Exceptional Model Mining meets Multi-objective Optimization

Alexandre Millot * Rémy Cazabet † Jean-François Boulicaut*

Abstract

Exceptional Model Mining (EMM) is a local pattern mining framework that generalizes subgroup discovery. In EMM, we look for subsets of objects - subgroups - whose model deviates significantly from the same model fitted on the overall dataset. Multi-objective Optimization (MOO) is an area of Multiple Criteria Decision Making where two or more functions need to be optimized at the same time and the goal is to find the best compromise between the concurrent objectives. We introduce a new model class for EMM in a MOO setting called Exceptional Pareto Front Mining. We design fitting quality measures that take into account both the distance between models and the relevance of the subgroups. We propose a beam search for top-K EMM whose added-value is studied on both synthetic and real life datasets. Among others, we discuss a use case on hyperparameter optimization in machine learning for both regression and multi-label classification.

1 Introduction

Exceptional Model Mining (EMM) has been recently proposed [7]. It is a generalization of subgroup discovery [12, 13]. Given labeled data, subgroup discovery aims at discovering subsets of objects - subgroups - described by interesting descriptions or patterns according to a quality measure computed for the target variable. The measure has to capture deviations between the target variable distribution on the selected subset of objects and the distribution on the overall dataset. In EMM, we look for subgroups whose model deviates significantly from the same model fitted on the entire dataset. Where subgroup discovery is inherently limited to a unique target concept, EMM is able to handle data where two or more targets exist. It supports the discovery of more complex interactions between variables. An example of complex interactions between variables can be found in multi-objective optimization (MOO) [4]. It is part of the Multiple Criteria Decision Making area where two or more functions need to be optimized simultaneously. If an order of importance can be defined between the objectives, the problem can be investigated as uni-objective either by giving weights to each objective and then summing the weighted objective values or by optimizing the objectives one by one following a predefined order. However, when no order can be defined a priori, the use of a method based on Pareto optimization has to be investigated [5, 24]. It is based on the dominance between solutions of the objective space. A solution is said to be non-dominated - or Pareto optimal - if it is impossible to improve an objective without degrading another. The set of Pareto optimal solutions is known as the Pareto front. The result of a MOO algorithm then involves not one but a set of equal solutions - the Pareto front.

We introduce a new model class for EMM based on the discovery of exceptional Pareto fronts. We look for deviations in the shape of the Pareto front left by the absence of a subgroup of objects compared to the Pareto front computed on the overall dataset. We have to design new quality measures that take into account both the distance between Pareto fronts and the subgroup relevancy. We propose a beam search strategy to mine high quality exceptional Pareto fronts in an efficient way. We show the relevance of our approach on both synthetic and real life data. Among others, we discuss an application to hyperparameter optimization in machine learning.

This paper is organized as follows. Section 2 formalizes our mining task. In Section 3, we discuss related work. We detail our contribution in Section 4 before introducing our experimental results in Section 5. Finally, Section 6 concludes.

2 Preliminaries

2.1 Exceptional Model Mining. EMM is a generalization of subgroup discovery that can handle more than one target attribute by using model classes. In EMM, a dataset (G, M, T) is a set of objects G, a set of attributes M and a set of targets T. In a given dataset, the set of attributes M contains real and categorical attributes while the set of targets mainly depends on the model class at hand. For us, the domain of any target $t \in T$ is a finite set: when considering numerical values, only occurring values are part of the domain.

*Univ de Lyon, CNRS, INSA Lyon, LIRIS, UMR5205, F-69621 Villeurbanne, France. Email: firstname.lastname@insa-lyon.fr

†Univ de Lyon, CNRS, Université Lyon 1, LIRIS, UMR5205, F-69622 Villeurbanne, France. Email: firstname.lastname@univ-lyon1.fr

DEFINITION 2.1. *A subgroup p can be described either by its intent - the description of the subgroup in terms of attribute values - or by its extent - the coverage of the subgroup in the dataset. The intent of a subgroup p is given by $p_d = \langle \varphi_1, ..., \varphi_{|M|} \rangle$ where each φ_i is a restriction on the domain value of $m_i \in M$. The intent p_d of subgroup p covers the set of objects denoted $ext(p_d) \subseteq G$.*

For example, given attributes m_1 (resp. m_2) with domain values $\{a, b, c\}$ (resp. $\{1, 2.7, 6.2\}$), we could find a subgroup whose intent is $\langle m_1 = a,\ m_2 = 2.7 \rangle$.

In subgroup discovery, we have only one target. The quality of a subgroup is usually defined as the discrepancy between the distribution of the target variable in the subgroup and its distribution over the entire dataset. Since important discrepancies can easily be achieved with small subsets of objects, a factor taking into account the size of the subgroup can be used as well (see, e.g., the popular Weighted Relative Accuracy measure). Exceptional Model Mining enables for two or more target variables depending on the chosen model class. In our current setting, given a dataset (G, M, T), the interestingness of a subgroup p is measured by a numerical value that quantifies the deviation between the model fitted on the subgroup and the same model fitted on the overall dataset. In most common EMM algorithms, the search space of subgroups is traversed in a general to specific way. At each stage, a specialization operator is applied to create more complex subgroups by addition of a restriction on an attribute.

2.2 Multi-objective Optimization. Many real world optimization problems are intrinsically multi-objectives.

DEFINITION 2.2. *A multi-objective optimization problem can be defined as follows:*

$$Minimize\ F(x) = (f_1(x), ..., f_n(x))^T, x \in M$$

where M is the attribute space and x is an attribute vector. $F(x)$ consists of n objective functions $f_i : M \to \mathbb{R}, i \in \{1, ..., n\}$, where $\mathbb{R}^n$ is the objective space.

However, the objectives usually conflict with each other and the improvement of an objective might lead to a degradation for others. For this reason, we lack a single solution that enables the optimization of all objectives at the same time. When no order or relevance can be defined a priori on the different objectives, a Pareto optimization method is required. It is based on the dominance between solutions of the objective space.

DEFINITION 2.3. *A vector $a = (a_1, ..., a_n)^T$ is said to dominate a vector $b = (b_1, ..., b_n)^T$, denoted $a \prec b$ if and only if $\forall i \in \{1, ..., n\}, u_i \leq v_i$ and $u \neq v$.*

A non-dominated solution is called Pareto optimal.

DEFINITION 2.4. *A solution x is called Pareto optimal if and only if $\nexists y \in M$ such that $F(y) \prec F(x)$. The set of all Pareto optimal solutions is called the Pareto Front $PF = \{F(x) | x \in M | \nexists y \in M, F(y) \prec F(x)\}$.*

3 Related Work

Although we are not aware of previous proposals connecting EMM to multi-objective optimization, related topics have been seriously investigated. Subgroup discovery has been mainly concerned with problems involving a unique target concept. When it comes to subgroup discovery and the optimization of a numerical target, few works exist. Among them, we find SD-Map* [1] and OSMIND [18] that both find optimal subgroups according to a quality measure calculated on a numerical target. In [18], it is shown that subgroup discovery can be used to mine high quality subgroups that optimize a numerical variable. However, such subgroup discovery algorithms are unable to deal with multiple numerical targets and can not be used in a multi-objective optimization setting. EMM generalizes subgroup discovery for multiple targets [7]. Several algorithms have been developed to build upon classical beam search approaches. To this end, heuristic [14], exhaustive [16] and sampling-based [19] methods were introduced to produce better patterns or to compute them faster. The type of model mined in EMM is crucial. Since any type of model can technically be used, many approaches involving different models have been proposed. In [8], the authors mine exceptional regression models, but one finds also the search for exceptional Bayesian networks [9] or exceptional correlations [6]. Regarding the association of subgroup discovery and multi-objective optimization, a few approaches have been proposed [3, 15, 22, 23]. [3] proposes an evolutionary algorithm to mine subgroups offering the best trade-offs between multiple quality measures (e.g., support, confidence, unusualness). The notion of skyline patterns is exploited in [23] to mine high quality patterns according to multiple measures at the same time: the user only needs to input the measures he is interested in and the algorithm returns the best patterns by exploiting the concept of dominance. However, these approaches work on defining the dominance between subgroups, while we are interested in the dominance between observations.

We have to design resilient quality measures to compare Pareto fronts. In the multi-objective optimization literature (see, e.g., [17]), numerous quality measures have been introduced to evaluate the distance between the true Pareto front and the Pareto fronts resulting from the optimization algorithms. Among others, an

averaged version of the Hausdorff distance was shown to give promising results [20]. [11] investigates the selection of a subset or a unique solution from the Pareto front when it is needed.

4 Contributions

4.1 Approach. We want to build a model class for EMM in a MOO setting: we propose to look for Exceptional Pareto Front Models (EPFM). In a given dataset, we define the true Pareto front - denoted PF^{true} - as the set of all non-dominated objects over the whole dataset. In typical EMM approaches, an exceptional model is computed directly on the objects of the subgroup. Then a quality measure is used to measure the deviation between the model built on the subgroup and the same model built on the whole dataset.

Our goal hereafter is to capture subgroups representing local phenomena with the highest influence on the shape of PF^{true}, meaning that we need to measure the effects on PF^{true} of removing these objects from the data. Therefore, when a subgroup is generated, we remove all its objects from the dataset. Then, we compute the new Pareto front PF on the remaining data. Finally, we can compute the deviation between PF^{true} - i.e., the Pareto front for the dataset - and PF.

Let us first define which objects of each Pareto front are taken into account when computing distances between Pareto fronts.

DEFINITION 4.1. *Given Pareto fronts PF_1 and PF_2, the Partial Pareto Front $PPF(PF_1, PF_2)$ is equal to:*

$$\{x \in PF_1 | \nexists y \in PF_2, x = y\}$$

The PPF is defined as the subset of objects of a Pareto front that are not in the set of objects of the other Pareto front. A PPF can be computed either for PF^{true} by keeping its objects which are not in PF or for PF by keeping its objects which are not in PF^{true}. Figure 1 depicts the PPFs of PF (left) and PF^{true} (right). In Figure 1 (left), the PPF of the model - denoted by PPF_model - is the set of objects of PF (i.e., the Pareto front of the data that is left once the subgroup has been removed) which do not belong to the Pareto front of the dataset - denoted by PF_true.

Conversely, in Figure 1 (right), the PPF of the dataset - denoted by PPF_true - is the set of objects of PF_true which do not belong to the Pareto front of the model - denoted by PF_model.

In our figures, NDP or ND stand for normal data point, SG denotes a subgroup, PF_true represents the best known Pareto front and PF_model represents the Pareto front of a subgroup.

4.2 Designing quality measures

4.2.1 Measuring distances between Pareto fronts. Multi-objective optimization needs for algorithms that approximate as well as possible the true Pareto front for any given problem. Many quality measures have been introduced to estimate the quality of the computed Pareto front compared to the true Pareto front or to an ideal point [17]. Thanks to some of these measures, the distance between two Pareto fronts can be computed. In traditional multi-objective optimization measures, only the distance between either the true Pareto front and the approximate Pareto front or the approximate Pareto front and the true Pareto front is computed. However, [20] shows that taking into account both distances provides measures that are more resilient to outliers and uncommonly shaped Pareto fronts. Therefore, we compute both the Euclidean distance between the partial Pareto front of the subgroup PPF and the Pareto front of the overall dataset PF^{true}, and the Euclidean distance between the partial Pareto front of the overall dataset PPF^{true} and the Pareto front of the subgroup PF. Then, the largest one is kept as the true distance. Furthermore, it is important to normalize each of the objectives such that they contribute equally to the measure. We normalize each of them to get a value between 0 and 1 using the standard scaling $x'_j = (x_j - min_j)/(max_j - min_j)$, where min_j and max_j are respectively the minimum and maximum of Objective j. In our figures, we use non-normalized ranges for a better understanding.

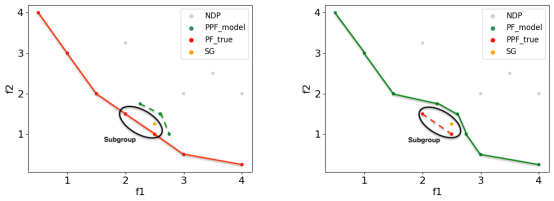


Figure 1: Partial Pareto fronts of PF (left) and PF^{true} (right).

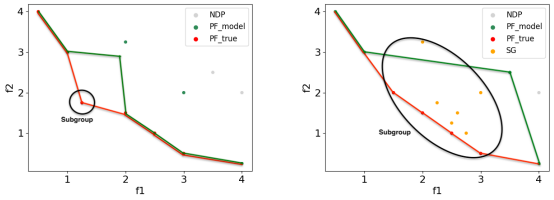


Figure 2: Low entropy (left) and large (right) subgroups.

Our measure is based on the popular Hausdorff Distance that estimates how far two subsets of a metric space are from each other: informally, it is defined as the largest of all the distances from a point in one subset to its closest point in the other subset.

DEFINITION 4.2. *The Hausdorff Distance between PF and PF^{true} is defined as:*

$$HD(PF, PF^{true}) = max(max(mind(PPF, PF^{true})), max(mind(PPF^{true}, PF)))$$

The Median Hausdorff Distance between PF and PF^{true} is defined as:

$$MHD(PF, PF^{true}) = max(med(mind(PPF, PF^{true})), med(mind(PPF^{true}, PF)))$$

mind computes the minimal Euclidean distance from each point of the partial Pareto front to the other Pareto front, max returns the largest value in a set of distances and med returns the median value in a set of distances.

Let us now consider a modified version of the Averaged Hausdorff Distance (AHD) introduced in [20].

DEFINITION 4.3. *The Averaged Hausdorff Distance $AHD(PF, PF^{true})$ between PF and PF^{true} is:*

$$max(\frac{1}{N}\sum_{i=1}^{N}(mind(PPF_i, PF^{true})), \frac{1}{M}\sum_{i=1}^{M}(mind(PPF_i^{true}, PF)))$$

N is the number of objects of PPF and M is the number of objects of PPF^{true}. mind computes the minimal Euclidean distance from object i of the partial Pareto front to the other Pareto front. The average of all minimal distances is then computed. Finally, max takes the largest distance of the two.

4.2.2 A generic quality measure for EPFM. Unfortunately, knowing the distance between Pareto fronts may not be enough to mine interesting subgroups. Indeed, an issue can arise when either outliers are apart of the true Pareto front or when the density of objects is very low close to some part of the Pareto front. Indeed, in such cases, the removal of subgroups with very few objects on the true Pareto front can create unwanted large deviations in the Pareto front of the model leading to overfitting and trivial subgroups. Figure 2 (left) depicts an example of this phenomenon. Furthermore, since our method relies on evaluating the effect on the true Pareto front of removing subgroups from the dataset, subgroups with a large coverage can create large deviations or even completely change the Pareto front of the dataset. Despite of their high quality, such subgroups are not interesting. Figure 2 (right) depicts an example of this phenomenon. To summarize, given the previously defined distance measures, we can get either very large or very small subgroups.

To deal with the first issue, we propose to use the entropy of the split between the objects of the Pareto front which are not part of the subgroup, and those who are.

DEFINITION 4.4. *The entropy of a subgroup p is $ent(p) = -\frac{n}{N}\lg\left(\frac{n}{N}\right) - \frac{N-n}{N}\lg\left(\frac{N-n}{N}\right)$ where $\lg$ denotes the binary logarithm, N is the total number of objects on the true Pareto front and n is the number of objects of p that belong to the true Pareto front.*

The entropy favors balanced splits over unbalanced ones. It returns 0 when the subgroup has no point on the true Pareto front or the subgroup covers the whole true Pareto front. It returns 1 when a perfect 50/50 split is achieved. This way, our quality measure is driven toward finding more relevant subgroups with enough objects on the true Pareto front. It is worth noting that this introduces a bias against subgroups which cover most of the true Pareto front (or the whole Pareto front) although these subgroups might be interesting.

To deal with the second issue (i.e., unwanted large subgroups), let us introduce a locality factor.

DEFINITION 4.5. *The locality factor of a subgroup p is $loc(p) = 1 - \left(\frac{n}{N}\right)$ where N is the total number of objects of the dataset and n is the number of objects of p.*

This locality factor favors smaller subgroups over larger ones. It is especially useful for cases where objects can be removed from a subgroup without modifying the Pareto fronts. We can now define our aggregated measure to take into account both the distance between Pareto fronts and the relevance of the subgroups.

DEFINITION 4.6. *Our aggregated quality measure q_{disrel} for a subgroup p is defined as:*

$$q_{disrel}(p) = dist(p) \times ent(p) \times loc(p)$$

$dist(p)$ can be any measure of distance between the Pareto fronts. $ent(p)$ is the entropy of the subgroup p and $loc(p)$ denotes its locality.

4.3 Algorithm. Our search space exploration method is based on a top-K beam search [7]. The evaluation part of the process is by far the most costly here. To compute the Pareto front of a subgroup, we employ a greedy approach where each object not in the subgroup is compared to all the objects not in the subgroup to check whether it is dominated by at least one other object. If it is not dominated by any other object, we add it to the Pareto front. Finally, we implemented a simple pruning technique that leads to a large reduction in the number of subgroups that need to be evaluated. Indeed, for a subgroup to be interesting, its removal has to create a deviation in the shape of the Pareto front. Due to the nature of the dominance relation, the removal of any object not on the Pareto front cannot lead to a change in the Pareto front. It means that only subgroups that contain at least one object that belong to the dataset Pareto front are of interest. As a result, during our search, we ignore any subgroup and their specializations if it does not contain an object that belongs to the dataset Pareto front.

5 Experiments

Let us now consider experiments on both synthetic and real life datasets. The source code and datasets used in our experiments are available at `https://bit.ly/3oXcSqO`. In the following experiments, the beam width was set to 10 and the search depth to 5. These parameters were chosen to explore the search space as much as possible while keeping the running times in an acceptable range. When not specified, the quality measure is q_{disrel} with HD. In the figures, both red and orange objects belong to the best subgroup.

5.1 Synthetic data

5.1.1 Fonseca-Fleming. Numerous test functions for multi-objective algorithms have been proposed in the literature. The true Pareto front of these functions is usually known and they are designed such that Pareto front approximation by algorithms is difficult. We consider here the Fonseca-Fleming function [10] that implies 3 descriptive variables from $\{x_1, x_2, x_3\}$ and 2 objectives. It is described by functions f_1 and f_2 that both need to be minimized:

$$f_1(p) = 1 - exp\left(-\sum_{i=1}^{3}\left(x_i - \frac{1}{\sqrt{3}}\right)\right), x_i \in [-4, 4]$$

$$f_2(p) = 1 - exp\left(-\sum_{i=1}^{3}\left(x_i + \frac{1}{\sqrt{3}}\right)\right), x_i \in [-4, 4]$$

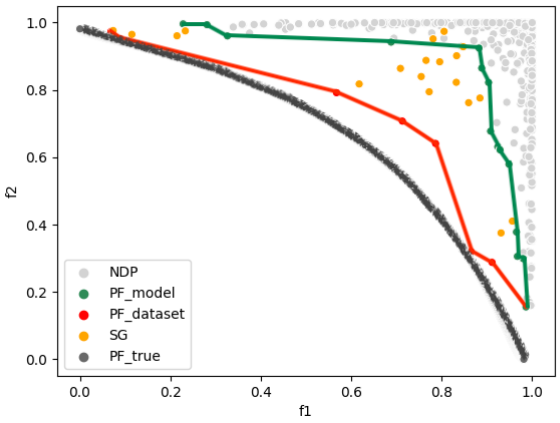


Figure 3: Best computed model on the Fonseca dataset (PF_model) compared to the known true Pareto front of the Fonseca-Fleming function (PF_true). PF_dataset represents the Pareto front of the overall dataset.

The goal of this experiment is to show how our algorithm is able to discover exceptional models leading to better results. Here, it means finding a subgroup with a description in the attribute space that leads to a better approximation of the true Pareto front. To do this, we generated a dataset made of 5000 random data points and ran our method.

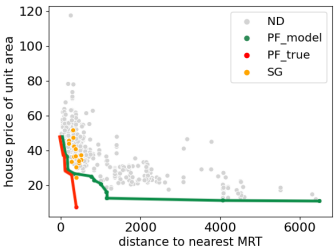
As can be seen in Figure 3, our algorithm is able to discover an exceptional model that provides actionable information on how to better approximate the true Pareto front of the Fonseca-Fleming function. Indeed, removing this subgroup from the dataset (i.e., removing the red and orange objects) leads to a way worse approximation of the true Pareto front. Moreover, this exceptional model can be described in terms of descriptive attributes. Here, the description of the subgroup is $x_1 \in [-0.798, 0.801]$ *and* $x_2 \in [-0.8, 0.8]$ *and* $x_3 \in [-0.8, 0.799]$. It supports its interpretation and gives explicit indications on which values for which variables lead to better points to improve the true Pareto front approximation. What is interesting here is that these interpretations can be extended to use cases where the true Pareto front is unknown. Indeed, our method is able to discover exceptional models that produce critical and useful information to further improve the optimization model, to explore yet unexploited zones of the objective space or to determine which solutions of the Pareto front are better than others.

5.1.2 Crop data. Let us now consider the impact of discretization on the quality of the discovered models.

We use here the Python Crop Simulation Environment PCSE[1] to generate a dataset of 300 plant growth recipes made of 9 numerical attributes and 2 numerical target labels which need to be optimized. We then generated several datasets by using discretization techniques on the main dataset. We used equal-width and equal-frequency, two of the most well-known discretization techniques and for each technique, we tried respectively 2, 3, 5, 10, 15 and 20 cut-points. It leaves us with 12 datasets on which we can experiment with our method to study the effect of discretization on the quality of the discovered models. We run our algorithm with the three described distance measures and only retained the best subgroup found for each run. The results can be found in Table 1. The overall best model found for each distance measure is highlighted in red. Although the discretization technique seems to have some impact on the quality of the best exceptional models (equal-width best model for both HD and AHD, and equal-frequency best model for MHD), the main issue w.r.t. quality seems to be the number of cut-points. Indeed, in almost all cases, the quality of the best model decreases as the number of cut-points increases. Finding large enough subgroups to create significant deviations in the Pareto front indeed becomes harder as the number of values that can be taken by each attribute grows.

5.2 Real world data. Let us now consider use cases that are less common in the multi-objective optimization community. Here, the data is limited to the available one (i.e., it cannot be easily extended) and the underlying model is unknown, making it impossible to run something else than a Pareto front computation.

5.2.1 Real Estate. This first dataset has been extracted from the UCI repository[2]. It concerns over 400 sales of houses in Taiwan between 2012 and 2013. It is made of 4 descriptive variables (latitude, longitude, house age, and number of convenience stores in the living circle on foot) and 2 objective variables: the price of the house and the distance to the closest massive rapid transit station which both need to be minimized. We run our method and retain the Top 3 exceptional models. Figure 4 gives an illustration of the best computed model on the left and the description of the Top 3 models on the right. *SG Desc* provides a subgroup intent and *Qual* denotes its quality. Here, our method allows us to discover a more interesting part (i.e., a subset) of the Pareto front that possesses a description in the description space of the attributes. The objects of

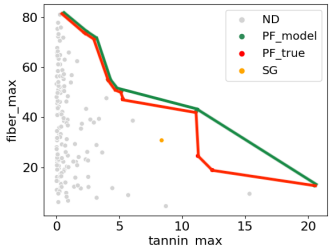


Rank	SG Desc	Qual
1	latitude ∈ [24.949, 24.965] and nb_conv_stores ∈ [4.0, 6.0]	0.79
2	latitude ∈ [24.949, 24.965] and longitude ∈ [121.529, 121.548]	0.76
3	house_age ∈ [8.76, 17.52] and longitude ∈ [121.529, 121.548]	0.75

Figure 4: Representation of the best computed model (left) and Top-3 subgroups (right) in Real Estate data.

PF_true that belong to the subgroup can be seen as better solutions than the rest of PF_true since their removal leads to a large deviation in the shape of the Pareto front (i.e., PF_model in Figure 4). In other terms, it can be used to find houses (including their location and characteristics) that offer a more interesting trade-off between price and distance to the nearest transport station.

5.2.2 Plant defenses. Let us now consider the trade-off between physical and chemical defense in plant seeds. The dataset `Plant`, made of 163 observations, was extracted from the Datadryad website[3]. Each observation is described by the family and the mass of the plant seed. The objective variables are the fiber - physical defense - and the tannin contents - chemical defense - that both need to be maximized. Again, we compute the Top 3 models. Their description is in Figure 5 (right) and an illustration of the best computed model is in Figure 5 (left). Once again, our method enables the discovery of more interesting parts of the Pareto front that possess descriptions in the attribute description space. The best subgroup is described by $family =' Melastomataceae'$. It means that removing plant seeds that belong to this family of plants leads to



Rank	SG Desc	Qual
1	family = 'Melastomataceae'	0.26
2	family = 'Combretaceae'	0.13
3	family = 'Menispermaceae'	0.11

Figure 5: (left) Representation of the best computed model (right) Top-3 subgroups in `Plant`.

[1] `https://pcse.readthedocs.io/en/stable/index.html`
[2] `https://archive.ics.uci.edu/ml/datasets.php`
[3] `https://datadryad.org/stash`

Table 1: Impact of the discretization on the quality of the best model for our 3 distance measures.

Measure \ Disc. tech.	Equal-Width						Equal-Frequency					
	2	3	5	10	15	20	2	3	5	10	15	20
q_{disrel} with AHD	0.033	0.02	0.02	0.01	0.01	0.01	0.02	0.03	0.02	0.01	0.01	0.01
q_{disrel} with HD	0.14	0.08	0.05	0.03	0.02	0.02	0.1	0.06	0.05	0.03	0.02	0.02
q_{disrel} with MHD	0.028	0.02	0.02	0.01	0.01	0.01	0.02	0.029	0.02	0.01	0.01	0.01

a large deviation in the Pareto front. In other terms, we identified a family of plants with a more interesting trade-off between physical and chemical defenses than other families.

5.3 Hyperparameter optimization for Machine Learning. We propose to apply our algorithm when one needs to optimize multiple metrics at the same time (e.g., precision and recall, bias and variance, quality and runtime, accuracy and interpretability). A metric can be any measure that needs to be optimized (e.g., a quality measure or the complexity of a model). Since multiple metrics need to be optimized at the same time, a trade-off has to be found. It has already been shown in the literature that one metric can often not be enough to assess the quality of a model [2, 21]. We show how we can discover exceptional models that lead to better trade-offs and higher quality learning models.

5.3.1 Regression. Let us first consider the optimization of the hyperparameters of a neural network, and more precisely of a multi-layer perceptron regressor. We use the `California Housing` dataset from the scikit-learn library[4]. It is made of 20640 observations, 9 variables and the goal is to predict the sale price of each house. We retain 9 hyperparameters and discretize each of them into a list of values to sample from. For every run of the neural network, we sample random values from the list of each hyperparameter. To evaluate the quality of the neural network depending on the hyperparameter values, we run the model 200 times and, for each run, we compute the maximum residual error and the explained variance of the model. Finally, we build a dataset made of 200 observations with 9 descriptive variables - the hyperparameter values of each run - and 2 objective variables - the maximum residual error and the explained variance - which both need to be minimized. We then use our algorithm to discover exceptional models. The best computed model is in Figure 6. We have a subgroup that creates a large deviation in the Pareto front. Its description is $learning\_rate\_init = 0.01$ and it contains several objects of PF_true that lead to better trade-offs between

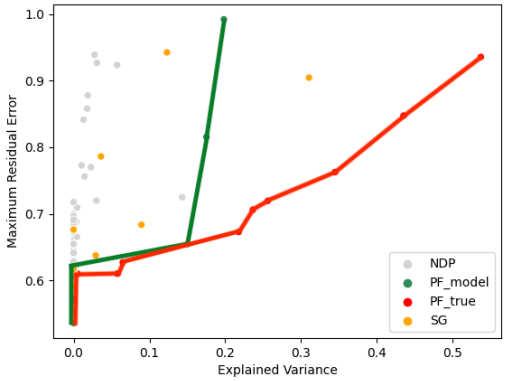


Figure 6: Best model in `California Housing`.

explained variance and maximum residual error for the neural network (i.e., removing those objects leaves neural networks with poorer trade-offs). This is interesting: we now know that running our neural network with a learning rate of 0.01 will lead to higher quality models. It can also be used as a basis for further hyperparameter optimization.

5.3.2 Multi-label classification. We now consider the optimization of hyperparameters for a multi-label classification task using a random forest classifier. We use the popular `yeast` dataset from the OpenML[5] repository. It is made of 2416 observations, 103 descriptive variables, and 14 binary labels to classify. We use 5 hyperparameters that are discretized into a list of values to sample from. For each run of the classifier, we select random values from the list of each hyperparameter. We run the classifier 200 times with different sets of hyperparameter values for each run and we assess the quality of the model by computing the recall and the precision of each model. Indeed, it is known that both precision and recall are important in classification tasks and that a trade-off between the two measures has to be found. Indeed, to do this, the F1 measure has been proposed. However, simplifying the problem

[4] `https://bit.ly/38UGJe9`

[5] `https://www.openml.org/`

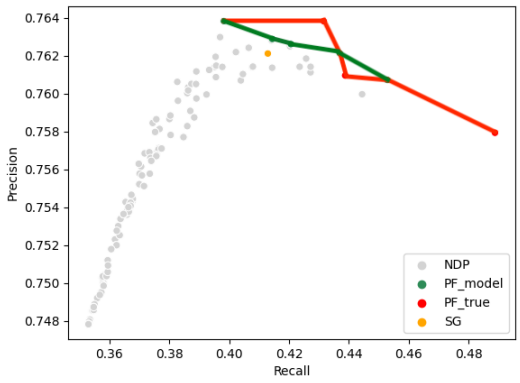


Figure 7: Best model in `yeast`.

of optimizing both measures into optimizing only one is a well-known trope in multi-objective optimization and it has been shown that it can lead to suboptimal results and loss of information. In other terms, we argue that optimizing both precision and recall at the same time is better than trying to optimize the F1 measure only. We finally build a dataset made of the 200 runs of the classifier with 5 descriptive variables - the hyperparameter values - and 2 objectives - the precision and recall - that need to be maximized. Using our algorithm to mine the best exceptional models in the data, we get the results in Figure 7. The best discovered subgroup, described by $n\_estimators = 900$ *and* $min\_samples\_leaf = 0.01$ creates deviations in multiple spots of the Pareto front. The objects of the PF_true that belong to the subgroup not only have a common description, but also offer a good trade-off between recall and precision. It can be used to prune the hyperparameter search space for further optimization of the classifiers, or be used to build high quality multi-label classifiers with an interesting trade-off between recall and precision.

So far, we have considered two objectives only. Our approach can however be generalized to more objectives. For instance, let us consider the same settings as with the previous example though adding the running time of each classifier as a third objective. Our goal is to look for models that maximize both the precision and recall and at the same time minimize the running time. Since we have now a larger objective space, we increase the number of multi-label classifier executions to 400 to make sure that the dataset provides a good enough cover of the objective space. After building our new dataset made of 400 observations, 5 descriptive variables and 3 objectives to be optimized, we run our algorithm to return the best computed model. When dealing with Pareto fronts which are more than two-dimensional, one way to study their characteristics is to use scatter plots and visualize the pair-wise relationship of objectives (see Figure 8). As can be seen on each of the 3 scatter plots, the removal of the subgroup leads to a large deviation in all 3 pair-wise relationships that compose the overall Pareto front. The corresponding subgroup is described by $min\_samples\_leaf = 0.01$ *and* $min\_samples\_split = 0.02$. From Figure 8, we can infer that the objects which compose the subgroup highly optimize the recall but show poorer results on execution time. This information as well as the subgroup description can be used to investigate the reasons why optimizing the recall leads to overall higher execution times, while the same relationship does not exist between precision and execution time. Next, if concessions can be made on the degree of optimization of the execution time (i.e., we still want solutions on the Pareto front but other objectives can be prioritized when a conflict occurs), the subgroup can be exploited to further optimize the classifiers by looking for solutions which both optimize the recall and precision while keeping the execution time as low as possible.

6 Conclusion.

We propose a new model class for Exceptional Model Mining in a multi-objective optimization setting. We look for deviations in the shape of the Pareto front created by the absence of a subgroup of objects compared to the same Pareto front computed on the whole dataset. We designed a new generic quality measure that combines both the distance between Pareto fronts and the relevance of the subgroup in the data. We propose a beam search strategy for top-K EMM with an interesting though simple pruning strategy. Thanks to experiments on both synthetic and real life data, we show how our method can be used for multiple purposes. On typical multi-objective optimization scenarios, it can be used to identify key features in the description space leading to a better approximation of the true Pareto front. On less common scenarios with limited data and unknown underlying models, it can be used either to identify a subspace of the current Pareto front where data might be missing or to select a subset of more interesting solutions of the Pareto front with an explicit and concise description in the attribute description space. We also introduced a use case on hyperparameter optimization for machine learning. It would be interesting to improve the proposed quality measure and its theoretical basis as well as to investigate the use of skyline patterns to discover exceptional models with the best trade-offs between multiple constraints.

Acknowledgments. Our research is partially funded by the FUI programme (project DUF 4.0, 2018-2021).

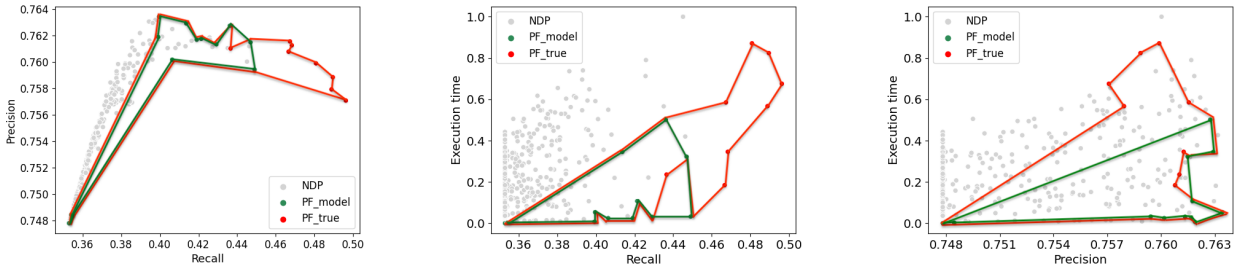


Figure 8: Scatter plots of the best computed model showing the pair-wise relationship between objectives.

References

[1] M. Atzmueller, and F. Puppe, *SD-Map–A fast algorithm for exhaustive subgroup discovery*, In PKDD 2006, pp. 6–17, Springer.

[2] J.C.F. Caballero, F.J. Martínez, C. Hervás, and P.A. Gutiérrez, *Sensitivity versus accuracy in multiclass problems using memetic Pareto evolutionary neural networks*, IEEE Trans. on Neural Networks 21, 5(2010), pp. 750–770.

[3] C.J. Carmona, P. González, M.J. del Jesus, and F. Herrera, *NMEEF-SD: non-dominated multiobjective evolutionary algorithm for extracting fuzzy rules in subgroup discovery*, IEEE Trans. on Fuzzy Systems 18, 5(2010), pp. 958–970.

[4] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, *A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II*, In PPSN 2000, pp. 849–858, Springer.

[5] K. Deb, *Multi-objective optimization*, In Search Methodologies, pp. 403–449, Springer, 2014.

[6] L. Downar, and W. Duivesteijn, *Exceptionally monotone models—the rank correlation model class for exceptional model mining*, Knowl. Inf. Syst. 51, 2(2017), pp. 369–394.

[7] W. Duivesteijn, A.J. Feelders, and A. Knobbe, *Exceptional model mining*, Data Min. Knowl. Discov. 30, 1(2016), pp. 47–98.

[8] W. Duivesteijn, A. Feelders, and A. Knobbe, *Different slopes for different folks: mining for exceptional regression models with cook's distance*, In ACM SIGKDD 2012, pp. 868–876.

[9] W. Duivesteijn, A. Knobbe, A. Feelders, and M. van Leeuwen, *Subgroup discovery meets bayesian networks–an exceptional model mining approach*, In IEEE ICDM 2010, pp. 158–167.

[10] C.M. Fonseca, and P.J. Fleming, *Multiobjective genetic algorithms made easy: selection sharing and mating restriction*, In GALESIA 1995, pp. 45–52.

[11] D. de la Fuente, M.A Vega-Rodríguez, and C.J Pérez, *Automatic selection of a single solution from the Pareto front to identify key players in social networks*, Knowledge-based Systems, 160 (2018), pp. 228–236.

[12] F. Herrera, C.J. Carmona, P. González, and M.J. Del Jesus, *An overview on subgroup discovery: foundations and applications*, Knowl. Inf. Syst. 29, 3(2011), pp. 495–525.

[13] W. Klösgen, *Explora: A multipattern and multistrategy discovery assistant*, In Advances in Knowledge Discovery and Data Mining, pp. 249–271, 1996.

[14] T.E. Krak, and A. Feelders, *Exceptional model mining with tree-constrained gradient ascent*, In SIAM Data Mining 2015, pp. 487–495.

[15] M. Van Leeuwen, and A. Ukkonen. *Discovering skylines of subgroup sets*, In ECML PKDD 2013, pp. 272–287. Springer.

[16] F. Lemmerich, M. Becker, and M. Atzmueller, *Generic pattern trees for exhaustive exceptional model mining*, In ECML PKDD 2012, pp. 277–292, Springer.

[17] M. Li, and X. Yao, *Quality evaluation of solution sets in multiobjective optimisation: A survey*, ACM Computing Surveys 52, 2(2019), pp.1–38.

[18] A. Millot, R. Cazabet, and J.F. Boulicaut, *Optimal subgroup discovery in purely numerical data*, In PaKDD 2020, pp. 112–124, Springer.

[19] S. Moens, and M. Boley, *Instant exceptional model mining using weighted controlled pattern sampling*, In IDA 2014, pp. 203–214, Springer.

[20] O. Schutze, X. Esquivel, A. Lara, and C.A.C. Coello, *Using the averaged Hausdorff distance as a performance measure in evolutionary multiobjective optimization*, IEEE Trans. on Evolutionary Computation 16, 4(2012), pp. 504–522.

[21] C. Shi, X. Kong, P.S. Yu, and B. Wang, *Multi-objective multi-label classification*, In SIAM Data Mining 2012, pp. 355–366.

[22] S. Srinivasan, and S. Ramakrishnan, *Evolutionary multi objective optimization for rule mining: a review*, Artif. Intell. Rev. 36, 3(2011), p. 205–248.

[23] W. Ugarte, P. Boizumault, B. Crémilleux, A. Lepailleur, S. Loudni, M. Plantevit, C. Raïssi, and A. Soulet, *Skypattern mining: From pattern condensed representations to dynamic constraint satisfaction problems*, Artif. Intell., 244, pp.48–69, 2017.

[24] A. Zhou, B.Y. Qu, H. Li, S.Z. Zhao, P.N. Suganthan, and Q Zhang, *Multiobjective evolutionary algorithms: A survey of the state of the art*, Swarm and Evol. Computat. 1, 1(2011), pp. 32–49.