Group Assignment

Knowledge Representation

Summary. The project assignment will be devoted to writing Prolog programs answering counterfactual explanations related to fairness in a pattern classification dataset. These topics have significant relevance in modern Artificial Intelligence and will help illustrate the practical usability of the declarative programming paradigm. The project will consist of two parts: (1) designing a Prolog knowledge base given a raw file containing symbolic terms and confidence values and (2) implementing Prolog queries to answer counterfactual explanations for interesting scenarios. Students must upload a single text file containing the Prolog knowledge base, the queries resolving the counterfactual explanation questions, and concise comments discussing relevant implementation details and the solutions provided by each query. This assignment will count for 20% of the course grade but passing is not mandatory to pass the course.

Why is this project relevant?

In recent years, the performance of machine learning algorithms in solving complex tasks from a high volume of structured and unstructured data has caught the attention of industry, governments and society. These techniques are increasingly deployed in specific decision-making tasks that affect humans directly, such as medical diagnosis or treatment, recidivism prediction, personalized recommendations, recruiting, etc. Therefore, there is a clear need to ensure the meaningful and responsible use of machine learning models. Regrettably, the best-performing machine learning techniques (such the ones based on deep learning) tend to build less transparent models, obstructing trust and raising questions about the accountability and fairness of the decisions.

Algorithmic decision-making is used by organizations partly because machines seem neutral and unbiased. Such an assumption is rather naive since bias can be encoded into historical data by capturing the discriminatory beliefs of people involved in the data generation process. Artificial Intelligence (AI) algorithms then use the contaminated data to facilitate life-changing decisions such as accepting loan applications or suggesting appropriate medical treatments. Therefore, detecting bias in data used by Machine Learning (ML) systems is of paramount importance to the welfare of both society and individuals.

Bias in decision-making tasks[1] can be expressed implicitly or explicitly. Direct discrimination (or explicit bias) occurs when decisions are influenced by sensitive or protected features like gender, race, or marital status. Indirect discrimination (or implicit bias) occurs when decisions are influenced by non-sensitive features that strongly correlate with sensitive ones, thus resulting in non-favorable outcomes towards underprivileged groups. Implicit bias is also referred to in the literature as unconscious bias. It expresses unintentional forms of discrimination that infect decision-making systematically and remains hard to address during data collection, labeling, or at an algorithmic level.

In this project, you will implement a (simplified) symbolic explanation module to answer counterfactual explanations using the Prolog programming language. These explanations will help determine whether individuals are discriminated against in decision-making processes.

German credit dataset

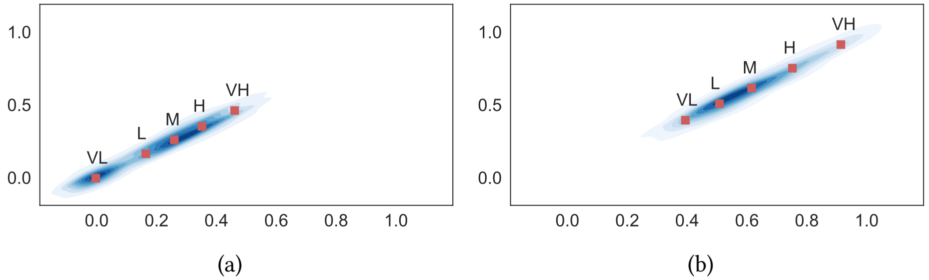
In this project, you will use the German credit dataset, which is used for classifying loan applicants at a bank as good or bad credit risks. It contains information about 1000 applicants, with 700 belonging to the good class and 300 to the bad class. Applicants are described by 20 qualitative and quantitative features: status of existing checking account, credit duration, credit history, purpose, credit amount, savings account/bonds, present employment since, installment rate in percentage of disposable income, personal status and sex, other debtors/guarantors, present residence since, property, age in years, other installment plans, housing, number of existing credits at this bank, job, number of people being liable to provide maintenance for, and foreign worker.

[1] This video provides a general overview of bias and fairness.

Based on the literature, the features *age* and *personal status and sex* are considered protected features. Protected characteristics or features are specific aspects of a person's identity defined by the Equality Act 2010. The protection aspect relates to protection from discrimination. It should also be mentioned that gender/female and age/young (younger than 25 years old) are the protected groups. Overall, when it comes to knowledge encoding , a protected group can be represented by a symbolic value for a protected feature.

Transforming the dataset

Unfortunately, we cannot simply use this dataset as it appears on the Internet. While the values of qualitative features can be included in a Prolog knowledge base, performing symbolic reasoning with continuous features might be far from ideal. Instead, we need to transform these quantitative (numerical) features into qualitative (symbolic) ones. The instructors have transformed these features by using the approach proposed in the paper entitled: Prolog-based agnostic explanation module for structured pattern classification. Overall, the algorithm uses fuzzy logic theory to transform numerical features into symbolic knowledge representations and quantify the uncertainty of such a transformation. This can be done by building fuzzy clusters denoted by fuzzy prototypes that are associated with pre-defined symbolic terms that fulfill an order relation. Figure 1 shows the fuzzy prototypes for two normalized numerical features. These prototypes are ordered along the identity line (where axes depict the same values for that feature, thus making the visualization space bi-dimensional). In these examples, we use the following symbolic terms to label the prototypes: very low (VL), low (L), medium (M), high (H) and very high (VH).



(a)

(b)

Figure 1: Fuzzy prototypes (red squares) for two numerical features, which are labeled with predefined symbolic terms. The blue area denotes the numerical values visualized in a 2D space.

Once the fuzzy prototypes have been obtained for each feature, we build pairs with the form (p_j, l_j) where the first component is the prototype and the second is the corresponding linguistic term. Next, we can assign each numerical value x_i of a given feature with an appropriate symbolic term. This is done by selecting the closest prototype p_j to x_i and then matching that point with the corresponding symbolic term l_j. Moreover, we can determine the confidence of such a matching as the membership degree of x_i to the j-th fuzzy cluster.

As mentioned, we should specify a set of symbolic terms for each numerical feature we want to transform into symbolic ones. In the case of the feature *credit duration*, we will use the following symbolic terms: very short-term (VS), short-term (S), medium-term (M), long-term (L), and very long-term (VL). In the case of *present residence since*, we will use the following symbolic terms: long time ago (L), some time ago (S), fairly recent (F), recently (R), and very recent (VR). For the numerical features representing quantities, we will use very low (VL), low (L), medium (M), high (H) and very high (VH).

Note. It is worth mentioning that students are not required to fully understand the whole process of deriving the symbolic terms to complete the assignment since that algorithm goes beyond the course scope.

The transformed dataset (encoded as a CSV file for easy manipulation) and the feature descriptions can be downloaded from Canvas. In the dataset, each row (or instance) represents a credit application, while columns enclose the symbolic terms and confidence values for each problem feature. Confidence values are in the $[0, 1]$ interval and can be understood as a quality measure. In addition, each instance is labeled with a decision class (good or bad credit risk) and its corresponding confidence value. Finally, each instance is associated with a global confidence value that indicates how much that instance conflicts with others in the dataset. For example, a confidence value of one suggests that the instance does not conflict with any other instances in the dataset.

Assignment details

As mentioned, the programming project will consist of two parts: (1) designing a Prolog knowledge base from the transformed German credit dataset containing symbolic terms and confidence values and (2) implementing Prolog queries to answer counterfactual explanations for some scenarios.

Part 1. Build a Prolog knowledge base using the symbolic terms and confidence values describing the transformed German credit dataset. The knowledge base must support queries involving the problem features, the different confidence values associated with each instance or the decision class. For example, you might want to define a rule for each instance where the antecedent denotes the feature values and their confidence values, while the consequent encodes the decision class and its confidence value. Moreover, the global confidence value of each instance can be encoded somewhere as the rule confidence.

After completing the first part of the assignment, students can then proceed to implement Prolog queries to answer counterfactual explanations for two scenarios concerning bias in the transformed German credit dataset.

Before going any further, it seems useful to introduce the *counterfactual explanation* notion. Overall, a counterfactual explanation describes a causal situation in the form: “If X had not occurred, Y would not have occurred”. For example: “If I had not taken a sip of this hot coffee, I would not have burned my tongue”. The event Y is that I burned my tongue, while the cause X is that I had a hot coffee. Counterfactual thinking requires imagining a hypothetical reality that contradicts the observed facts (for example, a world in which I have not drunk the hot coffee), hence the name *counterfactual*. In this project, counterfactual explanations can be used to explain predictions of individual instances and investigate the scenarios concerning the profiles below. The event is the predicted outcome of an instance, the causes are the particular feature values of this instance that were input to the model and caused a certain prediction.

Part 2(a) *Are women discriminated against?*

An AI-powered decision system built using the German credit dataset denies the loan to a female applicant (see profile below). However, the loan requested by one of her male friends was granted (the ’present residence since’ field differs in both applications but that is deemed not relevant). Write a Prolog query to determine if the female applicant was discriminated against due to her gender by the AI-powered decision system. In addition, you should determine all confidence values supporting the answer provided by the program.

First sensitive profile (female applicant): A young female whose existing checking account at the bank is ‘<0DM‘, the credit duration is ‘very short‘, her credit history is labeled as ‘existing credits paid back duly until now‘, the purpose of the loan is to ‘buy a new car‘, the credit amount is encoded as ‘very low‘, the savings account/bonds is ‘<100DM‘, her present employment falls in the category ‘1 <=...< 4 years‘, the installment rate in percentage of disposable income is encoded as ‘very high‘, her marital status is ‘separated‘, she does not have ‘other debtors/guarantors‘, her present residence date is ‘fairly recent‘, she ‘owns a real state property, and she does not have other installment plans. Moreover, the applicant is the ‘owner‘ of her house, her number of existing credits at this bank is labeled as ‘very low‘, she is an ‘unskilled resident‘, the number of people being liable to provide maintenance for is labeled as ‘very low‘ by the bank, she does not have a telephone, and she is a ‘foreign worker‘.

Part 2(b) *Are young people discriminated against?*

An AI-powered decision system using the German credit dataset denies the loan to a young male applicant (see profile below). However, the loan requested by one of his older friends was granted (the ’present employment since’ field differs in both applications, which might be relevant for this scenario). Write a Prolog query to determine if the applicant was discriminated against due to his young age. In addition, you should determine all confidence values supporting the answer provided by the Prolog program.

Second sensitive profile (young applicant): A young male whose existing checking account at the bank is ‘>=200DM‘, the credit duration is ‘long‘, his credit history is labeled as ‘existing credits paid back duly until now‘, the purpose of the loan is to ‘buy a radio/television‘, the credit amount is encoded as ‘medium‘, the savings account/bonds is ‘<100DM‘, his present employment duration falls in the category ‘1 <=...< 4 years‘, the installment rate in percentage of disposable income is encoded as ‘very high‘, his marital status is ‘single‘, he does not have ‘other debtors/guarantors‘, his present residence date is ‘fairly recent‘, he ‘owns a car‘, and he does not have other installment plans. Moreover, the applicant is the ‘owner‘ of his house, his number of existing credits at this bank is labeled as ‘very low‘, he is a ‘skilled employee/official‘, the number of people being liable to provide maintenance for is labeled as ‘very low‘, he does not have a registered telephone, and he is a ‘foreign worker‘.

To solve the tasks concerning counterfactual explanations, you might need to manipulate the knowledge base dynamically (it depends on the selected knowledge base design). This can be done using the following predicates: `dynamic` [2], `assertz` [3] and `retractall` [4]. The first predicate informs the Prolog interpreter that the definition of the predicate(s) may change during execution (using `assertz` and/or `retractall`). The second predicate asserts a clause (fact or rule) into the database. Finally, the third predicate removes all facts or clauses for which the head unifies the specified head. Recall that the way you will use (or not) these predicates depends entirely on the solutions to the tasks.

Submission and grading

Students will work in groups of two (minimum), three or four students (maximum). No exceptions will be made. Doing the project alone is not allowed since we intend to promote the discussion and collaboration among students, yet each student is responsible for being enrolled in a group. However, we will provide assistance if you have difficulty to find a group.

Deliverable. Students are requested to submit a .pl file containing the knowledge base, the Prolog queries, and concrete answers to tasks 2a) and 2b). The queries and answers can be included as comments to avoid compilation issues. Likewise, the file should include the names of all group members as a comment (a template for guidance will be provided). The knowledge base must be uploaded to Canvas by any group member. Submissions via email will not be accepted unless some exceptional situation comes to play.

Grading. In order to get a pass, the knowledge base must contain clauses representing the information describing the transformed German credit dataset, the program must compile, and the queries must run and provide correct answers. Please notice that there could be different correct approaches to build the knowledge base, and all of them will be accepted as long as the queries provide correct answers. Minor design issues such as inefficient knowledge representations or unnecessary clauses might be tolerated. Failing projects will receive a concise feedback on the main issues that led to a fail decision. Notice that passing this project assignment is not mandatory to pass the course successfully, which means that there will be no resit opportunity for it.

[2] https://www.swi-prolog.org/pldoc/man?predicate=dynamic/1

[3] https://www.swi-prolog.org/pldoc/man?predicate=assertz/1

[4] https://www.swi-prolog.org/pldoc/man?predicate=retractall/1

Communication

The communication of this course will be implemented via the discussion section of Canvas. This includes important announcements, complementary materials and answering either general or technical questions.

Contacting the instructors. Students are kindly asked to refrain from contacting the instructors via email to ask questions about the project organization and related contents. Of course, you can contact them via email if there is a private issue that you feel your fellow students should not be aware of.

How to ask your questions. Students should post all questions concerning the project in the corresponding discussion entry of Canvas. However, students are not allowed to ask the instructor to revise their projects or provide feedback prior to submission. Please make sure that questions are short, sharp and to the point, and that similar questions have not been posted. If deemed necessary, you can include code snippets to provide some extra context to your question. Students are encouraged to answer the questions in the discussion section. The instructor will always validate the answers and further expand them whenever opportune. Overall, students can expect a reply within three working days after the question being posted on Canvas (during working hours).

Important dates

The deadline for submitting the knowledge base through Canvas is strict. Please be aware that there will be one opportunity to submit the project assignment. Below you can find the timeline with relevant dates.

14.03.2023	The project assignment is released on Canvas
09.05.2023	The project deliverable is uploaded to Canvas
23.05.2023	The pass/no-pass decisions are released