Fair Feature Selection with a Lexicographic Multi-Objective Genetic Algorithm

James Brookhouse[0000−0002−9802−7070] and Alex Freitas[0000−0001−9825−4700]

School of Computing, University of Kent, Canterbury, UK
james@brookhou.se, A.A.Freitas@kent.ac.uk

Abstract. There is growing interest in learning from data classifiers whose predictions are both accurate and fair, avoiding discrimination against sub-groups of people based e.g. on gender or race. This paper proposes a new Lexicographic multi-objective Genetic Algorithm for Fair Feature Selection (LGAFFS). LGAFFS selects a subset of relevant features which is optimised for a given classification algorithm, by simultaneously optimising one measure of accuracy and four measures of fairness. This is achieved by using a lexicographic multi-objective optimisation approach where the objective of optimising accuracy has higher priority over the objective of optimising the four fairness measures. LGAFFS was used to select features in a pre-processing phase for a random forest algorithm. The experiments compared LGAFFS' performance against two feature selection approaches: (a) the baseline approach of letting the random forest algorithm use all features, i.e. no feature selection in a pre-processing phase; and (b) a Sequential Forward Selection method. The results showed that LGAFFS significantly improved fairness measures in several cases, with no significant difference regarding predictive accuracy, across all experiments.

1 Introduction

Recently, there has been an increased focus on the fairness of the decisions made by automated processes [1,2]; since algorithms that learn from biased data often produce biased predictive models. We address fairness in the classification task of machine learning, where a predictive feature (e.g. gender or race) is set as a sensitive feature. The values of a sensitive feature are used to split individuals (instances in a dataset) into protected and unprotected groups. The protected group contains individuals likely to be victims of discrimination, who are more likely to obtain a negative outcome (class label) than the unprotected group.

A large number of fairness measures have been proposed to capture some notion of fairness in a model learned from data [3,4]. These fairness measures can be categorised into group-level and individual-level fairness measures.

An example of a group-level fairness metrics is the discrimination score [5], which measures the difference between the predicted positive-class probabilities of the protected and unprotected groups. Some group-level metrics of fairness measure the difference between the false positive error rate and/or the false negative error rate between the protected and unprotected groups [6]. Group-level fairness measures have the limitation of not considering fairness at the individual level; i.e., they do not penalise models where two very similar individuals within the same group unfairly receive different outcomes (class labels).

An individual-level fairness metric avoids this limitation, by measuring similarities among individuals. Consistency is an individual fairness metric which compares an individual to its k-nearest neighbours; if all of an individual's neighbours have the same class as the current individual, this test is considered maximally satisfied for that individual, this is then repeated for each individual and an average taken [7]. However, as the number of features grows, the notion of "nearest neighbours" become increasingly meaningless, as the distances between individuals tend to increase, leading to comparisons being made between increasingly different individuals.

In practice, no single fairness measure can be deemed the best in general, and it has also been proved that there is a clear trade-off among some fairness measures, which cannot be simultaneously optimised [6,8].

Hence, intuitively it makes sense to use multiple fairness measures, with different pros and cons, and try to optimise those multiple measures at the same time, in order to achieve more robust fairness results. This is precisely the focus of this paper, where we propose a new multi-objective Genetic Algorithm (GA) for fair feature selection, The GA uses the lexicographic approach to optimise two objectives in decreasing priority order: predictive accuracy and fairness. The accuracy objective involves one measure, but the fairness objective is more complex and involves four measures. Hence, we propose a new procedure for aggregating four fairness measures into a single fairness objective by systematically considering all permutations of lexicographic ordering of those four measures, as described in detail later.

The GA selects a subset of relevant features for a given classification algorithm in a data pre-processing phase [9]. This is a difficult task for two reasons. First, the search space's size is exponential in the number of features, with 2^{n} – 1 candidate solutions (feature subsets), where n is the number of features in the dataset (the "– 1" discounts the empty feature subset). Second, intuitively the search space is rugged (highly non-convex) with many local optima, even in a single-objective scenario, with the problem being aggravated in the multi-objective scenario.

We focus on GAs for two mains reasons. First, they are robust global search methods, being less likely to get trapped into local optima in the search space, by comparison with conventional local search methods [10,11], and so they tend to cope better with feature interaction (a key issue in feature selection). Second, the fact that they evolve a population of candidate solutions facilitates multi-objective optimisation [12,13], as proposed in this work.

This paper is organised as follows. Section 2 describes the proposed multi-objective genetic algorithm for fair feature selection. Section 3 describes the datasets used in the experiments and the experimental setup. Section 4 reports experimental results and Section 5 presents the conclusions and future work.

Algorithm 1: Ramped Population Initialisation

```
  Data: population_size, MIN_P, MAX_P
  Result: Population of Individuals
1 Function initialise_population():
2     step_size = (MAX_P − MIN_P) / population_size
3     for i to population_size do
4         p = MIN_P + (i * step_size)
5     population += Individual.initialise(p)
6     return population
```

2 A Lexicographic-Optimisation Genetic Algorithm for Fair Feature Selection

This section describes our new Lexicographic-optimisation Genetic Algorithm for Fair Feature Selection (LGAFFS), which selects a subset of relevant features for a classification algorithm in a data pre-processing phase. LGAFFS selects individuals for reproduction based on the principle of lexicographic optimisation to combine predictive accuracy and fairness measures, as described later.

In LGAFFS, each individual of the population represents a candidate feature subset. More precisely, each individual consists of a string of N bits (genes), where N is the number of features in the dataset, and the i-th gene takes the value 1 or 0 to indicate whether or not (respectively) the i-th feature is selected.

LGAFFS follows a wrapper approach to feature selection [9], where a base classification algorithm is used to learn a classification model based on the feature subset selected by an individual, and that model's quality (in terms of accuracy and fairness) is used to compute that individual's fitness. Hence, the GA aims at finding the best subset of features for the base classification algorithm. Fitness computation is performed by using a well-known internal cross-validation procedure, which uses only the training set (i.e. not using the test set).

LGAFFS uses uniform crossover and bit-flip mutation as genetic operators to generate new individuals in each generation. However, the population initialisation, tournament selection and elitism selection are non-standard procedures, and hence these are described in detail in the next subsections.

2.1 Population Initialisation

When creating the initial population, each individual has a different probability that each gene (feature) will be selected or not. This ramping initialisation is described in Algorithm 1. As shown in line 4, each individual has the probability (p) that a gene (feature) will be switched on increased by `step_size` compared to the previous individual, where `step_size` is defined in line 2 as a function of the maximum and minimum probabilities for a feature to be selected – denoted `MAX_P` and `MIN_P`, which are input arguments for Algorithm 1.

The motivation for this ramped population initialisation procedure is to promote diversity in the population. If all individuals had the same probability p

Algorithm 2: Pseudo-code of Lexicographic Tournament selection.

```
Data: Instances, Population, $\epsilon$, fair_win_$\epsilon$
Function tournament_selection():
    i1, i2 = select_random_individuals()
    if not |i1.accuracy - i2.accuracy| > $\epsilon$ then
        i1_win, i2_win = fairness_aggregation(i1,i2) // See Algorithm 3
        if |i1_win - i2_win| > fair_win_$\epsilon$ then
            return fairest_individual
    return best_accuracy_individual
```

that a single gene is switched on or off, then, as the number of genes (features) in an individual increases, the number of features selected (switched on genes) in each individual would tend to converge to $p \times N_{genes}$, the mean of a binomial distribution, where p is the probability of each gene being switched on and N_{genes} is the number of genes. Hence, all individuals would tend to have a similar number of selected features, and individuals with low or high numbers of selected features in the initial population would be rare, limiting the search of these areas. The ramped population initialisation avoids this problem, giving each individual a different probability p of switching a gene, sweeping from `MIN_P` to `MAX_P`.

2.2 Lexicographic Tournament Selection

Lexicographic tournament selection, with tournament size of two, is used to select individuals for reproduction. In the lexicographic-optimisation approach [14], we compare the two individuals in a tournament considering the objectives in decreasing order of priority. Let V_1 and V_2 denote the values of the current objective for individuals 1 and 2. When those two individuals are compared based on the first objective, if $|V_1 - V_2| > \epsilon$ (where ϵ is a very small value), then the best individual is the tournament winner. Otherwise, the two individuals' objective values are deemed equivalent (negligible difference) based on that objective; then the next objective is considered in the same way, and so on. This is repeated until a significant (greater than ϵ) difference is observed and a best individual selected. If there is no significant difference between two individuals for all objectives, then the individual with the best value of the highest priority objective is selected.

The pseudo-code for this is shown in Algorithm 2. When comparing two individuals, the lexicographic approach requires the objectives to be ordered. LGAFFS considers accuracy as the highest-priority objective to be optimised, followed by a lower-priority set of fairness measures which are aggregated into a single fairness objective to be optimised as described in section 2.4.

Note that the lexicographic method avoids the specification of ad-hoc weights to each objective, which would be the case if using a weighted sum of objectives [14]. The lexicographic approach simply requires that an order of priority for the objectives be defined; and intuitively it is easier for users to specify a priority order of objectives than ad-hoc numerical weights. The lexicographic approach requires a small threshold parameter (ϵ); but again, it is intuitively easier for users to specify this parameter than to specify ad-hoc weights for each objective.

Note that an alternative to the lexicographic approach would be the well-known Pareto dominance approach [12]. However, the Pareto approach is not suitable for our feature selection task where the objective of accuracy has higher priority than the objective of fairness, since the Pareto approach ignores this objective prioritisation. In particular, if we used the Pareto approach, once the fairest model is found it would tend to be preserved by the selection operator and remain in the Pareto front along the GA run even if its accuracy was very low; but that model would be a bad solution, given the objectives' priority order. In this case, the GA would waste computational resources searching on areas of the Pareto front around bad solutions (like areas with maximal fairness but low accuracy). In contrast, a lexicographic approach would never select that fairest model due to its very low accuracy (as the highest-priority objective).

2.3 The Four Fairness Measures and the Accuracy Measure

No single fairness measure captures all nuances of a fair model, so LGAFFS optimises four fairness measures to get more robust fairness results. To define these measures we use the following nomenclature:

- S: Protected/sensitive feature: 0 → unprotected group, 1 → protected group
- $\hat{Y}$: the predicted class; Y: the actual class; taking class labels 1 or 0
- TP, FP, TN, FN: Number of True Positives, False Positives, True Negatives and False Negatives, respectively

The first measure is the discrimination score (DS) [5], which is defined as:

$$DS = 1 - \left| P(\hat{Y} = 1 | S = 0) - P(\hat{Y} = 1 | S = 1) \right| \tag{1}$$

DS is a group-level fairness measure that takes the optimal value of 1 if both protected and unprotected groups have an equal probability of being assigned to the positive class by the classifier. If DS is used on unbalanced datasets, those where the data shows a large difference between the probability of a positive outcome for both groups, to satisfy DS will require a reduction in accuracy. In this case the lexicographic approach is robust to such selective pressures as the ordering of the objectives prioritises accuracy over the fairness measures.

The second measure used is consistency[7], defined as:

$$C = 1 - \frac{1}{Nk} \sum_{i} \sum_{j \in kNN(x_n)} |\hat{y}_i - \hat{y}_j| \tag{2}$$

Consistency is an individual-level similarity metric that compares the class predicted by a classifier to each instance in the dataset to the class predicted by the classifier to that instance's k nearest instances (neighbours) in the dataset. If all these neighbours have the same predicted class as the current instance, then that instance is considered consistent. The measure computes the average degree of consistency over all instances in the dataset. A fully consistent model has a consistency of 1 and an inconsistent model has a value of 0.

Thirdly, the False Positive Error Rate Balance Score (FPERBS) [6,15] is:

$$FPERBS = 1 - \left| \frac{FP_{S=0}}{FP_{S=0} + TN_{S=0}} - \frac{FP_{S=1}}{FP_{S=1} + TN_{S=1}} \right| \tag{3}$$

FPERBS measures the difference in the probability that a truly negative instance is incorrectly predicted as positive between protected and unprotected groups. Fourthly, the False Negative Error Rate Balance Score (FNERBS) [6,16,17] measures the difference in the probability that a truly positive instance is incorrectly predicted as negative between protected and unprotected groups:

$$FNERBS = 1 - \left| \frac{FN_{S=0}}{FN_{S=0} + TP_{S=0}} - \frac{FN_{S=1}}{FN_{S=1} + TP_{S=1}} \right| \tag{4}$$

A score of 1 indicates an optimally fair result for both FPERBS and FNERBS.

As the accuracy measure to be optimised, LGAFFS uses the geometric mean of Sensitivity and Specificity (Equation 5). This measure was chosen because it incentivises the correct classification of both positive-class and negative-class instances, to counteract pressure from the fearness measures to produce maximally fair models that trivially predict the same class for all instances.

$$\begin{gathered} Sensitivity = \frac{TP}{TP + TN}, \qquad Specificity = \frac{TN}{TN + FP}, \\ GM_{Sen \times Spec} = \sqrt{Sensitivity \cdot Specificity} \end{gathered} \tag{5}$$

2.4 Aggregating fairness measures

As discussed earlier, LGAFFS optimises one accuracy measure and four fairness measures. We consider accuracy as the highest-priority objective (as usual in machine learning), and the four fairness measures as lower-priority objectives. Among those fairness measures, there is no consensus in the literature about what is the best one, and so it would be "unfair" to prioritise one fairness measure over the others. Hence, we aggregate the four fairness measures into a single objective to be optimised by the GA (in addition to the accuracy objective), by computing all possible 24 (4!) permutations of the four fairness measures.

Algorithm 3 shows how two individuals are compared regarding fairness. Each permutation defines a lexicographic order for the fairness measures which can be evaluated to find the first significant difference between the individuals, at which point the best individual is given a win. A significant difference is one greater than the very small ϵ. After all permutations have been evaluated, the individual with the higher number of wins is declared the best individual overall.

Algorithm 3: Aggregating fairness measures.

```
Data: Ind_1, Ind_2, $\epsilon$
Result: Number of wins for each individual
Function fairness_aggregation():
    i1_win = i2_win = 0
    permutations = generate_permutations(measures)
    forall permutations do
        forall permutation.measures do
            i1, i2 = compute_fairness_measure(measure, Ind_1, Ind_2)
            if $|i1 - i2| > \epsilon$ then
                if $i1 > i2$ then
                    i1_win++
                    break // Exit inner forall
                else
                    i2_win++
                    break // Exit inner forall
    return i1_win, i2_win
```

2.5 Lexicographic elitism

Recall that the lexicographic approach requires the ranking of objectives, and our GA prioritises the accuracy objective (the geometric mean of Sensitivity and Specificity) over the four fairness measures. The fairness measures are aggregated into a single objective (see Section 2.4). To find the `best individual` the procedure in Algorithm 4 is used for implementing elitism.

First the population is sorted by accuracy (line 2 of Algorithm 4), where any individuals with accuracy within ϵ of the most accurate individual are shortlisted for fairness comparison (line 4). These shortlisted individuals have their average rank of fairness computed across all 24 permutations generated as described in Section 2.4. For each permutation of the four fairness measures, the set of

Algorithm 4: Lexicographic Selection of the Best Individual

```
  Data: population, $\epsilon$, fair_rank_$\epsilon$
  Result: Best individual
1 Function get_best_individual():
2     accuracy_rank = sort_population_by_accuracy(population)
3     best_accur_indiv = accuracy_rank.head()
4     individuals = select_all_individuals_within_accuracy_$\epsilon$(accuracy_rank)
5     avg_fair_rank = average_rank_of_fairness_permutations(individuals)
6     if (avg_fair_rank.head().average_rank − best_accur_indiv.average_rank) >
        fair_rank_$\epsilon$ then
7         return avg_fair_rank.head()
8     else
9         return best_accur_indiv
```

Table 1. Datasets used in all experiments, detailing the number of instances, features and the sensitive features for each dataset.

Data set	Instances	Features	Sensitive Features
Adult Income (US Census)	48842	14	Race, Gender, Age
German Credit	1000	20	Age, Gender
Credit Card Default	30000	24	Gender
Communities and Crime	1994	128	Race
Student Performance (Portuguese)	650	30	Age, Gender, Relationship
Student Performance (Maths)	396	30	Age, Gender, Relationship
ProPublica recidivism	6167	52	Race, Gender

shortlisted individuals is arranged by its lexicographic order, where the first (last) measure in the permutation is considered the most (least) important. Fairness values within the threshold ϵ are considered equivalent and the less important metrics are considered until a significant difference is found.

If the fairest shortlisted individual (with the lowest average rank) has a significantly better rank than the most accurate shortlisted individual (i.e. the difference between their average ranks is greater than fair_rank_ϵ), the former is selected by elitism as the best individual in preference over the most accurate individual – since the difference in accuracy between those two shortlisted individuals is considered non-significant, i.e., within ϵ. Otherwise, the most accurate shortlisted individual is selected by elitism.

LGAFFS' Python code is available at `https://github.com/bunu/LGAFFS`.

2.6 Related Work

Quadrianto et al. [18] and Valvidia et al. [19] proposed a GA for fair classification. Both GAs were designed for optimising (hyper)-parameters of a classification algorithm, rather than feature selection; and both GAs use Pareto dominance rather than the lexicographic approach used here. The Pareto approach is sound in general, but as noted earlier, it is not suitable for our feature selection task prioritising accuracy over fairness. La Cava and Moore [20] proposed genetic programming (GP) for feature construction, which can implicitly perform feature selection, but feature construction has a much larger search space than feature selection. Their GP uses lexicase selection, a broadly lexicographic approach. However, instead of ordering the objectives based on user-defined priorities like in LGAFFS; their GP uses *randomised* lexicographic orderings of different subgroups of instances (with different sensitive feature values). The GP evaluates multiple fairness-violation events, each for a different subgroup of instances; but each event is evaluated by the same fairness formula: the difference of error rates (either FP or FN error rates) between all instances and a sub-group of instances. In addition, unlike those three algorithms, LGAFFS combines group-level and individual-level fairness measures, increasing fairness robustness.

Table 2. Results for Research Question 1: Comparing the performance of random forest trained with the features selected by LGAFFS in a pre-processing phase against the performance of random forest trained with all features. Showing the values for all five measures being optimised by LGAFFS.

	Sensitive	$GM_{Sen \times Spec}$		Discrimination Score		Consistency		FPERBS		FNERBS	
Dataset	Feature	LGAFFS	All feats	LGAFFS	All feats	LGAFFS	All feats	LGAFFS	All feats	LGAFFS	All feats
Adult	Age	0.6475	**0.7602**	**0.8485**	0.7557	**0.8656**	0.7887	**0.9522**	0.9084	**0.9189**	0.6416
Adult	Race	0.7409	**0.7632**	**0.9356**	0.8955	**0.8201**	0.7889	**0.9862**	0.9589	**0.9902**	0.9004
Adult	Sex	0.7420	**0.7623**	**0.8498**	0.8152	**0.8163**	0.7877	**0.9490**	0.9217	**0.9416**	0.9110
German Credit	Age	**0.5901**	0.5774	**0.9361**	0.8394	0.7642	**0.8032**	**0.8633**	0.7626	0.8911	**0.8968**
German Credit	Gender	**0.6036**	0.5637	**0.9399**	0.9087	0.7510	**0.8114**	**0.8993**	0.8489	0.9230	**0.9349**
Student Maths	Age	**0.9208**	0.9046	0.7964	**0.8169**	0.8444	**0.8483**	**0.9022**	0.8781	0.8951	**0.8974**
Student Maths	Dalc	0.8951	**0.9072**	**0.7563**	0.7214	0.8377	**0.8447**	0.8051	**0.8312**	0.8157	**0.8160**
Student Maths	Famrel	**0.9052**	0.8914	0.7039	**0.7148**	0.8361	**0.8412**	**0.8760**	0.8480	**0.9222**	0.9189
Student Maths	Romantic	0.8984	**0.9008**	**0.9076**	0.9055	**0.8468**	0.8443	**0.9151**	0.9090	**0.9106**	0.8989
Student Maths	Sex	**0.9027**	0.8977	0.8397	**0.8652**	0.8387	**0.8488**	0.7646	**0.8233**	0.9012	**0.9251**
Student Maths	Walc	0.9000	**0.9012**	**0.8364**	0.8023	0.8376	**0.8503**	0.8206	**0.8362**	**0.9196**	0.8741
Student Portuguese	Age	**0.8196**	0.7864	**0.8638**	0.8536	0.9106	**0.9192**	**0.7038**	0.6686	**0.9578**	0.9428
Student Portuguese	Dalc	**0.7867**	0.7834	0.8470	**0.8758**	0.9128	**0.9251**	**0.6230**	0.5511	0.9088	**0.9376**
Student Portuguese	Famrel	0.8031	**0.8035**	0.8019	**0.8305**	0.9097	**0.9177**	**0.6450**	0.6286	0.9205	**0.9583**
Student Portuguese	Romantic	**0.7846**	0.7825	0.9370	**0.9452**	**0.9211**	0.9189	0.7608	**0.7775**	0.9686	**0.9796**
Student Portuguese	Sex	**0.8110**	0.7994	0.9200	**0.9313**	0.9134	**0.9239**	**0.7646**	0.7029	**0.9708**	0.9650
Student Portuguese	Walc	**0.8035**	0.7831	0.9271	**0.9380**	0.9186	**0.9205**	0.7214	**0.7248**	**0.9679**	0.9671
Communities & Crime	Race	0.8303	**0.8419**	**0.6623**	0.5957	0.6056	**0.6126**	**0.9182**	0.8553	**0.8313**	0.7762
Default of Credit	Sex	0.5896	**0.5910**	**0.9755**	0.9724	0.8354	**0.8395**	0.9739	**0.9776**	**0.9852**	0.9832
Propublica Recidivisim	Race	0.7306	**0.7515**	**0.8324**	0.7744	**0.6849**	0.6836	**0.9022**	0.8169	**0.9105**	0.8834
Propublica Recidivisim	Sex	0.7113	**0.7518**	**0.9408**	0.9189	0.6756	**0.6876**	**0.9254**	0.9125	**0.9451**	0.9400
Number of Wins		10	11	13	8	6	15	15	6	13	8
Wilcoxon Signed-Rank Test		0.9442		0.06288		0.07346		▲ **0.0088**		0.18024	

3 Datasets and Experimental Setup

Table 1 describes the 7 binary classification datasets used. When a dataset has multiple sensitive features – a sensitive feature is one which represents a protected characteristic and/or a group that is unfairly treated – the algorithm is ran multiple times using a different sensitive feature each time. 6 datasets are from the UCI Machine Learning repository [21]. The 7th dataset is from ProPublica, investigating biases in predicting if criminals would re-offend [2].

For all datasets, except Adult Income, the experiments use a well-known 10-fold cross-validation procedure. Adult Income is already partitioned into a training and test set, so this partition is used instead of cross validation. LGAFFS' parameters were not optimised and were set as follows: ϵ: 0.01 (threshold for significant differences in Algorithms 2, 3 and 4), fair_rank_ϵ: 1 (threshold for significant fairness-rank differences in Algorithm 4), fair_win_ϵ: 1 (threshold for significant difference in the number of wins among 24 permutations of fairness measures in Algorithm 2); population_size: 100, MAX_P: 0.5 and MIN_P: 0.1 (for population initialisation in Algorithm 1), internal cross validation folds: 3, max_iterations: 50, tournament_size: 2, crossover_probability: 0.9, mutation_probability: 0.05.

4 Experimental Results

We addresses two research questions. First, we compare the use of LGAFFS to select features in a pre-processing phase against the baseline of no feature selection in that phase. Second, we compare LGAFFS to the popular Sequential Forward Selection (SFS) method. Both LGAFFS and SFS use the wrapper approach to feature selection; i.e., they repeatedly use a base classification algorithm to evaluate feature subsets. The base algorithm was Random Forest from scikit-learn [22], with default parameter settings; which was chosen because it is a very popular and powerful classification algorithm. Note that the Random Forest algorithm performs embedded feature selection (during its run), but that feature selection considers only accuracy; whilst using LGAFFS to perform feature selection in a pre-processing phase we optimise both accuracy and fairness.

We also calculated the Pearson's linear correlation coefficient for each of the 6 pairs of fairness measures for LGAFFS, the coefficients were: 0.71 for (DS,FNERBS), -0.59 for (C,FPERBS), 0.42 for (C,FNERBS), 0.24 for (DS,C), 0.08 for (DS,FPERBS) and 0.02 for (FPERBS,FNERBS). So, 4 pairs of fairness measures have an absolute value of correlation smaller than 0.5.

4.1 RQ1: Does LGAFFS select a better subset than the full set?

The first research question asks whether using LGAFFS to select features in a pre-processing phase leads to better results than the baseline approach of not performing any feature selection. That is, does the random forest algorithm perform better (regarding accuracy and fairness) when it is trained with the features selected by LGAFFS or when it is trained with the full feature set?

Table 3. Results for Research Question 2: Comparing the performance of random forest trained with the features selected by LGAFFS vs. Sequential Forward Selection (both selecting features in a pre-processing phase). Showing the values for all five measures being optimised by LGAFFS.

	Sensitive	$GM_{Sen \times Spec}$		Discrimination Score		Consistency		FPERBS		FNERBS	
Dataset	Feature	LGAFFS	SFS	LGAFFS	SFS	LGAFFS	SFS	LGAFFS	SFS	LGAFFS	SFS
Adult	Age	0.6475	**0.7482**	**0.8485**	0.7968	**0.8656**	0.8142	**0.9522**	0.9437	**0.9189**	0.8306
Adult	Race	0.7409	**0.7465**	0.9356	**0.9358**	**0.8201**	0.8149	0.9862	**0.9887**	**0.9902**	0.9444
Adult	Sex	0.7420	**0.7480**	**0.8498**	0.8331	**0.8163**	0.8145	**0.9490**	0.9396	**0.9416**	0.9057
German Credit	Age	**0.5901**	0.4653	**0.9361**	0.8872	0.7642	**0.8110**	**0.8633**	0.8095	**0.8911**	0.8890
German Credit	Gender	**0.6036**	0.4851	**0.9399**	0.9237	0.7510	**0.8212**	**0.8993**	0.8457	**0.9230**	0.9202
Student Maths	Age	**0.9208**	0.9041	**0.7964**	0.7954	**0.8444**	0.8407	**0.9022**	0.8468	**0.8951**	0.8619
Student Maths	Dalc	**0.8951**	0.8921	**0.7563**	0.6893	0.8377	**0.8392**	**0.8051**	0.7940	0.8157	**0.8177**
Student Maths	Famrel	0.9052	**0.9069**	**0.7039**	0.6707	0.8361	**0.8367**	**0.8760**	0.7993	**0.9222**	0.9096
Student Maths	Romantic	0.8984	**0.9033**	**0.9076**	0.9008	**0.8468**	0.8311	**0.9151**	0.8906	0.9106	**0.9170**
Student Maths	Sex	0.9027	**0.9265**	0.8397	**0.8468**	**0.8387**	0.8361	0.7646	**0.8602**	0.9012	**0.9521**
Student Maths	Walc	0.9000	**0.9084**	**0.8364**	0.8151	**0.8376**	0.8347	0.8206	**0.8796**	**0.9196**	0.9026
Student Portuguese	Age	**0.8196**	0.7812	0.8638	**0.8909**	**0.9106**	0.9050	0.7038	**0.7445**	**0.9578**	0.9576
Student Portuguese	Dalc	0.7867	**0.7917**	0.8470	**0.8580**	**0.9128**	0.9090	**0.6230**	0.6138	0.9088	**0.9460**
Student Portuguese	Famrel	**0.8031**	0.7946	**0.8019**	0.7392	**0.9097**	0.9050	**0.6450**	0.5911	**0.9205**	0.8538
Student Portuguese	Romantic	**0.7846**	0.7833	0.9370	**0.9378**	**0.9211**	0.9062	**0.7608**	0.7273	0.9686	**0.9716**
Student Portuguese	Sex	**0.8110**	0.8017	**0.9200**	0.9148	**0.9134**	0.9032	**0.7646**	0.7504	**0.9708**	0.9554
Student Portuguese	Walc	**0.8035**	0.7845	**0.9271**	0.9179	**0.9186**	0.9087	0.7214	**0.7393**	**0.9679**	0.9541
Communities and Crime	Race	**0.8303**	0.7923	**0.6623**	0.5740	**0.6056**	0.6049	**0.9182**	0.7935	**0.8313**	0.6992
Default of Credit	Sex	**0.5896**	0.5639	0.9755	**0.9816**	0.8354	**0.8580**	0.9739	**0.9807**	0.9852	**0.9889**
Propublica Recidivisim	Race	**0.7306**	0.7200	**0.8324**	0.7763	**0.6849**	0.6617	**0.9022**	0.8026	**0.9105**	0.8769
Propublica Recidivisim	Sex	0.7113	**0.7199**	**0.9408**	0.8660	**0.6756**	0.6661	**0.9254**	0.8492	**0.9451**	0.9058
Number of Wins		12	9	15	6	16	5	15	6	15	6
Wilcoxon Signed-Rank Test		0.17068		▲ **0.00634**		0.05876		▲ **0.04236**		▲ **0.030**	

Table 2 show the experimental results for this research question. In this table, the first two columns show the dataset and the sensitive feature. The following ten columns show the accuracy and fairness results of training the Random Forest algorithm with features selected by LGAFFS or with the full feature set.

In each row of this table (i.e. for each pair of a dataset and a sensitive feature), for each pair of columns comparing the accuracy or fairness of LGAFFS against the full feature set, the best result is shown in boldface. The last but one row of the table shows the number of wins for each approach for each of the five measures of performance, whilst the last row shows the p-value obtained by the Wilcoxon signed-rank statistical significance test. Statistically significant results, at the conventional significance level of $\alpha = 0.05$, are marked with a red triangle. In Table 2, there was no substantial difference in the number of wins regarding accuracy. LGAFFS achieved substantially more wins in three of the four fairness measures, with statistical significance in one measure: FPERBS.

4.2 RQ2: Does LGAFFS perform better than SFS?

The second question involves the comparison of LGAFFS to a popular local search-based feature selection method, viz. Sequential Forward Selection (SFS). The SFS method is not aware of the 4 fairness measures; it is just optimising the accuracy measure, i.e., the geometric mean of sensitivity and specificity.

Table 3 presents the results for the Random Forest algorithm when using LGAFFS or SFS to select features. LGAFFS achieved more wins in all 5 measures, with statistical significance shown in 3 of the 4 fairness measures.

5 Conclusions

We have proposed a new lexicographic-optimisation Genetic Algorithm for fair feature selection, which selects a feature subset optimised for a classification algorithm based on both predictive accuracy and 4 fairness measures capturing different aspects of fairness, including both group-level and individual-level fairness. No single fairness measure reflects all the nuances of fairness; LGAFFS optimises multiple fairness measures to obtain more robust fairness results.

LGAFFS was compared with 2 other feature selection approaches (no feature selection and Sequential Forward Selection) using Random Forest as the classification algorithm. There was no significant difference in the predictive accuracies of models learned when using LGAFFS versus the 2 other approaches. Regarding fairness, when comparing LGAFFS against the 2 other approaches across the 4 fairness measures, LGAFFS achieved significantly better results in 4 of the 8 comparisons, and there was no significant differences between LGAFFS and the 2 other approaches in the other 4 comparisons.

Future work could include extending SFS to make it a fairness-aware method.

Acknowledgements: This work was funded by a research grant from The Leverhulme Trust, UK, reference number RPG-2020-145.

References

1. J. L. Skeem and C. T. Lowenkamp, "Risk, race, & recidivism: Predictive bias and disparate impact.(2016)," *Criminology*, vol. 54, p. 680, 2016.
2. J. Angwin, J. Larson, S. Mattu, and L. Kirchner. (2016) Machine bias: There's software used across the country to predict future criminals. and it's biased against blacks. [Online]. Available: https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing
3. N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan, "A survey on bias and fairness in machine learning," *arXiv preprint arXiv:1908.09635*, 2019.
4. S. Verma and J. Rubin, "Fairness definitions explained," in *2018 IEEE/ACM International Workshop on Software Fairness (FairWare)*. IEEE, 2018, pp. 1–7.
5. T. Calders and S. Verwer, "Three naive bayes approaches for discrimination-free classification," *Data Mining and Knowledge Discovery*, vol. 21, no. 2, pp. 277–292, 2010.
6. A. Chouldechova, "Fair prediction with disparate impact: A study of bias in recidivism prediction instruments," *Big data*, vol. 5, no. 2, pp. 153–163, 2017.
7. R. Zemel, Y. Wu, K. Swersky, T. Pitassi, and C. Dwork, "Learning fair representations," in *International Conference on Machine Learning*, 2013, pp. 325–333.
8. J. Kleinberg, S. Mullainathan, and M. Raghavan, "Inherent trade-offs in the fair determination of risk scores," *arXiv preprint arXiv:1609.05807*, 2016.
9. J. Li, K. Cheng, S. Wang, F. Morstatter, R. Trevino, J. Tang, and H. Liu, "Feature selection: a data perspective," *ACM Computing Surveys*, vol. 50, no. 6, pp. 94:1–94:45, 2017.
10. A. Telikani, A. Tahmassebi, W. Banzhaf, and A. Gandomi, "Evolutionary machine learning: a survey," *ACM Computing Surveys*, vol. 54, no. 8, pp. 161:1–161:35, 2021.
11. A. Freitas, *Data Mining and Knowledge Discovery with Evolutionary Algorithms*. Springer, 2002.
12. K. Deb, *Multi-objective optimization using evolutionary algorithms*. John Wiley & Sons, 2002.
13. Y. Tian, L. Si, X. Zhang, R. Cheng, C. He, K. C. Tan, and Y. Jin, "Evolutionary large-scale multi-objective optimization: a survey," *ACM Computing Surveys*, vol. 54, no. 8, pp. 174:1–174:34, 2021.
14. A. A. Freitas, "A critical review of multi-objective optimization in data mining: a position paper," *ACM SIGKDD Explorations Newsletter*, vol. 6, no. 2, pp. 77–86, 2004.
15. S. Corbett-Davies and S. Goel, "The measure and mismeasure of fairness: A critical review of fair machine learning," *arXiv preprint arXiv:1808.00023*, 2018.
16. M. Hardt, E. Price, and N. Srebro, "Equality of opportunity in supervised learning," in *Advances in neural information processing systems*, 2016, pp. 3315–3323.
17. M. J. Kusner, J. Loftus, C. Russell, and R. Silva, "Counterfactual fairness," in *Advances in neural information processing systems*, 2017, pp. 4066–4076.
18. N. Quadrianto and V. Sharmanska, "Recycling privileged learning and distributed matching for fairness," in *Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017)*, 2017, pp. 677–688.
19. A. Valdivia, J. Sanchez-Monedero, and J. Casillas, "How fair can we go in machine learning? assessing the boundaries of accuracy and fairness," *International Journal of Intelligent Systems*, vol. 36, no. 4, pp. 1619–1643, 2021.
20. W. La Cava and J. Moore, "Genetic programming approaches to learning fair classifiers," in *Proc. Genetic and Evolutionary Computation Conference (GECCO-2020)*, 2020, pp. 967–975.

21. D. Dua and C. Graff, “UCI machine learning repository,” 2017. [Online]. Available: http://archive.ics.uci.edu/ml
22. F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, “Scikit-learn: Machine learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.