

Robust Explanations for Private Support Vector Machines*

Rami Mochaourab[†] Sugandh Sinha[‡] Stanley Greenstein[§]
Panagiotis Papapetrou[¶]

Abstract

We consider counterfactual explanations for private 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. For such classifiers, counterfactual explanations need to be robust against the uncertainties in the SVM weights in order to ensure, with high confidence, that the classification of the data instance to be explained is different than its explanation. We model the uncertainties in the SVM weights through a random vector, and formulate the explanation problem as an optimization problem with probabilistic constraint. Subsequently, we characterize the problem's deterministic equivalent and study its solution. For linear SVMs, the problem is a convex second-order cone program. For non-linear SVMs, the problem is non-convex. Thus, we propose a sub-optimal solution that is based on the bisection method. The results show that, contrary to non-robust explanations, the quality of explanations from the robust solution degrades with increasing privacy in order to guarantee a prespecified confidence level for correct classifications.

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) [1], namely, *explainability*, *privacy*, and *accuracy*.¹ What becomes apparent

*This paper is under consideration at Pattern Recognition Letters.

[†]Digital Systems Division, RISE Research Institutes of Sweden, rami.mochaourab@ri.se

[‡]Digital Systems Division, RISE Research Institutes of Sweden, sugandh.sinha@ri.se

[§]Dept. of Law, Stockholm University, stanley.greenstein@juridicum.su.se

[¶]Dept. of Computer and Systems Sciences, Stockholm University, panagiotis@dsv.su.se

¹References to these social values can be found in Recital 71, Article 25 and Article 5(1)(d) as expanded by the Article 29 Data Protection Working Party Adopted on 3 October 2017

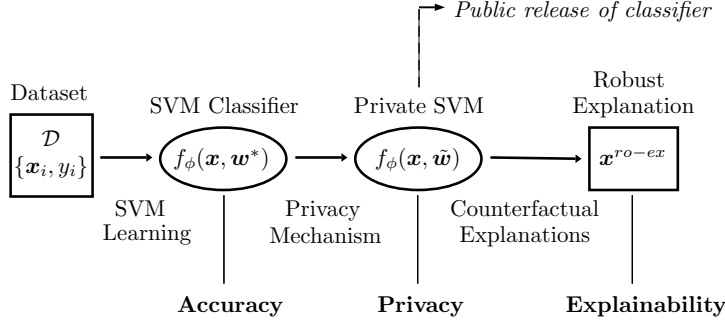


Figure 1: Illustration of the relationship between accuracy, privacy, and explainability associated with an SVM classifier trained on a dataset \mathcal{D} . A privacy mechanism is applied to preserve the privacy of the data before the public release of the SVM classifier. Explanations need to suitably take into account the introduced uncertainty through the privacy mechanism.

when 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:

1. Accuracy is targeted when learning an SVM classifier through the data from a dataset.
2. Privacy is guaranteed by applying a privacy preserving mechanism. We will use the mechanism developed in [2] which guarantees a strong notion of privacy called *differential privacy* [3].
3. Explainability of predictions for specific data instances enhances the transparency and the trust in the classifier predictions. We will consider *counterfactual explanations* [4, 5], which is a class of explainability methods that quantify the necessary changes to a considered data instance in order to change its classification. Such explainability methods fall under the category of *post hoc* interpretations [6, 7], since they do not require model retraining.

In this work, we will propose robust counterfactual explanations that *exploit the characteristics of the SVM classifier and the applied privacy mechanism*. The privacy mechanism proposed in [2] perturbs the SVM weights through additive Laplace noise. As a result, privacy is achieved by establishing uncertainty about the true classifier weights. For constructing explanations, we

respectively. It is also noteworthy that an in-depth discussion of what exactly these social values entail is beyond the confines of this paper.

suitably model the uncertainty in the SVM weights through random variables. Then, we formulate counterfactual explanations as a optimization problem with probabilistic constraints [8], and characterize its deterministic equivalent problem. For linear SVMs, the deterministic problem is a second-order cone program (SOCP) and can be solved efficiently. For the non-linear SVM case, the problem is non-convex. We therefore propose an efficient sub-optimal algorithm to find robust explanations utilizing the existence of class specific prototypes. The use of prototypes in the context of counterfactual explanations has been previously applied in [9] for the purpose of ensuring that explanations lie within the data domain and can be computed efficiently.

Experimental results illustrate the trade-offs between accuracy, privacy, and explainability using the UCI Breast Cancer Wisconsin Dataset [10]. In particular, we show that, contrary to non-robust explanations, the changes to an instance required by its robust explanation increase with the level of privacy. This degradation in the explanation quality occurs in order to guarantee, with prespecified confidence, that the explanations are counterfactuals, i.e., their classification is different than the instance, with respect to the non-private and unknown classifier. To the best of our knowledge, this is the first work to study the design of explanations for privacy preserving classifiers.

2 Preliminaries

In this section, we will describe the dataset and the SVM learning problem. Then, we will review the privacy preserving mechanism for SVM proposed in [2].

2.1 Dataset and SVM Classification

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

$$(\mathbf{x}_i, y_i), \quad i = 1, \dots, n, \quad (1)$$

where each tuple (\mathbf{x}_i, y_i) consists of a *features vector* $\mathbf{x}_i \in \mathbb{R}^L$ and its associated *class label* $y_i \in \{-1, 1\}$.

Dataset \mathcal{D} is used to learn an SVM classifier [11, Ch. 12] that can efficiently separate the two classes of data points through a separating hyperplane. The optimization problem for the SVM with hinge loss and parameter $C \geq 0$, is:

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

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

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

Here, the *feature mapping* $\phi : \mathbb{R}^L \rightarrow \mathbb{R}^F$, $F \geq L$, enlarges the feature space of the data points to improve the separability of the two classes of data points through a hyperplane [11, Ch. 12.3]. 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. Let, \mathbf{w}^* be the optimal solution to this problem, then the *binary classification* of a given data point \mathbf{x} is

$$h_\phi(\mathbf{x}, \mathbf{w}^*) := \text{sign}[f_\phi(\mathbf{x}, \mathbf{w}^*)], \quad (4)$$

where $\text{sign}[a] = -1$, if $a < 0$ and $\text{sign}[a] = 1$, if $a \geq 0$.

If the number of features F is larger than the number of available data points n , then it is more efficient to solve the dual problem of Eq. (2):

$$\underset{\boldsymbol{\alpha} \in \mathbb{R}^n}{\text{maximize}} \quad \sum_i^n \alpha_i - \frac{1}{2} \sum_i^n \sum_j^n \alpha_i \alpha_j y_i y_j \phi(\mathbf{x}_i)^\top \phi(\mathbf{x}_j) \quad (5a)$$

$$s.t. \quad 0 \leq \alpha_i \leq C/n, \quad i = 1, \dots, n, \quad (5b)$$

since then the number of optimization variables is smaller. Another advantage in solving the dual is the possibility to apply the kernel trick [11, Ch. 12.3], where a kernel function $k(\mathbf{x}_i, \mathbf{x}_j)$ replaces $\phi(\mathbf{x}_i)^\top \phi(\mathbf{x}_j)$ in Eq. (5a).

The optimal solution of the primal problem in Eq. (2) is then computed from the solution of the dual problem in Eq. (5) as:

$$\mathbf{w}^* = \sum_i^n \alpha_i^* y_i \phi(\mathbf{x}_i). \quad (6)$$

From Eq. (3) it can be observed that in order to perform SVM classification, all we need is \mathbf{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 the information in \mathbf{w}^* . Therefore, it is required to apply a privacy preserving mechanism before the public release of the classifier, as is illustrated in Figure 1.

2.2 Privacy Preserving Mechanism for SVM

In this section, we will describe the privacy preserving mechanism proposed in [2, Sec. 3] for SVMs with finite dimensional feature mappings. The proposed mechanism that guarantees differential privacy essentially perturbs the SVM optimal weights $\mathbf{w}^* \in \mathbb{R}^F$ by additive Laplace noise.

More formally, let $M : \mathfrak{D} \rightarrow \mathcal{R}$ 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 dual problem in Eq. (5)). Define *neighboring datasets* as the datasets in \mathfrak{D} that differ by one data point entry. Then, for a given $\beta > 0$, a randomized mechanism M provides β -differential privacy [3, Def. 2.4] if for any two neighboring datasets $\mathcal{D}_1, \mathcal{D}_2 \in \mathfrak{D}$ and all response subsets $\mathcal{S} \subseteq \mathcal{R}$ it holds

$$\Pr [M(\mathcal{D}_1) \in \mathcal{S}] \leq \exp(\beta) \Pr [M(\mathcal{D}_2) \in \mathcal{S}]. \quad (7)$$

From [2, Thm. 10], the perturbed SVM weight vector

$$\tilde{\mathbf{w}} := \mathbf{w}^* + \boldsymbol{\mu}, \quad (8)$$

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

$$\mu_i \sim \text{Lap}(0, \lambda), i = 1, \dots, F, \quad (9)$$

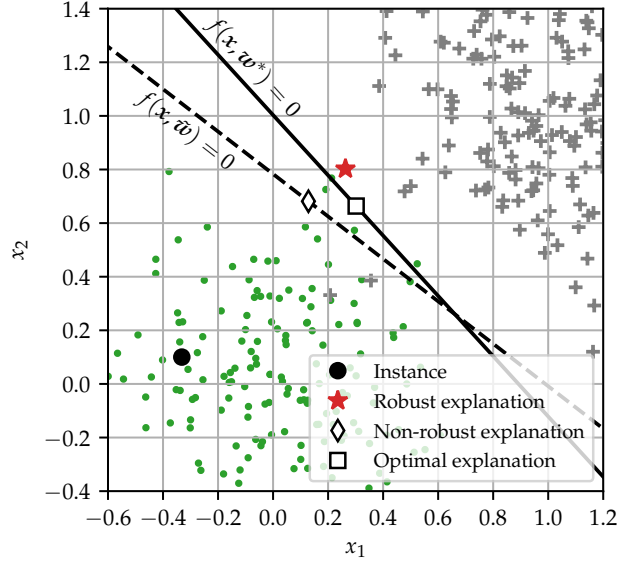


Figure 2: Illustration for linear classification with SVM and private SVM, and the different associated explanations using the Euclidean norm as distance measure. The data points are generated from two bivariate Gaussian distributions with means $[0, 0]$ and $[1, 1]$, and same covariance $0.1\mathbf{I}$.

achieves β -differential privacy for

$$\lambda \geq 4C\kappa\sqrt{F}/(\beta n), \quad (10)$$

where κ satisfies $\phi(\mathbf{x})^\top \phi(\mathbf{x}) \leq \kappa^2$ for all $\mathbf{x} \in \mathbb{R}^L$.

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 [2] by introducing a condition called (ϵ, δ) -useful mechanism:

$$\Pr \left[\sup_{\mathbf{x} \in \mathcal{X}} |f_\phi(\mathbf{x}, \tilde{\mathbf{w}}) - f_\phi(\mathbf{x}, \mathbf{w}^*)| \leq \epsilon \right] \geq 1 - \delta. \quad (11)$$

where $\epsilon > 0, \delta \in (0, 1)$, and the set $\mathcal{X} \subseteq \mathbb{R}^L$ contains all the data points. From [2, Thm. 11], the condition above is satisfied for $0 \leq \lambda \leq \frac{\epsilon}{2\Phi(F - \ln \delta)}$, where $|\phi_i(\mathbf{x})| \leq \Phi$ for all $\mathbf{x} \in \mathcal{X}, i = 1, \dots, F$, and some $\Phi > 0$.

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

3 Counterfactual Explanation

The concept of counterfactual explanations was proposed in [4, Eq. 2] for general ML classifiers. In the following definition we use the original definition, but formulated for the SVM.

Definition 1 (Counterfactual Explanation). *Given an SVM classifier with weight vector \mathbf{w} , a counterfactual explanation for the classification $y' = h(\mathbf{x}', \mathbf{w})$ of a given data instance \mathbf{x}' is the solution of the following problem*

$$\underset{\mathbf{x} \in \mathbb{R}^L}{\text{minimize}} \quad d(\mathbf{x}, \mathbf{x}') \quad (12a)$$

$$\text{s.t.} \quad y' f_\phi(\mathbf{x}, \mathbf{w}) \leq 0, \quad (12b)$$

where $d(\mathbf{x}, \mathbf{x}')$ is a distance between \mathbf{x} and \mathbf{x}' , assumed to be convex in the first argument, and $f_\phi(\mathbf{x}, \mathbf{w})$ is defined in Eq. (3).

The solution of problem (12) is the closest point to \mathbf{x}' , according to the distance d , with the constraint that its class is different than y' . Notice that since we consider binary classification, the constraint in Eq. (12b) is equivalent to $h(\mathbf{x}, \mathbf{w}) \neq y'$.

In Figure 2, we illustrate different counterfactual explanations for the linear SVM classifiers with optimal weights \mathbf{w}^* and perturbed weights $\tilde{\mathbf{w}}$. The optimal and non-robust explanations are the closest points to the instance that lie on the respective decision boundaries. It can be seen that the non-robust explanation 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. We will study robust explanations next that will take into account the uncertainty in the perturbed weights.

3.1 Robust Explanations

The private SVM mechanism releases noisy versions of the optimal \mathbf{w}^* according to Eq. (8). Thus, there exists uncertainty about the correctness of the classification with $\tilde{\mathbf{w}}$, which diminishes the effectiveness of the counterfactual explanation unless this uncertainty is taken into account. Therefore, we will model the uncertainty about \mathbf{w}^* through the random vector $\boldsymbol{\xi} = \tilde{\mathbf{w}} - \boldsymbol{\mu}$. From Eq. (9), it follows that

$$\boldsymbol{\xi} \sim \text{mvLap}(\tilde{\mathbf{w}}, 2\lambda^2 \mathbf{I}), \quad (13)$$

where $\text{mvLap}(\mathbf{l}, \boldsymbol{\Sigma})$ is the multivariate Laplace distribution with location \mathbf{l} and covariance matrix $\boldsymbol{\Sigma}$. Subsequently, we will replace the constraint in (12b) with the probabilistic constraint

$$\Pr[y' f_\phi(\mathbf{x}, \boldsymbol{\xi}) \leq 0] \geq p. \quad (14)$$

The probability p in Eq. (14) is typically selected close to 1. In the following, we will characterize the deterministic equivalent of the constraint in Eq. (14).

Proposition 1. *The robust counterfactual explanation problem*

$$\underset{\mathbf{x} \in \mathbb{R}^L}{\text{minimize}} \quad d(\mathbf{x}, \mathbf{x}') \quad (15a)$$

$$\text{s.t.} \quad \Pr[y' f_\phi(\mathbf{x}, \boldsymbol{\xi}) \leq 0] \geq p, \quad (15b)$$

with $p \in [1/2, 1]$ is equivalent to

$$\underset{\mathbf{x} \in \mathbb{R}^L}{\text{minimize}} \quad d(\mathbf{x}, \mathbf{x}') \quad (16a)$$

$$\text{s.t.} \quad y' \phi(\mathbf{x})^\top \tilde{\mathbf{w}} - \lambda \sqrt{2} \ln(2(1-p)) \|\phi(\mathbf{x})\| \leq 0. \quad (16b)$$

Proof. We need to prove that the probabilistic constraint in Eq. (15b) can be reformulated to Eq. (16b). From (13), the multivariate Laplace distribution $\text{mvLap}(\tilde{\mathbf{w}}, 2\lambda^2 \mathbf{I})$ is *symmetric* since the variance does not depend on the mean. A symmetric multivariate Laplace distribution is *elliptically symmetric* [12]. Consequently, the structure of Eq. (16b) follows from [13, Lemma 2.2], and for the multivariate Laplace distribution, the derivation follows similar steps as in [14, Ex. 2.2]. \square

The constraint in Eq. (16b) includes two terms. The first term is the same as in Eq. (12b) 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 $|\phi(\mathbf{x})^\top \tilde{\mathbf{w}}|$. Notice that for $p = 0.5$, the second term is zero and Eq. (16b) becomes the same as that in the non-robust case in Eq. (12b).

For linear SVM, i.e., $\phi(\mathbf{x}) = \mathbf{x}$, problem (16) is rewritten as

$$\underset{\mathbf{x} \in \mathbb{R}^L}{\text{minimize}} \quad d(\mathbf{x}, \mathbf{x}') \quad \text{s.t.} \quad \|\mathbf{x}\| \leq \frac{y'}{\lambda\sqrt{2}\ln(2(1-p))} \mathbf{x}^\top \tilde{\mathbf{w}}, \quad (17)$$

which is a SOCP problem and can be solved efficiently using convex optimization solvers. The implementations for this letter have been done using CVXPY [15, 16], and the explanations plotted in Figure 2 were found by solving this problem.

The general problem in (16) is however not convex. Therefore, we will consider next finding a suboptimal solution that can be computed efficiently. Define the function

$$g(\mathbf{x}) := y'\phi(\mathbf{x})^\top \tilde{\mathbf{w}} - \lambda\sqrt{2}\ln(2(1-p))\|\phi(\mathbf{x})\|, \quad (18)$$

which is the left hand side of Eq. (16b). A *root* for the function g would qualify as a robust explanation since it satisfies the constraint in (16b) with equality. In order to find a root for g , we will use the *bisection method*. As a prerequisite, this method requires the knowledge of two input values to the function that give opposite signs. Clearly, for the given data instance \mathbf{x}' , $g(\mathbf{x}')$ is positive. The second required input vector should necessarily be of opposite class to \mathbf{x}' 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* [9].

Unlike in [9], 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 by knowing the characteristics 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. Let the prototypes for class 1 and -1 be \mathbf{z}_1 and \mathbf{z}_{-1} , respectively. In the process of finding the prototypes, it is desired that the classification for these points has sufficient confidence, i.e.,

$$|f_\phi(\mathbf{z}_y, \tilde{\mathbf{w}})| \geq -\lambda\sqrt{2}\ln(2(1-p))\|\phi(\mathbf{z}_y)\|, \quad y \in \{1, -1\}. \quad (19)$$

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. In each iteration

Algorithm 1 Bisection method for finding an explanation

```
1: Input: Instance and classification  $(\mathbf{x}', y')$ , prototype  $\mathbf{z}_{-y'}$ .  
2: Initialize:  $\mathbf{x}^{ub} = \mathbf{z}_{-y'}$ ,  $\mathbf{x}^{lb} = \mathbf{x}'$   
3: while  $\|\mathbf{x}^{ub} - \mathbf{x}^{lb}\| > \epsilon$  do  
4:    $\mathbf{x} \leftarrow (\mathbf{x}^{ub} + \mathbf{x}^{lb})/2$   
5:   if  $g(\mathbf{x}) < 0$  then  
6:      $\mathbf{x}^{ub} \leftarrow \mathbf{x}$   
7:   else  
8:      $\mathbf{x}^{lb} \leftarrow \mathbf{x}$   
9: Output:  $\mathbf{x}^{ro-ex} \leftarrow \mathbf{x}$ .
```

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.

4 Experimental Results

We illustrate our approach by using the publicly available UCI Breast Cancer Wisconsin (Diagnostic) dataset [10]. The dataset includes 569 instances, each with 30 features and the binary diagnosis: benign (class -1) or malignant (class 1). The code to reproduce all the figures is available at [17].

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 of the dataset with the 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 [18] with dimensions $F = 100$.² For the implementation of the feature mapping, we have used the library in [19]. 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. (16), is the Euclidean norm, i.e., $d(\mathbf{x}, \mathbf{x}') = \|\mathbf{x} - \mathbf{x}'\|$. The prototypes are selected to be the data mean of each class.

Accuracy vs privacy. Figure 3, depicts the trade-off between average accuracy and privacy of the private SVM, with an average taken over 10^4 random realizations of Laplace noise. 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 SVM average performance.

²Note, that in this work 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 [2, Section 4] for the general case of translation-invariant kernels.

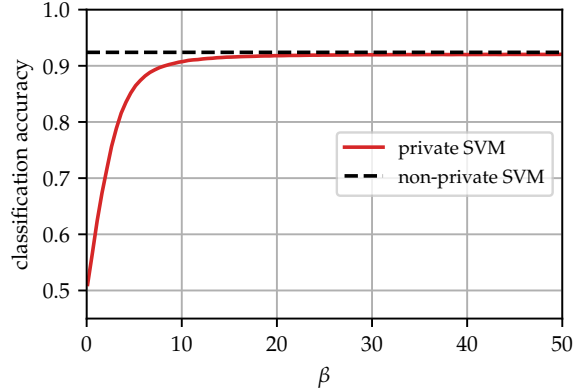
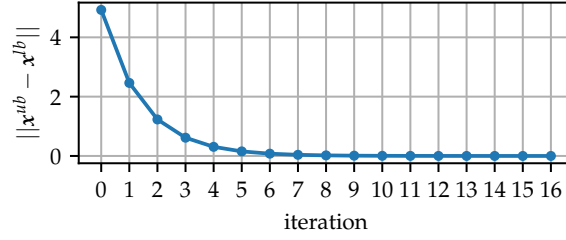


Figure 3: Average accuracy of private SVM for different values of β in (7). The confidence value is set to $p = 0.9$ for the probabilistic constraint in (14).

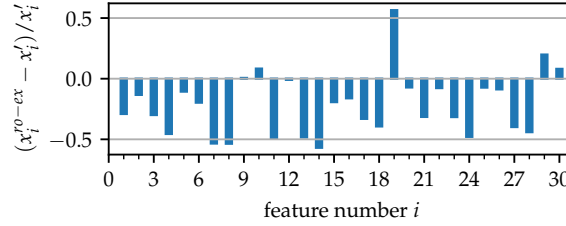
Convergence and explainability. We set $\beta = 5, p = 0.9$. Then, we select a random instance \mathbf{x}' from the test set with label $y' = 1$ (malignant), and apply Algorithm 1 to calculate an explanation \mathbf{x}^{ro-ex} for its classification. Figure 4a shows the low number of iterations needed for Algorithm 1 to converge. The found explanation \mathbf{x}^{ro-ex} quantifies the changes to each feature of \mathbf{x}' in order to change the classifier prediction. Figure 4b 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 be increased by around half its value, while several other feature values need to be halved in order to alter the prediction from malignant to benign.

Explainability vs privacy. The average distance between the counterfactual explanation and the instance is calculated depending on β (for $p = 0.9$) in Figure 5a, and depending on p (for $\beta = 0.5$) in Figure 5b. This average distance for robust counterfactual explanations is high for small values of β , as is shown in Figure 5a. This is due to the large uncertainty through the large noise variance. The non-robust explanation has similar distances as for non-private SVM since the noise has zero mean. In Figure 5b, the effects of p on the average distance are shown. For $p = 0.5$, robust explanation is identical with non-robust explanation as mentioned before. For large confidence values p , the robust explanation converges to the prototype data point, and is furthest away from the instance.

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 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. (12b) 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 6. In Figure 6a and Figure 6b, the summary statistics for the left hand side of (12b) 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 6b) for β less than around 4 is due to the convergence of the explanation to the prototype.



(a) Convergence of Algorithm 1.



(b) The counterfactual explanation is utilized to quantify the necessary changes in the instance’s features in order to alter its SVM classifier prediction.

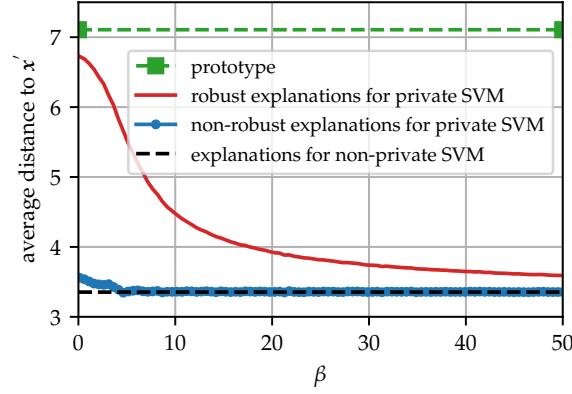
Figure 4: Convergence of Algorithm 1 and the explanation of an instance.

5 Conclusions

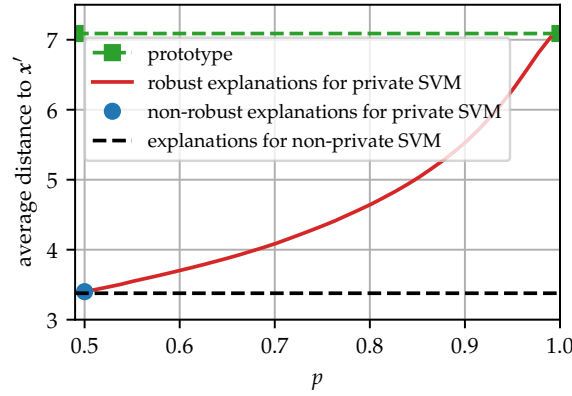
The above findings highlight the difficulties associated with embedding the social and ethical values mandated by regulatory instruments into ML algorithms. An ensuing conclusion is that a conscious decision may be required to promote one social value at the expense of another, the context in which the technology is being operated potentially being a deciding factor. These issues are highlighted in this work through the study of robust counterfactual explanations for privacy preserving SVMs. After suitably modelling the problem, we have studied its solution for linear and non-linear SVMs. While the problem can be solved optimally for the linear case, we have proposed an efficient suboptimal solution based on the bisection method for the general case. The advantages for using robust explanations are illustrated on the UCI Breast Cancer Wisconsin dataset. In particular, our robustness model guarantees a tuneable level of confidence that the counterfactuals are of opposite class to that of the instance we want to explain.

6 Acknowledgments

This work has been supported by KTH Digital Futures within the project “EXTREMUM: Explainable and Ethical Machine Learning for Knowledge Discovery from Medical Data Sources” (<https://www.digitalfutures.kth.se>).



(a) $p = 0.9$

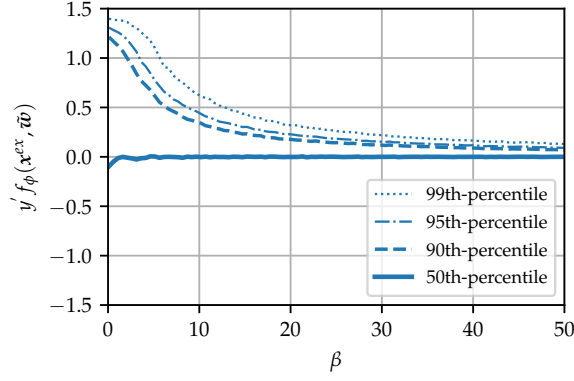


(b) $\beta = 5$

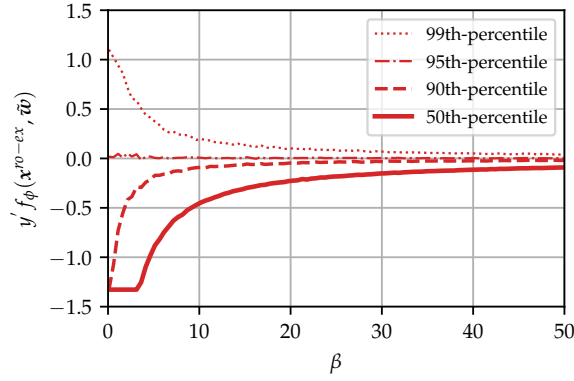
Figure 5: Robust explanations lie between the instance \mathbf{x}' and the selected prototype. This distance decreases for larger β and increases for larger probability p .

References

- [1] 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).
- [2] B. I. P. Rubinstein, P. L. Bartlett, L. Huang, N. Taft, Learning in a large function space: Privacy-preserving mechanisms for SVM learning, *Journal of Privacy and Confidentiality* 4 (1) (Jul. 2012).
- [3] C. Dwork, A. Roth, The algorithmic foundations of differential privacy, *Found. Trends Theor. Comput. Sci.* 9 (3–4) (2014) 211–407.



(a) Non-robust explanations



(b) Robust explanations

Figure 6: Comparison between robust and non-robust explanations on the violation of the constraint in (12b) for private SVM.

- [4] C. R. Sandra Wachter, Brent Mittelstadt, Counterfactual explanations without opening the black box: Automated decisions and the GDPR, *Harvard Journal of Law & Technology*, forthcoming 31 (2) (2018).
- [5] C. Molnar, *Interpretable Machine Learning - A Guide for Making Black Box Models Explainable*, 2019, <https://christophm.github.io/interpretable-ml-book/>.
- [6] W. J. Murdoch, C. Singh, K. Kumbier, R. Abbasi-Asl, B. Yu, Definitions, methods, and applications in interpretable machine learning, *Proceedings of the National Academy of Sciences* 116 (44) (2019) 22071–22080.
- [7] A. Barredo Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. Garcia, S. Gil-Lopez, D. Molina, R. Benjamins, R. Chatila, F. Herrera, Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI, *Information Fusion* 58 (2020) 82–115.

- [8] A. Shapiro, D. Dentcheva, A. Ruszczyński, Lectures on Stochastic Programming: Modeling and Theory, Second Edition, Society for Industrial and Applied Mathematics, Philadelphia, PA, 2014.
- [9] A. V. Looveren, J. Klaise, Interpretable counterfactual explanations guided by prototypes, CoRR abs/1907.02584 (2019).
URL <http://arxiv.org/abs/1907.02584>
- [10] D. Dua, C. Graff, UCI machine learning repository, University of California, Irvine, School of Information and Computer Sciences (2017).
URL <http://archive.ics.uci.edu/ml>
- [11] T. Hastie, R. Tibshirani, J. Friedman, The elements of statistical learning: data mining, inference and prediction, 2nd Edition, Springer Series in Statistics, Springer-Verlag New York, 2009.
- [12] S. Kotz, T. J. Kozubowski, K. Podgórski, Symmetric Multivariate Laplace Distribution, Birkhäuser Boston, Boston, MA, 2001, Ch. 5.
- [13] R. Henrion, Structural properties of linear probabilistic constraints, Optimization 56 (2007) 425 – 440.
- [14] S. Peng, Chance constrained problem and its applications, Theses, Université Paris Saclay (COMUE) ; Xi'an Jiaotong University (Jun. 2019).
URL <https://tel.archives-ouvertes.fr/tel-02303045>
- [15] S. Diamond, S. Boyd, CVXPY: A Python-embedded modeling language for convex optimization, Journal of Machine Learning Research 17 (83) (2016) 1–5.
- [16] A. Agrawal, R. Verschueren, S. Diamond, S. Boyd, A rewriting system for convex optimization problems, Journal of Control and Decision 5 (1) (2018) 42–60.
- [17] R. Mochaourab, Robust-Explanation-SVM, <https://github.com/rami-mochaourab/robust-explanation-SVM> (2021).
- [18] A. Rahimi, B. Recht, Random features for large-scale kernel machines, in: Proceedings of the 20th International Conference on Neural Information Processing Systems, NIPS’07, Curran Associates Inc., Red Hook, NY, USA, 2007, p. 1177–1184.
- [19] K. Atarashi, pyrfm: A library for random feature maps in python (2019).
URL <https://neonnnnn.github.io/pyrfm/>