In 2019, a Machine Learning (ML) model was created that derived the NOC with an accuracy of roughly 82% [1]. This Random Forest (RF) model was trained based on 590 profiles of 2-5 person mixtures, obtained from **TODO:find how many**  donors. The data used for training was not the original electrophoresis results, but consists of 19 features such as the MAC, locus-specific information, and other statistical features of the data.

**Decision Tree [2]**

Derive NOC from

* Decision tree with

Tested various ML approached (RF / MLP / LDA), showing similar performance to the RF19 model. They obtained very high performance (96%) with a RFC 35 model.

Difference with [2] is “Benschop et al. used 1174 unique donors to construct 590 profiles [20], whereas the PROVEDIt dataset only had 26 unique donors within the 766 profiles used”

This means that the classifiers probably overfit to certain donors.

“In conclusion, the decision tree method for NoC assignment has been shown to be over 77% accurate, with increasing performance with improved stutter and artefact filters”

They used a decision tree to classify peaks as stutter or allele.

**Background STR mixture interpretation [3]**

Information about how statistical analysis is done to determine the LR with the Hd and Hp. Showing that the LR is still the de-facto standard method.

“The peak height information is of benefit for analyzing mixed profiles.”

“The effect of incorrect estimation of the number of donors (caused by allele sharing) to the LR value was examined by Benschop (…) and was illustrated to exert a great effect on the LR” [4]

**Background about NFI-used software for LR calculation DNAStatistX**

Shows the importance of correct NOC estimations: under-assigned number of contributors can cause the model the fail calculating the LR because the observed peaks cannot be well explained.

Also includes the NOC model + the generic RF11 model (with a lower accuracy of ~

Also includes the LoCIM method for inferring the major contributor.

# Evaluation

Dependence between features must be visualized in explanations

The quality of explanations is sometimes evaluated by performing a quantitative evaluation of a user study. Users are asked to perform a certain task and the explanations help support this task. How well and how fast the humans can accomplish the task is measured as accuracy and efficiency respectively [5, 6]. Subjectively, users were asked for their preference of explanation type in a 1 versus 1 fashion and asked to provide reasons.

Explanations should have few features, as humans pick usually just a few reasons. They should be specific to the problem at hand, and every instance should be explained in the same deterministic way [7]. Deterministic, or consistent feature attributions [8].

Exploration using several visual aids [9].

Research into XAI has shown the need for comparison and evaluation of methods [10-16], and the recent interest in the implementation of counterfactual explanations [10-14]. Although there are a few key components highlighted by these surveys, they also mention that the evaluation must be done specifically to certain applications [14]. One could specify a specific goal to be achieved by the explanations which should be tested [15]. Also the relevance of explanations to a certain audience [16].

What does the NOC machine learning problem look like?

The dataset consists of 590 samples of mixtures between 1 and 5 contributors.

Any machine learning model learns to map the profile to a single output .

Where are the input features of a profile, which are all continuous variables. Currently in the RFC19 model. The target constitutes a multi-class classification problem in the current model, and for any other classifier. The target could be changed to be to correspond to a regression model.

Which model-agnostic counterfactual explanation techniques exist and what problems can they be applied to?

One of the first papers on counterfactual explanations was written to bring up discussion on the “right to explanation” in automated decisions made by black-box algorithms [17].

Two reasons that the authors mention which are also relevant to the current case:

1. Understand why a decision was made
2. Understand how a different decision could be made if certain conditions were changed

In short, a counterfactual explanation is a statement that shows how the instance would have to change to yield a different outcome, or formally:

“Score *p* was returned because variables *V* had values (*v*1, *v*2 , . . .) associated with them. If *V* instead had values (*v*1*'*, *v*2*'*, . . .), and all other variables had remained constant, score *p'* would have been returned.”[17]

For one instance, multiple possible counterfactuals exist, and it is often the question which one is most appropriate. Counterfactuals are purposefully constructed to be minimal, meaning that minimal changes are made to the instance in question. In this way, the changed instance can still be related to the original. It is desirable to present multiple explanations, which correspond to altering different aspects of the original instance. These aspects could also be case-specific.

To generate counterfactuals, we minimize:

Where is the original instance, and is the counterfactual, which we want to find to be as close to as defined by a certain distance function . The outcome of the prediction by the model should be the desired outcome . The desired outcome is balanced to the distance by a weight , where a larger value favors the desired outcome, and a lower value favors less change made to the original instance.

The choice of the distance function is important. The authors found the or Manhattan distance, weighed by the inverse median absolute deviation to be most useful. It is used to normalize each feature, and is robust to outliers. However, the authors recognize that this distance metric should be catered towards specific problems, audiences, and data.

Any suitable optimization algorithm can be used to solve this problem. If access to the gradients of the machine learning model are given, optimization is faster. However, since we aim to look at model-agnostic methods, we disregard this statement.

“Principally, counterfactuals bypass the substantial challenge of explaining the internal workings of complex machine learning systems”

“As the working memory of humans can contain around seven distinct items”

They tested on the LSAT dataset, which is a regression dataset predicting their entrance exam scores (3 input variables).

Pima Diabetes Database: whether Pima women are likely to develop diabetes or not (8 input variables).

One of the authors of this original paper, continued to make contributions with regards to **mixed input features** (categorical, numerical, other embeddings) which are not quite relevant to our problem; and **diverse explanations** [18]. The latter entails that once a type of counterfactual is generated, future counterfactuals do not make the same feature changes.

They also use a natural language form with automatically filled-in values to create a more natural explanation rather than a data point and its feature values.

Tested on the LSAT dataset again, showing that the heavily unbalanced dataset was counteracted with this approach: only 6% of students were black, but this strong bias was found.

Also showed how the original method by Wachter et al. uses a brute-force method which is not viable for a large dataset.

Only applicable to **linear classifiers**.

Evaluate based on **domain knowledge** if feature importance values are viable and legal. They claim that this approach yields “coherent and informative diverse explanations” based on reasoning about the problem. They claim that a user study would be most appropriate to evaluate whether explanations help people understand why a certain decision has been made.

Code: <https://bitbucket.org/ChrisRussell/diverse-coherent-explanations/>

Since then, many varieties of counterfactual methods have been generated.

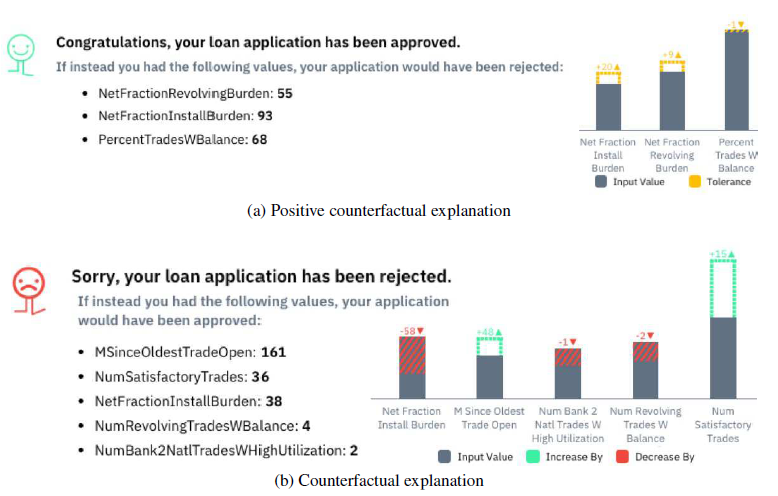
One method proposes to expand on Wachter et al. by adding ***positive counterfactuals***, which were catered towards the HELOC loan applications dataset [19]. These explanations are created when the desired outcome is already achieved, but a certain margin of value ranges is presented for which the outcome will also hold. They achieve this by **setting the target to represent the decision boundary, which would entail** .

In our case, that would still yield the same explanations since we have no explicit positive or negative case.

They balance the prediction target with the distance measure using a **lambda parameter** which is optimized using a tolerated mismatch between target and actual prediction.

A second change they propose, is **to weigh certain features** based on their relevance or importance. This is achieved by **adding a weight vector to the original distance metric** [19]. They use two strategies to obtain a relevance score for each feature. The first is using global feature importance (ANOVA) scores between feature and target, which should result in a smaller set of feature changes. **We can use the SHAP difference here.** The second is based on K-Nearest Neighbors to find instances close to the original stance, but with the desired result. The changed features can then be weighted according to this local area, giving more value to features that have historically been known to vary (since neighboring points exist).

**Evaluated** on different models, on a 5000-sample dataset: “we generate a counterfactual explanation for each of them, and compute the average counterfactuals size”

**Nice visualization **

This original notion of a counterfactual explanation was translated into a more conversation-like statement [20].

*“Your loan application has been declined. If you were a skilled employee instead of an unskilled –*

*resident, your loan application would be accepted.”*

A different approach to finding factual and counterfactual explanations is based around building a simple classifier around a local neighborhood of the point of interest [21]. This local neighborhood is created with a genetic algorithm, after which this feature space is described using a decision tree. The path down the tree that corresponds to the current label thus explains the actual prediction, while any path that leads to a different outcome could be interpreted as a counterfactual. They measure minimal changes to be the number of split conditions that need to be changed in order to change the prediction.

In comparison to LIME and Anchors, this approach thus supports contrastive explanations. Also the local neighborhood is explored through a genetic algorithm, which produces higher quality training data.

The genetic algorithm is still based around a notion of finding instances that look like x, but are different than x (in feature values). They generate both instances with the same, and different outcomes to x.

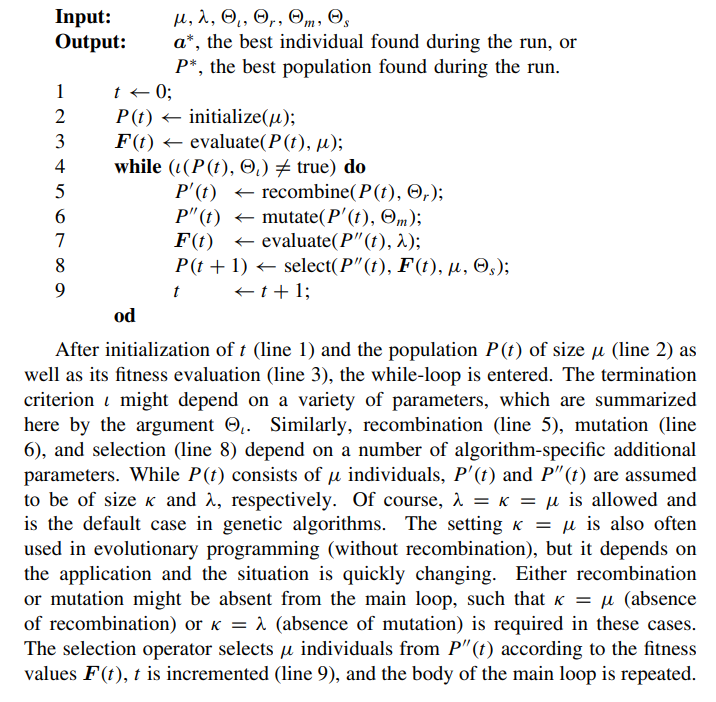


Figure 1: Basic evolutionary algorithm

* Fitness function: defined based on distance to the original instance
* Crossover/recombination: defined based on 2 features crossing over from parents to children.
* Mutation: defined based on 2 features mutating slightly from parent to child.

Evaluated based on correct classifications of the instances with the decision tree mimic model as compared to the original black-box. Evaluated on various (binary) classification datasets, mostly obtaining outputs with higher fidelity than LIME, even for the local neighborhood.

Where LIME generates something like: ““if the duration in months had been higher than the value it is for x, the prediction would have been, on average, 0.11 less 0 (or 0.11 more 1)”, which is not very intuitive to understand. In comparison to Anchors: “ANCHOR requires the apriori **discretization** of continuous features, while LORE benefits of the capabilities of decision trees to split continuous features”. “In summary, ANCHOR shows better precision than LORE at the expenses of generality and stability of the produced explanations.”

CADEX

We can also use the **gradient of the loss** with respect to the input, which can be followed over the input space with an optimizer (e.g. Adam) until we find an instance x\*, that typically lies on the decision boundary between y^ and y\* [22].

**A mask is used to only change certain attributes**. We pick which feature to change by first applying the original gradient function, and sorting which features are changed most. Then select the top n features to change, and incorporate those only into the mask. This mask is set to 1 for those features and multiplied by the gradient at each iteration of gradient descent (after obtaining the gradient, before applying it to the weights of the optimizer).

German loan dataset which contains 1000 observations. Run several times with number of features allowed to change: (5; 7; 10).

CADEX was compared to finding **training samples** with a different classification with the nearest L2 distance, and found that CADEX finds “**closer**” samples to the original. Yes, but are these viable data points?

Evaluated on **number of CFs found**

CADEX also finds attributes to change that SHAP does not consider to be important. However, this might be a wrong conclusion because SHAP is based on the classifier, and CF methods on the data.

They tested transferability to other classifiers, finding an 86% **agreement rate between RF and NN**.

Code: <https://github.com/spore1/cadex>

Some have used existing techniques like SHAP to formulate contrastive explanations [23]. They formulate their counterfactual as “‘Why [predicted-class]”, and “Why not [desired-class]?’”. Using the instance of interest, SHAP values are generated for the target class.

First SHAP values are generated for the target- and predicted class.

Then, counterfactual instances (in the training data) are identified using a nearest-neighbor approach (n=50 at a time), only changing features that contribute negatively for the target class.

A concise effort has been made to summarize the most important desiderata of counterfactual explanations [24]. For *completeness*, it is important to make emphasize the conditions of the counterfactual. This means that the explanation must include the phrase “had <variables x, y> changed to be <value a, b>, the classification outcome would have changed*, provided you do not change <variable z>”*. This also entails *contextfulness*. Others are: “soundness – truthfulness of the explanation with respect to the predictive model; interactiveness – interactive explanations are better than static ones; actionability – explanations that give the user suggestions how to change the model’s prediction are preferred; coherence – explanations should agree with the user’s mental model; complexity – the complexity of an explanation should be tuned to the user’s ability and knowledge; and parsimony – shorter explanations are more comprehensive.”

An investigation of the dis- and advantages of counterfactual explanations has shown some insight into the results [25]. It is good to show multiple counterfactuals. It is good to be able to show different levels of parsimony. It is good to show context.

An analysis of the assumptions made by counterfactual arguments has shown that changes deemed to be “actionable”, might not have the desired results; e.g. increasing income, might cause a person to change their job, which would decrease their length of employment [26]. This also highlights the fact that features are often correlated, and changing one value might impact another.

Another point the authors make is that in order to come up with a distance metric, the feature values must be normalized to make them comparable. However, one feature might have a more diverse distribution of values than another, or a certain feature might have a large amount of zeros for instance. These are often based on the training data, but might not align with the worldly reality; e.g. is one year of work experience equal to an increase in salary of 1000eu or 5000eu. It is better to recommend a feature change of something that has little cost to change, though this could conceal real reasons for a certain decision of immutable features such as race.

Here, again it is highlighted that there also must be a notion of what must *not* be changed.

Multi-Objective Counterfactuals (MOC) method [27]. Uses a distance metric based on four objectives, which are optimized simultaneously.

Counterfactual explanation x’ for x:

1. the prediction f(x’) is close to the desired outcome y’
2. x’ is close to x in feature space
3. x’ differs from x in only a few features
4. x’ is a plausible data point based on the training data

Based on L1 distance (2) for numerical features. The final objective (4) is based around finding the L1 distance between x’ and the k (=1) nearest neighboring data points (as an empirical approximation whether it is a plausible data point given the training data).

To minimize this joint problem, NSGA-II was adapted. For the crowding distance sorting, the method not only considers distances in the objective space, but also in the input space. In this way, they are kept if the feature values vary greatly in addition to their objective values. This corresponds better to the users’ desires.

Actionable features are also taken into account, meaning that feature values are capped to represent more realistic values.

Only evaluated on the 4 objectives posed, in comparison to other state-of-the-art methods (dice, recourse, whatif, tweaking).

Counterfactual Local Explanations viA Regression (CLEAR) [28]. They highlight the fact that when only generating counterfactuals, there is no notion of why the current prediction was made or how much each current feature contributed to the prediction. They indicate that instead perhaps a neighborhood could be defined around x. and use LIME to guide.

The method is based around

1. creating counterfactuals by searching through the feature space, starting from the input x
2. generating a **synthetic dataset by sampling features**, creating neighborhood around x (using Euclidean distance).
3. Perform **regression** on the neighborhood dataset.

They only change **one feature** to get counterfactuals.

They evaluate in comparison to LIME and based on a fidelity measure, which I don’t really understand.

[29] **generate CFs** based on a lin-SEDC algorithm performed on the top SHAP/LIME defined features.

(1) we propose two novel model agnostic explanation algorithms, creating them via the **combination of counterfactual explanations and additive feature attribution methods (LIME-C and SHAP-C)**; (2) we define quantitative evaluation criteria that proxy the effectiveness and efficiency of these algorithms; (3) we perform an in-depth evaluation of the explanation quality of LIME-C and SHAP-C when applied to high-dimensional behavioral and textual data and benchmark their performance against the SEDC algorithm, and lastly, (4) we propose changes to the model-agnostic methods for generating counterfactuals, and discuss research directions stemming out of our findings.

Evaluate on:

“1. Effectiveness

– Switching point: number of features that need to be removed before the classification changes. The switching point equals the size of the counterfactual explanation.

– Percentage explained: fraction of positively predicted instances for which a counterfactual explanation smaller than 30 features is found.

2. Efficiency

– Computation time: number of seconds it takes to generate an explanation.”

Created an interactive user-based system, that supports a range of *why* questions asked by a user through chat, called Glass-Box [30]. It provides class-contrastive counterfactual explanations, presented in natural language. They used the German credit dataset, which has a binary classification. The class contract in a multi-class classification could be specified.

Three types of questions are allowed: *Why* gives a plain counterfactual. *Why given* gives another counterfactual where some variables are fixed.

Desiderata based on [24]:

* F3: Explanation Target, F4: Explanation Breadth/Scope, F7: Relation to the Predictive System
* O7: Function of the Explanation, O8: Causality vs. Actionability
* U3: Contextfullness, U6: Chronology, U7: Coherence, U8: Novelty, U9: Complexity, U10: Personalisation, U11: Parsimony

They allow the **explanation to go further than a single data point, to a subspace** of the data.

Tested on AI people (6) and lay persons (11). They were satisfied with the explanations regarding understanding and content.

They used a decision tree to be the ML model and was used to generate the counterfactuals.

Counterfactual Explanations for Robustness, Transparency, Interpretability, and Fairness of Artificial Intelligence models (**CERTIFAI**) [31].

They use a **genetic algorithm** based on minimum distance (L1 distance) between counterfactual and instance to be explained. They sample on the CF side of the decision boundary by randomly sampling, and then applying the GA to improve the set. They only constrain feature values to be in the **min/max range** either set by user/training data.

They use their counterfactuals to evaluate the fairness and robustness of different classifiers.

**LOOK AT THEIR WAY OF DEFINING THE SEARCH SPACE / ALGO WITH SET NOTATION**

When sampling counterfactual instances it is important to bring in domain knowledge, especially if the problem is not that simple [32]. This domain knowledge includes taking advantage of the **empirical distribution** of the data to sample instances that are likely, by capturing the distribution of feature values and feature correlations.

Feature importance techniques fall into two main categories; perturbation- and gradient-based methods [33]. The most prevalent method using perturbations is SHAP. “The highlight of Shapley values is that they enjoy axiomatic uniqueness guarantees. Unfortunately, calculating the exact Shapley value is exponential in d, input dimensionality; however, the literature has proposed approximate methods using weighted linear regression [34], Monte Carlo approximation [61], centroid aggregation [11], and graph-structured factorization [14].”. “review the Shapley value explanations to ensure that the model exhibits expected behavior (i.e., the model uses the same features that a human would for the same task)”. “Organization A found that in practice LIME explanations [50] give unexpected explanations that do not align with human intuition. Recent work [71] shows that the fragility of LIME explanations can be traced to the sampling variance when explaining a singular data point and to the explanation sensitivity to sample size and sampling proximity”. In summary, Shapley values are easy to deploy and have solid motivations from game theory.

As for counterfactuals; they can be applied to any model and data type. Some counterfactuals might not be feasible. It is therefore important to use training data and constraints to guide the process.

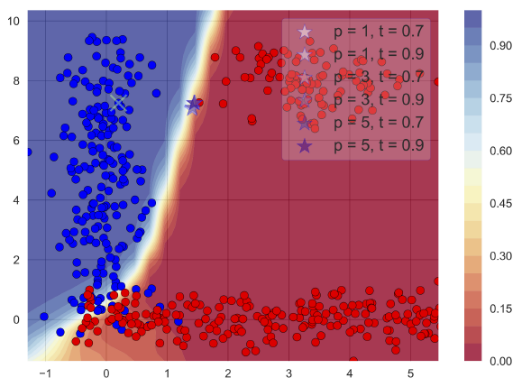
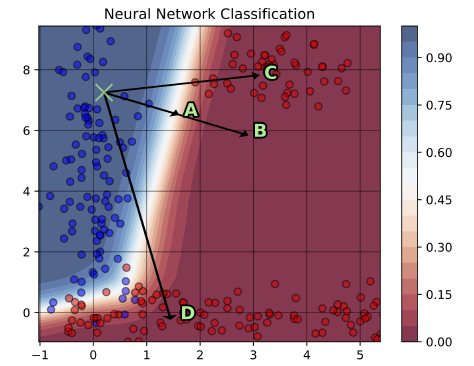
They recommend to determine clear desiderata:

* Identify the target for which the explanation is intended.
* Understand what they are looking for.
* Understand how they will engage with the explanation.

In a paper where counterfactual sets are presented, some desirable properties are listed [34]. One of these is the notion of set coverage, meaning that the **counterfactual set should include as many training samples as possible** (similarly defined in LORE). They also mention the bounds of the counterfactuals so that they produce **feasible instances**, as well as **provide ranges** instead of unbounded changes. For example; 25C < temp < 35C, instead of >25C.

Also evaluated on distance to factual sample (0-1 using L1-norm)

Feasible and Actionable Counterfactual Explanations (FACE) was developed to overcome the problem of generating counterfactuals that do not align with the **underlying data distributions** [35]. They also focus on actionability of the found counterfactual and argue that there should be an obtainable path to the desired new data point. They argue that D is therefore the best option, though I think C would be more appropriate for our purpose. A is generated by minimizing the L2 norm, and B represents a larger classification margin.



The danger of A and B is that these instances might still be classified wrong due to the low-density areas, which ML models are known to perform unreliably in.

The focus of this paper is to *transform* one data point to another, thereby focusing on actionability.

Their only focus for evaluation is to visualize the path taken by their algorithm. They do show that the counterfactuals by Wachter et al. (left picture) do not fall close to the data distribution.

ViCE: visual counterfactual explanations [36]. Their desiderata:

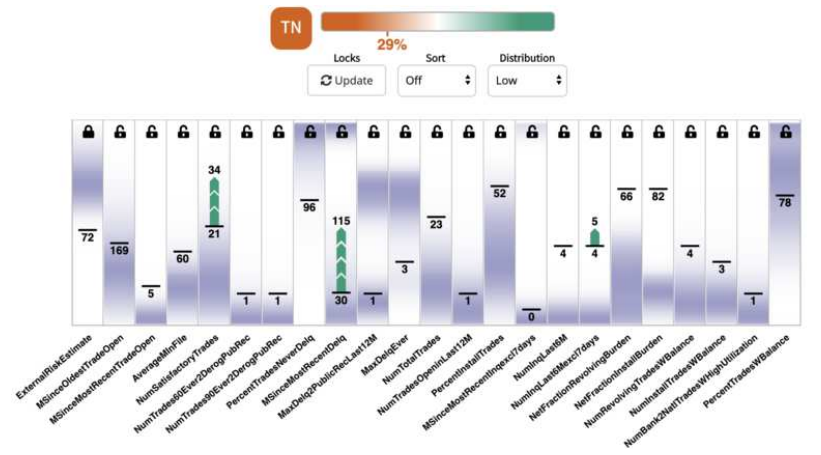
* How does the instance in question lie in comparison to the data?
* Which features are most relevant to the model?
* Which are changes in feature values that would alter the model’s prediction?
* How can we select actionable changes?

Find counterfactuals by finding a minimal set of features, and minimal amounts of changes.

They **discretize the dataset** by fitting a gaussian on each feature, splitting the values into *n* bins such that the middle *n-2* capture 4 standard deviations from the mean, extreme bins capture data points beyond that.

Their algorithm works **by greedily moving feature values across bins** until the predicted class is changed, or until a constraint is reached on the number of features changed / number of bins changed. They start with the instance of interest, and move along the available features to the bins below and above, choosing the one that elicits the most change in the model’s output.

Model-agnostic.



They evaluated on the Home Equity Line of Credit (HELOC) dataset, using the desiderate mentioned previously. Very shortly!!

“Q1 Data distribution - How do the values of the instance compare to those across the rest of the dataset? Example: If a student has a GRE score of 320, how does it compare to the scores of their peers?

Q2 Relevant features -Which features have the most considerable effect on the model’s prediction? Example: Identifying what variables in a patient’s blood work are significant contributors to a negative diagnosis.

Q3 Possible changes - Are there changes that could alter the model’s current prediction? Example: If an applicant was rejected for a loan, what changes in their profile would be required for the application to be accepted?

Q4 Actionable changes - Is it possible to change only a subset of actionable features to change the model’s prediction? Example: If a graduate school applicant knows certain features cannot be changed such as Gender or Age, is it possible to generate an alternative explanation without altering these features?”

Cannot be used for multi-class in this moment.

**DiCE** proposes diversity as an important constraint for actionable counterfactuals [37]. Evaluation framework for CF methods.

They assume that the **ML model is differentiable** and that the **output is binary**. They transform all **features to be 0-1**.

They define **diversity** using a determinantal point processes (DPP), which is used for subset selection with diversity. They define proximity as the **mean l1 distance** over the set. They allow for the user to select any features they do not wish to change.

In summary, this is minimization over the target, the distance, and (neg) diversity. Which they optimize using **gradient descent** with a max of 5000 steps.

They make an interesting point about the target class, even though that is binary; when you try to optimize to the target (1 instead of 0), you penalize when the prediction is 0.8 for instance, even though that is a good option. They therefore use a hinge loss (loss is 0 for 100% correct, then linearly increase the loss for more wrong classifications).

For distance: “For continuous features, we define dist as the mean of feature-wise ℓ1 distances between the CF example and the original input. Since features can span different ranges, we divide each feature-wise distance by the median absolute deviation (MAD) of the feature’s values in the training set, following Wachter et al.”

They balanced distance and diversity using hyperparameters, for which a gridsearch was performed.

Code: <https://github.com/interpretml/DiCE/tree/master/dice_ml>

Usually CFs are evaluated in a qualitative fashion. Ultimately, they should help a user.

Evaluate on:

* Validity = nr of CFs with the target outcome (f(c)>0.5)
* Proximity = the one mentioned above.
* Sparsity = ..
* Qualitative evaluation = seeing if they make sense
* User understanding = by generating a 1-NN classifier based on . in comparison to LIME

They mention that you need to conduct a user study where CF examples are compared against past approaches, to provide better explanations.

**QUITE SIMILAR APPROACH WITH MULTIPLE EVALUATIONS; YET NOT IN METHOD**

An assessment of what are *good* counterfactuals for various contexts [38]. They show that for a case of a drinking app that determines whether you are over the limit or not, the factual case is not as effective as the counterfactual case; showing that it displays more about the causal dependencies, and actionable information. “counterfactuals can tell you about the feature differences that affect the decision boundary around a prediction.”. Have the same definition for counterfactuals:

“*Score y was returned because variables V had values (v1, v2,.. ). If V had values (v1', v2' ... ), and all others remain constant, score y' would have been returned.*”

Problems:

1. **Prolixity**: many systems generate a variety of counterfactuals using random perturbation and search, from which a “best” must be chosen. They often use l1 with MAD of each feature.
2. **Sparcity**: modify the fewest features, preferably with 1-2 features.
3. **Plausibility**: must be valid data points (= the least solved)

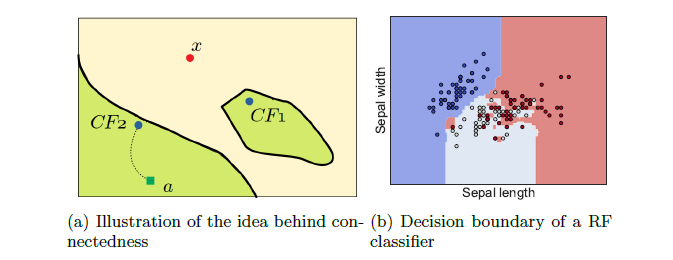
“**Plausibility is the least solved of the three problems facing counterfactuals**; for instance, many researchers propose to “lock” features (e.g., to not allow gender change) or to allow users to provide inputs on feature weights (e.g., using sliders at the interface on, say, salary boundaries). However, attempts to find an automated solution to the plausibility problem are thin on the ground. Here, we propose that, rather than generating counterfactuals by “blind” random perturbation, an XAI system should use the training-data/casebase to find suitable counterfactuals; as these counterfactuals are “real experiences” from the problem domain, they have an inherent plausibility”

Nr. of feature differences between all pairwise comparisons of cases of various datasets. Good counterfactuals are rare (with <3 feature changes): <1%; most have >5 feature changes. For numerical features, it might be good to introduce a **matching tolerance**. The Cleveland heart dataset seems to be the most similar to our problem: 303 cases, 13 features, 5 classes. Also evaluate on distance.

**They develop an approach to find a nearest data point that *does* have a good counterfactual, therefore making sure that the counterfactual is plausible since it came from the training data.**

They did not perform a user study!

A study on justification of counterfactuals [39]. “*Given a classifier f : X ! Y trained on a dataset X, a counterfactual example e <in> X is justified by an instance a <in> X correctly predicted if f(e) = f(a) and if there exists a continuous path h between e and a such that no decision boundary of f is met. Formally, e is justified by a <in> X if: <there is an> h : [0; 1] ! X such that: (i) h is continuous, (ii) h(0) = a, (iii) h(1) = e and (iv) <all>t <in> [0; 1]; f(h(t)) = f(e).”*



They argue that CF2 is better than CF1; in B you can see an application for it (overfit classifier). “CF2 can be connected to a ground-truth instance a without crossing the decision boundary of f and is therefore justified.”

**So instead of choosing instances from the training data per se; they make sure that a counterfactual instance can be “connected” to a ground truth data point without crossing the decision boundary.**

They have written software to asses if counterfactuals are connected to ground-truth data. They sample a circle around an instance, and cluster them using DBSCAN. If they belong to the same cluster as the ground-truth instance, they are justified.

Shows that LORE often creates justified explanations.

<https://github.com/thibaultlaugel/truce>

“As shown by our experiments on real-world data, our algorithm is: i) model-agnostic ((non-)llinear, (non-)differentiable, (non-)convex); ii) data-type-agnostic (heterogeneous features); iii) distance-agnostic (l0; l1; linf, and combinations thereof); iv) able to generate plausible and diverse counterfactuals for any sample (i.e., 100% coverage); and v) at provably optimal distances.” [40].

“it is widely agreed that a good explanation should provide answers to the following two questions [Doshi-Velez and Kim, 2017, Gunning, 2019, Wachter et al., 2017b]: (i) “why the model outputs a certain prediction for a given individual? ”; and, (ii) “what features describing the individual would need to change to achieve the desired output? ””

MACE: <https://github.com/amirhk/mace>

Distance = for categorical and numerical, normalized.

Plausibility = data points fall into observes range from the original data.

Diversity = distance between old and new CF values must be greater than a certain distance.

Evaluated using standard datasets on **distance** (norm distance) and **plausibility** (coverage = % of factual samples for which plausible (in type and range) counterfactuals can be generated)

Desiderata [41]:

* Validity: meaning it has the desired class (originally by wachter et al)
* Proximity
* Actionability
* Sparsity
* Data manifold closeness
* Causality “a counterfactual should maintain any known causal relations between features”
* Diversity

Challenges:

* The evaluation for counterfactual explanations must be done using a user study.
* Counterfactual explanations should be integrated with visualization features.
* Generate robust counterfactual explanations (not overfit to model).
* Unify counterfactuals with traditional XAI

GeCo: [42]

Using a **GA**: “A good explanation 𝒙𝑐 𝑓 should differ from 𝒙 by only a few features; counterfactual examples 𝒙𝑐 𝑓 that require the customer to change too many features are of little interest. Based on this observation, we propose a genetic algorithm, which we customize to search the space of counterfactuals by prioritizing those that have fewer changes. Starting from a population consisting of just the given entity 𝒙, the algorithm repeatedly updates the population by applying the operations crossover and mutation, and then selecting the best counterfactuals for the new generation. It stops when it reaches a sufficient number of examples on which the classifier returns the “good” (desired) outcome.”

Compares to MACE, CERTIFAI, WIT (what if tool)

Claim that **Mace is not model agnostic**.

Want to create **feasible** and **plausible** CFs.

**Feasibility is defined by using constraints on the features**

**Evaluation**: Benchmark on 5000 instances on:

1. The consistency of the explanations,i.e., does the classifier return the good outcome for counterfactual 𝒙𝑐 𝑓 ;
2. The distance between 𝒙𝑐 𝑓 and the original instance 𝒙;
3. The number of features changed in 𝒙𝑐 𝑓 .

Anchors [6]

An anchor A is a set of predicates that returns true for an instance x if all the predicates hold.

e.g. if x = “This movie is not bad.”, f(x) = Positive, A(x) = 1 where A = (“not”, “bad”)

D(·|A) is the conditional distribution where rule A applies.

A is an anchor if .

Where A(x) = 1 means that the anchor A holds for instance x. And the anchor A is a sufficient condition for f(x) with high () probability, for a sample z from D(z|a), for a certain prediction f(x) = f(z).

“anchors are by construction faithful, adapting their coverage to the model’s behavior and making their boundaries clear.”

D is defined by a validation dataset (training set?). By fixing A, then sampling the rest of the row, D(z|A) is defined.

For “hard” predictions around the decision boundaries, the generated Anchors can be longer.

“Short, disjoint rules are easier to interpret than hierarchies like decision lists or trees (Lakkaraju, Bach, and Leskovec 2016)”

Need to compute: “an anchor A is a set of feature predicates on x that achieves prec(A) ≥ τ”

Which is intractable precisely. Instead we look for a probability constraint P(prec(A) ≥ τ ) ≥ 1 – δ

“If multiple anchors meet this criterion, those that describe the behavior of a larger part of the input space are preferred, i.e. ones with the largest coverage. . Formally, we define the coverage of an anchor as the probability that it applies to samples from D, i.e. cov(A) = ED(z) [A(z)].”.

We thus define this search for an anchor as the following combinatorial optimization problem: max A s.t. P (prec(A)≥τ)≥1−δ cov(A).

## How to weigh the total closeness scores

* Weigh features according to feature importance
* Weigh distance and features changed together
* Weigh using hyperparameters ([42] and MACE)

1. Benschop, C.C.G., et al., *Automated estimation of the number of contributors in autosomal short tandem repeat profiles using a machine learning approach.* Forensic Science International: Genetics, 2019. **43**: p. 102150.

2. Kruijver, M., et al., *Estimating the number of contributors to a DNA profile using decision trees.* Forensic Science International: Genetics.

3. Tao, R., et al., *Separation/extraction, detection, and interpretation of DNA mixtures in forensic science (review).* International Journal of Legal Medicine, 2018. **132**.

4. Benschop, C.C.G., et al., *The effect of varying the number of contributors on likelihood ratios for complex DNA mixtures.* Forensic Science International: Genetics, 2015. **19**: p. 92-99.

5. Kim, B., O. Koyejo, and R. Khanna. *Examples are not enough, learn to criticize! Criticism for Interpretability*. in *NIPS*. 2016.

6. Ribeiro, M.T., S. Singh, and C. Guestrin. *Anchors: High-Precision Model-Agnostic Explanations*. in *AAAI*. 2018.

7. Lakkaraju, H., et al., *Interpretable & Explorable Approximations of Black Box Models.* 2017.

8. Lundberg, S. and S.-I. Lee, *A Unified Approach to Interpreting Model Predictions*. 2017.

9. Lundberg, S.M., et al., *Explainable machine-learning predictions for the prevention of hypoxaemia during surgery.* Nat Biomed Eng, 2018. **2**(10): p. 749-760.

10. Du, M., N. Liu, and X. Hu, *Techniques for interpretable machine learning.* Communications of the ACM, 2020. **63**(1): p. 68-77.

11. Adadi, A. and M. Berrada, *Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI).* IEEE Access, 2018. **6**: p. 52138-52160.

12. Barredo Arrieta, A., et al., *Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI.* Information Fusion, 2020. **58**: p. 82-115.

13. Carvalho, D.V., E.M. Pereira, and J.S. Cardoso, *Machine learning interpretability: A survey on methods and metrics.* Electronics (Switzerland), 2019. **8**(8).

14. Gilpin, L.H., et al. *Explaining Explanations: An Overview of Interpretability of Machine Learning*. in *2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA)*. 2018.

15. Lipton, Z.C., *The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery.* Queue, 2018. **16**(3).

16. Murdoch, W.J., et al., *Definitions, methods, and applications in interpretable machine learning.* Proceedings of the National Academy of Sciences of the United States of America, 2019. **116**(44): p. 22071-22080.

17. Wachter, S., B. Mittelstadt, and C. Russell, *Counterfactual Explanations Without Opening the Black Box: Automated Decisions and the GDPR.* Harvard journal of law & technology, 2018. **31**: p. 841-887.

18. Russell, C. *Efficient search for diverse coherent explanations*. 2019.

19. Grath, R.M., et al., *Interpretable Credit Application Predictions With Counterfactual Explanations.* ArXiv, 2018. **abs/1811.05245**.

20. Sokol, K. and P. Flach. *Conversational Explanations of Machine Learning Predictions Through Class-contrastive Counterfactual Statements*. 2018.

21. Guidotti, R., et al., *Factual and Counterfactual Explanations for Black Box Decision Making.* IEEE Intelligent Systems, 2019. **34**(6): p. 14-23.

22. Moore, J., N. Hammerla, and C. Watkins, *Explaining deep learning models with constrained adversarial examples*. 2019. p. 43-56.

23. Rathi, S., *Generating Counterfactual and Contrastive Explanations using SHAP*. 2019.

24. Sokol, K. and P. Flach. *Desiderata for interpretability: Explaining decision tree predictions with counterfactuals*. 2019.

25. Sokol, K. and P. Flach. *Counterfactual explanations of machine learning predictions: Opportunities and challenges for AI safety*. 2019.

26. Barocas, S., A.D. Selbst, and M. Raghavan. *The hidden assumptions behind counterfactual explanations and principal reasons*. 2020.

27. Dandl, S., et al. *Multi-Objective Counterfactual Explanations*. in *Parallel Problem Solving from Nature – PPSN XVI*. 2020. Cham: Springer International Publishing.

28. White, A. and A. Garcez. *Measurable Counterfactual Local Explanations for Any Classifier*. in *ECAI*. 2020.

29. Ramon, Y., et al., *A comparison of instance-level counterfactual explanation algorithms for behavioral and textual data: SEDC, LIME-C and SHAP-C.* Advances in Data Analysis and Classification, 2020. **14**(4): p. 801-819.

30. Sokol, K. and P. Flach, *One Explanation Does Not Fit All: The Promise of Interactive Explanations for Machine Learning Transparency.* KI - Kunstliche Intelligenz, 2020. **34**(2): p. 235-250.

31. Sharma, S., J. Henderson, and J. Ghosh. *CERTIFAI: A common framework to provide explanations and analyse the fairness and robustness of black-box models*. 2020.

32. Bertossi, L., *Score-Based Explanations in Data Management and Machine Learning*. 2020. p. 17-31.

33. Bhatt, U., et al. *Explainable machine learning in deployment*. 2020.

34. Fernández, R.R., et al., *Random forest explainability using counterfactual sets.* Information Fusion, 2020. **63**: p. 196-207.

35. Poyiadzi, R., et al., *FACE: Feasible and Actionable Counterfactual Explanations*. 2020. 344-350.

36. Gomez, O., et al. *ViCE*. 2020.

37. Mothilal, R.K., A. Sharma, and C. Tan. *Explaining machine learning classifiers through diverse counterfactual explanations*. 2020.

38. Keane, M. and B. Smyth, *Good Counterfactuals and Where to Find Them: A Case-Based Technique for Generating Counterfactuals for Explainable AI (XAI)*. 2020.

39. Laugel, T., et al., *Unjustified Classification Regions and Counterfactual Explanations in Machine Learning*. 2020. p. 37-54.

40. Karimi, A.-H., et al. *Model-agnostic counterfactual explanations for consequential decisions*. in *International Conference on Artificial Intelligence and Statistics*. 2020. PMLR.

41. Verma, S., J.P. Dickerson, and K. Hines, *Counterfactual Explanations for Machine Learning: A Review.* ArXiv, 2020. **abs/2010.10596**.

42. Schleich, M., et al., *GeCo: Quality Counterfactual Explanations in Real Time*. 2021.