Multi-objective Counterfactuals in Bayesian Classifiers with Estimation of Distribution Algorithms

Daniel Zaragoza-Pellicer DANIEL.ZARAGOZAP@ALUMNOS.UPM.ES
Concha Bielza MCBIELZA@FI.UPM.ES
Pedro Larrañaga PEDRO.LARRANAGA@FI.UPM.ES
Departamento de Inteligencia Artificial, Universidad Politécnica de Madrid, Boadilla del Monte, 28660 Madrid (Spain)

Editors: J.H.P. Kwisthout & S. Renooij

Abstract

Counterfactual explanations are a very popular and effective method to convey interpretability in supervised classification models. These explanations answer the question of which change is needed in the input data to obtain a desired output. Computing good counterfactuals involves achieving some key objectives, such as validity, minimality, similarity or plausibility. Our proposal consists of using estimation of distribution algorithms for approximating counterfactual explanations within Bayesian classifiers. They are experimentally compared with a genetic algorithm, both with a single-objective and with a multi-objective formulation. Different types of Bayesian classifiers will be evaluated to find the differences in their explanations and we will use their results together to provide more accurate explanations. The experiments show how estimation of distribution algorithms are faster and achieve better results with a single-objective whereas they are competitive in the multi-objective version.

Keywords: Counterfactual explanations, estimation of distribution algorithms, genetic algorithms, Bayesian classifiers

1. Introduction

Each year artificial intelligence and machine learning models become increasingly popular and powerful, but at the same time their complexity renders them less interpretable. Explainable artificial intelligence has gained protagonism, enhancing interpretable models and providing post-hoc explanations for black box models. This interpretability is becoming an indispensable requirement for people to trust these models, which is why within this field there are a large variety of methods and models developed to find explanations (Holzinger et al. (2022), Molnar (2022), Linardatos et al. (2020)). Dwivedi et al. (2023) describe the different types of techniques in which explainability can be approached depending on the objective pursued, such as explaining the reasoning of the model, explaining the importance of different variables or explaining specific instances.

Among the techniques explaining the reasoning of the model, we will focus on counterfactual explanations within supervised classification. They answer the question of which change is needed in the input data to obtain a desired output. Obtaining good counterfactual explanations may not be straightforward, hence many methods have been developed (Verma et al. (2020), Guidotti (2022)). Heuristic search based approaches usually involve minimizing a cost function accounting for the sought objective(s). Wachter et al. (2017) is

one of the first works to propose to optimize a function, combining the distance between the input and the generated counterfactual, and the counterfactual estimated output class probability. Dhurandhar et al. (2018) add plausibility in their cost function and Mothilal et al. (2020) focus on plausibility and solution diversity, penalizing similar solutions. As for the heuristic search used, Lash et al. (2017) use genetic algorithms and local search, Moore et al. (2019) apply gradient-descent methods and Lucic et al. (2020) use Monte Carlo simulation. Many existing methods combine different objectives into a single one to fulfil the desired properties, at the expense of losing information when combined. Other methods pose the problem as multi-objective optimization.

This paper seeks to approach counterfactuals with estimation of distributions algorithms (EDAs). EDAs (Larrañaga and Lozano (2002)) are evolutionary algorithms that, at each generation, build a probabilistic model from the best solutions found. They follow a process similar to genetic algorithms (GAs) but eliminating crossover and mutation, since the probabilistic model will take care of generating new solutions. There are many different EDAs (Hauschild and Pelikan (2011)), either for continuous or discrete data, or depending on the probabilistic model they learn. We will compare the EDA implementation with the GA-based multi-objective method for obtaining counterfactuals proposed by Dandl et al. (2020). Moreover, we will compare both the multi-objective and single-objective versions of both algorithms, looking at how they compare with each other and what advantages are obtained by using multi-objective functions. A group of Bayesian classifiers will be used to observe how the explanations obtained may differ depending on which model or combinations of models are used.

The paper is organised as follows. In Section 2 we define the background notation and definitions. We present our proposal in Section 3, where we define the algorithms and classifiers used. Experiments are shown in Section 4, and conclusions in Section 5.

2. Background

2.1. Counterfactuals

Different authors have defined counterfactual explanations from a variety of points of view. For example, Guidotti (2022) formalized the problem with the objective of minimizing the change of the input variables for achieving a different prediction.

Definition 1 ***(Guidotti (2022))*** *Given a classifier ϕ that outputs the decision $c = \phi(\boldsymbol{x})$ for an instance $\boldsymbol{x}$, a counterfactual explanation consists of an instance $\boldsymbol{x}'$ such that the decision for ϕ on $\boldsymbol{x}'$ is different from c, i.e., $\phi(\boldsymbol{x}') \neq c$, and such that the difference between $\boldsymbol{x}$ and $\boldsymbol{x}'$ is minimal.*

To find the best possible explanation, a counterfactual explainer will be used, which will be in charge of finding $\mathbf{x}'$ that meets the constraints of Definition 1. Thus, the only thing that remains to be defined is what it means for the difference between $\mathbf{x}$ and $\mathbf{x}'$ to be minimal. Each method for calculating counterfactuals has defined which objectives are important to obtain that minimal difference while being a high quality counterfactual. The objectives that we consider most important are:

- *Validity*: the classification output has to be different from the original one.

- *Minimality*: the number of changes between $\mathbf{x}'$ and $\mathbf{x}$, and the distance between $\mathbf{x}'$ and $\mathbf{x}$ should be as small as possible.
- *Plausibility*: $\mathbf{x}'$ should be coherent with an observation population, i.e, $\mathbf{x}'$ can occur with the given data.

2.2. Estimation of distribution algorithms

EDAs are evolutionary algorithms that, at each generation, explore the solution space by sampling a probabilistic model constructed from the best solutions found. The EDA procedure is outlined in Algorithm 1. EDAs work with a population of candidate solutions, which are scored using a cost function (line 3). This function ranks the solutions and the best ones are selected to learn the probabilistic model (lines 4 and 5). Then a new population is sampled from the model (line 6) and the process is repeated until a termination criterion is met (line 2). From this basic procedure a multitude of variants have been developed, adapting it to different data types and more complex problems.

Algorithm 1: EDA procedure

Input: Population size, cost function, selection rate
Output: Best individual and cost
1 Initial population
2 **for** $t = 1, 2, \ldots$ *until stopping criterion is met* **do**
3 Evaluate population using a cost function
4 Select individuals
5 Learn a probabilistic model from the best individuals
6 Sample new individuals from the probabilistic model
7 **end**

Our proposal is to create a multi-objective EDA by modifying the cost function and the process for selecting individuals of a UMDA (Mühlenbein and Paass (1996)) and an EBNA (Etxeberria and Larrañaga (1999)) for categorical data. The individual selection system used is based on non-dominated sorting and crowding distance. Then we will compare these algorithms with the genetic algorithm NSGA2, where both use the same cost function. In addition to these two algorithms, single-objective versions of both will be used.

3. Our proposal

Let $\mathbf{X} = (X_1, \ldots, X_n)$ denote the predictor (categorical) features from N labeled instances and $\mathcal{D} = \{(\mathbf{x}^1, c^1), \ldots, (\mathbf{x}^N, c^N)\}$ the dataset, where for each $\mathbf{x}^i = (x_1^i, \ldots, x_n^i), i = 1, \ldots, N$, we have the respective value c^i of a class variable C with labels in the domain $\Omega_C = \{c_1, \ldots, c_R\}$. The domain of each X_i is accordingly denoted Ω_{X_i}.

Definition 2 *Given $\phi : \Omega_{X_i} \times \cdots \times \Omega_{X_n} \to \Omega_C$ a Bayesian classifier (Bielza and Larrañaga (2014)), and $(\boldsymbol{x}^*, c)$ an instance of $\mathcal{D}$, a counterfactual explanation $\boldsymbol{x}'$ for $\boldsymbol{x}^*$ is a solution of the multi-objective problem*

$$\min_{\boldsymbol{x}} \boldsymbol{f}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), f_3(\boldsymbol{x}), f_4(\boldsymbol{x})) \tag{1}$$

where:

- $f_1(\mathbf{x}')$ is prediction objective, defined as the Manhattan distance between the class-posterior distribution of the counterfactual $\mathbf{x}'$, $\mathbf{P}(C|\mathbf{x}')$, and the distribution corresponding to the desired outcome (i.e., a vector $\mathbf{P}'$ with all zeros except for a 1 in the position corresponding to the desired class), the prediction probability and desired probability outcome, calculated through L_1 norm,

$$f_1(\mathbf{x}) = \sum_{i=1}^{n} |\mathbf{P}'_i - \mathbf{P}_i(C|\mathbf{x}')| \tag{2}$$

- $f_2(\mathbf{x}')$ is the distance objective, defined as the distance between the input instance $\mathbf{x}^*$ and the counterfactual $\mathbf{x}'$, calculated using the Gower distance (Gower (1971)) d_G,

$$d_G(\mathbf{x}', \mathbf{x}^*) = \sum_{i=1}^{n} d_i(x'_i, x^*_i)/n \tag{3}$$

  where the distance d_i per feature in the summation varies depending on whether the feature is categorical, where the distance d_i is 0 if $x'_i = x^*_i$ and 1 otherwise, or numeric, where we use the normalized Manhattan distance for d_i.

- $f_3(\mathbf{x}')$ is the number of feature changes from the input instance $\mathbf{x}^*$ to $\mathbf{x}'$,

$$f_3(\mathbf{x}) = \sum_{i=1}^{n} \mathbb{I}_{x'_i \neq x^*_i} \tag{4}$$

- $f_4(\mathbf{x}')$ is the plausibility of $\mathbf{x}'$ given $\mathcal{D}$, which is the distance between $\mathbf{x}'$ and its nearest instance in $\mathcal{D}$, given by the same distance as $f_2(\mathbf{x})$.

Our proposal is to approach this multi-objective problem with EDAs (MOEDAs), where the individual selection is based on non-dominated sorting, which consists of classifying solutions based on Pareto dominance, and crowding distance, a measure used to estimate the density of solutions surrounding a solution over the objective space. Dandl et al. (2020) used the genetic algorithm NSGA2 (Deb et al. (2000)) to optimize a related function with four objectives. We will compare this algorithm and its single-objective counterpart against our proposal.

To compute the counterfactuals we build a group of models with five Bayesian classifiers, see Figure 1. The classifiers used are naive Bayes (NB), semi-naive Bayes (SNB), tree augmented naive Bayes (TAN), hill-climbing tree augmented naive Bayes (TAN-HC) and k-dependence Bayesian classifier (KDB). NB assumes that the predictive variables are conditionally independent given the class, SNB relaxes the NB assumption by allowing dependencies within some groups of variables, TAN uses a tree structure for the dependencies of the variables, TAN-HC finds this structure in a wrapper-like manner, with hill-climbing search and KDB allows each variable to have k parents. To ensure that the generated counterfactual is as good as possible, the classifiers are first filtered based on whether their predicted class is correct. Only in this case the counterfactual will be computed. After

this selection, the models are sorted by their classification accuracy and the model with the highest accuracy is used. Alternatively, we can also use the results of more than one model, so different solutions can be obtained. Note that there is a possibility that no solution is found as no model passes the first filter; however this restriction will allow to avoid generating counterfactuals that are not accurate.

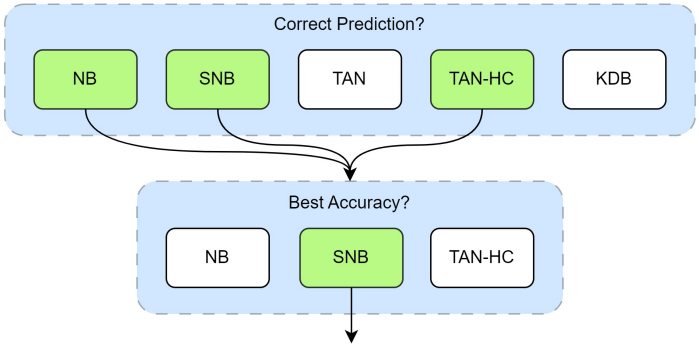


Figure 1: Model filtering and selection. NB=Naive Bayes, SNB=Semi-naive Bayes, TAN=Tree augmented naive Bayes, TAN-HC=Hill-climbing tree augmented naive Bayes, KDB=k-dependence Bayesian classifier

4. Experiments

4.1. Implementation and algorithms

The implementation for counterfactual computation can be found uploaded on GitHub[1]. It is implemented in Python using the libraries Pymoo (Blank and Deb (2020)) for genetic algorithms and EDAspy (Soloviev et al. (2024)) for EDAs. The Bayesian classifiers were implemented in R in the bnclassify package (Mihaljević et al. (2018)). All experiments were conducted on the same hardware (intel i5-12500H and 16GB RAM).

Four different algorithms have been used to find the best counterfactual, splitting them into single-objective and multi-objective:

- The single-objective algorithms used are a basic genetic algorithm, a univariate marginal distribution algorithm (Mühlenbein and Paass (1996)) (UMDA) and an estimation of Bayesian network algorithm (Etxeberria and Larrañaga (1999)) (EBNA), where both aim at minimizing only the distance (f_2). UMDA is an EDA that assumes that all variables are independent and thus their joint probability can be factorized as a product of univariate marginal probabilities, while EBNA uses BNs to capture and exploit the dependencies between variables in the solution space.

- The multi-objective algorithms are NSGA2, MOUMDA and MOEBNA. All algorithms will use the four objectives described in Section 3.

1. https://github.com/DanielZaragozaP/counterfactual_ensemble.

Each algorithm will start from a random initial population, and all algorithms will use the same number of evaluations (20) and population size (20). The results will consist of the average of all runs between datasets. Differences in prediction (f_1), distance (f_2), plausibility (f_4) and run time will be shown, where the objective (f_3) of minimizing the number of variable changes will not be presented, since in general a smaller distance implies fewer variable changes.

4.2. Datasets

The selected datasets have all discrete variables and without missing values. The datasets have been obtained from the UCI Machine Learning Repository (Kelly et al. (2023)) and from the OpenML repository (Vanschoren et al. (2014)). The datasets have been selected to contain different number of instances and features to see how the algorithms performs under different situations, see Table 1. In each dataset, 90% of the data was used for training and the remaining 10% for testing. The counterfactuals were calculated from the test data, specifically using 100 instances, except for datasets where 10% of instances is lower than 100, where we used all test instances instead. The counterfactual comparisons are calculated taking those test data as an input and a random class as the desired class.

Dataset	#Instances	#Features	#Classes
`Tic-tac-toe`	958	9	2
`Car evaluation`	1728	6	4
`Chess (kr vs kp)`	3196	35	2
`Mushroom`	8124	22	2
`Nursery`	12960	8	3
`Monk 1`	556	6	2
`Monk 2`	601	6	2
`Monk 3`	554	6	2
`Letter`	20000	16	26
`Phishing Websites`	11100	30	2

Table 1: Description of benchmark datasets

4.3. Results with the best model

The first comparison contrasts the results of the single-objective EDA with the GA and then the two multi-objective counterparts, all using the filtered model with the highest accuracy. In Table 2 we can observe the gain or loss (in %) when using EDAs with respect to GAs. In single-objective (first two rows), we can see how UMDA improves GA, while EBNA only improves it in prediction. In the multi-objective case (last two rows), we observe how NSGA2 improves in general over both EDAs, except in prediction, where UMDA obtains a large improvement over NSGA2, implying that UMDA is more confident that the explanations obtained are well classified.

In addition to these differences, another important factor is the execution time of each algorithm. Note that if more than one model is used it will be necessary to sum the times taken by all models. Table 3 shows the average time per model over all the datasets used. In

	Distance	Prediction	Plausibility
UMDA vs GA	33.88%	19.29%	−1.05%
EBNA vs GA	−16.88%	9.21%	−21.24%
MOUMDA vs NSGA2	−6.98%	169.36%	−26.69%
MOEBNA vs NSGA2	−34.19%	−30.76%	−34.54%

Table 2: Percentage of improvement (per objective) of EDAs vs GAs, in single-objective and multi-objective problems

the single-objective case, UMDA is almost twice faster than GA, while both are much faster than EBNA, taking more than five times longer than GA. In the multi-objective scenario, a similar pattern is observed, where MOUMDA is the fastest but with a smaller gap to NSGA2 than in single-objective. MOEBNA is still the slowest algorithm, although with a slightly smaller gap than in the single-objective case. An important detail here is that if you do not require the results with the best predictions and plausibilities, you can perform an UMDA with all five Bayesian classifiers in the same amount of time as a MOUMDA or EBNA with only one model.

	UMDA	EBNA	GA	MOUMDA	MOEBNA	NSGA2
Time	1.62 ± 1.72	16.73 ± 20.82	2.89 ± 2.20	6.22 ± 5.36	24.08 ± 26.57	6.87 ± 6.06

Table 3: Average execution time (s) and standard deviation of each algorithm over all datasets

4.4. Results with the two best models

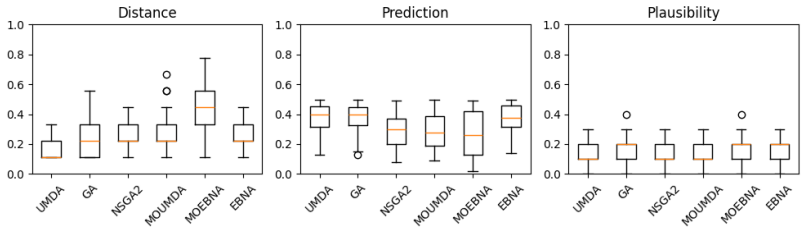
Rather than using the best model, we now observe what happens if we use more than one model to calculate the counterfactual. It is worth noting that some models may achieve the same or almost identical accuracy, this will depend on the dataset. First we analyze the possible gain when calculating the counterfactuals by adding the second best model, see Table 4. It can be seen how the distance improves with all the algorithms between 13% and 22%, while the overall prediction deteriorates and plausibility shows a slight improvement. Although confidence in the prediction is lost by using an additional model, this gain in distance can mean a considerable improvement in the counterfactual obtained, also maintaining plausibility. MOUMDA is the algorithm that takes the most advantage with the use of two models, being the one that obtains the biggest improvement in distance and plausibility.

To observe in a more visual way the effect of using two models in each algorithm, Figure 2 shows the result for the `Tic-tac-toe` dataset. The boxplots show the execution of 96 test cases where the algorithm tried to calculate the counterfactual. The results are similar to those seen in Table 4, although it is worth paying attention to the MOUMDA improvement in distance where its results are close to those seen with the single-objective

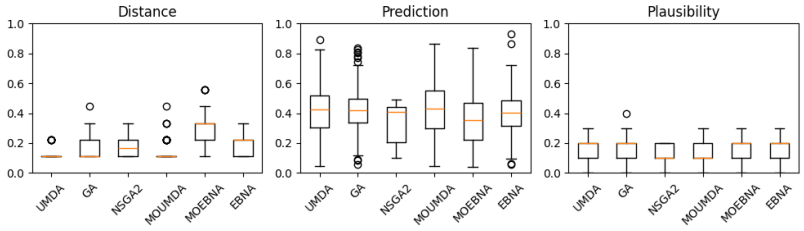
	Distance	Prediction	Plausibility
UMDA	14.44%	−25.77%	5.85%
EBNA	14.41%	−20.29%	6.40%
GA	14.42%	−23.20%	1.09%
MOUMDA	22.52%	−34.70%	20.13%
MOEBNA	14.75%	−27.51%	7.57%
NSGA2	13.09%	−23.11%	−0.09%

Table 4: Gain (in%) when using the results of two best models versus only the best model

EDA, taking into account that the worsening of the precision is also observed. It can be seen that switching to another model does not improve the plausibility.



(*a*) Best model



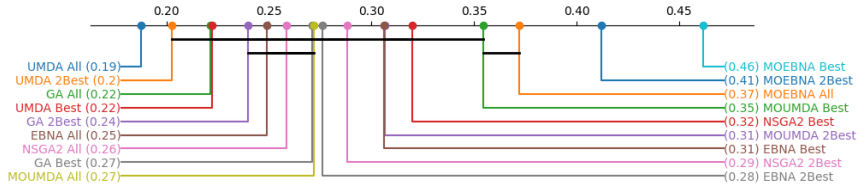
(*b*) Two best models

Figure 2: Results from all executions in the `Tic-tac-toe` dataset using the best accuracy model (*a*) and the two best models (*b*)

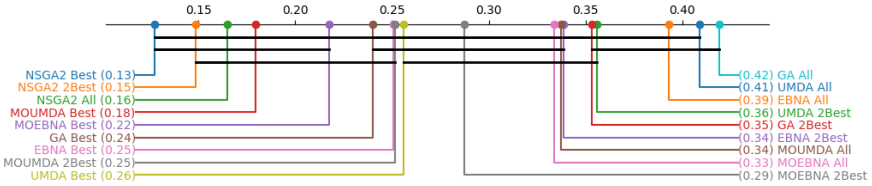
4.5. Results with all models

By having two models it is possible to obtain an improvement in distance without significantly worsening the rest of the objectives, so it is worth checking what happens if all five Bayesian classifiers available are used. To check this we will analyse how the different algorithms compare with the best model, two best models and all models. Figure 3 shows the comparison in distance, prediction and average plausibility over all datasets. The values on the axis indicate the average objective value obtained by the corresponding algorithm and models used over all the datasets, where smaller the better. The lines linking different results mean that they do not show a statistically significant difference, calculated using the Friedman test followed by the post-hoc Nemenyi test. Starting with the distance, Figure 3(*a*), it is possible to see how the best results are obtained by the single-objective

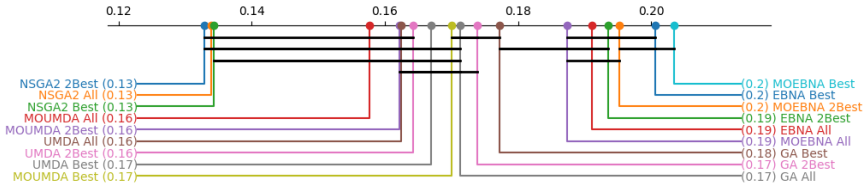
algorithms, where the best is the UMDA with all the models. The multi-objective versions come after, alternating between NSGA2 and MOUMDA, while MOEBNA are the last ones. In the case of prediction and plausibility, as expected, the multi-objective versions are ahead of the single-objective ones, and in these objectives not always having all the models improves the results. In prediction, Figure 3(b), NSGA2 obtains the best results followed by the EDAs with the best model, while in plausibility, Figure 3(c), the gap between NSGA2 and EDAs is more remarkable, but all results are really close for this objective. Note that in all objectives the version with the two best models is better or it is close to the results obtained by all models, so adding these models does not provide significant improvement given that they add execution time.



(a) Distance comparison



(b) Prediction comparison



(c) Plausibility comparison

Figure 3: Critical difference diagrams of distance, prediction and plausibility

4.6. Counterfactual Example

This section includes an example of a counterfactual explanation computation to better understand how the different algorithms work. The dataset used is `Car evaluation`(Table 1), which consists of a dataset of car specifications and the output is how acceptable the car is, based on the specifications (unacceptable, acceptable, good, very good). The features consist of the price of the car, maintenance cost, number of doors, passenger capacity, boot capacity and safety. We start from the initial instance that appears at the beginning of Table 5, which correct prediction is unacceptable, and we search for the changes necessary to obtain acceptable as a prediction. In Table 5 the result with the best model is observed, in this particular case the k-dependence Bayesian classifier (with k=2). The features that

have not been modified in each algorithm are marked with a hyphen. The results show that a change in maintenance and safety are essential to change the prediction output. Moreover, some models obtain results with a smaller distance than others, the best ones being single-objective EDAs and NSGA2 since their distance is the smallest. Also, models with higher distance have more feature changes, so they are worse solutions although having similar prediction and plausibility values. On the other hand, in Table 6 the same counterfactuals can be seen but using the results from all models, showing in the first column which model(s) obtain the best solution. Note that there is no consistency between models, since all models appear except the semi-naive Bayes, due to the fact that there are few features. In this case it can be seen how practically all the algorithms obtain the same result, i.e., maintenance should be reduced to medium and safety to high.

Algorithm	price	maint	doors	persons	lugboot	safety	dist	prec	plau
initial	high	vhigh	5more	more	small	low	-	-	-
UMDA (KDB)	-	med	-	-	-	high	0.14	0.24	0.00
EBNA (KDB)	-	med	-	-	-	high	0.14	0.24	0.00
GA (KDB)	-	med	-	-	big	med	0.22	0.23	0.00
MOUMDA (KDB)	-	med	-	-	med	med	0.30	0.30	0.00
MOEBNA (KDB)	low	-	-	-	-	high	0.19	0.23	0.00
NSGA2 (KDB)	-	med	-	-	-	high	0.14	0.24	0.00

Table 5: Counterfactual explanation example obtained using the best model

Algorithm	price	maint	doors	persons	lugboot	safety	dist	prec	plau
initial	high	vhigh	5more	more	small	low	-	-	-
UMDA (KDB,TAN)	-	med	-	-	-	high	0.14	0.24	0.00
EBNA (KDB,TAN)	-	med	-	-	-	high	0.14	0.24	0.00
GA (TAN-HC)	med	-	-	-	-	high	0.14	0.22	0.05
MOUMDA (NB)	-	med	-	-	-	high	0.14	0.24	0.00
MOEBNA (TAN)	med	-	-	-	-	high	0.14	0.22	0.05
NSGA2 (KDB)	-	med	-	-	-	high	0.14	0.24	0.00

Table 6: Counterfactual explanation example obtained using all models

5. Conclusion

In this paper we have approached counterfactual explanations with estimation of distribution algorithms using Bayesian classifiers. We compared our single-objective and multi-objective solutions with the genetic algorithms counterparts. Regarding the number of models the best configuration in the experiments is to use the two best models regardless of which algorithm is being used and the NSGA2 algorithms obtain the best results on average taking into account distance, prediction and plausibility. However, if the priority is execution time, it is recommendable to use a single-objective UMDA, even if the results are slightly worse. MOUMDA obtains good results, very close to those obtained by NSGA2. Due to this, and the fact that it is slightly faster than NSGA2, sometimes you can take advantage of its use.

As for future work, it would be interesting to see how to improve MOEDAs to better deal with cases with few variables while maintaining the results in the other datasets. We

could also use other types of classification models or use continuous predictor variables, changing the distances to better suit the numerical space. Also the plausibility calculation could be improved with more robust and advanced methods. In addition, an alternative could be searched for when none of the classifiers is able to find a solution, either because no model is able to predict the input correctly or because the algorithms do not find a solution.

Acknowledgments

This paper was supported by the Spanish Ministry of Science and Innovation through the PID2022-139977NB-I00 project and TED2021-131310B-I00 projects and by the grant to the ELLIS Unit Madrid by the Autonomous Region of Madrid.

References

C. Bielza and P. Larrañaga. Discrete Bayesian network classifiers: A survey. *ACM Computing Surveys*, 47(1):1–43, 2014.

J. Blank and K. Deb. PYMOO: Multi-objective optimization in Python. *IEEE Access*, 8: 89497–89509, 2020.

S. Dandl, C. Molnar, M. Binder, and B. Bischl. Multi-objective counterfactual explanations. In *International Conference on Parallel Problem Solving from Nature*, pages 448–469. Springer, 2020.

K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. In *Proceeding of the 6th International Conference of Parallel Problem Solving from Nature VI*, pages 849–858. Springer, 2000.

A. Dhurandhar, P.-Y. Chen, R. Luss, C.-C. Tu, P. Ting, K. Shanmugam, and P. Das. Explanations based on the missing: Towards contrastive explanations with pertinent negatives. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

R. Dwivedi, D. Dave, H. Naik, S. Singhal, R. Omer, P. Patel, B. Qian, Z. Wen, T. Shah, G. Morgan, and R. Ranjan. Explainable AI (xAI): Core ideas, techniques, and solutions. *ACM Computing Surveys*, 55(9):1–33, 2023.

R. Etxeberria and P. Larrañaga. Global optimization using Bayesian networks. In *Proceedings of the 2nd Symposium on Artificial Intelligence*, 1999.

J. C. Gower. A general coefficient of similarity and some of its properties. *Biometrics*, pages 857–871, 1971.

R. Guidotti. Counterfactual explanations and how to find them: Literature review and benchmarking. *Data Mining and Knowledge Discovery*, pages 1–55, 2022.

M. Hauschild and M. Pelikan. An introduction and survey of estimation of distribution algorithms. *Swarm and Evolutionary Computation*, 1(3):111–128, 2011.

A. Holzinger, A. Saranti, C. Molnar, P. Biecek, and W. Samek. Explainable AI methods - a brief overview. In *International Workshop on Extending Explainable AI beyond Deep Models and Classifiers*, pages 13–38. Springer, 2022.

M. Kelly, R. Longjohn, and K. Nottingham. UCI machine learning repository, 2023. URL http://archive.ics.uci.edu/ml.

P. Larrañaga and J. A. Lozano, editors. *Estimation of Distribution Algorithms: A new Tool for Evolutionary Computation*. Kluwer Academic Publisher, 2002.

M. T. Lash, Q. Lin, N. Street, J. G. Robinson, and J. Ohlmann. Generalized inverse classification. In *Proceedings of the 2017 SIAM International Conference on Data Mining*, pages 162–170. SIAM, 2017.

P. Linardatos, V. Papastefanopoulos, and S. Kotsiantis. Explainable AI: A review of machine learning interpretability methods. *Entropy*, 23(1):18, 2020.

A. Lucic, H. Haned, and M. de Rijke. Why does my model fail? Contrastive local explanations for retail forecasting. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pages 90–98, 2020.

B. Mihaljević, C. Bielza, and P. Larrañaga. bnclassify: Learning Bayesian network classifiers. *The R Journal*, 10(2):455–468, 2018.

C. Molnar. *Interpretable Machine Learning*. Lulu.com, 2nd edition, 2022.

J. Moore, N. Hammerla, and C. Watkins. Explaining deep learning models with constrained adversarial examples. In *Procedings of the 16th Pacific Rim International Conference on Artificial Intelligence Part I 16*, pages 43–56. Springer, 2019.

R. K. Mothilal, A. Sharma, and C. Tan. Explaining machine learning classifiers through diverse counterfactual explanations. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pages 607–617, 2020.

H. Mühlenbein and G. Paass. From recombination of genes to the estimation of distributions I. Binary parameters. In *International Conference on Parallel Problem Solving from Nature*, pages 178–187. Springer, 1996.

V. P. Soloviev, P. Larrañaga, and C. Bielza. Edaspy: An extensible python package for estimation of distribution algorithms. *Neurocomputing*, page 128043, 2024.

J. Vanschoren, J. N. Van Rijn, B. Bischl, and L. Torgo. Openml: Networked science in machine learning. *ACM SIGKDD Explorations Newsletter*, 15(2):49–60, 2014.

S. Verma, J. Dickerson, and K. Hines. Counterfactual explanations for machine learning: A review. *arXiv preprint arXiv:2010.10596*, 2, 2020.

S. Wachter, B. Mittelstadt, and C. Russell. Counterfactual explanations without opening the black box: Automated decisions and the GDPR. *Harvard Journal of Law & Technology*, 31:841, 2017.