Robust Counterfactual Explanations for Privacy-Preserving SVM

Anonymous Authors[1]

Abstract

We consider counterfactual explanations for privacy-preserving support vector machines (SVM), where the privacy mechanism that publicly releases the classifier guarantees differential privacy. While privacy preservation is essential when dealing with sensitive data, there is a consequent degradation in the classification accuracy due to the introduced perturbations in the classifier weights. Therefore, counterfactual explanations need to be made robust against such perturbations in order to ensure, with high confidence, that the explanations are valid. In this work, we suitably model the uncertainties in the SVM weights and formulate the robust counterfactual explanation problem. Then, we study optimal and efficient suboptimal algorithms for its solution. Experimental results illustrate the connections between privacy levels, classifier accuracy, and the confidence levels that validate the counterfactual explanations.

1. Introduction

Despite their efficiency in solving complex problems, machine learning (ML) algorithms and models are seldom value-neutral to the extent that they include social and ethical values. Even when such values are integrated into the models they may be mandated by regulatory frameworks, such as traditional laws or policy documents. This paper aims to illustrate the relational nexus between social and ethical values in a technical context. This is done by focusing on three values advocated by the General Data Protection Regulation (GDPR) (Reg, 2016), namely, *explainability*,[1] *privacy*,[2] and *accuracy*.[3] What becomes apparent when

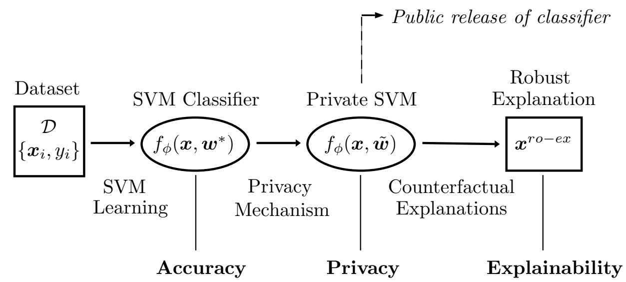

Figure 1. Illustration for the relationship between accuracy, privacy, and explainability considered in this work.

attempting to transform the above social and ethical values from the natural language of the law into the mathematical language of ML algorithms is that this may be challenging and even technically unattainable. The above social and ethical values have been chosen as their transformation into ML rules clearly illuminates the challenge of aligning these competing social and ethical values promoted by the law into a technical format, a conclusion being that the simultaneous promotion of all these three values is potentially mathematically unattainable.

Figure 1 gives an overview on how the three mentioned social values are related within this work: Accuracy is targeted when learning an SVM classifier from a dataset. Privacy is guaranteed through the privacy preserving mechanism. The explainability of predictions is done by constructing *counterfactual explanations* for each specific data instance. Counterfactual explanations (Sandra Wachter, 2018; Molnar, 2019) is a class of *post hoc* explainability methods that quantify the necessary changes to a considered data instance to change its classification.

In this work, we will propose counterfactual explanations that exploit the characteristics of the SVM classifier as well as the applied privacy mechanism. The privacy mechanism proposed in (Rubinstein et al., 2012) perturbs the SVM weights through additive Laplace noise. As a result, privacy

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

[1]References to this social value can be found in Recital 71.

[2]References to this social value can be found in Article 25.

[3]References to this social value can be found in Article 5(1)(d) as expanded upon by the Article 29 Data Protection Working Party (Guidelines on Automated individual decision-making and Profiling for the purposes of Regulation 2016/679, Adopted on 3 October 2017). It is also noteworthy that an in-depth discussion of what exactly the social values referred to in footnotes 2, 3 and 4 actually entail is beyond the confines of this paper.

is achieved by establishing uncertainty about the true classifier weights. For constructing explanations, we suitably model the uncertainty in the SVM weights through random variables. Then, we formulate counterfactual explanations as a optimization problem with probabilistic constraints (Shapiro et al., 2014), and characterize its deterministic equivalent. For linear SVMs, the deterministic problem is a convex second-order cone program (SOCP). For the non-linear SVM case, we propose an efficient sub-optimal algorithm to find robust explanations utilizing the existence of class specific prototypes. Experimental results illustrate the trade-offs between accuracy, privacy, and explainability.

2. Preliminaries

In this section, we will describe the dataset and the SVM learning problem. Then, we will review the privacy preserving mechanism proposed in (Rubinstein et al., 2012).

Consider a dataset $\mathcal{D}$ consisting of a collection of n tuples

$$(\boldsymbol{x}_i, y_i), \quad i = 1, \ldots, n, \tag{1}$$

where each tuple $(\boldsymbol{x}_i, y_i)$ consists of a *features vector* $\boldsymbol{x}_i \in \mathbb{R}^L$ and its associated *class label* $y_i \in \{-1, 1\}$. Dataset $\mathcal{D}$ is used to learn an SVM classifier (Hastie et al., 2009) that can efficiently separate the two classes of data points through a separating hyperplane. The optimization problem for SVM with hinge loss and parameter $C \geq 0$ is:

$$\min_{\boldsymbol{w} \in \mathbb{R}^F} \frac{1}{2}\|\boldsymbol{w}\|^2 + \frac{C}{n}\sum_{i=1}^{n}[1 - y_i f_\phi(\boldsymbol{x}_i, \boldsymbol{w})]_+, \tag{2}$$

where the weights $\boldsymbol{w}$ geometrically correspond to the vector perpendicular to the separating hyperplane, $[a]_+ := \max\{0, a\}$, and f_ϕ is the *classifier* function:

$$f_\phi(\boldsymbol{x}, \boldsymbol{w}) := \phi(\boldsymbol{x})^\top \boldsymbol{w}. \tag{3}$$

Here, the *feature mapping* $\phi : \mathbb{R}^L \to \mathbb{R}^F$, $F \geq L$, enlarges the feature space of the data points to improve the separability of the two classes of data points through a hyperplane (Hastie et al., 2009). We assume in this work that F is finite.

The minimization problem defined in Eq. (2) can be formulated as a quadratic program and solved efficiently.[4] Let, $\boldsymbol{w}^*$ be the optimal solution to this problem, then the *binary classification* of a data point $\boldsymbol{x}$ is the sign of $f_\phi(\boldsymbol{x}, \boldsymbol{w}^*)$.

From Eq. (3) it can be observed that in order to perform SVM classification, all we need is $\boldsymbol{w}^*$ and the feature mapping ϕ. In applications where the dataset includes sensitive information, the public release of the SVM classifier may lead to privacy breaches through publishing $\boldsymbol{w}^*$. Therefore, it is required to apply a privacy preserving mechanism before publishing the classifier, as is shown in Figure 1.

[4] If the number of features F is much larger than the number of data points n, then it is more efficient to solve the dual problem.

We will use the privacy preserving mechanism proposed in (Rubinstein et al., 2012) for SVMs with finite dimensional feature mappings. This mechanism guarantees differential privacy by perturbing the SVM optimal weights $\boldsymbol{w}^* \in \mathbb{R}^F$ through additive Laplace noise. Formally, let $M : \mathfrak{D} \to \mathcal{R}$ be a *randomized mechanism*, where $\mathfrak{D}$ is the set of all datasets and $\mathcal{R}$ is the response set of the mechanism M (defined as the solution space of the SVM problem). Define *neighboring datasets* as the datasets in $\mathfrak{D}$ that differ by one data point entry. Then, for a given $\beta > 0$, a mechanism M provides β-differential privacy (Dwork & Roth, 2014) if for any two neighboring datasets $\mathcal{D}_1, \mathcal{D}_2 \in \mathfrak{D}$ and all subsets $\mathcal{S} \subseteq \mathcal{R}$ it holds $\Pr[M(\mathcal{D}_1) \in \mathcal{S}] \leq \exp(\beta) \Pr[M(\mathcal{D}_2) \in \mathcal{S}]$.

From Theorem 10 in (Rubinstein et al., 2012), the perturbed SVM weight vector

$$\tilde{\boldsymbol{w}} := \boldsymbol{w}^* + \boldsymbol{\mu}, \tag{4}$$

where $\boldsymbol{\mu}$ is a vector of iid Laplace random variables

$$\mu_i \sim \text{Lap}(0, \lambda), i = 1, \ldots, F, \tag{5}$$

guarantees $\beta-$differential privacy for $\lambda \geq 4C\kappa\sqrt{F}/(\beta n)$, where κ satisfies $\phi(\boldsymbol{x})^\top\phi(\boldsymbol{x}) \leq \kappa^2$ for all $\boldsymbol{x} \in \mathbb{R}^L$.[5]

In the following, we will assume that the following information is available for calculating the counterfactual explanations: the SVM weights $\tilde{\boldsymbol{w}}$, the data-independent details for constructing ϕ, and the noise scale λ.

3. Robust Counterfactual Explanation

The concept of counterfactual explanations was proposed in (Sandra Wachter, 2018) for general ML classifiers. The following definition corresponds to binary SVM classifiers: Given an SVM classifier with weight vector $\boldsymbol{w}$, a counterfactual explanation for the classification y' of a given data instance $\boldsymbol{x}'$ is the solution of

$$\min_{\boldsymbol{x} \in \mathbb{R}^L} \; d(\boldsymbol{x}, \boldsymbol{x}') \quad s.t. \; y' f_\phi(\boldsymbol{x}, \boldsymbol{w}) \leq 0, \tag{6}$$

where $d(\boldsymbol{x}, \boldsymbol{x}')$ is a distance between $\boldsymbol{x}$ and $\boldsymbol{x}'$ and $f_\phi(\boldsymbol{x}, \boldsymbol{w})$ is defined in Eq. (3). In words, a counterfactual explanation is the closest point to $\boldsymbol{x}'$, in the sense of d, which has a different class than y'.

In Figure 2, we illustrate different counterfactual explanations for two linear SVM classifiers, one with optimal

[5] By perturbing the optimal weight vector, the accuracy of the SVM classifier will be degraded. For this purpose, it is important to deliver guarantees on the classification accuracy by upper bounding the noise scale λ. This is done in (Rubinstein et al., 2012) by introducing a condition called (ϵ, δ)-useful mechanism. We will rely on experimental validation for the accuracy and omit the description of the theoretical bounds here due to space constraints.

weights $\boldsymbol{w}^*$ and one with perturbed weights $\tilde{\boldsymbol{w}}$. For both optimal and perturbed classifiers, the explanations are the closest points to the instance and lie on the respective decision boundaries. It can be seen that the non-robust explanation on the perturbed boundary is closer to the instance compared to the optimal explanation and thus also has the same classification as the instance when using the optimal classifier. Hence, the non-robust explanation may not be credible or valid. We will next study robust explanations which take into account the perturbations.

According to Eq. (4), the private SVM mechanism releases noisy versions of the optimal $\boldsymbol{w}^*$. Thus, there exists uncertainty about the correctness of the classification with $\tilde{\boldsymbol{w}}$, which diminishes the effectiveness of the counterfactual explanation unless this uncertainty is taken into account. Therefore, we will model the uncertainty about $\boldsymbol{w}^*$ through the random vector $\boldsymbol{\xi} = \tilde{\boldsymbol{w}} - \boldsymbol{\mu}$. From Eq. (5), it follows that

$$\boldsymbol{\xi} \sim \mathrm{mvLap}\big(\tilde{\boldsymbol{w}}, 2\lambda^2 \boldsymbol{I}\big), \tag{7}$$

where $\mathrm{mvLap}(\boldsymbol{l}, \boldsymbol{\Sigma})$ is the multivariate Laplace distribution with location $\boldsymbol{l}$ and covariance $\boldsymbol{\Sigma}$. Subsequently, we can formulate the *robust counterfactual explanation* problem as

$$\min_{\boldsymbol{x} \in \mathbb{R}^L} \; d(\boldsymbol{x}, \boldsymbol{x}') \quad s.t. \; \Pr\left[y' f_\phi(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0\right] \geq p, \tag{8}$$

where we have replaced the constraint in Eq. (6) with a probabilistic constraint. The next result provides a reformulation for the constraint above.

Proposition 1. *The deterministic equivalent of the probabilistic constraint in Eq.* (8)*, with* $p \in [1/2, 1]$*, is*

$$\underbrace{y' f_\phi(\boldsymbol{x}, \tilde{\boldsymbol{w}}) - \lambda\sqrt{2}\ln(2(1-p))\|\phi(\boldsymbol{x})\|}_{g(\boldsymbol{x})} \leq 0. \tag{9}$$

Proof. From Eq. (7), the multivariate Laplace distribution $\mathrm{mvLap}\big(\tilde{\boldsymbol{w}}, 2\lambda^2 \boldsymbol{I}\big)$ is *symmetric* since the variance does not depend on the mean. A symmetric multivariate Laplace distribution is *elliptically symmetric* (Kotz et al., 2001). Consequently, the structure of Eq. (9) follows from Lemma 2.2 in (Henrion, 2007), and for the multivariate Laplace distribution, the derivation follows similar steps as in Example 2.2 in (Peng, 2019). □

The left hand side of Eq. (9) includes two terms: The first term is the same as in the constraint in Eq. (6) and requires that the solution of the problem has a different class than y'. The second term establishes robustness by enforcing stronger confidence in the SVM prediction, i.e., larger $|f_\phi(\boldsymbol{x}, \tilde{\boldsymbol{w}})|$. Notice that for $p = 0.5$, this second term is zero, i.e., the constraint becomes identical to that of the non-robust case.

For linear SVM, i.e., $\phi(\boldsymbol{x}) = \boldsymbol{x}$, the constraint in Eq. (9) can be rewritten as $\|\boldsymbol{x}\| \leq \frac{y'}{\lambda\sqrt{2}\ln(2(1-p))}\boldsymbol{x}^\top \tilde{\boldsymbol{w}}$, which is a convex second order cone constraint. Considering a convex distance function d in its first argument, then the robust counterfactual explanation problem in Eq. (8) for linear SVM can be solved efficiently using convex optimization solvers. For this work, we use CVXPY (Diamond & Boyd, 2016; Agrawal et al., 2018), and Figure 2 shows the explanations found by solving this problem with $d(\boldsymbol{x}, \boldsymbol{x}') = \|\boldsymbol{x} - \boldsymbol{x}'\|$.

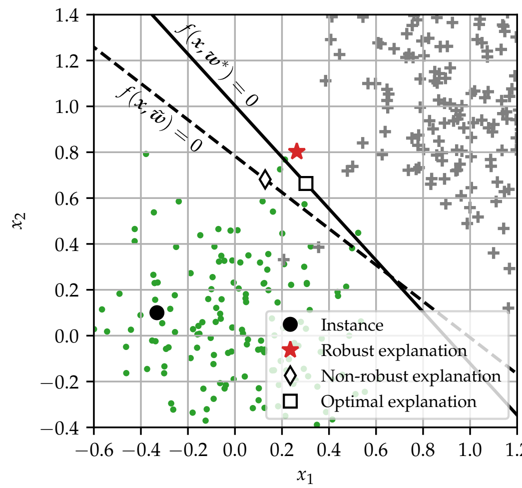

Figure 2. Illustration for SVM and private SVM linear classifications and the associated explanations using the Euclidean norm as distance measure. The data points are generated from two bivariate Guassian distributions with means $[0, 0]$ and $[1, 1]$, and same covariance $0.1\boldsymbol{I}$.

For non-linear SVM, the problem is generally not convex. Therefore, we will next consider a suboptimal solution that can be computed efficiently. Notice that a *root* for the function g defined in Eq. (9) would qualify as a robust explanation since it satisfies the constraint in Eq. (9) with equality. In order to find a root for g, we will use the *bisection method* (McNamee & Pan, 2013). As a prerequisite, this method requires two input data points that have different classes. Clearly, for the given data instance $\boldsymbol{x}'$, $g(\boldsymbol{x}')$ is positive. The second required input vector should necessarily be of opposite class in order for g to be negative. We will discuss next the availability of such input that we will here refer to as a *prototype* (Looveren & Klaise, 2019).

Unlike in (Looveren & Klaise, 2019), we do not have access to test data to construct these prototypes due to privacy issues. However, we argue that if we consider prototypes as representatives of their classes, the "domain expert" that provides the explanations should be able to estimate these from experience and knowledge of the data for each class. If this is not the case, we assume that the prototypes can be constructed by generating random data instances and studying their classification. The description of the well known bisection method is relegated to the Appendix.

4. Experimental Results

We illustrate our approach by using the publicly available UCI Breast Cancer Wisconsin (Diagnostic) dataset (Dua & Graff, 2017). The dataset includes 569 instances, each with 30 features and the binary diagnosis: benign (class -1) or malignant (class 1). This dataset is one of several datasets typically used when evaluating privacy preserving algorithms, e.g. (Farokhi, 2020). The code to reproduce all the figures is available at (Mochaourab, 2021).

We randomly split the dataset once into a training (70% of total) and a test set (30% of total). The results were qualitatively similar for different random splits with same splitting ratio. Moreover, we normalize the training data to have zero mean and unit variance, and the calculated normalization parameters are applied to the test data. Next, a feature mapping ϕ is generated using the Radial Basis Function (RBF) kernel approximation in (Rahimi & Recht, 2007) with dimensions $F = 100$.[6] For the implementation of the feature mapping, we have used the library in (Atarashi, 2019). The SVM classifiers learned for the plots are trained using the training set and their performance measured on the test set. The distance function used for the counterfactuals in Eq. (6), is the Eucleadian norm, i.e., $d(\boldsymbol{x}, \boldsymbol{x}') = \|\boldsymbol{x} - \boldsymbol{x}'\|$. The prototypes are selected as the data mean of each class. For calculating the average performance in the plots, we use 10^4 random realizations of Laplace noise.

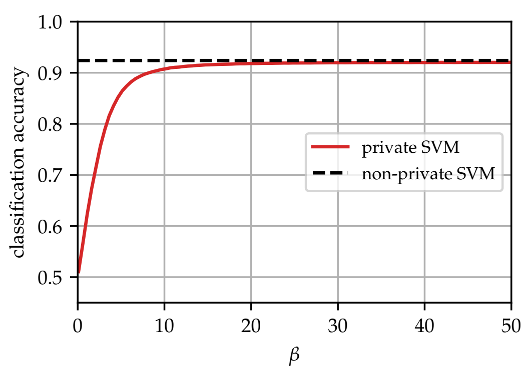

Figure 3. Tradeoff between average accuracy and privacy.

Figure 3, depicts the trade-off between average accuracy and privacy of the private SVM. The dashed line corresponds to the non-private case in which the SVM weights are not perturbed with noise. The average accuracy for the private SVM is lowest (≈ 0.5) for high privacy levels (very small β), and monotonically increases with β to eventually converge to the non-private average performance.

[6]Note that we have assumed finite dimensional feature mappings and hence we do not explicitly consider the approximation error in the feature mapping in relation to using the RBF kernel as is done in Section 4 in (Rubinstein et al., 2012) for the general case of translation-invariant kernels.

The average distance between the counterfactual explanation and the instance is calculated depending on β (for $p = 0.9$) in Figure 4, and depending on p (for $\beta = 0.5$) in Figure 5. This average distance for robust counterfactual explanations is high for small values of β, as is shown in Figure 4. This is due to the large uncertainty through the large noise variance. The non-robust explanations, which correspond to low confidence value of $p = 0.5$, have similar average distance as for non-private SVM since the noise has zero mean. In Figure 5, the tradeoff between confidence p and the average distance are shown. For large confidence values p, the robust explanation converges to the prototype data point, and is furthest away from the instance.

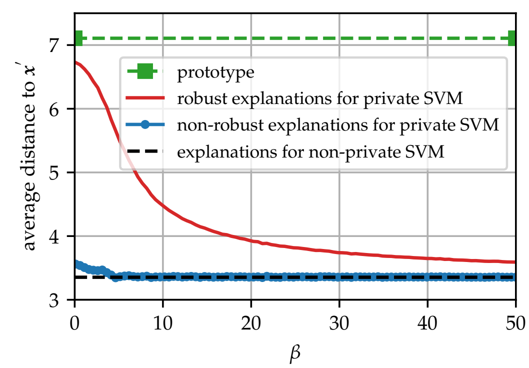

Figure 4. Average distance from explanation to instance ($p = 0.9$).

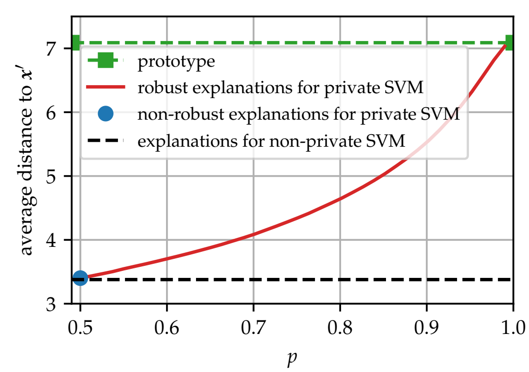

Figure 5. Average distance from explanation to instance ($\beta = 5$).

5. Conclusions

The above findings highlight the difficulties associated with embedding the social and ethical values mandated by regulatory instruments into ML algorithms. An ensuing conclusion is that a conscious decision may be required to promote one social value at the expense of another, the context in which the technology is being operated potentially being a deciding factor. These issues are highlighted in this work through the joint study of privacy and counterfactual explanations that are valid within desired levels of confidence.

References

Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation). OJ L 119, 4.5.2016, 2016.

Agrawal, A., Verschueren, R., Diamond, S., and Boyd, S. A rewriting system for convex optimization problems. *Journal of Control and Decision*, 5(1):42–60, 2018.

Atarashi, K. pyrfm: A library for random feature maps in python, 2019. URL https://neonnnnn.github.io/pyrfm/.

Diamond, S. and Boyd, S. CVXPY: A Python-embedded modeling language for convex optimization. *Journal of Machine Learning Research*, 17(83):1–5, 2016.

Dua, D. and Graff, C. UCI machine learning repository. University of California, Irvine, School of Information and Computer Sciences, 2017. URL http://archive.ics.uci.edu/ml.

Dwork, C. and Roth, A. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.*, 9(3–4):211–407, August 2014. ISSN 1551-305X.

Farokhi, F. Privacy-preserving public release of datasets for support vector machine classification. *IEEE Transactions on Big Data*, pp. 1–1, 2020. doi: 10.1109/TBDATA.2019.2963391.

Hastie, T., Tibshirani, R., and Friedman, J. *The elements of statistical learning: data mining, inference and prediction*. Springer Series in Statistics. Springer-Verlag New York, 2 edition, 2009.

Henrion, R. Structural properties of linear probabilistic constraints. *Optimization*, 56:425 – 440, 2007.

Kotz, S., Kozubowski, T. J., and Podgórski, K. *Symmetric Multivariate Laplace Distribution*, chapter 5, pp. 231–238. Birkhäuser Boston, Boston, MA, 2001. ISBN 978-1-4612-0173-1.

Looveren, A. V. and Klaise, J. Interpretable counterfactual explanations guided by prototypes. *CoRR*, abs/1907.02584, 2019. URL http://arxiv.org/abs/1907.02584.

McNamee, J. and Pan, V. Chapter 7 - Bisection and Interpolation Methods. In McNamee, J. and Pan, V. (eds.), *Numerical Methods for Roots of Polynomials - Part II*, volume 16 of *Studies in Computational Mathematics*, pp. 1–138. Elsevier, 2013.

Mochaourab, R. Robust-Explanation-SVM, 2021. https://github.com/rami-mochaourab/robust-explanation-SVM.

Molnar, C. *Interpretable Machine Learning - A Guide for Making Black Box Models Explainable*. 2019. https://christophm.github.io/interpretable-ml-book/.

Peng, S. *Chance constrained problem and its applications*. Theses, Université Paris Saclay (COmUE) ; Xi'an Jiaotong University, June 2019. URL https://tel.archives-ouvertes.fr/tel-02303045.

Rahimi, A. and Recht, B. Random features for large-scale kernel machines. In *Proceedings of the 20th International Conference on Neural Information Processing Systems*, NIPS'07, pp. 1177–1184, Red Hook, NY, USA, 2007. Curran Associates Inc. ISBN 9781605603520.

Rubinstein, B. I. P., Bartlett, P. L., Huang, L., and Taft, N. Learning in a large function space: Privacy-preserving mechanisms for SVM learning. *Journal of Privacy and Confidentiality*, 4(1), July 2012.

Sandra Wachter, Brent Mittelstadt, C. R. Counterfactual explanations without opening the black box: Automated decisions and the GDPR. *Harvard Journal of Law & Technology, forthcoming*, 31(2), 2018.

Shapiro, A., Dentcheva, D., and Ruszczyński, A. *Lectures on Stochastic Programming: Modeling and Theory, Second Edition*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2014.

Appendix: Bisection and Further Results

Algorithm 1 Bisection method for finding an explanation

Input: Data instance $(\boldsymbol{x}', y')$, prototype $\boldsymbol{z}_{-y'}$.
Initialize: $\boldsymbol{x}^{ub} = \boldsymbol{z}_{-y'}, \boldsymbol{x}^{lb} = \boldsymbol{x}'$
while $\|\boldsymbol{x}^{ub} - \boldsymbol{x}^{lb}\| > \epsilon$ **do**
  $\boldsymbol{x} \leftarrow (\boldsymbol{x}^{ub} + \boldsymbol{x}^{lb})/2$
  if $g(\boldsymbol{x}) < 0$ **then**
    $\boldsymbol{x}^{ub} \leftarrow \boldsymbol{x}$
  else
    $\boldsymbol{x}^{lb} \leftarrow \boldsymbol{x}$
Output: $\boldsymbol{x}^{ro-ex} \leftarrow \boldsymbol{x}$.

The steps for the bisection method are described in Algorithm 1. The lower and upper bounds for bisection are initialized according to the given data instance and the prototype from the opposite class, respectively. Here, the prototypes for class 1 and -1 are $\boldsymbol{z}_1$ and $\boldsymbol{z}_{-1}$, respectively. In the process of finding these prototypes, it is desired that the classification of these points has sufficient confidence, i.e., $|f_\phi(\boldsymbol{z}_y, \tilde{\boldsymbol{w}})| \geq -\lambda\sqrt{2}\ln(2(1-p))\|\phi(\boldsymbol{z}_y)\|$, for $y \in \{1, -1\}$. In each iteration of Algorithm 1, we check the classification of the midpoint of the interval between the upper and lower bounds. If this class is the same as the lower bound, then we replace the lower bound by the midpoint. Otherwise, we replace the upper bound. These steps are performed until the distance between the upper and lower bounds is lower than the threshold ϵ. The algorithm has linear convergence since the distance between the bounds is halved in each iteration.

Figure 6 shows the low number of iterations needed for Algorithm 1 to converge. Here, we set $\beta = 5, p = 0.9$. Then, we select a random instance $\boldsymbol{x}'$ from the test set with label $y' = 1$ (malignant), and apply Algorithm 1 to calculate an explanation $\boldsymbol{x}^{ro-ex}$ for its classification. The found explanation $\boldsymbol{x}^{ro-ex}$ quantifies the changes to each feature of $\boldsymbol{x}'$ in order to change the classifier prediction. Figure 7 shows these changes normalized over the instance's feature values. For example, for the selected instance, the explanation shows that feature number 19 needs to increase by around half its value, while other feature values need to be halved to alter the prediction from malignant to benign.

Clearly, it is desirable to find counterfactual explanations that are as close as possible to the instance to explain. Still, as we observe in Figure 4 and Figure 5, robust explanations are further away compared to the non-robust explanations, showing that privacy degrades the *quality* of explanations. The reason for that is, non-robust explanations violate the constraint in Eq. (6) with probability 0.5, while the robust explanations violate this constraint with probability p (which we here set to 0.9). This constraint violation is further studied in Figure 8 and Figure 9. There, the summary statistics for the left hand side of the constraint in (6) are plotted for the non-robust and robust explanations, respectively. These plots highlight the importance for considering robust explanations. Notice that the flattening of the 50-th percentile curve (Figure 9) for β less than around 4 is due to the convergence of the explanation to the prototype.

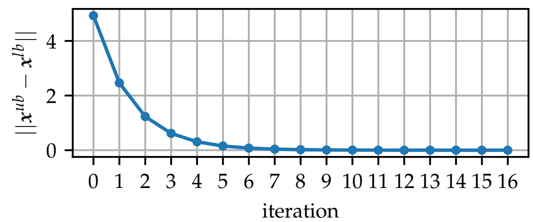

Figure 6. Convergence of Algorithm 1.

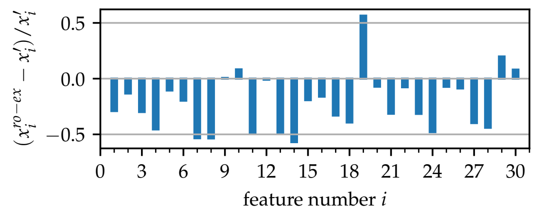

Figure 7. The counterfactual explanation quantifies the necessary changes in the instance's features to alter the prediction.

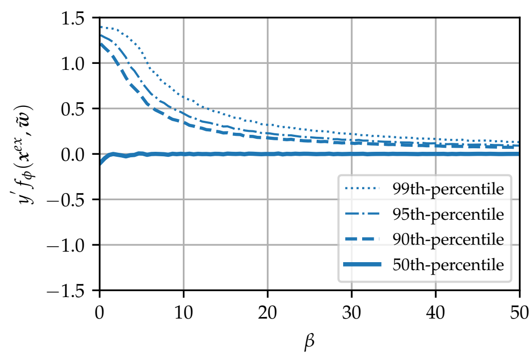

Figure 8. Constraint violation by non-robust explanations

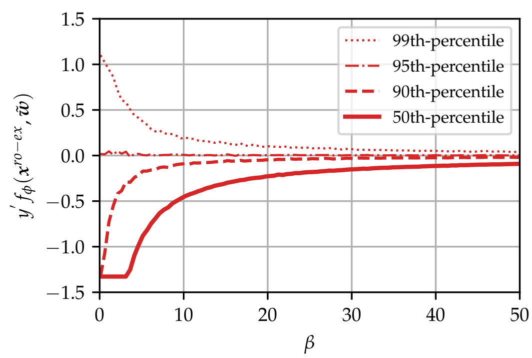

Figure 9. Constraint violation by robust explanations