

Symbolic AI (LFIT) for XAI to handle biases*

Extending LFIT to numerical domains for explaining Biases in ML

Javier Tello^{1,†}, Marina de la Cruz^{2,†}, Tony Ribeiro^{3,4,†}, Julian Fierrez^{1,†},
Aythami Morales^{1,†}, Ruben Tolosana^{1,†}, César Luis Alonso^{5,†} and Alfonso Ortega^{2*,†}

¹Escuela Politécnica Superior. Universidad Autónoma de Madrid

²Escuela Superior de Ingeniería y Tecnología ESIT, Universidad Internacional de la Rioja UNIR

³Laboratoire des Sciences du Numérique de Nantes; 44300, Nantes, France

⁴National Institute of Informatics, Tokyo 101-8430, Japan

⁵Departamento de Informática. Universidad de Oviedo

Abstract

LFIT is a well known declarative machine learning framework able to generate propositional logic twins of complex systems. It needs discrete input data. It has been successfully applied in further works to explain biases in different domains. This work aims to extend and improve LFIT capabilities on numeric domains.

Keywords

LFIT, symbolic machine learning, XAI, biases

1. Motivation

Machine learning algorithms, specially deep learning approaches, have been successfully applied to an increasing number of different areas and applications and have become a *de facto* standard in many domains. In this context, the term *machine learning* is a rather metonymical use to stand for *numerical or statistical* machine learning models. These algorithms feed on huge amount of data to learn how to get answers similar to those of the data seen, when facing unseen new data. They mimic, in this way, the process implicit in the data and, the only reason that they usually provide to explain their decisions is that *they choose the ones with higher probabilities* among all the options. But, to mimic the process implicit in the available data implies to reproduce it, including unfair biases, regarding gender or ethnicity, for example. There exist domains in which, due to ethical or legal constraints, this skewed behaviour is not acceptable as, for instance, [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14] reported.

Not only the scientific community is concerned by this issue, governmental initiatives have been also taken such as [15].

Aequitas 2023: Workshop on Fairness and Bias in AI | co-located with ECAI 2023, Kraków, Poland


*Corresponding author.

†These authors contributed equally.

✉ javier.tello@estudiante.uam.es (J. Tello); marina.delacruz@unir.net (M. d. l. Cruz); tony.ribeiro@ls2n.fr (T. Ribeiro); julian.fierrez@uam.es (J. Fierrez); aythami.morales@uam.es (A. Morales); ruben.tolosana@uam.es (R. Tolosana); calonso@uniovi.es (C. L. Alonso); alfonso.ortegadelapuerta@unir.net (A. Ortega)



© 2023 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

This scenario describes one of the main motivations for the emerging area that is being called *explainable artificial intelligence* or *XAI* [16]. In the last years, several conditions have been described to consider that a machine learning model is able to explain its decisions. Traditional *opaque* algorithms have been extended, enriched or sometimes wrapped in techniques to increase their explainability. [17] is one of the more complete surveys on this topic.

By the other hand, approaches based on discrete and declarative learning models coined, back in the late eighties [18], the terms *strong* and *ultra strong* to stand for essentially explainable (also called interpretable or transparent) machine learning models. Logic programming is one of the most powerful declarative paradigms that better fits for the *ultra strong* category.

Learning from interpretation transitions (LFIT [19]) is a very powerful declarative machine learning approach able to induce a propositional logic theory equivalent to the data which it feeds on. Its original goal was to produce a declarative (propositional logic) twin to some biological complex systems (metabolic pathways). The help of experts in Biology was needed to properly discretize the datasets describing the low-level chemical behaviour of the pathways. This contribution continues previous works of the authors ([20, 21]) as we describe in the proper section.

Both, pure statistical or pure declarative approaches are not the only option. Mixed approaches try to take advantage of the best characteristics of both of them. From our perspective and interests the most relevant are those that produce the representation of the knowledge learned by the algorithm with highest level of abstraction. One of these approaches is PSyKE [22]. PSyKE takes a step further than other previous approaches to extract knowledge from data sets by unifying them and translating them into first order (Prolog) theories. Its theoretical basis and guarantees rely on those of the underlying models, and most of them belongs to the statistical realm. Although we are mainly interested in declarative approaches PSyKE is focused in the unification of the extraction procedure to be expressed as first order logical theories what shares our general purpose approach.

One of the areas of interest for this work is the automatic assessment of personal profiles such as, for example, in automatic recruiting and hiring processes. Several recent works were focused on XAI for this domain [23, 24, 25, 26, 27].

Statistical machine learning models do their best on numerical data and usually need to transform categorical attributes into numeric, while declarative ones do the opposite. The benefit, or rather need, to discretize data before applying declarative models has been already reported, long time ago, such as in [28].

Our goal in this work is to improve the performance of LFIT for managing biases in automatic assessment of personal profiles and, by the other hand, to extend LFIT with some general purpose discretizers to be used in case of facing numerical information as it is the case in these scenarios.

The rest of the paper is organized as follows:

Section 2 describes the context of this research, paying special attention to how LFIT can be used to explain biases; available methods to design a general purpose discretizer for LFIT, both, unsupervised (biometric hashing) and supervised (CAIM); and a brief and intuitive description of LFIT. This section includes all the needed references.

Section 3 summarizes the main contributions of this paper that is focused on testing the viability of adding a general purpose discretizer to LFIT and improving its ability to handle

biases in demographic and biometric datasets, by means of this new module and by extending LFIT to overcome issues found in previous works.

Section 4 describes the steps taken, decisions made and experiments run to test the viability of adding a general purpose discretizer with CAIM and biometric hashing. In each case it has been checked that LFIT ability to explain biases is not worsen. Specific experiments have been designed to fully control and analyse each step to incorporate a complete version of biometric hashing to LFIT.

Section 5 is devoted to explain how LFIT itself has been extended to improve the accuracy of its explanations in general, but specially in demographic and biometric domains.

And finally section 6 summarizes the main conclusions of this research and describes our future plans and open lines.

2. State of the art

2.1. Explaining biases by symbolic XAI: LFIT

The authors of this contribution had some successful previous experience in explaining biases by means of LFIT [29, 20]. In this preliminary works, LFIT was applied for the first time for this purpose and some limitations and advisable further research lines were identified.

2.2. Discretization of numeric information in machine learning scenarios

The difficulty of handling numeric (continuous) information in some machine learning scenarios has been reported long time ago [30]. It is always possible to follow a naive approach and split numeric attributes with some *ad-hoc* technique. There exist several contributions that describe different approaches to discretization such as [31, 32, 33, 34, 35, 36, 37, 38, 39, 40] and some review on this topic, [41]. In these papers more systematic and automatic approaches are introduced, considering, for example, from rather simple equal width / frequency intervals to more sophisticated unsupervised and supervised approaches; some of them consider discretization as a special case of clustering; some others take into account statistical information drawn from the values of the numeric attributes as the aforementioned Chimerge [38] and its extension Chi^2 [39] that proposes the χ^2 test to relate the values of the attributes; other approaches are based on the Shannon's entropy concept, such as [40], that has been used for discretizing the inputs to TILDE [28], an ILP system used back in the late 90's. Entropy is a concept widely used in these domains by systems as ID3 [42] or FUSINTER [43].

To get a really general purpose discretizer for LFIT we have to include options both for supervised and unsupervised scenarios.

The next two sections explain our choices.

2.2.1. Unsupervised discretizers: biometric hashing

The most interesting unsupervised discretizer for this work is *biometric hashing* that was introduced, proposed and tested by some of the authors of this contribution in [44]. Although its goal is different from that of this paper, it has characteristics very important for us.

Biometric hash is based on *k-means* [45] with which shares its best-known limitation: the number k of clusters has to be provided.

Biometric hashing proceeds as follows: it designs, the best possible grouping of the attributes, with respect to its relevance, that have to be discretized among all their possible combinations, including also groupings that exclude some irrelevant attribute and the possibility for the groups to overlap by sharing some common subset of attributes. This optimization is delegated to an adequate algorithm. In [44] for example, a genetic engine is considered. *k-means* is used in each group and the group (its centroid) is labelled with Gray code to ensure that close groups differ in few bits. Finally, each value of the discretized attributes is labelled with the concatenation of the labels of its centroids of each groups. Biometric hashing generates, in this way, discrete values that keep the semantics included in the original attributes: the codes of close centroids differ in few bits. Keeping the semantics is the most important characteristic for us and it is why we have chosen biometric hashing as the unsupervised algorithm for our general purpose discretizer.

2.2.2. Supervised discretizers: CAIM

CAIM (Class-Attribute Interdependence Maximization) [31] locally maximizes the inter dependency between classes and attributes, and hence, it reduces the number of intervals needed to discretize. This is done by means of the definition of the CAIM criterion that normalizes this dependency between classes and attributes: the higher the criterion, the higher interdependency between class and attribute.

CAIM iterates, splitting the domains in each iteration, following the optimization of this criterion until no further improvement is accomplished.

CAIM is one of the discretizer better ranked in surveys such as [46] and, although it has been recently extended [37] and other models are also interesting, it still can be considered as a classic *de facto* standard supervised discretizer. We have chosen CAIM for our general purpose discretizer as a good option without any other experimental tests that could support and guarantee a better performance in any domain. Future experiments and developments will cope with this question. The goal of the current paper is to test the viability of a general purpose approach.

2.3. LFIT

Learning from interpretation transition (**LFIT**) [47] has been proposed to induce propositional logic twins of complex dynamic systems from the observation of its state transitions. From data captured from the domains of the systems, some discretization on them is needed to finally feed LFIT.

The LFIT framework actually includes a family of algorithms with their implementations that come from several extensions, generalizations and performance improvements: for memory-less deterministic systems [47], for systems with memory [48], for probabilistic systems [49], and for their multi-valued extensions [50, 51], for continuous data [52], and for learning system dynamics no matter their update semantics [53, 54]. LFIT can be used, then, to learn an equivalent propositional logic program that provides explanations for each given observation,

and thus, our bet has been to incorporate LFIT as an alternative to explain the decisions of opaque machine learning processes.

Another suggestive feature of LFIT is its ability to get in some way minimal propositional theories (programs) equivalent to the data seen. The rules learned by LFIT are prime implicant, all their condition a necessary to explain the outcome, making them the best candidates for actual causality explanation. In XAI scenarios, the relevance of explanations is an essential feature. Logical equivalence ensures relevance. Minimality ensures efficiency. Being the programs learnt by LFIT equivalent and minimal to the data, these programs' size and structure could be considered a metric of the complexity of the semantics implicit in the data.

From the family of LFIT algorithms, GULA [53, 54] and PRIDE [55] are the ones that we have chosen. In particular PRIDE that improves GULA's performance while keeping its theoretical properties we are interested in.

In the examples used to introduce PRIDE, the description of the census [56] dataset included in table 1 will be used. LFIT shares its logical notation with Prolog.

Roughly speaking we can consider that LFIT initially creates a clause to represent each single row of the dataset. The values of its input attributes are translated into the clause's body and the target generated into its head.

Listing 1 shows an example of LFIT rules for the census dataset.

Listing 1: Prolog version of rules learnt by LFIT in the case study related to Table 1

```
class(0) : -age(3), education(6), maritalStatus(0), occupation(0).
class(0) : -age(4), workclass(0), education(1), occupation(8), relationship(0), nativeCountry(0).

class(1) : -education(7), maritalStatus(5).
class(1) : -age(2), education(8), occupation(10).
class(1) : -age(1), education(3), maritalStatus(2), occupation(9).
```

The key concepts of the learning engine of LFIT are *satate matching* and *rule dominance*.

Informally, LFIT uses these two concepts to remove from the theory those rules that are more specific than others that includes them.

After iterating on the set of examples and ensuring that all of them are covered, LIFT gets the set of minimal rules equivalent to the set of examples. It is minimal because it includes the more general rules and also because all the conditions of their bodies are needed, that is, if any of them is removed, some examples end up not covered by the theory.

Formal and detailed explanations of LFIT and all these concepts can be found in the aforementioned literature. We will explain them by means of examples:

In Listing 2 you can see how clause R_1 dominates R_2 because they have the same head ($class(0)$) and R_1 's body is contained in that of R_2 .

Listing 2: Example of rule domination

```
 $R_1$  : class(0) : -education(6), maritalStatus(0).
```

```
 $R_2$  : class(0) : -age(3), education(6), maritalStatus(0), occupation(0).
```

We have informally used so far the expression *a clause covers an example*. From a formal viewpoint examples are considered *states*, clauses and rules are synonymous and *to cover* is formally expressed as *n state and a rule match*.

Listing 3 shows an example of rule-state matching: state s_1 and rule R_1 does because s_1 is contained in R_1 's body.

Listing 3: Example of rule-state matching

s_1 : *age(3), education(6), maritalStatus(0), occupation(0)*

R_1 : *class(0): - education(6), maritalStatus(0)*.

From the declarative viewpoint of LFIT, the focus is on the qualitative guarantee of learning a logical version equivalent to the observed system. Regarding equivalence, the version is equivalent or it is not. If the model fails in 1% of the examples, equivalence is lost in the same way than if it had failed in 20% or 60% of the examples.

From the viewpoint of the statistical approaches it is very important to take into account the *amounts*. For example, the output of deep-learning classifiers is based on a quantitative criterion such as to choose the label with the highest probability.

It could seem that the qualitative behaviour of LFIT does not matter; but this is not exactly true.

LFIT can easily collect qualitative information, such as how many states (input examples) match each rule. This numerical information can be used as weights, both to better explain and understand the process, but also to incorporate predicting capabilities to the declarative version. This option has been explained and explored in [57, 58].

3. Contributions of this paper

Main contributions of this papers can be summarised in the following two points: to add to LFIT a general purpose discretizer that improves its ability to explain hidden semantic relationships on numeric domains; and extend LFIT itself to improve its expressiveness to explain demographic and biometric datasets, for example, to detect biases. Being more specific we can reformulate this two main goals in supporting or rejecting the following intuitive opinions or hypothesis:

- It is not clear that CAIM finds the best number of clusters (optimum) for any clustering algorithm.
- It seems that supervised discretizers outperform unsupervised ones, but unsupervised ones worth depending on the availability of the right number of clusters.
- It seems that the discretization used matters for LFIT
- It seems that biometric hashing used as a general purpose discretizer does not worsen the expressive power of other ad-hoc discretizers.

Table 1

Names, values and codification of the dataset about incomes. Attributes of type C take integer or real continuous values and they are uniformly discretised. Attributes of type D are originally discrete and are numerically coded from 0 to the maximum needed value.

Attribute	Meaning	Type	Codification
Age	Age of the individual (years)	C	{0, 1, ..., 7}
Workclass	Work type (self employment, private, ...)	D	{0, 1, ..., 6}
Fnlwgt	Demographic weight (row) from census	D	{0, 1, ..., 14}
Education	Highest academic degree	D	{0, 1, ..., 15}
Marital status	Civil status	D	{0, 1, ..., 3}
Occupation	Individual's job sector	D	{0, 1, ..., 13}
Relationship	Present individual's relationship	D	{0, 1, ..., 5}
Ethnicity	Ethnic group	D	{0, 1, ..., 4}
Sex		D	{0, 1}
Capital gain	Increase in individual's capital asset	C	{0, 1, ..., 9}
Capital loss	Decrease in individual's capital asset	C	{0, 1, ..., 4}
Hours per week	Spent on work (average)	C	{0, 1, ..., 9}
Native country	Country of origin	D	{0, 1, ..., 40}
Income level	Individual's class of income ($\leq 50, > 50$)	D	{0, 1}

- It seems that extending LFIT by incorporating from the dataset under consideration, some measure of the weights that actually each rule has, ends up in quantitatively more accurate programs.

In this paper we have used tree datasets:

The UCI wine dataset [59]: a well-known labeled numeric dataset about some Italian wines' characteristics. It contains thirteen continuous attributes: alcohol, malic acid, ash, alcalinity of ash, magnesium, total phenols, flavanoids, nonflavanoid phenols, proanthocyanins, color intensity, hue, OD280/OD315 of diluted wines, and proline; in addition to a categorical target about wine's quality that considers three different classes of wine.

The US 1994 census [56] that is described in table 1.

The FariCV db dataset detailed described in [29, 20].

4. General purpose discretizer for LFIT

Roughly speaking, from the machine learning viewpoint, any domain should be able to be faced either from supervised or from unsupervised approaches (or even from both). We have, hence, integrated with LFIT a general purpose discretizer that includes both options. The researcher can select the one that fits its domain.

From the unsupervised viewpoint we take in this work the first steps to check the viability of incorporating biometric hashing. From the supervised one we have chosen CAIM that could be considered as a *de facto* standard.

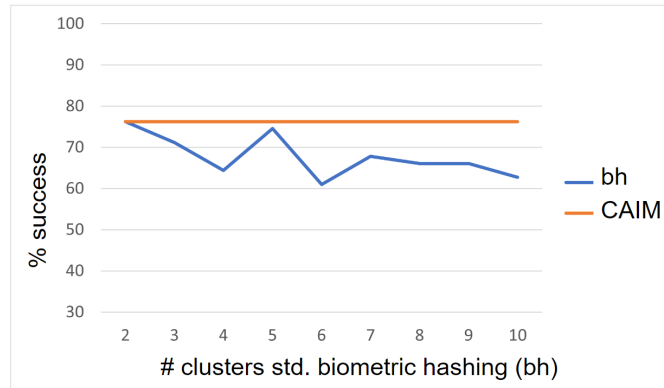


Figure 1: Precision of CAIM vs reduced biometric hashing on *wine* dataset (test subset)

4.1. Supervised vs unsupervised discretization for LFIT

4.1.1. Rough comparison of raw precision of LFIT predictions

In order to compare CAIM and biometric hashing, labeled datasets are needed. We have chosen the *wine* dataset [59]. We have, in addition, reduced biometric hashing to the standard grouping (as many groups as attributes and with a single attribute in each group) in order to highlight the differences with CAIM when separately discretizing each attribute. CAIM automatically suggests the most adequate number of clusters for each attribute. In this experiment, CAIM found that three clusters was the best number no matter which of the thirteen attributes. To compare standard biometric hashing with CAIM all the combinations of different number of clusters for each of the thirteen attributes should be checked. In this first proof of concept we decided to take a preliminary look at the behaviour of the discretizers analyzing by hand as many aspects as possible to be able to explain step by step what we find. We decided to *mimic* in some way CAIM's results using the same number of clusters for all the attributes and to compare CAIM vs the reduced biometric hashing considering 2 to 10 clusters. The performance of LFIT on the discretized dataset generated by each method was computed as the percentage of hits when predicting the class on the test portion of the dataset. Figure 1 compares both methods.

This preliminary test supports the intuitive initial assertion: ***supervised clustering should outperform unsupervised techniques***, it is shown that, in some way, CAIM's performance sets an upper bound that of basic biometric hashing. By the other hand. CAIM's results are 45 hits, of a total of 59 (76.27% of success) The other detailed values are provided in table 2. It can also be seen that other of our hypothesis is also supported: ***unsupervised methods worth when you find the right number of clusters***. In this case, both 5 and mainly 2 are comparable to CAIM. In the specific case of 2, both methods get the same performance.

A last consideration could be drawn from this experiment. Standard biometric hashing does better for 2 and 5 clusters than for 3. CAIM algorithm automatically chooses 3. So, some informal evidence is provided against the aforementioned intuition: ***CAIM finds a global optimum number of clusters useful for any other clustering algorithm***.

Table 2
Std. biometric hashing

std. biometric hashing			
# de clusters	Aciertos	Total	% de aciertos
2	45	59	76,27
3	42	59	71,19
4	38	59	64,41
5	44	59	74,58
6	36	59	61,02
7	40	59	67,80
8	39	59	66,10
9	39	59	66,10
10	37	59	62,71

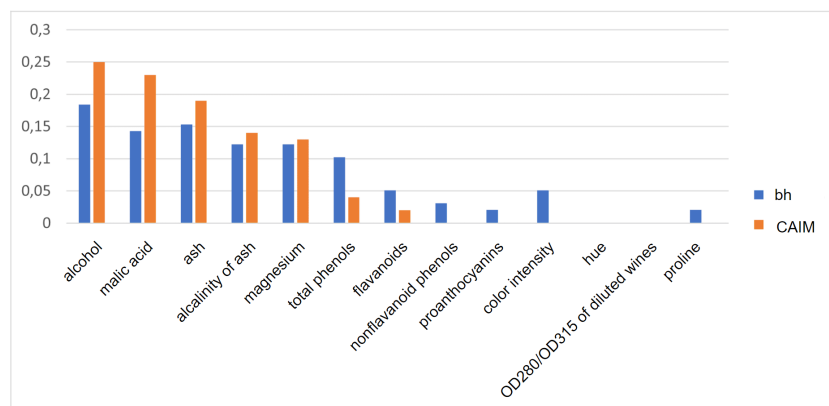


Figure 2: NPs of each attribute, CAIM+LFIT vs basic biometric hashing+LFIT

4.1.2. Assessment of the explainability of discretizers

To compare the information retained by each discretizer we repeat the kind of analysis of [20] based on *NP* with *wine* dataset. Figure 2 summarizes the results of this experiment. Remember that all the NPs for the same method add up 1 and, hence, the total of blue columns is 1 (same for the orange ones).

Figure 2 shows that both discretizers catch the most relevant set of attributes (alcohol, malic acid, ash, alcalinity of ash and magnesium). But some relevant differences are also clear:

The relevance of some attributes such as total phenols and flavanoids is the opposite depending on the discretizer.

Some other attributes disappear for CAIM (nonflavanoid phenols, proanthocyanins, color intensity or proline).

So, this first proof of concept supports the aforementioned hypothesis: ***the discretizer used matters***

4.2. Discretization and biases detection by LFIT

The main goal of our work is to improve the way in which LFIT handles biases. Discretization is important to tackle datasets with numeric attributes. But we have to ensure we do not worsen the already proven ability of LFIT to detect and explain biases.

To get it, we have reproduced the experiments run in [20] on the FairCV and US 1994 census databases.

4.2.1. Biases on FairCV db with general purpose discretizers

[20] describes how LFIT can be used to identify gender and ethnic biases by performing simple statistics comparing how many times these attributes increase and with which value between biased and unbiased datasets.

In this case we have replaced the equal width approach by the *basic biometric hashing*.

[20] contains a detailed description of the FairCV dataset. Among its characteristics, the most relevant for the current work is that it contains one numeric attribute (work experience) and one numeric target (the score given to the CV). Three different targets (scores) are given: an unbiased one, and two biased scores, respectively by gender and by ethnicity.

This characteristic (numeric targets) is incompatible with CAIM that requires discrete targets. So, the only option included in our systems for this kind of data is biometric hashing.

In this case, with just a single numeric attribute, biometric hashing reduces to *k-means* with Gray codes for the centroids.

The next decision in this case is the number of clusters. We have decided to use the same number than in [20] (6 for the input attribute and 4 for the targets). For checking the relevance of all the input attributes in the dataset, in [20] different scenarios were considered to include an input attribute at a time (from s_1 to s_{11} , this last one considers all the input attributes).

The parameters *AIP* (absolute increment percentage) were used. They, in fact, measure the increment of the absolute frequency of an attribute (how many times it appears in the conditions of the clauses learnt by LFIT) between two LFIT programs (one unbiased and another biased) with respect to the second one (the biased, in this case). The intuitive interpretation of these parameters is that the higher the AIP the attribute is a more important cause of bias.

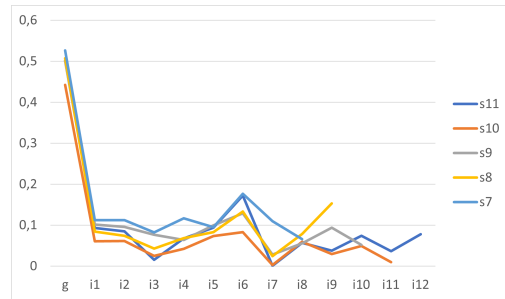


Figure 3: $AIP_{ns,s}^{s_7} - AIP_{ns,s}^{s_{11}}$ wrt biased-gender

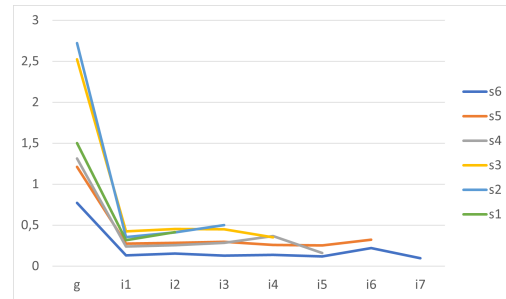


Figure 4: $AIP_{ns,s}^{s_1} - AIP_{ns,s}^{s_6}$ wrt biased-gender

Figures 4, 3, 6, 5 graphically show the *AIP* of each attribute. They are very similar to those in [20] and the interpretation is the same: the gender bias seems to be caused by the attribute

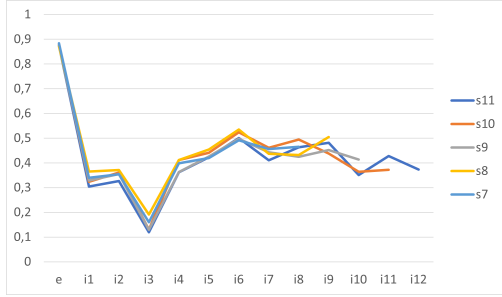


Figure 5: $AIP_{ns,s}^{s_7} - AIP_{ns,s}^{s_{11}}$ wrt biased ethnicity

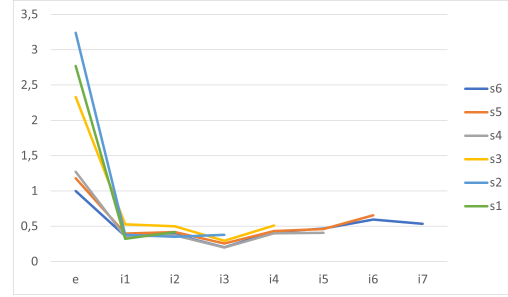


Figure 6: $AIP_{ns,s}^{s_1} - AIP_{ns,s}^{s_6}$ wrt biased ethnicity

gender and the ethnicity bias seems to be caused by *ethnic group* because they are those that increment their presence the most as a condition in the clauses of the LFIT program comparing the unbiased version with the biased one.

In [20] the relevance of each value of the attributes were also studied by means of the parameter *partial weight (PW)* for every attribute, that accumulates how many times each specific value of the attribute appears as argument in the conditions of clauses for every value of the target in the head of the clauses. Figures 11 and 12 show the *PWs* for *gender* and figures 13 and 14 show that of *ethnic group*. In figure 12, for example, *PW* for *gender* male and unbiased score 4 counts how many clauses in the program learnt by LFIT from unbiased scores include as condition *gender(male)* and as head *score(4)*.

These figures express the same information explained in [20]: the higher values of each color shows the specific score value for which the considered attribute value is more relevant. In the case of data biased by gender: female for lower scores (2 and 1), and male for higher. In the case of ethnic bias: lower scores (2 and 1) for ethnic groups different from Caucasian, and higher for Caucasian. You can also see in these figures how the higher scores increase more when biasing data for male and Caucasian ethnicity while lower scores does the same for male and other ethnic groups.

4.2.2. Biases on US 1994 census with general purpose discretizers

A similar procedure has been followed on census. In this case (the target is discrete) both approaches (CAIM and biometric hashing) are applicable.

[20] describes how LFIT can be used to identify gender and ethnic biases by performing simple statistics counting how many times these attributes occurs for high and low earnings.

We have decided to compare CAIM with standard basic biometric hashing on single attributes; and to keep the same number of clusters used in [20], that is, 8 for *age*, 10 for *capital gain* and *hours per week*; and 4 for *capital loss*. In this case *NP* parameter is used no normalize *PWs*. This dataset has no unbiased version so we keep the hypothesis of [20]: high or low income usually are biased by gender and ethnic group.

Figures 7, 8, 9, and 10 show the same behaviour seen in [20]: higher incomes are more probable for Caucasian males.

A closer comparison between CAIM and basic biometric hashing suggests that CAIM's

discretization could allow LFIT to amplify in some way the relevant information under consideration: in this case the role of gender and ethnic group to skew the incomes in US. This slightly supports one of our hypothesis: ***supervised discretizers should outperform unsupervised ones.***

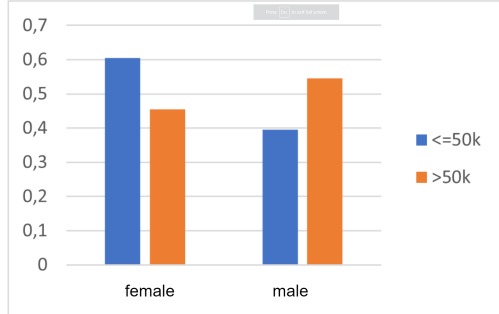


Figure 7: gender NP, basic biom. hashing in census

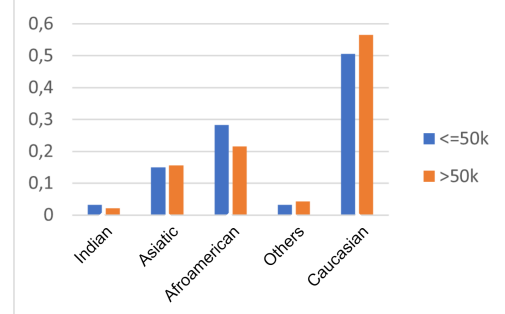


Figure 8: ethnic group NP, basic biom. in census

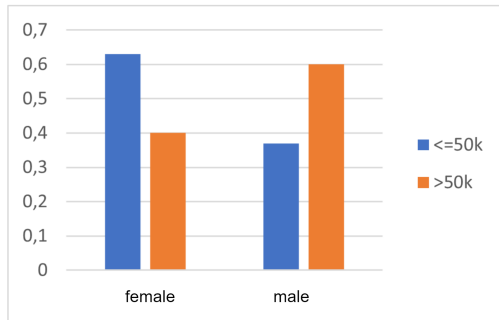


Figure 9: gender NP, CAIM in census

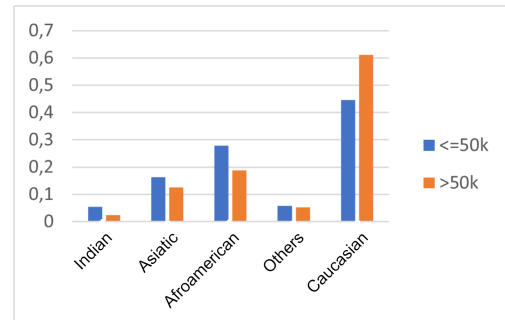


Figure 10: ethnic group NP, CAIM in census

Although more experiments have to be done, We can conclude that these results support one of our hypothesis: ***biometric hashing used as a general purpose discretizer does not worsen the expressive power of other ad-hoc discretizers.***

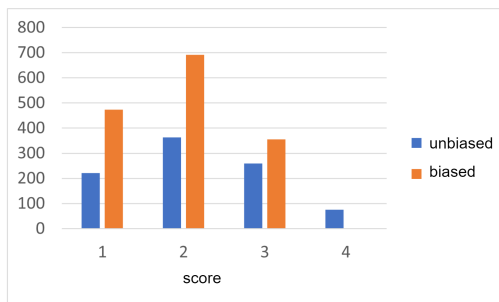


Figure 11: PW for female gender in FairCV db

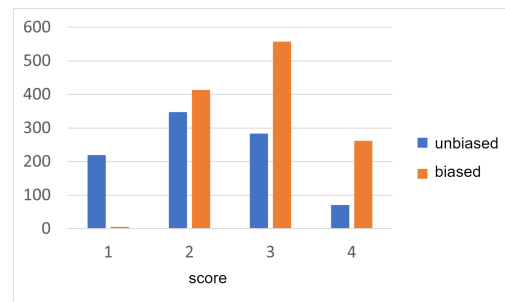


Figure 12: PW for male gender in FairCV db

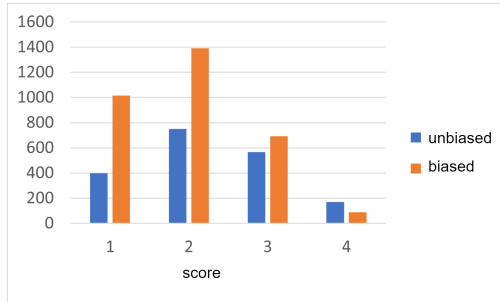


Figure 13: PW for other ethnic groups in FairCV db

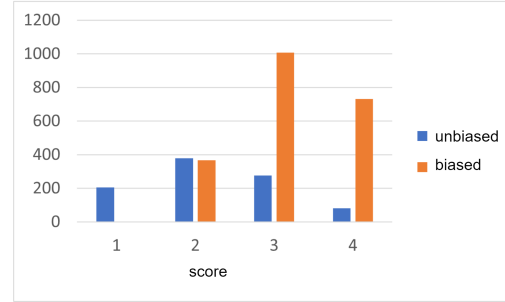


Figure 14: PW for Caucasian ethnic group in FairCV db

4.3. Biometric hash inspiration in the general purpose discretizer

Biometric hash's advantages have been previously described. Although the domain in which this recent technique has been proposed is different from ours, a general purpose discretizer should include features such as to be unsupervised, to automatically search for the best grouping of numerical attributes and to label the clusters' centroids in such a way that similar labels stand for close clusters. Regarding the first point, biometric hash is, in fact, based on *k-means*. Regarding the second one, it is able to, even both: discarding irrelevant attributes, but also considering overlapping groups to cope with complex relationships between them. Regarding the third one, biometric hash concatenates the gray codes of the projection of each centroid's on each group to get a label that ensures that close centroids differ in few bits.

To check its viability as a general discretizer, we propose a method inspired by biometric hashing adapting its features in this way:

In the basic case, a single numeric attribute is handled by biometric hash by applying *k-means* and labeling each centroid with gray codes. This is the standard procedure already considered in this work.

Biometric hashing includes a module to find the better grouping of characteristics. It is, in fact, an optimization problem usually solved by means of search techniques. We have decided to explicitly analyze all the possible groupings by hand in a proper dataset. In this first test, we have adopted two assumptions: all the attributes in the dataset are relevant and independent from each other. This implies some constraints to the grouping: all the attributes should belong to any group of the grouping (relevance) and only to one (independence).

After designing the best grouping of attributes, the biometric hashing concatenates the codes of all the centroids of each group in that grouping to generate a single label to each combination of the grouped attributes. Our general purpose approach has to potentially cope with datasets with lots of numerical attributes (as in the case, for example, of any scenario that includes images) what eventually could, if the best grouping includes too many groups of attributes, ends up with an attribute with lots of different values represented as large binary numbers, and hence, from a practical viewpoint, this attribute will be indistinguishable from any other numeric information. In this first proof of concept we have decided to keep each group in the grouping separate and not to concatenate their labels.

To check the viability of biometric hashing as a general purpose unsupervised discretizer

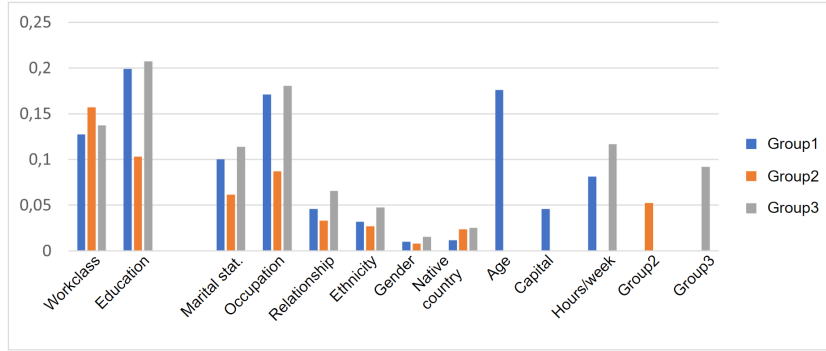


Figure 15: Comparison of NP for each attribute vs each grouping (g_1 , g_2 and g_3)

for LFIT, We have chosen the 1994 US census dataset [56] because it contains four numeric attributes (age, capital gain, capital loss and hours per week) that can easily be reduced to three by subtracting capital loss from gain to compute a single attribute (capital). If these three numeric attributes are respectively named n_1 , n_2 and n_3 , our two assumptions (relevance and independence) reduce, in fact, the possible combinations to the groups $g_1 = \{\{n_1\}, \{n_2\}, \{n_3\}\}$ (this is the reference grouping to compare with), $g_2 = \{\{n_1, n_2, n_3\}\}$, $g_3 = \{\{n_1, n_2\}, \{n_3\}\}$, $g_4 = \{\{n_1\}, \{n_2, n_3\}\}$ and $g_5 = \{\{n_1, n_3\}, \{n_2\}\}$.

Once we have seen that *k-means* with the standard grouping catches the same biases than *equal-width intervals* used in [20] we have looked by hand for the best grouping among those previously described. Figure 15 and 16 show the NP for each attribute compared with each group. The best grouping should be at least as good as g_1 . All the groupings seem to be useless because they apparently put together information whose combination is incoherent with the semantics of the dataset. In figure 16, nevertheless, you can see that g_5 is a quite reasonable grouping. g_5 's behaviour is similar to that of g_1 for all the attributes possible. In addition g_5 is the only one that seems to accumulate the information contained in the attributes it includes (age, and hours per week). The arrows in the figure show that NP for g_5 accumulates the information of both attributes (the height of its column seems to accumulate the sizes of the others). So, we can conclude that g_5 is able to put together its attributes without losing information.

5. Improving biases detection by LFIT

The other main goal of this paper is to improve the ability of LFIT to handle biases extending LFIT to specifically face datasets with the typical structure that of demographic and biometric scenarios.

Our proposal takes advantage of a characteristic of the LFIT's learning engine: each clause added to the learnt program, reduces the positive examples set in those that it can explain. So, it is possible to know how many examples are covered by each clause.

In addition, in biometric and demographic scenarios it is usual (and sometimes mandatory) the inclusion of some *final weight* information that represents how many real cases is covered

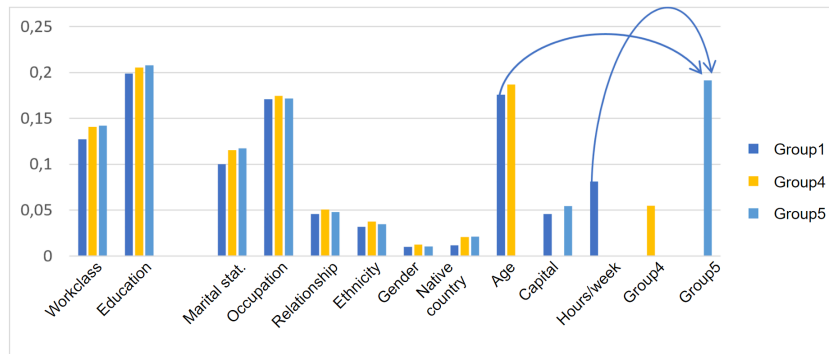


Figure 16: Comparison of NP for each attribute vs each grouping (g_1 , g_4 and g_5)

by each single row in the real dataset.

LIFT has been extended to accumulate the *final weight* of each row when the number of actual examples covered by each rule is computed.

Once LFIT provides the researchers with this value, all the parameters and graphics shown can be (re)calibrated to cope with the final actual weight that each clause has.

In this first proof of concept the experiments in section 4.2.2 have been repeated with the extended LFIT version.

Figures 17, 18, 19, and 20 shows the new version of the same figures of section 4.2.2 corresponding to the extended LFIT version re calibrated to take into account the actual weight of each rule.

It is clear that the qualitative information about the origin of the biases is the same but the quantitative values are much more accurate.

Although further experiments are needed, these preliminary results support one of our hypothesis: ***extended versions of LFIT that allow to incorporate from the dataset under consideration, some measure of the weights that actually each rule has, ends up in quantitatively more accurate programs.***

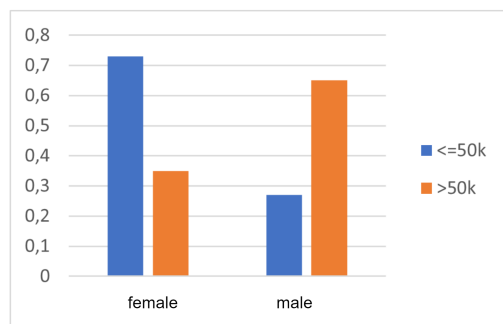


Figure 17: Weig. *gender NP* biom. hash in census

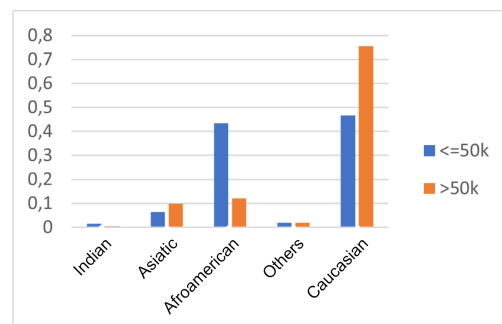


Figure 18: Weig. *ethnic group NP*, biom. hash in census

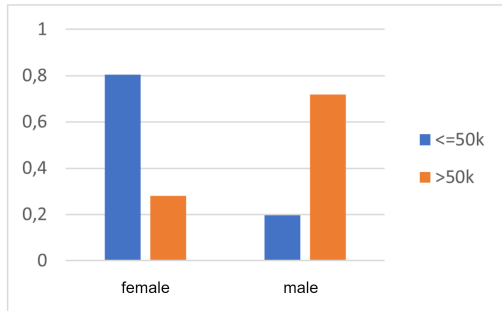


Figure 19: Weig. *gender NP*, CAIM in census

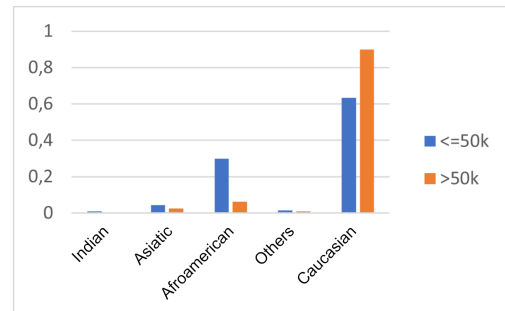


Figure 20: Weig. *ethnic group NP*, CAIM in census

6. Conclusions and further research lines

This paper represents a proof of concept for incorporating to LFIT a general purpose discretizer that includes CAIM and biometric hashing respectively as supervised and unsupervised methods. Experiments suggest that LFIT will be able to cope with general numeric datasets.

This research has also extended LFIT to improve the accuracy of its explanations when dealing with demographic or biometric datasets that usually include information of the final weight that each row of the dataset actually has.

In addition, the experiments done have supported the answer to all the questions introduced in section 3.

In the future we plan new experiments to deeply analyse all these questions and to apply the new extended LFIT version to numeric domains already introduced:

- Does CAIM find the best number of clusters?
- Do supervised discretizers outperform unsupervised ones?
- Is the discretization used able to get different LFIT results?
- Does biometric hashing used worse the expressive power of other ad-hoc discretizers?
- To incorporate some measure of the weights that actually each rule has ends up in quantitatively more accurate LFIT results?

Acknowledgments

Supported by project BBforTAI (PID2021-127641OB-I00MICINN/FEDER).

References

- [1] A. Acien, A. Morales, R. Vera-Rodriguez, I. Bartolome, J. Fierrez, Measuring the gender and ethnicity bias in deep models for face recognition, in: Proceedings of Iberoamerican Congress on pattern recognition (IbPRIA), IbPRIA, 2018.

- [2] P. Drozdzowski, C. Rathgeb, A. Dantcheva, N. Damer, C. Busch, Demographic bias in biometrics: a survey on an emerging challenge, *IEEE Trans Technol Soc* 1 (2020) 89–103.
- [3] S. Nagpal, M. Singh, R. Singh, M. Vatsa, N. Ratha, Deep learning for face recognition: pride or prejudiced?, *CoRR abs/1904.01219* (2019). URL: <https://arxiv.org/abs/1904.01219>.
- [4] J. Zhao, T. Wang, M. Yatskar, V. Ordoñez, C. K., Men also like shopping: reducing gender bias amplification using corpus-level constraints, in: *Proceedings of conference on empirical methods in natural language processing*, Association for Computational Linguistics, 2017, p. 2979–89.
- [5] S. Noble, *Algorithms of oppression: how search engines reinforce racism*, NYU Press, 2018.
- [6] L. Sweeney, Discrimination in online ad delivery: Google ads, black names and white names, racial discrimination, and click advertising, *Queue* 11 (2013) 10–29. URL: <https://doi.org/10.1145/2460276.2460278>. doi:10.1145/2460276.2460278.
- [7] M. Ali, P. Sapiezynski, M. Bogen, A. Korolova, A. Mislove, A. Rieke, Discrimination through optimization: how facebook’s ad delivery can lead to skewed outcomes, in: *Proceedings of the ACM conference on human–computer interaction*, Association for Computational Linguistics, 2019, p. 2979–89.
- [8] J. Angwin, J. Larson, S. Mattu, L. Kirchner, *Machine bias*, ProPublica, 2016.
- [9] M. Evans, A. Mathews, New york regulator probes united health algorithm for racial bias (2019).
- [10] W. Knight, *The apple card didn’t ‘see’ gender—and that’s the problem* (2019).
- [11] J. Buolamwini, T. Gebru, Gender shades: intersectional accuracy disparities in commercial gender classification, in: *Proceedings of the ACM conference on fairness, accountability, and transparency*, Association for Computational Linguistics, 2018.
- [12] M. Wang, W. Deng, Mitigating bias in face recognition using skewness-aware reinforcement learning, in: *IEEE conference on computer vision and pattern recognition (CVPR)*, IEEE, 2020, p. 9322–31.
- [13] I. Serna, A. Morales, J. Fierrez, M. Cebrian, N. Obradovich, I. Rahwan, Algorithmic discrimination: formulation and exploration in deep learning-based face biometrics, in: *Proceedings of the AAAI workshop on SafeAI: CEUR Workshop Proceedings, AAAI, 2020*, p. 9322–31.
- [14] G. Balakrishnan, Y. Xiong, W. Xia, P. Perona, Towards causal benchmarking of bias in face analysis algorithms, in: *European conference on computer vision (ECCV)*, Springer-Verlag, 2020, p. 547–63.
- [15] B. Goodman, S. Flaxman, Eu regulations on algorithmic decision-making and a “right to explanation.”, *AI Mag* 38 (2021) 50–57.
- [16] D. Castelvechi, Can we open the black box of ai?, *Nature News* 538 (2016) 20–23.
- [17] A. B. Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. García, S. Gil-López, D. Molina, R. Benjamins, et al., Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai, *Information fusion* 58 (2020) 82–115.
- [18] D. Michie, Machine learning in the next five years, in: D. H. Sleeman (Ed.), *Proceedings of the Third European Working Session on Learning, EWSL 1988*, Turing Institute, Glasgow, UK, October 3-5, 1988, Pitman Publishing, 1988, pp. 107–122.
- [19] T. Ribeiro, *Studies on learning dynamics of systems from state transitions*, 2015. PhD.

- [20] A. Ortega, J. Fierrez, A. Morales, Z. Wang, M. de la Cruz, C. L. Alonso, T. Ribeiro, Symbolic ai for xai: Evaluating lfit inductive programming for explaining biases in machine learning, *Computers* 10 (2021). URL: <https://www.mdpi.com/2073-431X/10/11/154>. doi:10.3390/computers10110154.
- [21] A. Ortega, J. Fierrez, A. Morales, Z. Wang, T. Ribeiro, Symbolic ai for xai: Evaluating lfit inductive programming for fair and explainable automatic recruitment, in: *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, 2021, pp. 78–87.
- [22] F. Sabbatini, G. Ciatto, R. Calegari, A. Omicini, Symbolic knowledge extraction from opaque ML predictors in PSyKE: Platform design & experiments, *Intelligenza Artificiale* 16 (2022) 27–48. URL: <https://content.iospress.com/articles/intelligenza-artificiale/ia220141>. doi:10.3233/IA-210120.
- [23] J. S. Black, P. van Esch, Ai-enabled recruiting: What is it and how should a manager use it?, *Business Horizons* 63 (2020) 215–226. URL: <https://www.sciencedirect.com/science/article/pii/S0007681319301612>. doi:<https://doi.org/10.1016/j.bushor.2019.12.001>.
- [24] M. Bertrand, S. Mullainathan, Are emily and greg more employable than lakisha and jamal? a field experiment on labor market discrimination, *American Economic Review* 94 (2004) 991–1013. URL: <https://www.aeaweb.org/articles?id=10.1257/0002828042002561>. doi:10.1257/0002828042002561.
- [25] M. Raghavan, S. Barocas, J. Kleinberg, K. Levy, Mitigating bias in algorithmic hiring: evaluating claims and practices, in: *Conference on fairness, accountability, and transparency*, ACM, Association for Computing Machinery, 2020, p. 469–81.
- [26] C. Schumann, J. Foster, N. Mattei, J. Dickerson, We need fairness and explainability in algorithmic hiring, in: *Proceedings of the 19th international conference on autonomous agents and multiagent systems*, International Foundation for Autonomous Agents and Multiagent Systems, 2020, p. 1716–20.
- [27] J. Sánchez-Monedero, L. Dencik, L. Edwards, What does it mean to ‘solve’ the problem of discrimination in hiring? social, technical and legal perspectives from the uk on automated hiring systems, in: *Conference on fairness, accountability, and transparency*, Association for Computing Machinery, 2020, p. 458–68.
- [28] H. Blockeel, L. De Raedt, Lookahead and discretization in ilp, in: *Inductive Logic Programming: 7th International Workshop, ILP-97 Prague, Czech Republic September 17–20, 1997 Proceedings* 7, Springer, 1997, pp. 77–84.
- [29] A. Ortega, J. Fierrez, A. Morales, Z. Wang, T. Ribeiro, Symbolic AI for XAI: evaluating LFIT inductive programming for fair and explainable automatic recruitment, in: *IEEE Winter Conference on Applications of Computer Vision Workshops, WACV Workshops 2021, Waikola, HI, USA, January 5-9, 2021, IEEE, 2021*, pp. 78–87. URL: <https://doi.org/10.1109/WACVW52041.2021.00013>. doi:10.1109/WACVW52041.2021.00013.
- [30] S. Kotsiantis, D. Kanellopoulos, Discretization techniques: A recent survey, *GESTS International Transactions on Computer Science and Engineering* 32 (2006) 47–58.
- [31] L. A. Kurgan, K. J. Cios, Caim discretization algorithm, *IEEE transactions on Knowledge and Data Engineering* 16 (2004) 145–153.
- [32] M. Boulle, Khiops: A statistical discretization method of continuous attributes, *Machine Learning* (2004) 53–69. URL: <https://doi.org/10.1023/B:MACH.0000019804.29836.05>. doi:10.1023/B:MACH.0000019804.29836.05.

- [33] K. M. Ho, P. D. Scott, An efficient global discretization method, in: X. Wu, R. Kotagiri, K. B. Korb (Eds.), *Research and Development in Knowledge Discovery and Data Mining*, Springer Berlin Heidelberg, Berlin, Heidelberg, 1998, pp. 383–384.
- [34] R. Kerber, Chimerge: Discretization of numeric attributes, in: *Proceedings of the Tenth National Conference on Artificial Intelligence, AAAI'92*, AAAI Press, 1992, p. 123–128.
- [35] Y. Yang, G. I. Webb, X. Wu, *Discretization Methods*, Springer US, Boston, MA, 2010, pp. 101–116. URL: https://doi.org/10.1007/978-0-387-09823-4_6. doi:10.1007/978-0-387-09823-4_6.
- [36] J. Dougherty, R. Kohavi, M. Sahami, Supervised and unsupervised discretization of continuous features, in: A. Prieditis, S. Russell (Eds.), *Machine Learning Proceedings 1995*, Morgan Kaufmann, San Francisco (CA), 1995, pp. 194–202. URL: <https://www.sciencedirect.com/science/article/pii/B9781558603776500323>. doi:<https://doi.org/10.1016/B978-1-55860-377-6.50032-3>.
- [37] A. Cano, D. T. Nguyen, S. Ventura, K. J. Cios, ur-caim: improved CAIM discretization for unbalanced and balanced data, *Soft Comput.* 20 (2016) 173–188. URL: <https://doi.org/10.1007/s00500-014-1488-1>. doi:10.1007/s00500-014-1488-1.
- [38] R. Kerber, Chimerge: Discretization of numeric attributes, in: *Proceedings of the Tenth National Conference on Artificial Intelligence, AAAI'92*, AAAI Press, 1992, p. 123–128.
- [39] H. Liu, R. Setiono, Feature selection via discretization, *IEEE Transactions on knowledge and Data Engineering* 9 (1997) 642–645.
- [40] U. M. Fayyad, K. B. Irani, Multi-interval discretization of continuous-valued attributes for classification learning, in: *International Joint Conference on Artificial Intelligence, 1993*.
- [41] H. Liu, F. Hussain, C. L. Tan, M. Dash, Discretization: An enabling technique, *Data mining and knowledge discovery* 6 (2002) 393–423.
- [42] J. R. Quinlan, Induction of decision trees, *Machine learning* 1 (1986) 81–106.
- [43] D. A. Zighed, S. Rabaséda, R. Rakotomalala, Fusinter: a method for discretization of continuous attributes, *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 6 (1998) 307–326.
- [44] M. R. Freire, J. Fierrez, J. Galbally, J. Ortega-Garcia, Biometric hashing based on genetic selection and its application to on-line signatures, in: *Advances in Biometrics: International Conference, ICB 2007, Seoul, Korea, August 27-29, 2007. Proceedings*, Springer, 2007, pp. 1134–1143.
- [45] J. MacQueen, Classification and analysis of multivariate observations, in: *5th Berkeley Symp. Math. Statist. Probability*, University of California Los Angeles LA USA, 1967, pp. 281–297.
- [46] S. Garcia, J. Luengo, J. A. Sáez, V. Lopez, F. Herrera, A survey of discretization techniques: Taxonomy and empirical analysis in supervised learning, *IEEE transactions on Knowledge and Data Engineering* 25 (2012) 734–750.
- [47] K. Inoue, T. Ribeiro, C. Sakama, Learning from interpretation transition, *Machine Learning* 94 (2014) 51–79.
- [48] T. Ribeiro, M. Magnin, K. Inoue, C. Sakama, Learning delayed influences of biological systems, *Frontiers in Bioengineering and Biotechnology* 2 (2015) 81.
- [49] D. Martínez Martínez, T. Ribeiro, K. Inoue, G. Alenyà Ribas, C. Torras, Learning probabilistic action models from interpretation transitions, in: *Proceedings of the Technical*

- Communications of the 31st International Conference on Logic Programming (ICLP 2015), 2015, pp. 1–14.
- [50] T. Ribeiro, M. Magnin, K. Inoue, C. Sakama, Learning multi-valued biological models with delayed influence from time-series observations, in: 2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA), 2015, pp. 25–31. doi:10.1109/ICMLA.2015.19.
 - [51] D. Martinez, G. Alenya, C. Torras, T. Ribeiro, K. Inoue, Learning relational dynamics of stochastic domains for planning, in: Proceedings of the 26th International Conference on Automated Planning and Scheduling, 2016.
 - [52] T. Ribeiro, S. Tournet, M. Folschette, M. Magnin, D. Borzacchiello, F. Chinesta, O. Roux, K. Inoue, Inductive learning from state transitions over continuous domains, in: N. Lachiche, C. Vrain (Eds.), Inductive Logic Programming, Springer, 2018, pp. 124–139.
 - [53] T. Ribeiro, M. Folschette, M. Magnin, O. Roux, K. Inoue, Learning dynamics with synchronous, asynchronous and general semantics, in: International Conference on Inductive Logic Programming, Springer, 2018, pp. 118–140.
 - [54] T. Ribeiro, M. Folschette, M. and Magnin, K. Inoue, Learning any semantics for dynamical systems represented by logic programs, 2020. URL: <https://hal.archives-ouvertes.fr/hal-02925942>, working paper or preprint.
 - [55] T. Ribeiro, M. Folschette, L. Trilling, N. Glade, K. Inoue, M. Magnin, O. Roux, Les enjeux de l'inférence de modèles dynamiques des systèmes biologiques à partir de séries temporelles, in: C. Lhoussaine, E. Remy (Eds.), Approches symboliques de la modélisation et de l'analyse des systèmes biologiques, ISTE Editions, 2020. In edition.
 - [56] R. Kohavi, Scaling up the accuracy of naive-bayes classifiers: a decision-tree hybrid, in: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, 1996.
 - [57] T. Ribeiro, M. Folschette, M. Magnin, K. Inoue, Learning any memory-less discrete semantics for dynamical systems represented by logic programs, Machine Learning (to appear) (2021).
 - [58] O. Iken, M. Folschette, T. Ribeiro, Automatic modeling of dynamical interactions within marine ecosystems, International Conference on Inductive Logic Programming (to appear as Late-breaking abstracts and poster) (2021).
 - [59] S. Aeberhard, M. Forina, Wine, UCI Machine Learning Repository, 1991. doi: <https://doi.org/10.24432/C5PC7J>.