An Interpretable SVM Classifier by Discrete-Weight Optimization

Fethi Jarray [*1] **Sabri Boughorbel** [*2]

Abstract

The main problem investigated in this paper is to learn "interpretable" linear classifiers from data. Interpretable models are captured using "discrete" linear functions. The learning problem is formulated as minimizing the cumulative zero-one loss of a discrete hyperplane, penalized by the standard L2 regularizer. This learning task is cast as a MILP problem, and solved using convex relaxation and rounding. Experiments on both synthetic and real-world datasets corroborate the interest of this approach. We benchmarked the proposed method against two classifiers: i- DILSVM a discrete version of SVM based a hinge-loss and ii- the traditional linear L1-norm SVM. Our algorithm outperforms DILSVM on several datasets in terms of accuracy and computational efficiency. It has close performance to the continuous SVM. These results suggest that the proposed classifier provides a good trade-off between performance and interpretability.

1. Introduction

Supervised machine learning has made major advances in the last decades. Significant improvements in term of performance have been achieved in a wide spectrum of applications. However the level of complexity has also grown to a level that only machine learning experts could interpret the output of the models. Thus there is a difficulty in adopting many of the classification models due to their complexity and lack of trust in classifier behavior once deployed in real applications. On the other hand, simple scoring systems are still much desired in many applications. For example, in the medical diagnosis area, scoring systems are widely used for diagnosis or outcome prediction such as Thrombolysis In Myocardial Infarction (TIMI) score, Apache II score for infant mortality in the ICU or CHADS2 score for Atrial Fibrillation Stroke Risk, etc. Credit scoring is also a related application where banks want to identify which customer attributes can predict good or bad credit risks. Providing tools to interpret the classifier output could help understanding which attributes are important for the prediction. Likert scale system is also being widely for survey evaluation. A typical Likert scale includes five levels (Strongly disagree, Disagree, Neither agree nor disagree, Agree, Strongly agree). A classifier that provides the level of agreement of each feature to an accurate classification can enhance its interpretability (Carrizosa et al., 2013; Ustun & Rudin, 2015; Letham et al., 2012).

The goal in supervised machine learning is to discover a function $f(\mathbf{x})$ called classifier that predicts a label y where $\mathbf{x}$ is a d-dimensional vector of values, often called feature vector and y is the label of the corresponding class of $\mathbf{x}$. The estimation of the function f is based on a training set $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$. For binary classification, the label y takes two values, e.g., -1 or 1. A hyperplane contains all the points in a d-dimensional space satisfying the following equation: $w_1 x_1 + w_2 x_2 + \ldots w_d x_d + b = 0$. Each coefficient w_i can be thought of as a weight or a rating on the corresponding feature. The vector containing all the weights $\mathbf{w} = (w_1, \ldots, w_d)$ is the weight vector. In the traditional learning problem, we seek to find real-valued weights that separate the two classes. Here, we restricts coefficients to integer coefficients, i.e. each coefficient can take value among a set $\mathbb{A}$ of arbitrary values.

Our objective is to build classifiers that are accurate, yet easy to interpret by human experts. Our model is based on linear Support Vector Machines with discrete coefficients. The proposed SVM model can be easily interpreted compared to classical SVM. We overcome the restriction in the work of (Carrizosa et al., 2013) where the space of discrete values is in the form of $\mathbb{A} = \{-a_K, \ldots, -a_1, 0, a_1, \ldots, a_K\}$, i.e., symmetric and includes zero. This paper extends also to previous work by Chevaleyre et al. where the classifier weights are limited to binary values (Chevaleyre et al., 2013).

The paper is organized as follows. In the next section, we

[*]Equal contribution [1]Laboratoire Cedric-CNAM, 292 rue St-Martin, 75003 Paris, France [2]Sidra Medical and Research Center, Doha, Qatar. Correspondence to: Fethi Jarray <fethi_jarray@yahoo.fr>, Sabri Boughorbel <sboughorbel@sidra.org>.

Proceedings of the 34^{th} International Conference on Machine Learning, Sydney, Australia, 2017. PMLR 70,

present a review of related work. In section 3, we propose a new Mixed Integer Linear Programming (MILP) formulation to improve the interpretability of SVM. In section 4, we propose an approximation scheme to solve the MILP problem. In section 5 and 6, we present and discuss the experimental results.

2. Related Work

The traditional SVM methods are continuous SVM as opposed to discrete SVM where each coefficient belongs to a discrete set. The earliest research in this area consisted in learning perceptrons with binary weights while the linear perceptron has arbitrary real-valued weights (Golea & Marchand, 1993; Fang & Venkatesh, 1996). The authors showed that the binary perceptron algorithms learn majority functions in linear time from small samples.

Chevaleyre et al. (Chevaleyre et al., 2013) proposed a general framework to build binary linear classifiers with $\{0,1\}$-valued weights. Their approach is quite general and can be applied with many learning algorithms. Firstly, it determines a fractional solution by replacing the discrete constraint by a weaker constraint, that each coefficient belongs to the interval $[0,1]$ instead of being in the set of $\{0,1\}$. Secondly, it applies randomized rounding and greedy rounding procedures to the fractional solution to get a discrete solution of the learning problem.

Carrizosa et al. (Carrizosa et al., 2013) proposed a Discrete Level Support Vector Machine (DILSVM) for rating features such that each coefficient w_j belongs to a discrete set $\mathbb{A} = \{a_1, \ldots, a_K\}$. Carrizosa et al. (Carrizosa et al., 2013) suppose that the set $\mathbb{A}$ is symmetric and contains 0 such as $\mathbb{A} = \{-2, -1, 0, 1, 2\}$. Firstly they formulate the learning problem as a Mixed Integer Linear problem (MILP) where each coefficient is restricted to take a discrete value. Secondly, the integral constraints are relaxed to turn the MILP to a linear program. Thirdly, the linear program is solved to obtain a partial continuous optimal solution. Finally rounding techniques are applied to construct a feasible solution to MILP. DILSVM evaluated with three or five discrete values has a comparable accuracy to the classical SVM with a gain in interpretability.

3. SVM-DISC Approach

We suppose that each coefficient w_j belongs to an arbitrary domain of values $\mathbb{A} = \{a_1, \ldots, a_K\}$. Carrizosa et al. (Carrizosa et al., 2013) suppose that the set $\mathbb{A}$ is symmetric such as $\mathbb{A} = \{-2, -1, 0, 1, 2\}$ and propose to minimize the hinge loss. The latter is a convex upper bound on 0-1 loss. Hence it guarantees nice convergence property for the algorithms. However the discrete version of SVM is intrinsically modeled by a non-convex program regardless of the loss function choice. Thus solving this problem with 0-1 loss is easier and more efficient than using the hinge loss (Ustun & Cynthia, 2016; Nguyen & Scott, 2013).

3.1. SVM-MILP

In Mixed Integer Programming, forcing constraints are generally required to force binary variables to 1 when other positive continuous variables are non-zero. For example, suppose x is a continuous variable and z is a binary variable. The forcing constraint $x \leq Mz$ will set z to 1 whenever x is positive where M is a large number, often called Big M. The value M should be at least as large as the largest possible value of the variable x, otherwise it will introduce an infeasibility. However, from an algorithmic point of view, M should not be too large, otherwise, it will be difficult for the integer solver to converge and it will also introduces potential round-off error.

Our new discrete SVM model can be written as follow:

$$SVM{-}discrete \begin{cases} min\ \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n} z_i \\ s.t. \\ y_i(\mathbf{w}\cdot\mathbf{x}_i + b) \geq 1 - \xi_i \quad i = 1, \ldots, n \\ \xi_i \leq M z_i \quad i = 1, \ldots, n \\ b \in \mathbb{R}; w_j \in \mathbb{A}; \xi_i \geq 0; z_i \in \{0,1\} \quad \forall j\ \forall i \end{cases} \tag{1}$$

where ξ_i is a positive slack variable such that if $0 < \xi_i < 1$ then instance i is between the margin and the correct side of hyperplane and if $\xi_i > 1$ then instance i is misclassified. C is a regularization parameter such that small C allows constraints to be easily ignored (large margin) and large C makes constraints hard to ignore (narrow margin). The variable z_i is an indicator variable associated with every example i, which takes value 1 if the example i is misclassified, 0 otherwise. The second constraint ensures that $z_i = 1$ if and only if $\xi_i > 0$ since we deal with a minimization problem.

The linearization technique consists in replacing a product by a new variable and adding a set of linear constraints that force the equality between the new variable and the product(Billionnet et al., 2013). To linearize the SVM-discrete model, we introduce the following binary variable α_{jk} such that $\alpha_{jk} = 1$ if $w_j = a_k$. So $w_j = \sum_{k=1}^{K} \alpha_{jk} a_k$ and $\|\mathbf{w}\|^2 = \sum_{j=1}^{d}\sum_{k=1}^{K} a_k^2 \alpha_{jk}$ because α_{jk} are binary and bitwise different. It is worth noting that the objective function is linear because a_k^2 are constant and not variables. We have also $\mathbf{w}\cdot\mathbf{x}_i = \sum_{j=1}^{d}\sum_{k=1}^{K} a_k \alpha_{jk} x_{ij}$. Thus the model SVM-discrete is equivalent to the following Mixed Integer Linear Program (MILP).

$$SVM{-}MILP \begin{cases} min\ \frac{1}{2}\sum_{j=1}^{d}\sum_{k=1}^{K} a_k^2\alpha_{jk} + C\sum_{i=1}^{n} z_i \\ s.t. \\ y_i(\sum_{j=1}^{d}\sum_{k=1}^{K} a_k\alpha_{jk}x_{ij} + b) \geq 1-\xi_i \ \forall i \\ \sum_{k=1}^{K}\alpha_{jk} = 1 \quad \forall j \\ \xi_i \leq M z_i \quad \forall i \\ z_i, \alpha_{jk} \in \{0,1\} \quad \forall i,j \\ b \in \mathbb{R}; \quad \xi_i \geq 0 \forall i \end{cases} \tag{2}$$

The second constraint makes sure that each coefficient w_j is assigned to exactly one discrete value a_k.

3.2. Bounds for SVM

We derive an upper bound on M for the slack variable ξ such that $\xi_i \leq M\ i = 1, \ldots, n$

Theorem 1. *Let $B = d\max_j |w_j| max_{ij}|x_{ij}|$. The program SVM-discrete has an optimal solution with $|b| \leq B+1$*

Proof. We note that $|\mathbf{w}\cdot\mathbf{x}_i| \leq \sum_{j=1}^{d}|w_j||x_{ij}| \leq dmax_j|w_j|max_{ij}|x_{ij}| = B$. Hence $\mathbf{w}\cdot\mathbf{x}_i + B \geq 0$ and $\mathbf{w}\cdot\mathbf{x}_i - B \leq 0$.

The separation constraint can be rewritten as:

For the positive instances: $\xi_i \geq 1 - \mathbf{w}\cdot\mathbf{x}_i - b$, for the negative instances: $\xi_i \geq 1 + \mathbf{w}\cdot\mathbf{x}_i + b$

Let $(\mathbf{w}, b, \xi, \mathbf{z})$ be an optimal solution for SVM-discrete. We will show that for a fixed vector $\mathbf{w}$, we can modify the variable b to reach the bound while obtaining another optimal solution.

Suppose that $b \geq B+1$. For the positive class, we have $\mathbf{w}\cdot\mathbf{x}_i + b \geq \mathbf{w}\cdot\mathbf{x}_i + B + 1 \geq 1$. So the positive class instances are correctly classified. For the negative instances, since $\xi_i \geq 1+\mathbf{w}\cdot\mathbf{x}_i+b$, we deduce that if we decrease b up to $B+1$ the slack variable ξ decreases while the positive instances remain correctly classified. So any value of b greater than $B+1$ can reduced to $B+1$.

Similarly, suppose that $b \leq -B-1$. For the negative class, we have $\mathbf{w}\cdot\mathbf{x}_i + b \leq \mathbf{w}\cdot\mathbf{x}_i - B - 1 \leq -1$. So the negative class instances are correctly classified. For the positive instances, since $\xi_i \geq 1 - \mathbf{w}\cdot\mathbf{x}_i - b$, we deduce that if we increase b up to $-B-1$ the slack variable ξ decreases while the negative instances remain correctly classified. So any value of b less than $-B-1$ can increased to $-B-1$. Finally, we have usually $|b| \leq B+1$. □

Theorem 2. *Let $M = 2B+2$. The program SVM-discrete has an optimal solution with $\xi_i \leq M$*

Proof. The separation constraint is $\xi_i \geq 1 - y_i(\mathbf{w}\cdot\mathbf{x}_i + b)$. Thus $\xi_i = max(1 - y_i(\mathbf{w}\cdot\mathbf{x}_i + b), 0)$. By considering the previous theorem, we have $1 - y_i(\mathbf{w}\cdot\mathbf{x}_i + b) \leq 1 + B + B + 1 = 2B + 2$. Hence $\xi_i \leq M$. □

For the discrete SVM, we have

$$B = d\max_j |w_j| \max_{ij} |x_{ij}| = d\max_k |w_k| \max_{ij} |x_{ij}| = da_K \max_{ij} |x_{ij}| \tag{3}$$

In particular if the training data $\mathbf{x}$ is normalized, i.e $|x_{ij}| \leq 1$ then

$$B = da_K, M = 2B + 2 \tag{4}$$

4. Solving SVM-MILP

The program SVM-MILP is NP-hard even for a set of two values $\mathbb{A} = \{a_1, a_2\}$. Due to the size of the model and its complexity, we propose a two-stage approach. First, we solve the associated relaxed linear program by using a linear programming solver such as CPLEX, COIN or GLPK. We replace the integrality constraint $z_i, \alpha_{jk} \in \{0,1\}$ by $z_i, \alpha_{jk} \in [0,1]$. Second, in order to obtain an integral solution to SVM-MILP, a rounding technique is used to derive the discrete solution for SVM-MILP.

4.1. Rounding Strategy

In the rounding procedure, for each coefficient w_j independently exactly one of the α_{jk} is set to 1 and the rest is set to 0. The fractional assignment α_{jk} will be rounded to an approximate integer assignment for SVM-MILP. We evaluated two approaches for the rounding strategy: a randomized and deterministic a approach. The latter gave better results. Therefore, we will use it in the remainder of this work. The probability for the coefficient w_j to be assigned by the rounding procedure to discrete value a_k is equal to the fractional value α_{jk}.

Algorithm 1 Algorithm for Training SVM-DISC.

Input: A tuplet of parameters $(C, a_1, \ldots, a_k)$
Output: An approximate classifier

1 Solve the linear relaxation of SVM-MILP and get the fractional optimal solution $(\alpha_{jk}, z_i, \xi_i, b)$ **for** $j = 1, \ldots, d$ **do**
2 Assign each coefficient w_j to exactly one discrete value a_k (set $w_j = a_k$) corresponding to the maximum probability from the distribution $\alpha_{jk}, \sum_{k=1}^{K}\alpha_{jk} = 1$.
3 Return the classifier $sign(\sum_{j=1}^{d}\sum_{k=1}^{K} a_k\alpha_{jk}x_j + b)$

5. Experimental Results

We compare the performance of our classifier with two others: 1- continuous L_1-norm SVM and 2- DILSVM

(Carrizosa et al., 2013). We evaluate the accuracy, interpretability and efficiency of the classifiers on 10 publicly available real-world datasets and one synthetic dataset (linearly separable points). The analysis is carried for $M = 2da_K \max_{ij} |x_{ij}| + 2$ as stated by equation (5) and two choices of the discrete set $\mathbb{A} = \{-1, 0, 1\}$ and $\mathbb{A} = \{-2, -1, 0, 1, 2\}$. The interpretability is measured by assessing the distribution of the discrete values and the sparsity of the classifiers. We used python linear programming toolkit PuLP for implementing Algorithm 1 and DILSVM (Mitchell et al., 2011). L_1-SVM is based LibSVM implementation (Chang & Lin, 2011). The python code for this work can be found at `https://github.com/bsabri/svm-disc`.

5.1. Synthetic Data

We generated three toy datasets of 100 samples each. The datasets are shown in the left column of Figure 1. The dataset on top (toy-data 1) is composed of two interleaving half circles. The middle one (toy-data 2) is formed with a circle inside another one. The third one in the bottom (toy-data 3) represents an overall separable two classes. The data points are generated using a model similar to the one used for Madelon dataset (Guyon, 2003). Dataset toy-data 2 is challenging for linear classification.

The comparison should reveal how our algorithm behaves compared with the continuous linear L_1-SVM. The classifier SVM-DISC is trained for the discrete set $\mathbb{A} = \{-2, -1, 0, 1, 2\}$. The middle and right columns of Figure 1 show the decision boundary for both linear SVM and SVM-DISC. As expected, both linear L_1-SVM and SVM-DISC fail to classify toydata-2 dataset as it is highly non linear dataset. For toy-data 1, the fitted classifier weight vector is $w = [1, 1]$. This means that both features x and y are equally important for the classification of the data points. The obtained accuracy (80%) is the same for L_1-SVM. For toy-data 2, both classifiers fail to classify the data. In general, high dimensional feature space helps improving the separability of the two classes and such scenario is not the most frequent. For toy-data 3, SVM-DISC reveals that dimension x is the most important for the classification and disregards the dimension y. It achieves also comparable accuracy to L_1-SVM. The analysis on synthetic data shows that despite the restricted capacity of the discrete classifier it is able to achieve comparable performance to the continuous SVM. Moreover the interpretation of the feature role in the classifier becomes very simple and easy to understand.

5.2. Real-Life Data

We compared the accuracy of SVM-DISC with respect to L_1-norm linear SVM and DILSVM on 10 real-world datasets and one synthetic dataset. The datasets are obtained from public resources such as UCI Machine Learning repository, openML and MLData repository (Asuncion & Newman, 2007; Sonnenburg et al.; Vanschoren et al., 2014; Guyon, 2003). We also evaluated the computational efficiency of SVM-DISC and DILSVM. We did not include L_1-norm SVM in the efficiency evaluation because its implementation has additional software optimization (optimized complied C code) that makes the comparison unfair. Moreover, continuous SVM optimization is expected to be faster than Mixed Linear Integer optimization as it is a convex problem.

For the evaluation, each dataset is split into a validation and test sets with a proportion of 60% and 40%. The hyper-parameter C is tuned using a 2-fold cross-validation on the validation set. The model is then fitted for the best parameter C using the whole validation dataset. For tuning the parameter C, the following set $C = \{2^{-4}, 2^{-2}, 2^0, 2^2, 2^4, 2^6, 2^8\}$ is used. The fitted classifier is evaluated on the test set. This procedure is repeated 5 times to obtain a mean and a standard deviation of the accuracy. Table 1 and Table 2 summarize the obtained results. Columns n and d in Table 1 correspond respectively to the number of examples and the number of features in each dataset. The description of each dataset can be found on the data source referenced above. As expected, L_1-norm linear SVM has better accuracy than the two discrete linear SVM. This is due to the higher complexity of the decision boundary. SVM-DISC outperforms DILSVM overall on the included dataset with 2% to 4% in terms of accuracy. The latter failed in particular properly to classify dataset *car*. The use of the sets $\mathbb{A} = \{-1, 0, 1\}$ and $\mathbb{A} = \{-2, -1, 0, 1, 2\}$ slightly changed the results for SVM-DISC.

Figure 2 presents histograms of $\mathbf{w}$ values for SVM-DISC and DILSVM. For each dataset, the histograms of the values in $\mathbf{w}$ are plotted. The higher the pick in zero the more sparse is the classifier. In addition to interpretability, discrete classifiers have the intrinsic property of applying feature selection. Features with weights $w_i = 0$ are irrelevant for the classification. In two datasets (splice and german), the whole vector $\mathbf{w}$ is zero. This could be explained by the high non-linearity in the data since also continuous SVM has failed to accurately classify these datasets. Features corresponding to positive a_k contributes positively and strongly to the classification.

In addition to evaluating the accuracy, we were also interested whether our formulation based on 0-1 loss has an advantage in terms of computational efficiency. We compared the CPU time for training the classifier on the validation data for the optimal hyper-parameter C. Figure 3 summarizes the comparison of SVM-DISC and DILSVM on CPU time required for training the classifiers on the dif-

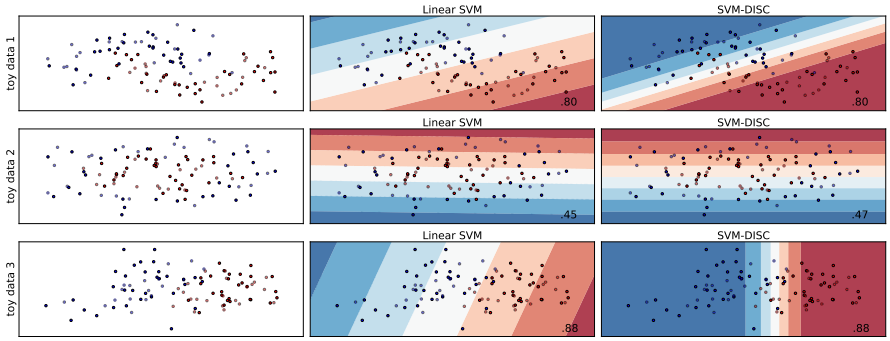

Figure 1. Comparison of SVM-DISC and Linear SVM on synthetic data. The values displayed on the right bottom are classification accuracy.

Table 1. Comparison of the classification accuracy for Linear SVM, SVM-DISC and DILSVM for $\mathbb{A} = \{-2, -1, 0, 1, 2\}$. n and d denote respectively the number of examples and features of the different datasets.

dataset	n	d	L_1-norm SVM	SVM-DISC	DILSVM
abalone	4177	10	78.60 $\pm$ 0.42	77.85 $\pm$ 0.85	77.99 $\pm$ 0.61
breast-cancer	683	10	96.64 $\pm$ 0.75	95.11 $\pm$ 1.08	92.77 $\pm$ 2.10
calhous	20640	8	84.15 $\pm$ 0.45	82.66 $\pm$ 0.30	81.04 $\pm$ 01.01
car	1594	16	94.36 $\pm$1.04	85.71 $\pm$ 1.01	66.61 $\pm$ 37.26
cod-rna	300	8	94.67 $\pm$ 2.54	88.50 $\pm$ 1.71	70.17 $\pm$ 39.29
german	1000	24	77.45 $\pm$1.35	66.75 $\pm$ 1.71	69.00 $\pm$ 2.04
linear-separable	1000	20	84.15 $\pm$1.55	85.25 $\pm$1.17	85.15 $\pm$1.02
nursery	12960	20	98.83 $\pm$ 0.10	97.64 $\pm$ 1.20	97.48 $\pm$ 0.83
spam	4601	57	92.91 $\pm$ 0.42	87.53 $\pm$ 1.45	83.78 $\pm$ 7.23
splice	2422	59	67.74 $\pm$ 0.99	67.74 $\pm$ 0.99	67.74 $\pm$ 0.99
thyroid	3163	25	98.07 $\pm$ 0.47	94.03 $\pm$ 3.52	93.19 $\pm$ 4.08
Average			87.96 $\pm$9.87	84.43$\pm$9.94	80.44$\pm$18.37

ferent benchmarking datasets. There is about 30% gain in computational efficiency in favor of SVM-DISC. The experiments were performed using a computer with 3 cores (i7-4600M CPU@2.90 GHz) with 8 GB of RAM.

6. Discussion

The advantage of SVM-DISC on DILSVM in terms of accuracy can be explained by the fact that SVM-DISC is better designed to maximize the accuracy. The use of the 0-1 loss function is a key factor for this improvement. The accuracy can be written $Accuracy = (n - \sum_i z_i)/n$. Since n is constant, maximize accuracy equivalent to minimize sum z_i. However in DILSVM, an approximation of accuracy is used by the hinge loss function. Concerning the computational complexity. SVM-DISC is faster than DILSVM possibly because the relaxed linear program for SVM-DISC is sparser than for DILSVM. In fact the objective function of SVM-DISC contains $dK + n$ variables whereas that of DILSVM contains $2dK + n$. Moreover the separation constraint plays a role in the computation efficiency. For SVM-DISC, it contains $dk + n + 1$ and for the DILSVM, it has $2dk + n + 1$ variables. Overall SVM-DISC comes with advantages in terms of accuracy and computational efficiency.

7. Conclusion

In this paper, we have proposed a new linear discrete SVM. The classifier coefficients take value in an arbitrary discrete values. The integer problem formulation is based on 0-1 loss function. Since this discrete problem is NP-hard, it is relaxed to a Mixed Integer Linear Programming problem. A rounding technique is used to transform the optimal fractional solution for MILP program to a discrete value. The obtained accuracy improves the state-of-the art results and is rather close to the continuous linear SVM. The proposed algorithm is also more efficient than DILSVM thanks to the formulation in terms of 0-1 loss function. As future work, we plan to investigate the extension of this paper to handle non linearity in data. For a larger choice of discrete set $\mathbb{A}$, the computational cost becomes very high. We would like to explore new approaches to reduce the computational

Table 2. Comparison of the classification accuracy for Linear SVM, SVM-DISC and DILSVM for $\mathbb{A} = \{-1, 0, 1\}$.

dataset	L_1-norm SVM	SVM-DISC	DILSVM
abalone	79.40 ± 1.09	74.36 ± 7.08	67.06 ± 16.68
breast-cancer	97.30 ± 0.80	93.50 ± 1.92	91.68 ± 7.30
calhous	84.24 ± 0.50	82.42 ± 2.02	81.34 ± 1.83
car	94.61 ± 0.76	86.83 ± 1.88	84.04 ± 0.28
cod-rna	95.00 ± 1.56	87.00 ± 1.39	86.17 ± 3.56
german	76.20 ± 1.16	70.30 ± 2.04	70.10 ± 1.59
linear-separable	84.40 ± 1.33	85.80 ± 1.71	85.50 ± 1.69
nursery	98.71 ± 0.20	98.00 ± 1.06	95.78 ± 1.10
spam	92.55 ± 0.51	85.59 ± 1.93	85.48 ± 1.87
splice	67.39 ± 1.72	67.39 ± 1.72	67.39 ± 1.72
thyroid	98.18 ± 0.36	93.46 ± 3.23	96.18 ± 3.09
Average	87.99 ± 10.00	84.05 ± 9.75	82.79 ± 11.37

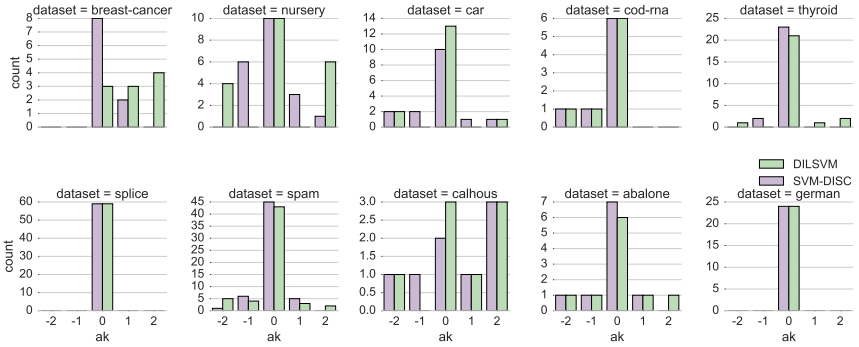

Figure 2. Histograms of the values in $\mathbf{w}$ for SVM-DISC and DILSVM for the different datasets. The classifiers are trained for $\mathbb{A} = \{-2, -1, 0, 1, 2\}$.

time for training the discrete classifier. We envisage also to study the theoretical properties of SVM-DISC such as consistency as well as deriving learning bounds.

References

Asuncion, Arthur and Newman, David. UCI machine learning repository, 2007.

Billionnet, Alain, Jarray, Fethi, Tlig, Ghassen, and Zagrouba, Ezzedine. Reconstructing convex matrices by integer programming approaches. *Journal of Mathematical Modelling and Algorithms*, 12(4):329–343, 2013.

Carrizosa, Emilio, Nogales-Gómez, Amaya, and Morales, Dolores Romero. Strongly agree or strongly disagree?: Rating features in support vector machines. Technical report, Technical report, Saïd Business School, University of Oxford, UK, 2013.

Chang, Chih-Chung and Lin, Chih-Jen. Libsvm: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27, 2011.

Chevaleyre, Yann, Koriche, Frédéerick, and Zucker, Jean-Daniel. Rounding methods for discrete linear classification. In *Proceedings of the 30th international conference on machine learning (ICML-13)*, pp. 651–659, 2013.

Fang, Shao C. and Venkatesh, Santosh S. Learning binary perceptrons perfectly efficiently. *Journal of Computer and System Sciences*, 52(2):374–389, 1996.

Golea, Mostefa and Marchand, Mario. On learning perceptrons with binary weights. *Neural Computation*, 5(5): 767–782, 1993.

Guyon, Isabelle. Design of experiments of the nips 2003 variable selection benchmark. In *NIPS 2003 workshop on feature extraction and feature selection*, 2003.

Letham, Benjamin, Rudin, Cynthia, McCormick, Tyler H, and Madigan, David. Building interpretable classifiers with rules using bayesian analysis. *Department of Statistics Technical Report tr609, University of Washington*, 2012.

Mitchell, Stuart, OSullivan, Michael, and Dunning,

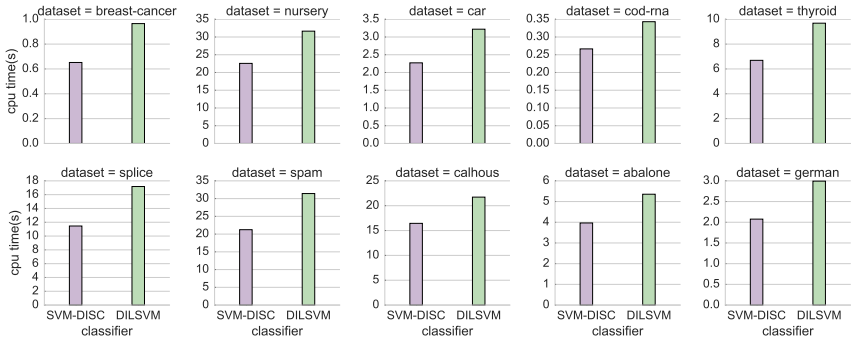

Figure 3. Comparison of CPU time in seconds for training SVM-DISC and DILSVM.

Iain. Pulp: a linear programming toolkit for python. *The University of Auckland, Auckland, New Zealand, http://www. optimization-online. org/DB_FILE/2011/09/3178. pdf*, 2011.

Nguyen, Tan and Scott, Sanner. Algorithms for direct 0-1 loss optimization in binary classification. In *Proceedings of the 30th international conference on machine learning*, pp. 1085–1093, 2013.

Sonnenburg, S, Ong, CS, Henschel, S, and Braun, M. Machine learning data set repository (2011). *URL http://mldata. org/.(Cited on pages 115 and 131.)*.

Ustun, Berk and Cynthia, Rudin. Supersparse linear integer models for optimized medical scoring systems. *Machine Learning*, 102(3):349–391, 2016.

Ustun, Berk and Rudin, Cynthia. Supersparse linear integer models for optimized medical scoring systems. *Machine Learning*, pp. 1–43, 2015.

Vanschoren, Joaquin, Van Rijn, Jan N, Bischl, Bernd, and Torgo, Luis. Openml: networked science in machine learning. *ACM SIGKDD Explorations Newsletter*, 15(2): 49–60, 2014.