ESANN 2020 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence
and Machine Learning. Online event, 2-4 October 2020, i6doc.com publ., ISBN 978-2-87587-074-2.
Available from http://www.i6doc.com/en/.

Efficient computation of counterfactual explanations of LVQ models

André Artelt* and Barbara Hammer †

CITEC - Cognitive Interaction Technology
Bielefeld University - Faculty of Technology
Inspiration 1, 33619 Bielefeld - Germany

Abstract. The increasing use of machine learning in practice and legal regulations like EU's GDPR cause the necessity to be able to explain the prediction and behavior of machine learning models. A prominent example of particularly intuitive explanations of AI models in the context of decision making are counterfactual explanations. Yet, it is still an open research problem how to efficiently compute counterfactual explanations for many models. We investigate how to efficiently compute counterfactual explanations for an important class of models, prototype-based classifiers such as learning vector quantization models. In particular, we derive specific convex and non-convex programs depending on the used metric.

1 Introduction

Due to the recent advances in machine learning (ML), ML models are being more and more used in practice and applied to real-world scenarios for decision making [1]. Essential demands for user acceptance as well as legal regulations like the EU's "General Data Protection Right" (GDPR) [2], that contains a "right to an explanation", make it indispensable to explain the output and behavior of ML models in a comprehensible way. As a consequence, many research approaches focused on the question how to realize explainability and transparency in machine learning in recent years [3]. There exist diverse methods for explaining ML models [4]: Local methods explain the decision for one specific input, while global methods refer to the whole model. Model-agnostic methods act as black-box methods, which do not need access to the training data or model internals, as opposed to model-specific technologies. Example-based explanations rely on references to a (set of) data points to explain a prediction or behavior of a model [5]. Alternatives refer to features, or generalizations thereof.

Counterfactual explanations constitute one popular instance of local agnostic example-based explanations [6]. Given an input, a counterfactual explanation represents a change of the original input that leads to a different (specific) prediction/behavior of the ML model. Counterfactual explanations are considered to be intuitive and useful because they can be associated to the minimum change/action which is required to achieve a desired outcome [6]. Existing counterfactual explanations are mostly model agnostic methods, which are universally applicable but computationally expensive [7–9]. In this work, we address the question how to efficiently compute counterfactuals for a popular class of models by referring to its specific structure.

*corresponding author: aartelt@techfak.uni-bielefeld.de

†We gratefully acknowledge funding from the VW-Foundation for the project *IMPACT* funded in the frame of the funding line *AI and its Implications for Future Society.*

Prototype based models such as learning vector quantization (LVQ) represent data by a set of representative samples [10]. LVQ models can be combined with metric learning and thereby increase the effectiveness of the model in case of few prototypes [11,12]. Furthermore, LVQ models can be used in many settings like life-long learning [13]. Here, we will consider the question how to efficiently compute counterfactual explanations of prototype-based classifiers, in particular LVQ models. By exploiting the special structure of such models, we are able to (i) propose model- and regularization-dependent methods for efficiently computing counterfactual explanations of LVQ models, (ii) and empirically demonstrate the efficiency of the modeling as regards speed-up as well as required amount of change in comparison to standard techniques. Further, the framework enables a straightforward incorporation of domain knowledge, which can be phrased as additional constraints.

2 Counterfactual explanations

Counterfactual explanations [6] (often just called counterfactuals) are an instance of example-based explanations [5]. A counterfactual states a change to some features/dimensions of a given input such that the resulting data point (called counterfactual) has a different (specified) prediction than the original input. The rational is considered to be intuitive, human-friendly and useful because it tells practitioners which minimum changes can lead to a desired outcome [6]. Formally, assume a prediction function h is given. Computing a counterfactual $\vec{x}' \in \mathbb{R}^d$ for a given input $\vec{x} \in \mathbb{R}^d$ is phrased as optimization problem [6]:

$$\underset{\vec{x}' \,\in \mathbb{R}^d}{\arg\min}\; \ell\left(h(\vec{x}'), y'\right) + C \cdot \theta(\vec{x}', \vec{x}) \tag{1}$$

where $\ell(\cdot)$ denotes the loss function, y' the requested prediction, and $\theta(\cdot)$ a penalty term for deviations of $\vec{x}'$ from the original input $\vec{x}$. $C > 0$ denotes the regularization strength. Common regularizations are the weighted Manhattan distance

$$\theta(\vec{x}', \vec{x}) = \sum_j \alpha_j \cdot |(\vec{x})_j - (\vec{x}')_j| \quad \text{where } \alpha_j > 0 \tag{2}$$

and the generalized L2 distance with a symmetric positive semi-definite (s.psd) matrix $\boldsymbol{\Omega}$

$$\theta(\vec{x}', \vec{x}) = \|\vec{x} - \vec{x}'\|_{\boldsymbol{\Omega}}^2 = (\vec{x} - \vec{x}')^\top \boldsymbol{\Omega}(\vec{x} - \vec{x}') \tag{3}$$

Depending on the model and the choice of $\ell(\cdot)$ and $\theta(\cdot)$, the final optimization problem might be differentiable or not. In the black box setting general optimization schemes such as Downhill-Simplex search are used [14]. Depending on the model and regularization type, the found solution might not be unique; this is usually referred to as *Rashomon effect* [14].

3 Learning vector quantization

In learning vector quantization (LVQ) models [10] we compute a set of labeled prototypes $\{(\vec{\mathrm{p}}_i, o_i)\}$ from a training data set of labeled real-valued vectors - we refer to the i-th prototype as $\vec{\mathrm{p}}_i$ and the corresponding label as o_i. A new data point is classified according to the winner-takes-it-all scheme:

$$h(\vec{x}) = o_i \quad \text{s.t. } \vec{\mathrm{p}}_i = \underset{\vec{\mathrm{p}}_j}{\arg\min}\ \mathrm{d}(\vec{x}, \vec{\mathrm{p}}_j) \tag{4}$$

where $\mathrm{d}(\cdot)$ denotes a distance function. In vanillas LVQ, this is independent chosen globally as the squared Euclidean distance $\mathrm{d}(\vec{x}, \vec{\mathrm{p}}_j) = (\vec{x} - \vec{\mathrm{p}}_j)^\top \mathbb{I}(\vec{x} - \vec{\mathrm{p}}_j)$. There exist extensions to a global quadratic form $\mathrm{d}(\vec{x}, \vec{\mathrm{p}}_j) = (\vec{x} - \vec{\mathrm{p}}_j)^\top \boldsymbol{\Omega}(\vec{x} - \vec{\mathrm{p}}_j)$, referred to as matrix-LVQ (GMLVQ), or a prototype specific quadratic form $\mathrm{d}(\vec{x}, \vec{\mathrm{p}}_j) = (\vec{x} - \vec{\mathrm{p}}_j)^\top \boldsymbol{\Omega}_j(\vec{x} - \vec{\mathrm{p}}_j)$, referred to as local-matrix LVQ (LGMLVQ). Since we are interested in the functional form of the model only, we do not refer to specific training mechanisms of these models [11, 12].

4 Counterfactual explanations of LVQ models

General approach: We aim for an efficient explicit formulation how to find counterfactuals, for diverse LVQ models. The specific form of LVQ models enables us to decompose the problem into a number of simpler ones as follows: Being a winner-takes all scheme, the nearest prototype $\vec{\mathrm{p}}_i$ of a counterfactual $\vec{x}'$ must be labeled $o_i = y'$. Hence a counterfactual $\vec{x}'$ of a given input $\vec{x}$ can be obtained by the following optimization problem

$$\underset{\vec{x}' \in \mathbb{R}^d}{\arg\min}\ \theta(\vec{x}', \vec{x}) \tag{5a}$$

$$\text{s.t. } \mathrm{d}(\vec{x}', \vec{\mathrm{p}}_i) \leq \mathrm{d}(\vec{x}', \vec{\mathrm{p}}_j) - \epsilon \quad \forall \vec{\mathrm{p}}_j \in \mathcal{P}(y') \tag{5b}$$

where $\mathcal{P}(y')$ denotes the set of all prototypes not labeled as y' and $\epsilon > 0$ is a small value preventing that the counterfactual lies exactly on the decision boundary. We solve this problem for each prototype $\vec{\mathrm{p}}_i$ with $o_i = y'$ and select the counterfactual $\vec{x}'$ yielding the smallest value of $\theta(\vec{x}', \vec{x})$ as solution. Note that problem Eq. (5) has always a feasible solution, the prototype $\vec{\mathrm{p}}_i$ itself. Furthermore, in contrast to Eq. (1), the formulation does not include hyperparameters.

For the weighted Manhattan distance as a regularization $\theta(\cdot)$, the objective Eq. (5a) becomes linear in $\vec{x}'$, where $\boldsymbol{\Upsilon}$ is the diagonal matrix with entries α_j and $\vec{\beta}$ is an auxiliary variable that can be discarded afterwards:

$$\min_{\vec{x}', \vec{\beta} \in \mathbb{R}^d} \vec{1}^\top \vec{\beta} \qquad \text{s.t. } \boldsymbol{\Upsilon}\vec{x}' - \boldsymbol{\Upsilon}\vec{x} \leq \vec{\beta}, \quad -\boldsymbol{\Upsilon}\vec{x}' + \boldsymbol{\Upsilon}\vec{x} \leq \vec{\beta}, \quad \vec{\beta} \geq \vec{0} \tag{6}$$

For the Euclidean distance Eq. (3) as regularization $\theta(\cdot)$, the objective Eq. (5a) can be written in a convex quadratic form:

$$\min_{\vec{x}' \in \mathbb{R}^d} \frac{1}{2}\vec{x}'^\top \vec{x}' - \vec{x}'^\top \vec{x} \tag{7}$$

In the subsequel we explore Eq. (5) for different regularizations $\theta(\cdot)$ and LVQ models, and investigate how to solve it efficiently.[1].

[1] See https://arxiv.org/abs/1908.00735 for details of the computation

ESANN 2020 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 2-4 October 2020, i6doc.com publ., ISBN 978-2-87587-074-2. Available from http://www.i6doc.com/en/.

Global quadratic form: For GMLVQ models, problem Eq. (5) becomes a linear program (LP) when using the weighted Manhattan distance as a regularizer, and a convex quadratic program (QP) problem when using the Euclidean distance. Both models can be solved efficiently and (up to equivalence) uniquely [15]. More precisely, the constraints Eq. (5b) can be written as a set of linear inequality constraints:

$$\vec{x}'^\top \vec{q}_{ij} + r_{ij} + \epsilon \leq 0 \quad \forall \vec{\mathrm{p}}_j \in \mathcal{P}(y') \tag{8}$$

where

$$\vec{q}_{ij} = \frac{1}{2}\left(\mathbf{\Omega}_j \vec{\mathrm{p}}_j - \mathbf{\Omega}_i \vec{\mathrm{p}}_i\right) \quad r_{ij} = \frac{1}{2}\left(\vec{\mathrm{p}}_i^\top \mathbf{\Omega}_i \vec{\mathrm{p}}_i - \vec{\mathrm{p}}_j^\top \mathbf{\Omega}_j \vec{\mathrm{p}}_j\right) \tag{9}$$

Local quadratic form: For LGMLVQ models with prototype specific distance matrix $\mathbf{\Omega}_{\mathrm{p}}$, the optimization problem Eq. (5) becomes a quadratically constrained quadratic program (QCQP) for Manhattan and Euclidean regularization, since the constraints Eq. (5b) become a set of quadratic constraints:

$$\frac{1}{2}\vec{x}'^\top \mathbf{Q}_{ij} \vec{x}' + \vec{x}'^\top \vec{q}_{ij} + r_{ij} + \epsilon \leq 0 \quad \forall \vec{\mathrm{p}}_j \in \mathcal{P}(y') \tag{10}$$

where

$$\mathbf{Q}_{ij} = \mathbf{\Omega}_i - \mathbf{\Omega}_j \tag{11}$$

Because we can not make any statement about the definiteness of $\mathbf{Q}_{ij}$, the constraints Eq. (10) are quadratic but non necessarily convex. Therefore, optimization might be challenging since non-convex QCQP is NP-hard in general [16]. However, there exist methods like the Suggest-Improve framework [16] that can efficiently find good solutions.

Experiments: We empirically confirm the efficiency of our proposed methods in comparison to black-box mechanisms by means of the following experiments: We use GLVQ, GMLVQ and LGMLVQ models with 3 prototypes per class for the "Breast Cancer Wisconsin (Diagnostic) Data Set" [17], the "Optical Recognition of Handwritten Digits Data Set" [18] and the "Ames Housing dataset" [19]. Thereby, we use PCA-preprocessing [2] to reduce the dimensionality of the digit data set to 10 and of the breast cancer data set to 5. We standardize the house data set and turned it into a binary classification problem[3] by setting the target to 1 if the price is greater or equal to 160k$ and 0 otherwise. The implementation of our proposed method for computing counterfactual explanations is available online[4]. We use the Suggest-Improve framework [16] for solving the non-convex

[2] This is for the purpose of better stability and better semantic meaning, since in the original domain already a small perturbation is sufficient for changing the class, since adversarial attacks exist even for linear functions in high dimensions if feature correlations are neglected. Since PCA can be approximately inverted, counterfactuals in PCA space can be lifted to the original data space.

[3] In addition, we select the following features: TotalBsmt, 1stFlr, 2ndFlr, GrLivA, WoodDeck, OpenP, 3SsnP, ScreenP and PoolA - When computing counterfactuals, we fix the last five features.

[4] `https://github.com/andreArtelt/efficient_computation_counterfactuals_lvq`

Data set	Breast cancer			Handwritten digits			House prices		
Method	DS	CMA	Ours	DS	CMA	Ours	DS	CMA	Ours
GLVQ	3.26	3.28	**1.96**	6.51	6.53	**3.99**	3.81	3.85	**3.32**
GMLVQ	2.71	6.49	**2.46**	21.34	11.63	**4.40**	5.06	8.63	**3.78**
LGMLVQ	*2.00*	*1.61*	**1.57**	*8.12*	*7.88*	**7.53**	*12.74*	*12.59*	**8.20**

Table 1: Mean Manhattan distance between the counterfactual and the original data point - best (smallest) values are **highlighted**. For LGMLVQ with DS or CMA-ES (marked italic), in 5% to 60% of the cases no solution was found.

QCQPs, where we pick the target prototype as initial solution in the Suggest-step and we use the penalty convex-concave procedure (CCP) in the Improve-step [16]. For comparison, we use the optimizer for computing counterfactual explanations of LVQ models as implemented in ceml [20] - where the distance to the nearest prototype with the requested label y' is minimized by Downhill-Simplex search or CMA-ES.

We report results for the Manhattan distance as regularizer - we used the Manhattan distance for enforcing a sparse solution. For each possible combination of model, data set and method, a 4-fold cross validation is conducted and the mean distance is reported. The results are listed in Table 1. In all cases, our method yields counterfactuals that are closer to the original data point than the one found by minimizing the original cost function Eq. (1) with Downhill-Simplex search (DS) or CMA-ES. In addition, our method is between 1.5 and 158.0 faster in comparison to DS/CMA-ES method. Furthermore, Downhill-Simplex and CMA-ES did not always find a counterfactual when dealing with LGMLVQ models. We would like to remark that our formulation can easily be extended by linear/quadratic constraints which can incorporate prior knowledge such as a maximum possible change of specific input features - see Table 2 for an example. Such extensions do not change the form of the optimization problem hence its complexity.

5 Conclusion

We proposed, and empirically evaluated, model- and regularization-dependent convex and non-convex programs for efficiently computing counterfactual explanations of LVQ models. We found that in many cases we get either a set of linear or convex quadratic programs which both can be solved efficiently. Only in the case of localized LVQ models we have to solve a set of non-convex quadratically constrained quadratic programs - we found that they can be efficiently approximately solved by using the Suggest-Improve framework.

Data point	TotalBsmt	1stFlr	2ndFlr	GrLivA	Label
Original	0	1120	468	1588	1
Counterfactual	0	366	1824	2225	0
Constrained Counterfactual	373	1454	1454	3125	0

Table 2: House prices, we obtain a "plausible" counterfactual by adding constrains, here the constraint "2ndFlr $\leq$ 1stFlr" is added.

References

[1] Amir E. Khandani, Adlar J. Kim, and Andrew Lo. Consumer credit-risk models via machine-learning algorithms. *Journal of Banking & Finance*, 34(11):2767–2787, 2010.

[2] European parliament and council. General data protection regulation. `https://eur-lex.europa.eu/eli/reg/2016/679/oj`, 2016.

[3] Leilani H. Gilpin, David Bau, Ben Z. Yuan, Ayesha Bajwa, Michael Specter, and Lalana Kagal. Explaining explanations: An overview of interpretability of machine learning. In *5th IEEE International Conference on Data Science and Advanced Analytics, DSAA 2018, Turin, Italy, October 1-3, 2018*, pages 80–89, 2018.

[4] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. A survey of methods for explaining black box models. *ACM Comput. Surv.*, 51(5):93:1–93:42, August 2018.

[5] A. Aamodt and E. Plaza. Case-based reasoning: Foundational issues, methodological variations, and systemapproaches. *AI communications*, 1994.

[6] Sandra Wachter, Brent D. Mittelstadt, and Chris Russell. Counterfactual explanations without opening the black box: Automated decisions and the GDPR. *CoRR*, abs/1711.00399, 2017.

[7] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Dino Pedreschi, Franco Turini, and Fosca Giannotti. Local rule-based explanations of black box decision systems. *CoRR*, abs/1805.10820, 2018.

[8] Shubham Sharma, Jette Henderson, and Joydeep Ghosh. CERTIFAI: counterfactual explanations for robustness, transparency, interpretability, and fairness of artificial intelligence models. *CoRR*, abs/1905.07857, 2019.

[9] Thibault Laugel, Marie-Jeanne Lesot, Christophe Marsala, Xavier Renard, and Marcin Detyniecki. Comparison-based inverse classification for interpretability in machine learning. In *Information Processing and Management of Uncertainty in Knowledge-Based Systems. Theory and Foundations - 17th International Conference, IPMU 2018, Cádiz, Spain, June 11-15, 2018, Proceedings, Part I*, pages 100–111, 2018.

[10] David Nova and Pablo A. Estévez. A review of learning vector quantization classifiers. *Neural Comput. Appl.*, 25(3-4):511–524, September 2014.

[11] Petra Schneider, Michael Biehl, and Barbara Hammer. Adaptive relevance matrices in learning vector quantization. *Neural Computation*, 21(12):3532–3561, 2009. PMID: 19764875.

[12] Petra Schneider, Michael Biehl, and Barbara Hammer. Distance learning in discriminative vector quantization. *Neural Computation*, 21(10):2942–2969, 2009. PMID: 19635012.

[13] Stephan Kirstein, Heiko Wersing, Horst-Michael Gross, and Edgar Körner. A life-long learning vector quantization approach for interactive learning of multiple categories. *Neural networks : the official journal of the International Neural Network Society*, 28:90–105, 04 2012.

[14] Christoph Molnar. *Interpretable Machine Learning*. 2019. `https://christophm.github.io/interpretable-ml-book/`.

[15] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, NY, USA, 2004.

[16] Jaehyun Park and Stephen Boyd. General heuristics for nonconvex quadratically constrained quadratic programming. *arXiv preprint arXiv:1703.07870*, 2017.

[17] Olvi L. Mangasarian William H. Wolberg, W. Nick Street. Breast cancer wisconsin (diagnostic) data set. `https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic)`, 1995.

[18] E. Alpaydin and C. Kaynak. Optical recognition of handwritten digits data set. `https://archive.ics.uci.edu/ml/datasets/Optical+Recognition+of+Handwritten+Digits`, 1998.

[19] Dean De Cock. Ames, iowa: Alternative to the boston housing data as an end of semester regression project. *Journal of Statistics Education*, 19(3), 2011.

[20] André Artelt. Ceml: Counterfactuals for explaining machine learning models - a python toolbox. `https://www.github.com/andreArtelt/ceml`, 2019.