently, its degree as a polynomial) minus one; for instance, $X_1^0 X_3^1$ defines a first order interaction. With little abuse of notation we let $1 \in \mathcal{I}$ be the empty interaction ($J = \emptyset$) and we allow for 0-order interactions, meaning that all dummies also fall in $\mathcal{I}$.

It is straight forward to see that

$$|\mathcal{I}| = (m_1 + 1) \cdot ... \cdot (m_p + 1)$$

in fact, when defining an interaction each X_k leave us with exactly $(m_k + 1)$ choices: discarding that variable or including one of the corresponding m_k dummies. In particular, if $m_1 = ... = m_p = m$, then the cardinality of $|\mathcal{I}| = (m + 1)^p$ grows exponentially with p.

For more readability, let us introduce a few more preliminary definitions:

- For an interaction $T \in \mathcal{I}$ we write $|T|$ to denote the number of dummy variables involved in the product representation of T;
- Given two interactions $T,\ S \in \mathcal{I}$ we say that T is a subinteraction of S if all its dummies appear in the product expansion of S; equivalently, $S = T \cdot Z$ for some $Z \in \mathcal{I}$;
- For $T, S \in \mathcal{I}$ we define their maximum common divisor, MCD(T, S), as the highest order interaction $Z \in \mathcal{I}$ that is both a subinteraction of T and S;

- We say that two interactions T and S are incompatible, and we write $T \perp S$, if they respectively include two different levels of a same random variable. Equivalently, $T \cdot S$ is identically zero, whichever is the joint distribution of the underlying r.v.s $(X_1, .., X_p)$.

2.2. Interaction Learning

We are given a dataset $\mathcal{D} = \{(x_{1,i}, ..., x_{p,i}, y_i)\}_{i=1}^n$ consisting of n i.i.d. observations of p categorical covariates $X_1, ..., X_p$ and a binary target variable Y. We assume to be in an imbalanced setting with respect to Y, that is we suppose the classes $\mathcal{O} = \{i \mid y_i = 1\}$ and $\mathcal{Z} = \{i \mid y_i = 0\}$ to come in remarkably different proportions. Without loss of generality, we consider the case of $\mathcal{O}$ being the minority class, which reflects the situation of $\mathbb{P}(Y = 1) \ll \mathbb{P}(Y = 0)$.

Our first purpose is to deduce from the data a suitable subcollection of interactions that is both small in size but also informative w.r.t. Y. To achieve such goal, we perform two steps: identification of all the candidates interactions through a targeted pattern search; choosing of the top K most relevant interactions via our novel dissimilarity-based interaction selection method, where K is user-specified and tipically satifies $K \ll |\mathcal{I}|$.

2.2.1. Targeted pattern search

In principle, starting from the data in $\mathcal{D}$, it is very easy to compute the sample values corresponding to any given $T \in \mathcal{I}$. However, as in general the cardinality of $\mathcal{I}$ is huge, it is seemingly impossible to study each conceivable interaction term alone; it is this redvery drawback that highlights the need of finding a preliminary subset of candidates $\hat{\mathcal{I}} \subset \mathcal{I}$.

To address this task, we propose to focus only on those interactions that appear (in the sense that they equal 1) within the minority class $\mathcal{O}$ with an empirical frequency that is higher than a fixed threshold $supp_{\min} > 0$.

This choice, which we further discuss in Section 3.3 and we will also refer to as *One-Class-Learning* (OCL), is mainly driven by the need of undertaking the imbalance between the two classes, but has also a few advantages in terms of computational cost.

In what follows, for an interaction $T \in \mathcal{I}$ we write t_i to intend its corresponding ith observations in the dataset. redAs aforementioned, we propose to scan the minority class data in order to define

$$\hat{\mathcal{I}} := \left\{ T \in \mathcal{I} \text{ such that } \frac{1}{|\mathcal{O}|} \sum_{i \in \mathcal{O}} t_i > \delta \right\}$$

Again, it is not feasible to actually compute such frequencies for all $T \in \mathcal{I}$, which poses the question on how to actually determine $\hat{\mathcal{I}}$. To overcome this issue, we notice that in our context there is a useful duality between the concepts of interactions and *itemsets*. The latter, which are also referred to as *patterns*, are tipically found in problems of association rule learning such as market basket analysis (Borgelt, 2012), intrusion detection (Bekti et al., 2017) and others. There, one is given a set of possible *items* and a list of *transactions*, which are nothing but sets of items. The goal is then to mine the

list of such transactions in order to find which subsets of items (itemsets) occur with a sufficient frequency (also called *support*).

In our framework, we observe that there is a one-to-one correspondence between itemsets and interactions, as these can be thought as being itemsets of dummy variables. In fact, if we interpret dummy variables as items, then the observations $\{(x_{1,i}^{(1)}, ..., x_{p,i}^{(m_p)})\}_{i=1}^n$ are nothing but transactions (where an item is picked if and only if its corresponding dummy equals 1); similarly, a certain pattern of dummies is present in the transaction if and only if the corresponding interaction term is nonzero.

This is very useful as it allows us to take advantage of all the wide variety of frequent item set mining algorithms, which are recently becoming more and more efficient (Borgelt, 2012; Shah and Meinshausen, 2014). In particular, we may reformulate the problem of computing $\hat{\mathcal{I}}$ as that of finding a list of patterns that appear with a frequency greater than $supp_{\min}$ (minimum support) within the minority class. Many possible implementations - such as the *Apriori* algorithm, which we later make use of - are available for our purpose, and it can be convenient to test different ones depending on the data.

After having constructed the candidate set $\hat{\mathcal{I}}$, preliminary to the next phase we build for each $T \in \hat{\mathcal{I}}$ the contingency table of (T, Y) along the whole dataset $\mathcal{D}$. Doing so implies at least computing the corresponding frequencies in the majority class, $\frac{1}{|\mathcal{Z}|}\sum_{i\in\mathcal{Z}} t_i$. We point out that, beside the possibly high-cardinality of $\hat{\mathcal{I}}$, this step can now be performed very efficiently: for more technical details, we refer to the Appendix, Section A.1.

2.2.2. A dissimilarity measure based feature selection algorithm

Knowing the contingency table (T, Y) for every $T \in \hat{\mathcal{I}}$ allows us to associate to each candidate interaction an odds-ratio:

$$\mathrm{OR}_T := \frac{\left(\sum_{i:\, t_i=1} y_i\right) / \left(\sum_{i:\, t_i=1}(1-y_i)\right)}{\left(\sum_{i:\, t_i=0} y_i\right) / \left(\sum_{i:\, t_i=0}(1-y_i)\right)}$$

We use such statistic to rank the candidate interactions in $\hat{\mathcal{I}}$: more precisely, we sort the patterns in descending order according to the quantity $\log|\mathrm{OR}_T|$, so that reciprocal odds-ratios are treated equally.

Next, given a fixed $K \in \mathbb{N}$, we propose a way to extract from the sorted list of candidates $\{T_k\}_{k=1}^L$ a subset of (at most) K interactions that is suitable for inferring on Y. Naively, one may think of selecting the first K patterns in the list. However, this approach may present several drawbacks (cfr. Section 3.3): due the combinatorial nature of high-order interactions, such strategy could easily result in a list of nested patterns (subinteractions) that carry similar information; if K is small, this may lead to overfitting-like phenomena, as the feature space is not explored sufficiently.

As a remedy, we introduce a dissimilarity measure, $d : \mathcal{I} \times \mathcal{I} \to [0, +\infty)$, by letting

$$d(T,S) := \begin{cases} |T| \vee |S| & T \perp S \\ |T| \vee |S| - |\mathrm{MCD}(T,S)| & \text{otherwise,} \end{cases}$$

where $x \vee y := \max\{x, y\}$. This allows us now to compare two different patterns. Notice that the definition of d does not rely on the data, instead it directly involves the formal structure of the interaction terms.

By definition, d returns higher dissimilarities for patterns that involve completely different variables (empty MCD) and for those that are incompatible. Because of these two properties, we consider d a suitable measure for exploring different regions of the feature space. For the sake of readability, we avoid diving here deeper into the mathematical properties that d enjoys: for a few straightforward insights we refer to the Appendix, Section B.

By making use of the dissimilarity measure d, we extract K patterns from the sorted list $\hat{\mathcal{I}}$ according to the following iterative procedure:

1. We remove from $\mathcal{I}$ the first interaction T, which corresponds to the one that maximizes $\log|\mathrm{OR}_T|$, and place it in a new list $\mathcal{I}_K$;
2. We search for those interactions in $\hat{\mathcal{I}}$ that are the most dissimilar from the ones in $\mathcal{I}_K$. In other words we determine $\mathrm{argmin}_{T\in\hat{\mathcal{I}}} \min_{S\in\mathcal{I}_K} d(T, S)$. Among these, we select the one with highest rank and move it from $\hat{\mathcal{I}}$ to $\mathcal{I}_K$;
3. We repeat the instructions in (2) until $\mathcal{I}_K$ contains K interactions.

In the end, we are left with a collection $\mathcal{I}_K = \{T_k\}_{k=1}^K$ of K interactions which can now be used as predictors in a logistic model with response Y. In other words, our final model reads

$$\mathrm{logit}\mathbb{P}\left(Y = 1|T_1, .., T_K\right) = \beta_0 + \sum_{k=1}^{K} \beta_k T_k \tag{1}$$

We conclude this paragraph with a final remark. Especially when the sample size is not substantial, it can be convenient to primarly filter the candidate list $\hat{\mathcal{I}}$ by removing those interactions that have an odds-ratio suspiciously close to 1. For example, as we did in our case study (cfr. Section 4.2), one may choose a $\gamma \in (0, 1)$ and discard all those patterns whose γ-level confidence interval for the odds-ratio contains the value 1. In general, this will not affect the ranking and the final steps described above, but can result in more robust models.

2.3. Variants

In this paragraph we explore two variants of our algorithm which aim at increasing interpretability and tractability of the final model. Although in our context of categorical predictors even high-order interactions remain highly interpretable (as aforementioned, they encode the co-presence of multiple levels of different variables), further reductions in the model structure can be of great interest in certain applications (cfr. Section 5).

Below we discuss two possible approaches: `Scores HOILP`, which condenses all the information into just a pair of variables; `Clusters HOILP`, that is instead a possible compromise between the other two, where the K identified interactions are grouped into multiple *Compatibility Clusters*. Note that, in both cases the number of terms in the

model is smaller than K (or at most equal). This makes the two alternatives a go-to solution in case we needed to include the information carried by several interactions (for instance, in a $p \gg n$ setting), without incurring in the risk of ending with an LR model that has too many terms w.r.t n.
As both variants necessitate of a further reshaping of the information carried by the selected interaction terms, we can expect their performances to be slightly worse w.r.t. the traditional `HOILP` where all terms are included independently. However, as mentioned beforehand - and further expanded in the Discussion (Section 5) - these variants present several advantages in terms of both interpretability and dimensionality reduction, two aspects that play a key role in certain research scenarios and make these algorithms widely applicable.

2.3.1. Risk and Protection Scores: `Scores HOILP`

In Section 2.2, we ranked the candidate interactions in a unique list using the log odds-ratio as criteria. However, we may avoid mixing together terms that exert opposite influences on the target and instead split the collection $\hat{\mathcal{I}}$ into two sublists

$$\hat{\mathcal{R}} := \{T \in \hat{\mathcal{I}} \mid \mathrm{OR}_T > 1\}, \quad \hat{\mathcal{P}} := \{T \in \hat{\mathcal{I}} \mid \mathrm{OR}_T < 1\}$$

which we may now sort separately, respectively by descending and ascending odds-ratios. We refer to the two collections as *risk* and *protection patterns*, a terminology that is mostly motivated by the majority of clinical applications.

Given an even integer K, we then apply our dissimilarity selection algorithm in order to extract two groups, $\hat{\mathcal{R}}_{K/2} \subset \hat{\mathcal{R}}$ and $\hat{\mathcal{P}}_{K/2} \subset \hat{\mathcal{P}}$, each of dimension $K/2$. Then, we construct the Risk Score R and Protection Score P as

$$R := \sum_{T \in \hat{\mathcal{R}}_{K/2}} T, \qquad P := \sum_{S \in \hat{\mathcal{P}}_{K/2}} S$$

The intuition between the two scores is that R (resp. P) counts the number of selected risk (protection) patterns that are present within a certain observation. In this way, the information relevant for inferring on Y gets squished into a single pair of highly interpretable variables. The final model is then

$$\mathrm{logit}\mathbb{P}\left(Y = 1 | R, P\right) = \mu_0 + \alpha R + \beta P \tag{2}$$

2.3.2. Compatibility Clusters: `Clusters HOILP`

The approach described in the previous section has the advantage of substituting the p original covariates X_i with two scores that carry information about possible interactions. However, compressing the two lists, $\hat{\mathcal{R}}_{K/2}$ and $\hat{\mathcal{P}}_{K/2}$, directly into the scores R and P could be too rough.

Here we propose a milder aggregation criteria, the *compatibility clusters*, which is based on the idea that we should not bind together incompatible patterns. In principle, in fact, no observation can ever present two incompatible patterns simultaneously; therefore, we may want to weight them differently in our model. Concretely, after having

defined $\hat{\mathcal{R}}_{K/2}$ and $\hat{\mathcal{P}}_{K/2}$ as in the previous section, we focus on each of them separately and propose a way to aggregate their terms. Starting from $\hat{\mathcal{R}}_{K/2}$, we wish to find a partition $\hat{\mathcal{R}}^1_{K/2} \cup ... \cup \hat{\mathcal{R}}^a_{K/2} = \hat{\mathcal{R}}_{K/2}$ for which: 1) all interactions within the same subgroup $\hat{\mathcal{R}}^j_{K/2}$ are compatible with one another; 2) given two subgroups there always exists at least a pairing of incompatible patterns, that is: no subgroups can be fused together without violating the first rule.

Stated as it is, there is no unique way of doing this. In our experiments, we decided to construct the groups in a way that their total number was as small as possible, thus maximazing the information compression. Further details on the implementation can be found in the Appendix, Section A.2.

Performing the described steps also for the protection patterns allows us to define the cluster scores

$$R_j = \sum_{T \in \hat{\mathcal{R}}^j_{K/2}} T, \quad P_j = \sum_{T \in \hat{\mathcal{P}}^i_{K/2}} T,$$

for $j = 1, .., a$ and $i = 1, .., b$. The final model reads:

$$\text{logit}\mathbb{P}\left(Y = 1 | R^1, .., R^a, P^1, .., P^b\right) = \mu_0 + \sum_{j=1}^{a} \alpha_j R_j + \sum_{i=1}^{b} \beta_i P_i \tag{3}$$

Comparing (3) with (2), we see that now we are splitting the risk and protection scores into more sub-scores, arguing that each one describes different, possibly incompatible, situations.

3. Simulation Study

This section is devoted to verify some of the statements made on the most relevant aspects of the proposed algorithm throughout the paper. In order to provide these empirical evidences in a controlled setting, we had to build a suitable simulation protocol. Indeed, to be meaningful, the following experiments required a dataset where the binary independent variable showed complex dependencies with multiple categorical covariates simultaneously, so that learning the right interactions would be essential to separate the two classes. In Section 3.1 we describe how the needed data was defined and simulated; then, on such data, we compare the performances of `HOILP` and its variants with those achieved by other logistic models that do not account for interaction effects.

The subsequent `experiments` are instead devoted to (i) verifying whether the *"targeted" search* (or OCL), focused on identifying patterns in the positive class only, could actually compete against the same algorithm fed with lists of interactions extracted from both classes. Then, (ii) we validate the value added by the *dissimilarity-based* feature selection in building the final set of interactions to include in the LR, w.r.t. picking the top K interactions ranked on ORs. Finally, we verify the robustness of `HOILP` to variations in (iii) sample size and (iv) class imbalance rate.

3.1. Simulated Data

For the definition of the datasets adopted in all the simulation experiments described in the following, we exploited a geometric model. This allowed us to synthetically introduce the aforementioned outcome-covariates dependency based on *multiple interactions*.

The underlying geometric model is defined as a double squared tiling structure (Fig.1). Each instance in the dataset is represented by a couple of triangular tiles, one picked from the left tiling and one from the right with uniform probability. Two example instances are reported in Fig.1.a. All the covariates in the dataset are binary representations of the position of one of the two tiles (from left or right tiling). Indeed, the position of the first tile on the left is described by 5 binary features defined as the following:

- R_1 (*right*) takes value 1 if the tile is in the right half, 0 otherwise.
- U_1 (*up*) takes value 1 if the tile is located in the upper half of the tiling, 0 otherwise.
- D_1 (*diagonal*) takes value 1 if the tile is located below the main diagonal of the tiling, 0 otherwise.
- A_1 (*anti-diagonal*) takes value 1 if the tile is located below the anti-diagonal, 0 otherwise.
- O_1 (*outside*) takes value 1 if the tile is outside the *inner square* (the 45-degrees rotated square - composed of 8 triangular tiles - that has its vertices on the mean points of the tiling's edges), 0 otherwise.

Similarly, R_2, U_2, D_2, A_2 and O_2 describe the position of the second tile in the right squared tiling. Notice that to recover the notation adopted in Section 2 one may simply let $X_1 := R_1$, $X_2 := U_1, \ldots,$ $X_{10} := O_2$.

The dependent variable Y, that defines the two classes ($y = \{0, 1\}$), signals whether at least one of the two tiles of the observations falls within one of the red areas highlighted in Fig. 1.b. As such, Y is univocally determined by the 10 covariates described above. However, in order for the classification problem to be less deterministic, we artificially added noise to the data by mislabeling some observations. In particular:

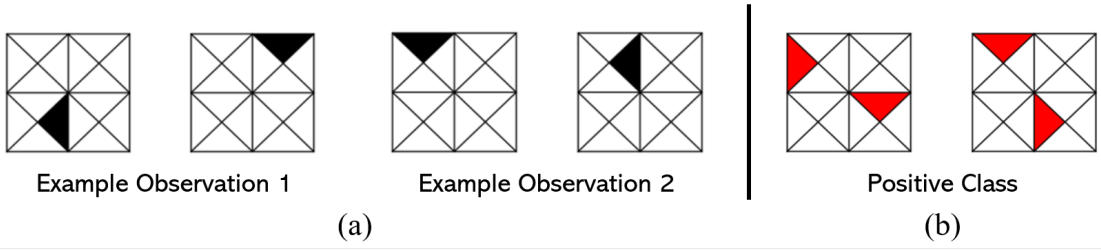

Fig. 1. Geometric model behind simulated data. On the left (a) are two illustrative observations in the dataset, represented by two tiles from each squared tiling. On the right (b) represents the tiles that determine the positive class.

- If both tiles of an instance i (left and right) fell in one *red area* (hence, $y_i = 1$), the label is changed to $y_i = 0$ with a probability $\tilde{p} = 0.005$.
- In all other cases, the value of y is changed to the opposite class with a probability $\tilde{q} = 0.05$.

There are several reasons motivating this construction of the dataset. First of all, note that the two classes are unbalanced by construction. Indeed, before introducing the noise, one has $\mathbb{P}(Y = 1) = 1 - \mathbb{P}(Y = 0) = 1 - \left(\frac{14}{16}\right)^2 \simeq 23.4\%$, as the two tiles are extracted independently and, due to uniform distribution, each one of them has a probability of 14/16 of not falling within a red zone. Secondly, strong dependencies exist within groups of covariates: for instance, if $R_{1,i} = U_{1,i} = 1$, then necessarily $A_{1,i} = 1$. Lastly, the minority class ($y_i = 1$) is built in such a way that it would be impossible to attain a good classification performance without introducing interaction terms in the model, as the red tiles need to be described with more variables simultaneosly to be properly identified in the tilings. It is indeed not surprising how the experiment in the following section makes evident the inaccuracy of a LR with no interaction terms on this data.

3.2. Learning with or without interactions

As mentioned, the simulated data was built with the aim of challenging the traditional LR models that try to classify observations exploiting the additive effect of the predictors alone. As a starting point, we hence compare the performance of a traditional LR and a Lasso Regression without interaction terms against our proposed algorithm in its three variants. For this experiment, we adopted the previously described procedure in order to generate 10 datasets, each of n =10,000 observations with $\mathbb{P}(Y = 1) \simeq 0.234$. We then splitted each dataset into training and test sets according to a 70/30 ratio. It is clear from Table 1 how considering only main effects and their additive contribution makes the model practically useless in discriminating between the two classes ($AUC = 0.49 \pm 0.015$ for both LR and Lasso). The Lasso Regression was performed imposing $\lambda = 10^{-5}$, because larger λ values shrinked the model to no term at all.

Note that our algorithm (here with $K = 10$ and $supp_{\min} = 10\%$) is yielding a very high score in all its variants. Nonetheless, it is not surprising to see `HOILP` using as-is interactions obtain the highest AUC.

One of the positive aspects that should not be undervalued about building statistical models with high-order interactions derived from categorical features, is the undeniable interpretability of the resulting model itself. This is especially true when applying `HOILP`, that returns an arbitrarily long list of high-order interaction terms selected in order to (i) be highly descriptive of one class in particular and (ii) cover as much diverse information as possible regarding such class - thanks to the dissimilarity-based selection. In Figure 2 we provide a concrete demonstration of this claim. We represented graphically the 10 interaction terms included in the model generated by `HOILP` in one of the trials of this experiment. The tiles in each pair of squared tilings are colored higlighting the areas defined by the dummies included in each of the $K = 10$ selected interactions. Red

Table 1. Results on the simulated dataset for `glinternet` (with varying number of interactions and consequent number of terms) and `HOILP` (last row).

Model	Terms	AUC	Sensitivity	Specificity	NPV	PPV
LR	20	0.494 ± 0.015	0.465 ± 0.069	0.603 ± 0.105	0.759 ± 0.015	0.299 ± 0.035
Lasso	9.900 ± 0.316	0.494 ± 0.015	0.465 ± 0.069	0.604 ± 0.105	0.759 ± 0.015	0.299 ± 0.034
`HOILP`	10	**0.919 ± 0.007**	**0.862 ± 0.015**	**0.945 ± 0.034**	**0.951 ± 0.006**	**0.855 ± 0.077**
`Scores HOILP`	**2**	0.835 ± 0.022	0.710 ± 0.052	0.845 ± 0.070	0.892 ± 0.016	0.634 ± 0.087
`Clusters HOILP`	7.400 ± 0.516	0.853 ± 0.031	0.726 ± 0.044	0.840 ± 0.070	0.896 ± 0.016	0.630 ± 0.092

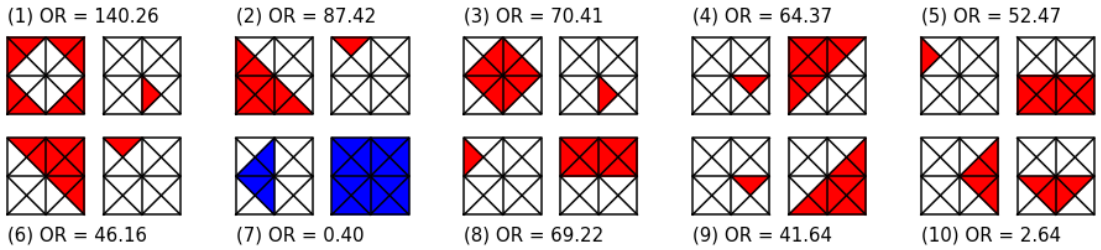

Fig. 2. K patterns identified by `HOILP` in one trial of the first simulation experiment (where $K = 10$). Tiles are colored according to the areas defined by the categorical terms involved in each of the selected interactions. Red patterns are considered *risk patterns* ($OR > 1$), while *protection patterns* ($0 < OR < 1$) are colored in blue.

and blue coloring depend on the OR. The order in which the terms are drawn (to be read by row), is the order in which the algorithm picked each of them. This means, for instance, that the only blue pattern was selected as 6^{th} interaction to be included. It can be easily noticed how each of the selected interaction describes precisely an aspect of the two classes. For instance, the first pattern spots one triangle in the right tiling that appears to be associated with the minority class, which is coherent with the original construction of the data (cfr. Figure 1.b). In contrast, the second selected interaction highlights a completely different (but still consistent) area. It is interesting to see how the only *protection pattern*, despite not telling much about the right tiling, precisely defines an area in the left one where no minority class observation should fall.

3.3. One-Class Targeted Search and the relevance of Dissimilarity-based Interaction Selection

Irrespectively of whether the objective were to fit a classification model, or to identify all relevant patterns, the search for high-order interactions is a computational challenge by nature because of its combinatorial character. As extensively discussed above, `HOILP` algorithm exploits the *interaction-item set* duality to extract relevant high-order interactions from the training data. To do that, it employs the widely recognized Apriori algorithm for frequent item set mining, focusing the search on the positive class observations only. Among other things, the experiment here conducted in our simulated setting is meant to support this approach by empirically demonstrating how it not only does not hinder the classification of both classes, but it may foster performance and save time w.r.t. extracting item sets from the two.

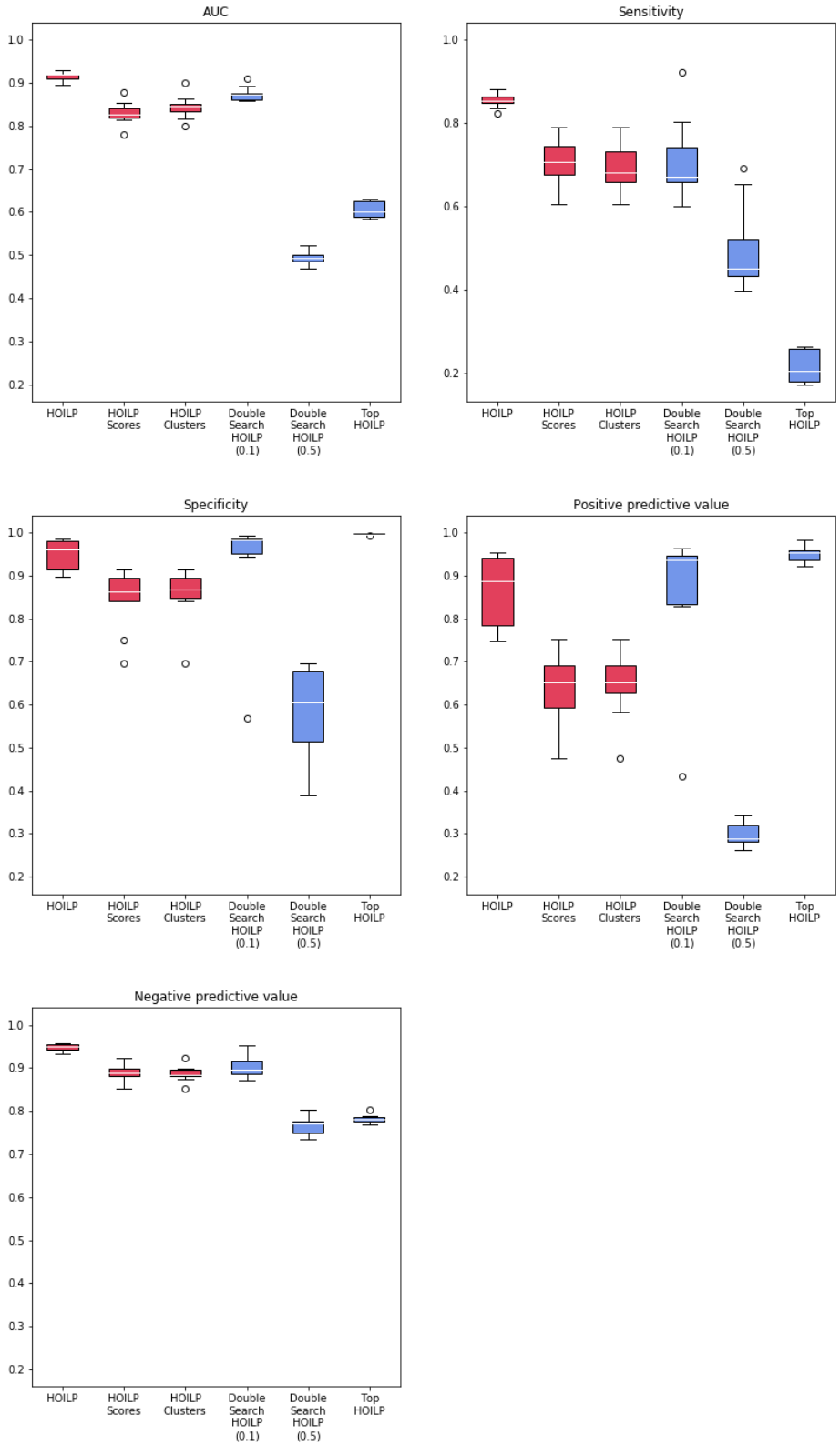

Fig. 3. In order, performance of HOILP in its three interaction-representation versions (Individual Interactions, Scores and Clusters), against DS-HOILP with $supp_{min} = 0.1$, DS-HOILP with $supp_{min} = 0.5$ and TOP HOILP on simulated data.

Once proven the validity of this first relevant aspect of the proposed algorithm, the second fundamental design choice that deserves an empirical evaluation is the dissimilarity-based feature selection method. Indeed, this peculiar passage builds arbitrarily long lists of features (interactions) by selecting the most diverse among those with the highest OR. This method was designed under the assumption that a more diverse set of interactions might reduce noise, generalize better on unseen minority class observations by capturing the whole underlying class' distribution and collect the most useful interactions to interpret and characterize this group. We tested these ideas within the same experiment as above.

We exploited the same 10 datasets described in 3.2 but trained four different algorithms:

(a) `HOILP` in its traditional *One-Class-Learning* (OCL) version, searching on minority class examples only with $supp_{min} = 0.1$. We tested the algorithm in all its three variants.
(b) *Two-Class-Learning* `HOILP` (referred to as Double Search `HOILP`), building the list of interactions searching separately in both classes. Similarly to OCL `HOILP` we imposed $supp_{min} = 0.1$.
(c) Double Search `HOILP` with $supp_{min} = 0.5$.
(d) To evaluate the substantiation of the hypotheses on the value added by the dissimilarity-based feature selection, the fourth algorithm is a version of the OCL `HOILP` that only picks the top K interaction terms from the ranked list (Top `HOILP`).

In Figure 3 we provide the results of this experiment. The most traditional implementation of `HOILP` outperforms all other versions basically on all the considered metrics. First of all, note that the DS-`HOILP` with the same $supp_{min}$ performs almost comparably in terms of AUC, but Sensitivity is strongly affected by searching for patterns within both classes. This result testifies in favour of our OCL approach to foster classification accuracy specifically on the underrepresented class.

Interestingly enough, the DS-`HOILP` with the highest support is the worst performer on all dimensions: at first sight, this is surprising as one would expect highly frequent patterns in both classes to separate the two at best. However, raising the $supp_{min}$ value, especially for what concerns the minority class, can result in a poor performance whenever the separating patterns are very different and scattered along the data.

For what concerns `Top HOILP`, its performance is particularly low in terms of Sensitivity. Comparing this metric with the others, it is straightforward to deduce that the algorithm's performance is hindered by a very high number of False Negatives ($Sensitivity = TP/(TP + FN) \simeq 0.2$).

For the purpose of our empirical demonstration, this latter result is an interesting one, that sheds a light on one of the multiple reasons why the dissimilarity-based interaction selection step helps in capturing the real underlying distribution of minority class observations in a generalized and robust manner. Indeed, the K interactions selected by `Top HOILP` with their high OR are very specific to certain minority class observations ($PPV \simeq 1$), thus guaranteeing a precise majority class classification as well ($Specificity \simeq 1$, as the identified patterns are extremely rare in the overrepresented class), but their generalizability to the whole minority class is quite low.

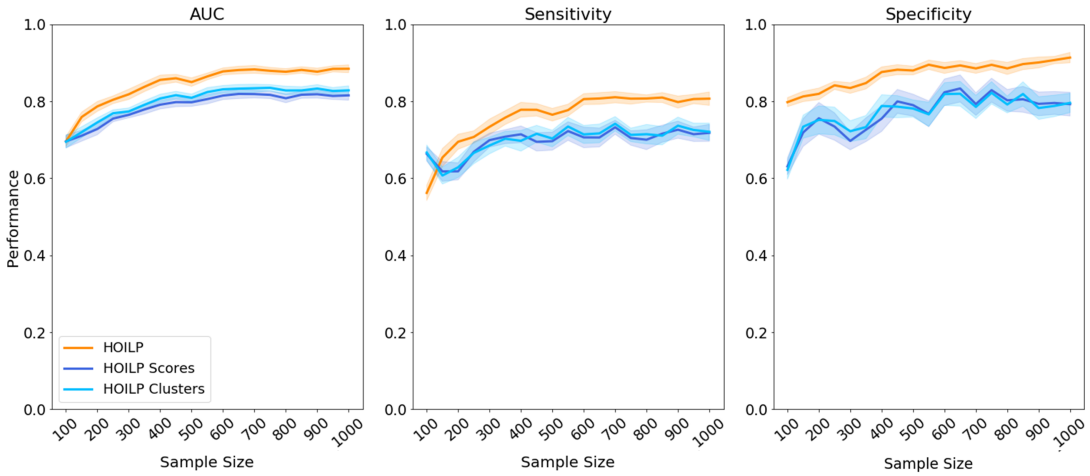

Fig. 4. Performance of `HOILP`, `Scores HOILP` and `Clusters HOILP` for varying sample sizes.

In other words, the algorithm is including a set of interactions that exist within the positive class only, but they all similarly describe one *subgroup* of this class - which appears to be quite distant from all other examples in the dataset. This situation may arise in the presence of outliers within the underrepresented class, or in case of an irregularly distributed minority class with two or more subgroups of observations. As a matter of fact, note that the simulated data has evident subgroups, as positive observations are defined by several distinct areas in the geometric space. This testifies in favour of the robustness of the dissimilarity-based feature selection method that would, by design, include at least one of the high OR patterns (thus describing the subgroup), but would then be forced to add *dissimilar* interaction terms, lowering the risk to overfit the group of outliers (or the subgroup), and providing a better generalization on the whole minority class population.

3.4. On the robustness to Sample Size

Nowadays, real-life research settings present an extremely wide range of different scenarios when it comes to sample size. Huge data sets in some domains are opposed to extremely limited samples in others, above all the healthcare or medical sector. Therefore, any novel statistical approach that aims at finding broad application needs to be flexible to the different situations it might encounter.

This experiment is meant to evaluate and discuss the performance of `HOILP` for extremely limited sample sizes. Indeed, as n grows larger, the only foreseeable drawback of the proposed methodology regards computational time. It has already been discussed what precautions were taken to limit the impact of extremely large datasets, therefore the focus here will be around the potential complexities in implementing `HOILP` on very few observations.

Indeed, mining item sets on one class only may easily induce low generalization capability of the identified interactions, if the sample is too small or poorly representative

of the real class' distribution.

Moreover, decreasing minority class sample size may bias the structure (and reduce the number) of *patterns* to evaluate for inclusion as interaction terms. Indeed, as the sample size gets smaller, fewer levels of each covariate may be represented in minority class examples.

With this experiment we wish to empirically demonstrate that thanks to the combined effect of the $supp_{min}$ parameter, the OCL search, and the dissimilarity-based FS, `HOILP` can handle extremely limited sample sizes granting a good performance on all evaluation metrics. Indeed, searching on minority class examples with a low threshold on $supp_{min}$ allows the algorithm to identify a sufficiently large set of interactions despite working on very few observations. Moreover, the dissimilarity-based selection lowers the risk to overfit the small pool of observed data points and include in the final model interactions that might generalize better on the overall population.

For the experiment we trained `HOILP`, `Scores HOILP` and `Clusters HOILP` on datasets of varying sample size with small n and $K = 10$. For each sample size ($n \in \{100, 1000\}$ with a step of 50 observations) we generated 50 training sets and test sets. In Figure 4 we provide the obtained results in terms of mean and standard deviation on several evaluation metrics. As the reader may notice, despite sample size gets significantly small and the number of terms included in the LR is rather limited (10 terms at most), the performance is not degrading significantly (lowest AUC score around 0.7 with the smallest sample of 100 observations only). Among the three, `HOILP` with ungrouped interactions performs significantly better than the other two versions. Again, this is not surprising as we are adding more terms to the LR without any information reprocessing. Nonetheless, the two other versions provide a solid performance as well, and might be useful in the case the number K of interactions had to grow larger, or to satisfy specific interpretability requirements.

3.5. On the robustness to Class Imbalance

The last relevant matter to discuss regarding the broad applicability of our proposed approach, is its capability to handle situations of strong class imbalance. As a matter of fact, the nature of the algorithm itself makes it almost unaffected by class imbalance ratios if the number of observations within minority class is sufficiently large. Indeed, the one-class-learning procedure vanishes the effect of class imbalance, that would instead make the problem rather intractable in case the algorithm searched on the whole dataset together.

Let us consider an extreme case in which the itemset i has empirical frequency $f_{i,1} \simeq 1$ among minority class and $f_{i,0} \simeq 0$ within majority observations. In what follows we write n_1 for the number of minority class observations and we let $p_1 := n_1/n$. Then, in order for the algorithm to include i (which is extremely relevant in discriminating between the two classes) in the list of interactions we should impose $supp_{min} < p_1$. In strongly imbalanced settings, p_1 can easily take values below 0.1. This would mean that including patterns specific to the positive class would force the imposition of extremely low $supp_{min}$ values, causing the list of potential interactions to explode. This would increase computational time, noise, and lower the probability of including strongly discriminative interactions in the model despite the high associated ORs.

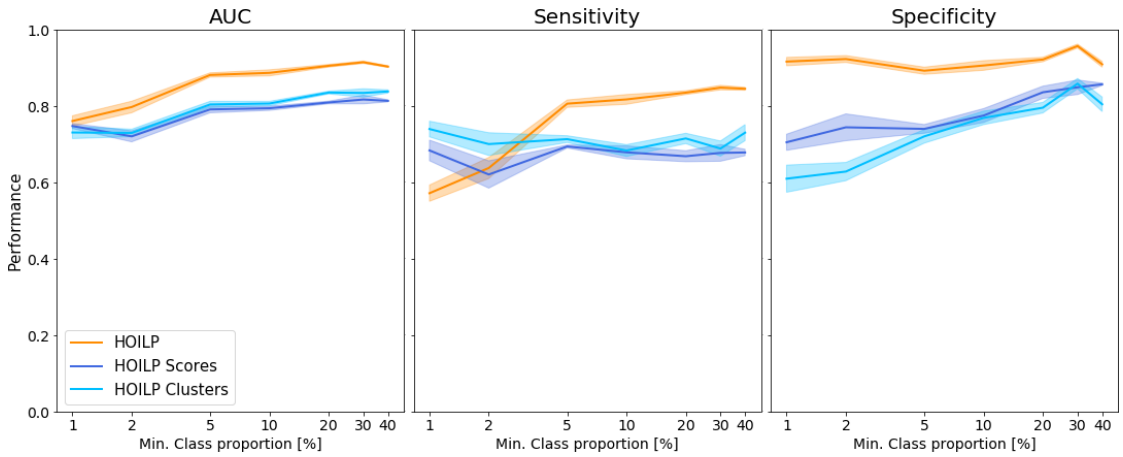

Fig. 5. Performance of `HOILP`, `Scores HOILP` and `Clusters HOILP` for varying imbalance ratios. The percentages on the x-axis (reported in log-scale) represent the portion of minority class observations on the whole dataset.

Instead, the OCL approach of `HOILP` overcomes this limitation and identifies relevant interactions irrespectively of the imbalance ratio. Indeed, in Figure 5 it can be noticed how none of the presented metrics are strongly affected by the incresingly imbalanced setting of the classification problem. For this experiment we generated 50 training and test set for each percentage of minority class observations, from $p_1 := n_1/n = 1\%$ to $p_1 = 40\%$, with $n = 5,000$. Minimum support was set to 0.1 and $K = 10$. Of course, in this case the algorithm was dealing with a rather large n. However, as we already discussed the performance for small sample sizes, we were more interested in observing the effect of the imbalance ratio only. The three variants demonstrate a solid performance for extreme imbalancement as well, and (especially for traditional `HOILP`) basically reach and robustly keep their best performance from 5% on.

4. Benchmark Experiments

As discussed in Section 1.1, our proposition lies at the crossing of two independent research lines, building on concepts borrowed from the best of both.

In particular, the group of works dealing with *significant pattern mining* provides a set of potential alternatives for the identification of the list of interaction to be filtered on a *dissimilarity* basis. As a first development of our methodology we decided to apply the most widely recognized item set mining algorithm, the Apriori algorithm. Nonetheless, future developments might include alternative pattern mining methods that might scale better, improve the efficiency or handle some data and problem-specific requirements more effectively. As a matter of fact, we see these methodologies more as potential additions to our algorithm, than actual competitors. Therefore, to compare the performance of our algorithm with a competing benchmark, we focused on the second line of research mentioned among related works. In particular, we chose the recent work of Lim and Hastie (2015), where the authors present Group-Lasso INTERaction-NET -

Table 2. Results on the simulated dataset for `glinternet` (with varying number of interactions and consequent number of terms) and `HOILP` (last row).

Param	AUC	Sensitivity	Specificity	Mean N Vars	Fitting Time [s]
$n_{inter} = 3$	0.843 ± 0.010	0.895 ± 0.008	0.760 ± 0.03	25.6	0.437 ± 0.057
$n_{inter} = 4$	0.867 ± 0.012	0.884 ± 0.023	0.810 ± 0.039	33.8	0.476 ± 0.049
$n_{inter} = 5$	0.878 ± 0.019	0.884 ± 0.012	0.837 ± 0.047	36.4	0.496 ± 0.047
$n_{inter} = 6$	0.891 ± 0.021	0.875 ± 0.016	0.879 ± 0.044	41.2	0.544 ± 0.063
$n_{inter} = 7$	0.899 ± 0.021	0.868 ± 0.018	0.916 ± 0.062	44	0.723 ± 0.505
$n_{inter} = 8$	0.920 ± 0.012	0.861 ± 0.013	0.968 ± 0.036	49.8	2.406 ± 1.490
$n_{inter} = 10$	0.924 ± 0.007	0.859 ± 0.011	0.985 ± 0.004	60.2	5.055 ± 1.198
K = 10	0.917 ± 0.009	0.854 ± 0.015	0.949 ± 0.036	10	1.185 ± 0.053

`glinternet`, a state-of-the-art approach when it comes to selecting interactions for LR models.

To compare `HOILP` with `glinternet` we applied both algorithms to simulated data and to a dataset from UCI Machine Learning Repository (Dua and Graff, 2017), namely Breast Cancer † dataset.

For readability and comparability of results there are a few considerations to make regarding `glinternet`. The algorithm is indeed a regularization-based interaction learning method and as such it does not allow for a precise control over the number of terms included in the model, especially in case of categorical covariates. In facts, the open source software‡ allows the user to define the number of interactions (n_{inter}) to include in the model, and then the algorithm iteratively relaxes the regularization to get as close as possible to n_{inter} interactions. The procedure stops when the algorithm reaches *at least* n_{iter}, meaning that the number of interactions may be larger than required. Moreover, `glinternet` includes main effects and interactions by enclosing the whole variable, with all its categorical levels. As a consequence, given A and B binary variables, including the interaction $A \times B$ requires 4 terms to be included in the LR model.

4.1. Benchmark comparison on simulated data

To test `HOILP`'s performance against the benchmark, we run `glinternet` on 10 simulated training and test set. To compare results on the basis of how many terms were introduced in the LR as well, we imposed several different n_{inter} values ($n_{inter} \in \{3, 4, 5, 6, 7, 8, 10\}$). Then, we collected the number of β parameters associated with each level of the selected covariates to count precisely the number of terms in the model. Table 2 reports the results of the experiment. Again, `HOILP` was trained with $K = 10$ and $supp_{\min} = 0.1$.

Our proposed algorithm performs notably better then the benchmark competitor on average, up until `glinternet` includes 10 interaction terms in the model. However, using 10 interactions translates in including on average 50 or 60 terms in the LR, as opposed to `HOILP` that is using 10 *pattern*-terms only. This makes the resulting model way more interpretable, and the estimated β parameters more robust.

† https://archive.ics.uci.edu/ml/datasets/breast+cancer
‡ cran.r-project.org/web/packages/glinternet/index.html

Table 3. Results on the Breast Cancer dataset for `HOILP` - (i) Individual interactions, (ii) Scores and (iii) Clusters, (iv) LR without interactions, (v) Lasso Regression and (vi) `glinternet` (with varying number of interactions and consequent number of terms).

Method	**Parameters**	**AUC**	**Sensitivity**	**Specificity**	**Mean N Terms**
`HOILP`	K = 4	**0.726 ± 0.068**	**0.692 ± 0.096**	0.683 ± 0.062	**4**
`Scores HOILP`	K = 12	**0.747 ± 0.064**	**0.692 ± 0.054**	0.725 ± 0.086	**2**
`Clusters HOILP`	K = 12	**0.746 ± 0.064**	**0.700 ± 0.057**	0.710 ± 0.084	**2.6 ± 0.516**
No Int. LR	-	0.643 ± 0.053	0.615 ± 0.083	0.677 ± 0.064	241
Lasso LR	$\lambda = 10^{-2}$	0.710 ± 0.086	0.665 ± 0.073	0.723 ± 0.064	12.2 ± 2.044
`glinternet`	$n_{inter} = 2$	0.715 ± 0.081	0.646 ± 0.149	0.765 ± 0.127	51.2
`glinternet`	$n_{inter} = 3$	0.716 ± 0.075	0.635 ± 0.123	0.772 ± 0.125	68.4
`glinternet`	$n_{inter} = 4$	0.712 ± 0.089	0.592 ± 0.206	0.810 ± 0.125	103.3
`glinternet`	$n_{inter} = 5$	0.710 ± 0.086	0.542 ± 0.175	0.863 ± 0.080	135.1
`glinternet`	$n_{inter} = 6$	0.705 ± 0.079	0.600 ± 0.126	0.808 ± 0.096	169.4
`glinternet`	$n_{inter} = 8$	0.701 ± 0.075	0.588 ± 0.106	0.813 ± 0.112	209.7
`glinternet`	$n_{inter} = 13$	0.687 ± 0.068	0.569 ± 0.155	0.818 ± 0.123	336.1

4.2. Breast Cancer Case Study

For this last benchmarking experiment it was chosen a freely available dataset from UCI Machine Learning Repository, namely Breast Cancer Dataset. This data set includes 201 instances of one class and 85 instances of another class. The instances are described by 9 categorical attributes, some of which are continuous binned into categories and some are nominal.

What makes this dataset interesting for this application is the rather small sample size (typical of a real-life medical experimental setting), and the varying number of levels per feature, ranging from 2 to 11. In Table 3 the results of the application of the three versions of `HOILP`, against `glinternet`, a LR model and a Lasso Regression both without interactions. Because of the small sample size, the cross-validation was performed by bootstrapping training and test set 10 times with splitting ratio 70/30.

Our algorithms were all trained with $supp_{\min} = 0.3$ and the additional use of 90% confidence intervals for the odds-ratios; we used different values for K, as reported in Table 3. `HOILP` and its variants consistently attain the best performance in terms of AUC and Sensitivity, using the smallest number of terms in the model. Lasso Regression and `glinternet` perform comparably, however the latter includes in the model an eccessive and disproportionate number of terms. Indeed, as the number of terms grows, the performance of `glinternet` worsens.

These results suggest that the interaction effect in describing the target variable in this particular dataset is not extremely evident, nor necessary. Indeed, the absence of interaction terms is not strong enough to hinder the classification capability of the Lasso. However, the co-occurrence of some specific dummy variables in the minority class seems to capture a relevant part of the variability in the data, allowing `HOILP` and its variants to provide the best accuracy with a very limited pool of patterns.

5. Discussion and Conclusions

In this paper we presented `HOILP`, an High-Order Interaction Learning algorithm via targeted Pattern search, to select high-order interactions among categorical covariates and build a LR model. Together with `HOILP`, we proposed two alternative strategies to represent and include the selected interactions in the model, namely `Scores HOILP` and `Clusters HOILP`, to respond to problem-specific needs of the research community using LR for inference and modeling. Throughout the sections we validated the claims regarding some relevant aspects of our proposed methodology through a wide set of experiment on simulated or real data. In particular, as `HOILP` was designed specifically to address a broad range of real-life data analysis issues, we wish to conclude the paper by recalling and discussing the most notable pros of the algorithm.

First and foremost, `HOILP` was designed to make LR models more powerful by introducing interaction terms between categorical covariates. However, to make the resulting model reliable for inference, a strict control over model dimensionality is needed. To this specific aim, the option to choose the precise number of interaction terms (K) to incude allows the user to tailor the LR fitting on the problem at hand, managing the risk of biased estimates and unreliable inference (Sur and Candès, 2019) induced by $p \gg n$ settings. Controlling for the dimensionality of the model surely makes `HOILP` extremely effective in small sample size settings. Nonetheless, the algorithm is fast and scalable enough to be applied to research scenarios where both p and n are large. Indeed, the proposed method provides a useful feature selection technique, that reduces the computational cost by focusing its pattern search on one class only (*targeted* search) without losing generalizability, and that returns lean and interpretable models in manageable running times.

Besides inference robustness, model dimensionality surely impacts interpretability as well. Most real-life application domains are willing to sacrifice a little on the performance side, to foster readability and explainability of results. `HOILP` addresses this aspect by providing an arbitrarily small and extremely descriptive set of discriminative interaction terms, that are easy-to-read and interpret. We provided a clear example via simulation, where the selected patterns described precisely the areas where the two classes were to be sought for. Moreover, as mentioned we introduced two alternatives to the principal algorithm (`Scores HOILP` and `Clusters HOILP`), whose aim is that of serving this need. Indeed, some potential applications of the two variants could be easily identified in the healthcare and lifescience domain, where n is traditionally very small, p may grow extremely large and interpretability is key. For instance, genome-wide association studies may require long lists of interactions to properly characterize a patient and introducing K terms in the model may reduce both interpretability and reliability of results (as even K may be larger than n). Furthermore, the *Risk* and *Protection Scores* of `Scores HOILP` provide a clear representation of the patterns and their role in profiling the population, while being an agile scoring system that can be easily accompanied by other descriptive covariates.

Another point of paramount importance, is the ability of `HOILP` and its variants to manage even extremely imbalanced settings. We already highlighted how oftentimes minority class represents the most interesting class to study, while researchers need to make reliable inference on a very small sample of this critical population. `HOILP`,

combining *targeted* search with dissimilarity-based interaction selection, demonstrated to be a powerful tool to capture the underlying minority class distribution in a robust and generalizable manner, even in presence of very few observations.

Lastly, it is relevant to address the fact that `HOILP` and its variants focus their attention on the search for interactions among categorical covariates. This apparent limitation is actually grounded on a solid and multifaceted rationale. First of all, as previously mentioned, fully categorical data are getting more and more common in several scenarios, such as life sciences. The tractability of these types of data within LR models with interactions is strongly affected by the exponential growth of the number of covariates to be considered as the number of levels grow. This same impact is less dramatic when dealing with continuous features. Nonetheless, `HOILP` was designed to solve this specific issue at best, while being a flexible and effective addition to a broader study framework with various data types as well. Indeed, once selected a restricted and powerful subset of K interaction terms from the categorical subset of variables (absorbing a great portion of computational complexity), the user can easily accompany them with additional numerical covariates - and their interactions. These further terms might indeed be selected - separately or jointly with the K `HOILP`'s terms - via other traditional techniques, that could work on a reduced number of features w.r.t. using the whole original data.

We also compared the performance of the proposed algorithm with that of a state-of-the-art interaction selection method, `glinternet`, demonstrating our method's superior performance in terms of accuracy, interpretability, and significance of the resulting model, as `HOILP` and its variants obtained higher scores with notably less terms in the model.

In conclusion, with its *targeted* approach, the introduction of a novel dissimilarity-based interaction selection, and its flexibility to be tailored around the specific user needs (i.e. number and order of interaction terms, interaction representation via Scores or Compatibility Clusters, confidence adjustment), `HOILP` results to be a powerful approach to face the complexities of applying LR to many real-life research domains.
Future works may be devoted to improving the algorithm's efficiency by including more refined item set mining methods at its core and expanding the methodology to include numerical covariates in a noise-adverse manner.

Appendix

In what follows, we assume to be given a dataset $\mathcal{D} = \{(x_{1,i}, ..., x_{p,i}, y_i)\}_{i=1}^{n}$ of n i.i.d. observations of p categorical covariates $X_1, ..., X_p$, resp. with $m_1, .., m_p$ levels, and a target variable Y. Again, we denote by $\mathcal{I}$ the set of all possible interaction terms (classical and degenerate) for the X_i's.

A. Algorithmic Details

A.1. Support computation

After the mining phase, one is left with a candidate subset $\hat{\mathcal{I}} \subset \mathcal{I}$. In order to compute the corresponding odds-ratios, the datamatrix $\hat{\mathbf{I}}$ has to be constructed; however, as the

number of candidate patterns is tipically very large, evaluating each interaction term by its formula can be unnecessarily expensive. Parallel computing may surely help, but we can also exploit the particular structure of patterns to grant a faster computation.

Without loss of generality, let us rename the dummie variables $\{X_1^{(1)}, ..., X_p^{(m_p)}\}$ as $\{Z_i\}_{i=1}^d$, where $d = \sum_{i=1}^p m_i$, and let $\mathbf{Z} \in \mathbb{R}^{n \times d}$ be the datamatrix of such dummies. For $|\hat{\mathcal{I}}| = L$ and $\hat{\mathcal{I}} = \{T_k\}_{k=1}^L$ we define the incompatibility matrix $\mathbf{M} \in \mathbb{R}^{d \times L}$ as (recall that here we allow dummies to be degenerate interactions)

$$\mathbf{M}_{i,k} := \begin{cases} 1 & Z_i \perp T_k \\ 0 & \text{otherwise} \end{cases} \qquad i = 1, .., d, \quad k = 1, .., L$$

If we consider $\mathbf{Z}, \mathbf{M}$ and $\hat{\mathbf{I}}$ as logical matrices (that is, matrices having entries in the boolean domain $\{0, 1\}$), then, because each observation always attains exactly one level for each variable, it is very easy to see that

$$\hat{\mathbf{I}} = \neg (\mathbf{Z} \cdot \mathbf{M}) \tag{4}$$

where the matrix product is intended in the boolean sense, whereas $\neg$ is the common notation for the logical negation operator (here applied entry-wise). Let us briefly sketch the proof for (4). Consider the jth observation for the kth interaction, $i_{j,k}$. By definition of logic sum and product, the corresponding term on the right-hand side of (4) is

$$\neg \left(\sum_{i=1}^{d} z_{j,i} m_{i,k} \right) = \neg \left(\exists i \in \{1, .., d\} \mid z_{j,i} \wedge m_{i,k} \right) =$$

$$= \neg \left(\exists i \in \{1, .., d\} \mid z_{j,i} \ \wedge \ Z_j \perp T_k \right) = \neg (\neg i_{j,k}) = i_{j,k}$$

in fact, the absence of a pattern is equivalent to the presence of (at least) an incompatible dummie.

A.2. Compatibility clusters computation

In Section 2.3.2, we described the algorithm `Clusters HOILP`, which condensates the information coming from the interaction terms into fewer variables associated to certain compatibility clusters. For more readability, let us recall the notation we used in that context. From the list of candidates $\hat{\mathcal{I}}$, we exploited the dissimilarity measure to extract K patterns (with K even). Half of those were selected from the risk interactions, resulting in the sublist $\hat{\mathcal{R}}_{K/2}$; similarly, the other half consisted of protective patterns, $\hat{\mathcal{P}}_{K/2}$. Next, our algorithm requires the partitioning of these two lists (separately) into smaller groups of patterns so that: each group is formed by pairwise compatible patterns; different groups always have -at least- two incompatible patterns. For the sake of simplicity, we address this problem only for the risk patterns.

To determine a suitable partition of $\hat{\mathcal{R}}_{K/2}$ into compatibility clusters $\hat{\mathcal{R}}_{K/2}^1, ..., \hat{\mathcal{R}}_{K/2}^a$, it can be convenient to consider the interactions as nodes of an undirected compatibility

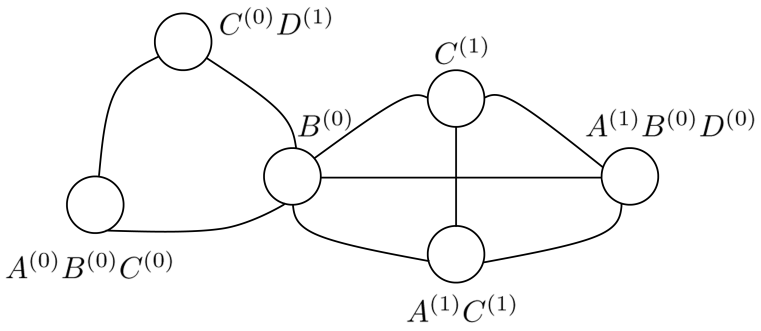

Fig. 6. Example of a compatibility graph. Several interactions terms (two of which are degenerate) involving the dummies of four $\{0,1\}$-valued random variables, A, B, C and D, define the nodes within the graph. Two nodes share a link if and only if the corresponding interactions are compatible. Our greedy decomposition breaks the graph into two clusters: the maximal clique $\{B^{(0)}, C^{(1)}, A^{(1)}C^{(1)}, A^{(1)}B^{(0)}D^{(0)}\}$ and the remaining pair $\{A^{(0)}B^{(0)}C^{(0)}, C^{(0)}D^{(1)}\}$.

graph $\mathcal{G}(\hat{\mathcal{R}}_{K/2})$, where two interactions are linked together if and only if they are compatible. In that way, our original problem becomes equivalent to that of partitioning a graph into cliques, with the additional constraint that such cliques should not be further joinable into a larger ones. As foretold, in general the solution to such puzzle is not unique and many partitions are feasible; in the optic of dimensionality reduction, one may thus want to strengthen the constraints and search for a minimal cover. However, the problem of finding a minimal clique cover for a graph is known to be NP-hard (Karp, 1972), so we partially avert these issue by opting for a greedy procedure. In short: iteratively, we find the maximal clique in $\mathcal{G}(\hat{\mathcal{R}}_{K/2})$, store its nodes in the new list $\hat{\mathcal{R}}^{j}_{K/2}$, and remove the clique from the graph; we keep this up until the graph becomes empty.

The determination of the maximal clique can be done in several ways. In our examples, we actually relayed on the more general Born-Kerbosch algorithm (Bron and Kerbosch, 1973), which in principle allows one to list *all* the maximal cliques within a graph. This algorithm is known for being computationally expensive on large graphs, but that was not our case as we typically picked small values for K. Whenever large values of K are needed, it may be convenient to relay on other maximal clique search algorithms (Tarjan and Trojanowski, 1977; Robson, 1986; Tomita and Kameda, 2007; Konc and Janežič, 2007).

A.3. Further technicalities

Here we address a few last technical details that can be considered when implementing `HOILP` or one of its variants.

First of all, even by exploiting the most sophisticated routines certain datasets may still yield too high computational times in the first training phase. Aside from replacing *Apriori* with other frequent itemsets mining algorithms, another possibility is that of

reducing the interaction search only to those patterns that have a length smaller than a fixed $L > 0$. This further constraint can be easily added as it does not impact on the subsequent steps of the algorithm. Fixing a maximal length L can also be used to intentionally bound the order of the interactions, which may be of interest depending on the context.

Finally, another possible issue is that of finding interactions T that along the dataset have an uncomputable rank, $\log |OR_T|$. This happens everytime the contingency table of (T, Y) has at least an empty cell. To deal with such situations, several approaches can be adopted. For example one may either: 1) further investigate the anomaly and decide whether to discard the interaction or force its presence in the final list of patterns; 2) adopt an adjusted version of the OR, for instance by employing the Haldane-Anscombe correction (Ruxton and Neuha, 2013), which consists in preliminary increasing each value in the contigency table by 0.5.

B. On the dissimilarity measure

In Section 2.2.2, we introduced a dissimilarity measure $d : \mathcal{I} \times \mathcal{I} \to [0, +\infty)$ for the purpose of comparing different patterns.

Here, we wish to point out a few theorical facts about d. First of all, the proposed dissimilarity measure defines a so called *semi-metric* over $\mathcal{I}$ (Wilson, 1931). In fact, as the MCD of two interactions is always a subinteraction of them both, it is straightforward to see that d satisfies:

1) $d(T, S) \geq 0$ for all $T, S \in \mathcal{I}$ (positivity);

2) $d(T, S) = 0 \iff T = S$ (identity of indiscernibles),
in fact $d(T, S) = 0$ if and only if either $|T| = |S| = 0$ or $|\text{MCD}(T, S)| = |T| = |S|$, which is equivalent to $T = S$;

3) $d(T, S) = d(S, T)$ (symmetry)

In general though, d does not define a metric over $\mathcal{I}$. Consider for instance the case of three binary variables A, B and C. Then, for $T := A^{(0)}B^{(0)}C^{(0)}$, $S = A^{(0)}B^{(0)}C^{(1)}$ and $Z = A^{(0)}B^{(0)}$, because $T \perp S$, one has

$$d(T, S) = 3 > 2 = d(T, Z) + d(Z, S)$$

which violates the triangular inequality.

References

Bekti, Cahyo, H., F. M. Rowi, K. Renny, P, and S. Achmad (2017). Network intrusion detection systems analysis using frequent item set mining algorithm fp-max and apriori. *Science Direct, Procedia Computer Science 124* (4th Information Systems International Conference 2017, ISICO 2017, 6-8 November 2017, Bali, Indonesia), 751–758.

Bien, J., J. Taylor, and R. Tibshirani (2013). A lasso for hierarchical interactions. *Annals of statistics 41*(3), 1111.

Boateng, E. Y. and D. A. Abaye (2019). A review of the logistic regression model with emphasis on medical research. *Journal of Data Analysis and Information Processing 7*(4), 190–207.

Borgelt, C. (2012). Frequent item set mining. *WIREs Data Mining Knowl Discov 2*(6), 437–456.

Bron, C. and J. Kerbosch (1973). Algorithm 457: finding all cliques of an undirected graph. *Complexity of Computer Computations 16*(9), 575–577.

Ceddia, G., L. N. Martino, A. Parodi, P. Secchi, S. Campaner, and M. Masseroli (2020). Association rule mining to identify transcription factor interactions in genomic regions. *Bioinformatics 36*(4), 1007–1013.

Chen, C. C., H. Schwender, J. Keith, R. Nunkesser, K. Mengersen, and P. Macrossan (2011). Methods for identifying snp interactions: a review on variations of logic regression, random forest and bayesian logistic regression. *IEEE/ACM transactions on computational biology and bioinformatics 8*(6), 1580–1591.

Dreiseitl, S. and L. Ohno-Machado (2002). Logistic regression and artificial neural network classification models: a methodology review. *Journal of biomedical informatics 35*(5-6), 352–359.

Dua, D. and C. Graff (2017). UCI machine learning repository.

Karp, R. M. (1972). Reducibility among combinatorial problems. *Complexity of Computer Computations*, 85–103.

Konc, J. and D. Janežič (2007). An improved branch and bound algorithm for the maximum clique problem. *MATCH Commun. Math. Comput. Chem. 58*, 569–590.

Lemon, S. C., J. Roy, M. A. Clark, P. D. Friedmann, and W. Rakowski (2003). Classification and regression tree analysis in public health: methodological review and comparison with logistic regression. *Annals of behavioral medicine 26*(3), 172–181.

Lim, M. and T. Hastie (2015). Learning interactions via hierarchical group-lasso regularization. *Journal of Computational and Graphical Statistics 24*(3), 627–654.

Llinares-López, F., L. Papaxanthos, D. Roqueiro, D. Bodenham, and K. Borgwardt (2019). Casmap: detection of statistically significant combinations of snps in association mapping. *Bioinformatics 35*(15), 2680–2682.

Llinares-López, F., M. Sugiyama, L. Papaxanthos, and K. Borgwardt (2015). Fast and memory-efficient significant pattern mining via permutation testing. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 725–734.

Moore, J. H. (2003). The ubiquitous nature of epistasis in determining susceptibility to common human diseases. *Human heredity 56*(1-3), 73–82.

Nicolau, J. (2007). An analysis of the 2002 presidential elections using logistic regression. *Brazilian Political Science Review 1*(1), 125–135.

Niu, L. (2020). A review of the application of logistic regression in educational research: common issues, implications, and suggestions. *Educational Review 72*(1), 41–67.

Onay, V. Ü., L. Briollais, J. A. Knight, E. Shi, Y. Wang, S. Wells, H. Li, I. Rajendram, I. L. Andrulis, and H. Ozcelik (2006). Snp-snp interactions in breast cancer susceptibility. *BMC cancer 6*(1), 114.

Papaxanthos, L., F. Llinares-López, D. Bodenham, and K. Borgwardt (2016). Finding significant combinations of features in the presence of categorical covariates. In *Advances in neural information processing systems*, pp. 2279–2287.

Pellegrina, L. and F. Vandin (2018). Efficient mining of the most significant patterns with permutation testing. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 2070–2079.

Radchenko, P. and G. M. James (2010). Variable selection using adaptive nonlinear interaction structures in high dimensions. *Journal of the American Statistical Association 105*(492), 1541–1553.

Robson, J. (1986). Algorithms for maximum independent sets. *Journal of Algorithms 7*(3), 425–440.

Rosasco, L., M. Santoro, S. Mosci, A. Verri, and S. Villa (2010). A regularization approach to nonlinear variable selection. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pp. 653–660.

Ruxton, G. and M. Neuha (2013). Review of alternative approaches to calculation of a confidence interval for the odds ratio of a 2×2 contingency table. *Methods in Ecology and Evolution 4*, 9–13.

Shah, R. D. and N. Meinshausen (2014). Random intersection trees. *Journal of Machine Learning Research 15*, 629–654.

Sloane, P. J. and I. Theodossiou (1994). The economics of low pay in britain: A logistic regression approach. *International Journal of Manpower 15*(2-3), 130–149.

Sugiyama, M. and K. M. Borgwardt (2019). Finding statistically significant interactions between continuous features. In *IJCAI*, pp. 3490–3498.

Sur, P. and E. J. Candès (2019). A modern maximum-likelihood theory for high-dimensional logistic regression. *Proceedings of the National Academy of Sciences 116*(29), 14516–14525.

Tarjan, R. and A. Trojanowski (1977). Finding a maximum independent set. *SIAM J. Comput. 6*(3), 537–546.

Tomita, E. and T. Kameda (2007). An efficient branch-and-bound algorithm for finding a maximum clique with computational experiments. *J Glob Optim 37*, 95–111.

Van Rossum, G. and F. L. Drake (2009). *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace.

Wilson, W. (1931). On semi-metric spaces. *American Journal of Mathematics 53*(2), 361–373.

Zaghdoudi, T. (2013). Bank failure prediction with logistic regression. *International Journal of Economics and Financial Issues 3*(2), 537.

MOX Technical Reports, last issues

Dipartimento di Matematica
Politecnico di Milano, Via Bonardi 9 - 20133 Milano (Italy)

55/2020 Botti, M.; Castanon Quiroz, D.; Di Pietro, D.A.; Harnist, A.
A Hybrid High-Order method for creeping flows of non-Newtonian fluids

56/2020 Botti, L.; Botti, M.; Di Pietro, D. A.;
A Hybrid High-Order method for multiple-network poroelasticity

57/2020 Regazzoni, F.; Quarteroni, A.
An oscillation-free fully partitioned scheme for the numerical modeling of cardiac active mechanics

58/2020 Beraha, M.; Pegoraro, M.; Peli, R.; Guglielmi, A
Spatially dependent mixture models via the Logistic Multivariate CAR prior

54/2020 Arnone, E.; Bernardi, M. S.; Sangalli, L. M.; Secchi, P.
Analysis of Telecom Italia mobile phone data by space-time regression with differential regularization

53/2020 Arnone, E.; Kneip, A.; Nobile, F.; Sangalli, L. M.
Some numerical test on the convergence rates of regression with differential regularization

52/2020 Arnone, E.; Kneip, A.; Nobile, F.; Sangalli, L. M.
Some first results on the consistency of spatial regression with partial differential equation regularization

51/2020 Ferraccioli, F.; Sangalli, L. M.; Arnone, E.,; Finos, L.
A functional data analysis approach to the estimation of densities over complex regions

50/2020 Bonaventura,L.; Gomez Marmol, M.
The TR-BDF2 method for second order problems in structural mechanics

49/2020 Bonaventura,L.; Garres Diaz,J.
Flexible and efficient discretizations of multilayer models with variable density