General Framework for Linear Binary Classification on Top Samples

Ing. Václav Mácha

July 24, 2020

Supervisor: doc. Ing. Václav Šmídl, Ph.D.
Supervisor specialist: Mgr. Lukáš Adam, Ph.D.

Abstract

Many binary classification problems minimize misclassification above (or below) a threshold. We show that instances of ranking problems, accuracy at the top or hypothesis testing may be written in this form. We propose a general framework to handle these classes of problems and show which known methods (both known and newly proposed) fall into this framework. We provide a theoretical analysis of this framework and mention selected possible pitfalls the methods may encounter. We suggest several numerical improvements including the implicit derivative and stochastic gradient descent. We provide an extensive numerical study. Based both on the theoretical properties and numerical experiments, we conclude the paper by suggesting which method should be used in which situation.

Contents

1 Introduction

General linear binary classification is a problem of finding the best linear separating hyperplane for two groups of samples. For sample $\boldsymbol{x}$, the linear classifier is defined as $f(\boldsymbol{w}) = \boldsymbol{w}^\top \boldsymbol{x} - t$, where $\boldsymbol{w} \in \mathbb{R}^d$ is the normal vector to the separating hyperplane and $t \in \mathbb{R}$ is a decision threshold. The most well-known example of such a classifier is a support vector machine [9], where the decision threshold t is considered as a free variable. However, many important binary classification problems focus on cases of maximizing performance only for a certain amount of samples with the highest scores $s = \boldsymbol{w}^\top \boldsymbol{x}$. In these cases, the threshold t is not a free variable, but the function of the scores. Multiple problem categories belong to this framework:

- *Ranking problems* [1, 12, 27, 21] select the most relevant samples and rank them. To each sample, a numerical score is assigned and the ranking is performed based on this score. Often, only scores above a threshold are considered.
- *Accuracy at the Top* [5, 14, 30] is similar to ranking problems. However, instead of ranking the most relevant samples, it only maximizes the accuracy (equivalently minimizes the misclassification) in these top samples. The prime examples of both categories include search engines or problems where identified samples undergo expensive post-processing such as human evaluation.
- *Hypothesis testing* [23] states a null and an alternative hypothesis. The Neyman-Pearson problem minimizes the Type II error (the null hypothesis is false but it fails to be rejected) while keeping the Type I error (the null hypothesis is true but is rejected) small. If the null hypothesis states that a sample has the positive label, then Type II error happens when a positive sample is below the threshold and thus minimizing the Type II error amounts to minimizing the positives below the threshold.

All these three applications may be written (possibly after a reformulation) in a similar form as a minimization of the false-negatives (misclassified positives) below a threshold. They only differ in the way they define the threshold. Despite this striking similarity, they are usually considered separately in the literature. The main goal of our work is to provide unified framework for these kind of problems and perform its theoretical and numerical analysis.

2 General framework

Let $\mathcal{X} = \mathcal{X}^+ \cup \mathcal{X}^- \subset \mathbb{R}^d$ be the set of all samples and let $\mathcal{I}$ be the set of all indexes defined as follows

$$\mathcal{I} = \mathcal{I}^+ \cup \mathcal{I}^-, \quad \mathcal{I}^+ = \left\{ i \mid \boldsymbol{x}_i \in \mathcal{X}^+ \right\}, \quad \mathcal{I}^- = \left\{ i \mid \boldsymbol{x}_i \in \mathcal{X}^- \right\}.$$

Furthermore, n, n^+, n^- denotes sizes of the sets $\mathcal{X}$, $\mathcal{X}^+$, $\mathcal{X}^-$ respectively. To be able to determine the missclassification above and below the threshold t, we define the true-positive, false-negative, true-negative and false-positive counts by

$$\begin{aligned} \operatorname{tp}(\boldsymbol{w}, t) &= \sum_{i \in \mathcal{I}^+} \left[\boldsymbol{w}^\top \boldsymbol{x}_i - t \geq 0 \right], \quad \operatorname{fn}(\boldsymbol{w}, t) = \sum_{i \in \mathcal{I}^+} \left[\boldsymbol{w}^\top \boldsymbol{x}_i - t < 0 \right], \\ \operatorname{tn}(\boldsymbol{w}, t) &= \sum_{i \in \mathcal{I}^-} \left[\boldsymbol{w}^\top \boldsymbol{x}_i - t < 0 \right], \quad \operatorname{fp}(\boldsymbol{w}, t) = \sum_{i \in \mathcal{I}^-} \left[\boldsymbol{w}^\top \boldsymbol{x}_i - t \geq 0 \right]. \end{aligned} \tag{1}$$

Here $[\cdot]$ is the 0-1 loss (also known as Iverson bracket or characteristic function) which is equal to 1 if the argument is true and to 0 otherwise. The discontinuity of the 0-1 loss in (1) is not a nice property and in general, it is difficult to handle optimization problems which contain discontinuous functions. Typical approach to get rid of this unwanted behavior is to employ a surrogate function to approximate the 0-1 loss. As an example of such a surrogate function, we can mention the hinge loss or the truncated quadratic loss defined by

$$l_{\text{hinge}}(s) = \max\{0, 1+s\}, \quad l_{\text{quadratic}}(s) = (\max\{0, 1+s\})^2.$$

In the text below, the symbol l denotes any convex non-negative non-decreasing function with $l(0) = 1$. Using the surrogate function, the counts (1) may be approximated by their surrogate counterparts

$$\begin{aligned} \overline{\text{tp}}(\boldsymbol{w}, t) &= \sum_{i \in \mathcal{I}^+} l(\boldsymbol{w}^\top \boldsymbol{x}_i - t), \quad \overline{\text{fn}}(\boldsymbol{w}, t) = \sum_{i \in \mathcal{I}^+} l(t - \boldsymbol{w}^\top \boldsymbol{x}_i), \\ \overline{\text{tn}}(\boldsymbol{w}, t) &= \sum_{i \in \mathcal{I}^-} l(t - \boldsymbol{w}^\top \boldsymbol{x}_i), \quad \overline{\text{fp}}(\boldsymbol{w}, t) = \sum_{i \in \mathcal{I}^-} l(\boldsymbol{w}^\top \boldsymbol{x}_i - t). \end{aligned} \tag{2}$$

Since $l(\cdot) \geq [\cdot]$, the surrogate counts (2) provide upper approximations of the true counts (1).

As we said before, the goal is to formulate a general problem which minimizes misclassified positive samples below the threshold, i.e. minimize false-negatives. We can use notation (1) and get the following problem

$$\begin{aligned} \underset{\boldsymbol{w}, t}{\text{minimize}} \quad & \frac{1}{n^+} \text{fn}(\boldsymbol{w}, t) \\ \text{subject to} \quad & \text{threshold } t \text{ is a function of } \{\boldsymbol{w}^\top \boldsymbol{x}\}. \end{aligned} \tag{3}$$

Replacing the true counts in (3) by their surrogate counterparts and adding a regularization term results in

$$\begin{aligned} \underset{\boldsymbol{w}, t}{\text{minimize}} \quad & \frac{\lambda}{2} \|\boldsymbol{w}\|^2 + \frac{1}{n^+} \overline{\text{fn}}(\boldsymbol{w}, t) \\ \text{subject to} \quad & \text{threshold } t \text{ is a function of } \{\boldsymbol{w}^\top \boldsymbol{x}\}, \end{aligned} \tag{4}$$

where $\lambda \geq 0$. Note that the objective function consists of the regularization term and the surrogate approximation of the false-negative counts. Furthermore, since we use surrogate approximation (2) of the true counts (1), the objective function is continuous. Another important property in numerical optimization is convexity. It ensures that the optimization problem has neither stationary points nor local minima and all points of interest are global minima. Moreover, many optimization algorithms have guaranteed convergence or faster convergence rates in the convex setting [6]. The following Theorem 2.1 allows us to reduce the analysis of the convexity of the objective function in problem (4) just to analysis of the $\boldsymbol{w} \mapsto t$ function.

Theorem 2.1. *If the threshold t is a convex function of the weights $\boldsymbol{w}$, then function $f(\boldsymbol{w}) = \overline{\text{fn}}(\boldsymbol{w}, t(\boldsymbol{w}))$ is convex*[1].

In the rest of this section, we list methods which fall into the framework of (3) and (4). We divide the methods into three categories and to all categories, we provide several methods which include known methods, new methods and modifications of known methods.

[1] All proofs of theorems and lemmas are available in Appendix on page 21.

2.1 Ranking problems

The goal of the ranking problems is to rank the relevant samples higher than the non-relevant ones. A prototypical example is the RankBoost [12] maximizing the area under the ROC curve, the Infinite Push [1] or the p-norm push [27] which concentrate on the high-ranked negatives and push them down. Since all these papers include pairwise comparisons of all samples, they can be used only for small datasets. This was alleviated in [21] by introducing the *TopPush* method, where the authors performed the limit $p \to \infty$ in p-norm push and obtained the linear complexity in the number of samples. Moreover, since the l_∞-norm is equal to the maximum, this method falls into our framework and can be easily written as the following maximization problem with the threshold equal to the largest score computed from negative samples

$$\begin{aligned} \underset{\boldsymbol{w},t}{\text{maximize}} \quad & \frac{1}{n^+}\text{tp}(\boldsymbol{w},t) \\ \text{subject to} \quad & t = \max_{i\in\mathcal{I}^-} \boldsymbol{w}^\top \boldsymbol{x_i}. \end{aligned} \tag{5}$$

Note that since $\text{tp}(\boldsymbol{w},t) + \text{fn}(\boldsymbol{w},t) = n^+$, maximizing the true-positives is equivalent to minimizing the false-negatives. Thus, we observe that (5) is equivalent to

$$\begin{aligned} \underset{\boldsymbol{w},t}{\text{minimize}} \quad & \frac{1}{n^+}\text{fn}(\boldsymbol{w},t) \\ \text{subject to} \quad & t = \max_{i\in\mathcal{I}^-} \boldsymbol{w}^\top \boldsymbol{x_i}. \end{aligned} \tag{6}$$

As t is a function of the scores $s = \boldsymbol{w}^\top \boldsymbol{x}$, problem (6) is a special case of (3). The *TopPush* method [21] replaces the false-negatives in (6) by their surrogate and adds regularization to arrive at

$$\begin{aligned} \underset{\boldsymbol{w},t}{\text{minimize}} \quad & \frac{\lambda}{2}\|\boldsymbol{w}\|^2 + \frac{1}{n^+}\overline{\text{fn}}(\boldsymbol{w},t) \\ \text{subject to} \quad & t = \max_{i\in\mathcal{I}^-} \boldsymbol{w}^\top \boldsymbol{x_i}. \end{aligned} \tag{\textit{TopPush}}$$

Note that this falls into the framework of (4).

As we will show in Section 3.1, the *TopPush* method is sensitive to outliers and mislabelled data. To robustify it, we follow the idea from [20] and propose to replace the largest negative score by the mean of K largest negative scores. This results in

$$\begin{aligned} \underset{\boldsymbol{w},t}{\text{minimize}} \quad & \frac{\lambda}{2}\|\boldsymbol{w}\|^2 + \frac{1}{n^+}\overline{\text{fn}}(\boldsymbol{w},t) \\ \text{subject to} \quad & t = \frac{1}{K}\sum_{i=1}^{K} s^-_{[i]}, \end{aligned} \tag{\textit{TopPushK}}$$

where s denotes scores corresponding to the samples, i.e. for each sample $\boldsymbol{x}$ score is defined as $s = \boldsymbol{w}^\top \boldsymbol{x}$. The lower index in brackets denotes the order of the score in the sorted sequence from the biggest to the smallest, i.e. $s_{[1]} \geq s_{[2]} \geq \ldots \geq s_{[n]}$. The minus in the upper index means that we select only negative samples.

Due to [20], we know that the mean of the K highest values of a vector is a convex function. Together with the theorem 2.1 this immediately implies that *TopPushK* is a convex problem. Moreover, it also implies that *TopPush* is a convex problem since it is a special case of the *TopPushK* for $K = 1$.

2.2 Accuracy at the top

Accuracy at the Top (τ-quantile) was formally defined in [5] and maximizes the number of relevant samples in the top τ-fraction of ranked samples. When the threshold equals the top τ-quantile of all scores, this problem falls into our framework. The early approaches aim at solving approximations, for example Joachims [17] optimizes a convex upper bound on the number of errors among the top samples. Due to the presence of exponentially many constraints, the method is computationally expensive.

Boyd [5] presented an SVM-like formulation which fixes the index of the quantile and solves n problems. While this removes the necessity to handle the (difficult) quantile constraint, the algorithm is computationally infeasible for a large number of samples. Kar [18] derived upper approximations, their error bounds and solved these approximations. Grill [14] proposed the projected gradient descent method where after each gradient step, the quantile is recomputed. Eban [11] suggested new methods for various criteria and argued that they keep desired properties such as convexity. Mackey [22] generalized this paper by considering a more general setting. Tasche [30] showed that accuracy at the top is maximized by thresholding the posterior probability of the relevant class.

The original formulation of the accuracy at the Top [5] minimizes false-positives above the top τ-quantile

$$\begin{aligned} \underset{\boldsymbol{w},t}{\text{minimize}} \quad & \frac{1}{n^-}\text{fp}(\boldsymbol{w},t) \\ \text{subject to} \quad & t \text{ is the top } \tau\text{-quantile: it solves (8)}, \end{aligned} \tag{7}$$

where the the empirical top τ-quantile is defined by the following formula

$$t_{\text{Q}}(\boldsymbol{w}) = \max\{t \mid \text{tp}(\boldsymbol{w},t) + \text{fp}(\boldsymbol{w},t) \geq n\tau\}. \tag{8}$$

Note that the objective function of the previous problem (7) is not the same as in our general framework (3). However, the following lemma describes the relationship between this objective function and the objective function in (3).

Lemma 2.2. *Denote by t the exact quantile from* (8)*. Then for all $\alpha \in [0,1]$ we have*

$$\text{fp}(\boldsymbol{w},t) = \alpha\text{fp}(\boldsymbol{w},t) + (1-\alpha)\text{fn}(\boldsymbol{w},t) + (1-\alpha)(n\tau - n^+) + (1-\alpha)(q-1), \tag{9}$$

where $q := \#\{\boldsymbol{x} \in \mathcal{X} \mid \ \boldsymbol{w}^\top \boldsymbol{x} = t\}$.

The right-hand side of (9) consists of three parts. The first one is a convex combination of false-positives and false-negatives and the second one is a constant term which has no impact on optimization. Finally, the third term $(1-\alpha)(q-1)$ equals the number of samples for which their classifier equals the quantile. However, this term is small in comparison with the true-positives and the false-negatives and can be neglected. Moreover, when the data are "truly" random such as when measurement errors are present, then $q = 1$ and this term vanishes completely. As a result of this, we may equivalently replace the (7) either by

$$\begin{aligned} \underset{\boldsymbol{w},t}{\text{minimize}} \quad & \frac{1}{n^+}\text{fn}(\boldsymbol{w},t) + \frac{1}{n^-}\text{fp}(\boldsymbol{w},t) \\ \text{subject to} \quad & t \text{ is the top } \tau\text{-quantile: it solves (8)}. \end{aligned} \tag{10}$$

or equivalently by

$$\begin{aligned} \underset{\boldsymbol{w},t}{\text{minimize}} \quad & \frac{1}{n^+}\text{fn}(\boldsymbol{w},t) \\ \text{subject to} \quad & t \text{ is the top } \tau\text{-quantile: it solves (8).} \end{aligned} \tag{11}$$

While the problem (11) falls into our framework (3), the problem (10) corresponds to the original definition from [5] and makes a base for the *Grill* method from [14]. This method replaces false-negative and false-positive counts in the objective by their surrogate counterparts (2). This leads to

$$\begin{aligned} \underset{\boldsymbol{w},t}{\text{minimize}} \quad & \frac{\lambda}{2}\|\boldsymbol{w}\|^2 + \frac{1}{n^+}\overline{\text{fn}}(\boldsymbol{w},t) + \frac{1}{n^-}\overline{\text{fp}}(\boldsymbol{w},t) \\ \text{subject to} \quad & t \text{ is the top } \tau\text{-quantile: it solves (8).} \end{aligned} \tag{*Grill*}$$

The main disadvantage of the *Grill* method is its non-convexity, which makes the optimization hard. This is caused by the empirical quantile (8) which is a non-convex function.

Since (8) finds the threshold t such that the fraction of samples above this threshold amounts τ, we can reformulate (8) into the following problem

$$t_{\text{Q}}(\boldsymbol{w}) \quad \text{solves} \quad \frac{1}{n}\sum_{i\in\mathcal{I}}\Big[\beta(\boldsymbol{w}^\top\boldsymbol{x}_i - t)\Big] = \tau, \tag{12}$$

where β is any positive scalar. This gives the idea to replace the counting function $[\cdot]$ by the surrogate function $l(\cdot)$ to arrive at the surrogate top τ-quantile

$$\bar{t}_{\text{Q}}(\boldsymbol{w}) \quad \text{solves} \quad \frac{1}{n}\sum_{i\in\mathcal{I}} l(\beta(\boldsymbol{w}^\top\boldsymbol{x}_i - t)) = \tau. \tag{13}$$

Then, we provide an alternative to *Grill* by replacing the true quantile by its surrogate counterpart and define propose problem

$$\begin{aligned} \underset{\boldsymbol{w},t}{\text{minimize}} \quad & \frac{\lambda}{2}\|\boldsymbol{w}\|^2 + \frac{1}{n^+}\overline{\text{fn}}(\boldsymbol{w},t) \\ \text{subject to} \quad & t \text{ is the surrogate top } \tau\text{-quantile: it solves (13).} \end{aligned} \tag{*Pat&Mat*}$$

Lemma A.2 shows that the surrogate quantile (13) is a convex approximation of the non-convex quantile (8). Moreover, due to Theorem 2.1 the problem (*Pat&Mat*) is convex and provides a good approximation to the Accuracy at the Top problem. Since this problem is easily solvable due to the convexity and requires almost no tuning, we named it *Pat&Mat* (Precision At the Top & Mostly Automated Tuning).

Remark 2.3. Note that *Grill* minimizes the convex combination of false-positives and false-negatives while (*Pat&Mat*) minimizes only the false-negatives. The reason for this will be evident in Section 3.1 and amounts to preservation of convexity.

The main purpose of the surrogate quantile (13) is to provide a convex approximation of the non-convex quantile (8). We propose another convex approximation based again on [20]. If we take the $n\tau$ largest scores $s = \boldsymbol{w}^\top\boldsymbol{x}$, $\forall\boldsymbol{x}\in\mathcal{X}$, and compute their mean, we arrive at

$$\begin{aligned} \underset{\boldsymbol{w},t}{\text{minimize}} \quad & \frac{\lambda}{2}\|\boldsymbol{w}\|^2 + \frac{1}{n^+}\overline{\text{fn}}(\boldsymbol{w},t) \\ \text{subject to} \quad & t = \frac{1}{n\tau}\sum_{i=1}^{n\tau} s_{[i]}, \end{aligned} \tag{*TopMean*}$$

where we use the same notation for the scores as we used in (*TopPushK*). There is a close connection between *TopPushK* and *TopMean* problems. The former provides stability to *TopPush* and thus, the threshold is computed from negative scores and K is small. On the other hand, the latter uses the threshold as an approximation of the empirical quantile (8) and thus, the threshold is computed from all scores and $K = n\tau$ is large.

2.3 Hypothesis testing

Another category falling into the framework of (3) and (4) is the Neyman-Pearson problem [23] which is closely related to hypothesis testing, where null H_0 and alternative H_1 hypotheses are given. In our case, the null hypothesis H_0 states that a sample $\boldsymbol{x}$ has the negative label. Type I error occurs when H_0 is true but is rejected, i.e. for us Type I error corresponds to false-positives. On the other hand, Type II error happens when H_0 is false but it fails to be rejected. In other words, Type II error corresponds to false-negatives. The standard technique is to minimize Type II error while a bound for Type I error is given. If the bound on Type I error equals τ, we may write this constraint as an empirical quantile defined as follows

$$t_{\mathrm{NP}}(\boldsymbol{w}) = \max\{t \mid \mathrm{fp}(\boldsymbol{w}, t) \geq n^- \tau\}. \tag{14}$$

Then, the Neyman-Pearson problem can be formulated as an minimization of the false negatives (Type II error) in the following form

$$\begin{aligned} \underset{\boldsymbol{w},t}{\text{minimize}} \quad & \frac{1}{n^+}\mathrm{fn}(\boldsymbol{w}, t) \\ \text{subject to} \quad & t \text{ is Type I error at level } \tau\text{: it solves (14).} \end{aligned} \tag{15}$$

Since (15) differs from (11) only by counting only the false-positives in (14) instead of counting all positives in (8), we can derive its three approximations in exactly the same way as in Section 2.2. For this reason, we provide only their brief description.

Replacing the true counts by their surrogates results in the Neyman-Pearson variant of the *Grill* method

$$\begin{aligned} \underset{\boldsymbol{w},t}{\text{minimize}} \quad & \frac{\lambda}{2}\|\boldsymbol{w}\|^2 + \frac{1}{n^+}\overline{\mathrm{fn}}(\boldsymbol{w}, t) + \frac{1}{n^-}\overline{\mathrm{fp}}(\boldsymbol{w}, t) \\ \text{subject to} \quad & t \text{ is the Neyman-Pearson threshold: it solves (14).} \end{aligned} \tag{*Grill-NP*}$$

Similarly as the surrogate quantile (13) is a convex approximation of the non-convex quantile (8), the surrogate Neyman-Pearson threshold given by

$$\bar{t}_{\mathrm{NP}}(\boldsymbol{w}) \quad \text{solves} \quad \frac{1}{n^-}\sum_{i \in \mathcal{I}^-} l(\beta(\boldsymbol{w}^\top \boldsymbol{x}_i - t)) = \tau. \tag{16}$$

is a convex approximation of the non-convex Neyman-Pearson threshold (14). Then, similarly to *Pat&Mat*, we write its Neyman-Pearson variant

$$\begin{aligned} \underset{\boldsymbol{w},t}{\text{minimize}} \quad & \frac{\lambda}{2}\|\boldsymbol{w}\|^2 + \frac{1}{n^+}\overline{\mathrm{fn}}(\boldsymbol{w}, t) \\ \text{subject to} \quad & t \text{ is the surrogate Neyman-Pearson threshold: it solves (16).} \end{aligned} \tag{*Pat&Mat-NP*}$$

Finally, the Neyman-Pearson alternative to *TopMean* reads

$$\begin{aligned} \underset{\boldsymbol{w},t}{\text{minimize}} \quad & \frac{\lambda}{2}\|\boldsymbol{w}\|^2 + \frac{1}{n^+}\overline{\text{fn}}(\boldsymbol{w},t) \\ \text{subject to} \quad & t = \frac{1}{n^-\tau}\sum_{i=1}^{n^-\tau} s_{[i]}, \end{aligned} \qquad (\textit{TopMean-NP})$$

where we use the same notation for the scores as we used in (*TopPushK*) and in (*TopMean*). We may see this problem in two different viewpoints. First, *TopMean-NP* provides a convex approximation of *Grill-NP*. Second, *TopMean-NP* has the same form as *TopPushK*. The only difference is that for *TopMean-NP* we have $K = n^-\tau$ while for *TopPushK* the value of K is small. Thus, even though we started from two different problems, we arrived at two approximations which differ only in the value of one parameter. This shows a close relation of the ranking problem and the Neyman-Pearson problem and the need for a unified theory to handle these tasks.

3 Theoretical analysis of the framework

In this section, we provide a theoretical analysis of the unified framework from Section 2. Since the convexity of the methods (with the exception of the *Grill* method which is not convex) was derived in the previous Section 2, we focus mainly on the undesirable feature of having the global minimum at $\boldsymbol{w} = \mathbf{0}$. Note that convexity makes solving the problem much easier while $\boldsymbol{w} = \mathbf{0}$ does not generate a sensible solution.

3.1 Degenerate behavior

This section provides a simple example of the degenerate behavior of the state of the art method *TopPush*. This example gives us motivation for the extensive theoretical analysis in Section 3.2 which is summarized in Table 2 on page 12,

Consider the case of n negative samples uniformly distributed in $[-1,0]\times[-1,1]$, n positive samples uniformly distributed in $[0,1]\times[-1,1]$ and one negative sample at $(2,0)$, see left part of Figure 1. If n is large, the point at $(2,0)$ is an outlier and the dataset is perfectly separable. The optimal separating hyperplane has the normal vector $\boldsymbol{w} = (1,0)$. Since the methods for the Neyman-Pearson problem are similar to those for the Accuracy at the Top problem, we consider only the five methods described in Sections 2.1 and 2.2 with the hinge loss and no regularization, i.e the regularization parameter is set to $\lambda = 0$. In this example, we show, that some of introduced methods has the optimal separating hyperplane with the normal vector $\boldsymbol{w} = (0,0)$.

Firstly consider the case $\boldsymbol{w}_1 = (0,0)$, then the computation of the threshold and the value of the objective function is easy. It is obvious that $\boldsymbol{w}_1^\top \boldsymbol{x} = 0$ for all $\boldsymbol{x}$ and thus for the threshold we have $t = 0$ for all methods with the exception of *Pat&Mat*, where the threshold is given as the solution of (13)

$$\tau = \frac{1}{n}\sum_{i\in\mathcal{I}} l(\beta(\boldsymbol{w}_1^\top \boldsymbol{x}_i - t)) = l(0-\beta t) = 1-\beta t,$$

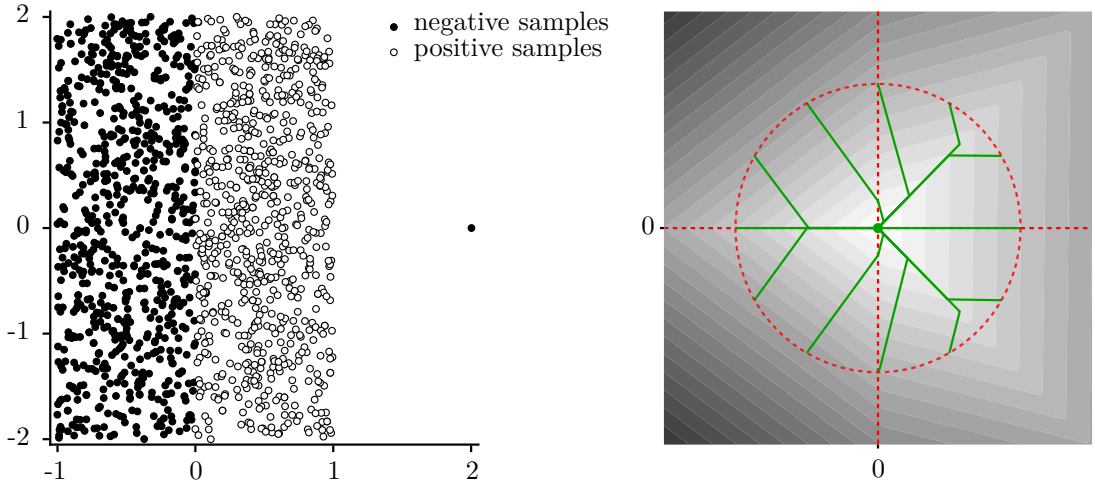

Figure 1: Left: distribution of positive (empty circle) and negative samples (full circles) for the example from Section 3.1. Right: contour plot for *TopPush* and its convergence to zero from 12 initial points.

which implies $t = \frac{1}{\beta}(1-\tau)$. Moreover, for $t \geq 0$ and for all methods with the exception of *Grill* the objective function equals to

$$\frac{1}{n^+}\mathrm{fn}(\boldsymbol{w}_1, t) = \frac{1}{n^+}\sum_{i\in\mathcal{I}^+} l(t-0) = l(t) = 1+t.$$

By substituting the value of the threshold into the objective function, we obtain $f(\boldsymbol{w}_1) = 1 + \frac{1}{\beta}(1-\tau)$ for *Pat&Mat* and $f(\boldsymbol{w}_1) = 1$ for the remaining methods with the exception of *Grill* where we have to add the false-positives to get $f(\boldsymbol{w}_1) = 2$.

Consider now solution $\boldsymbol{w}_2 = (1, 0)$. Since the *TopPush* method chooses the largest negative, it sets $t = 2$. *TopPushK* chooses the mean of K largest negatives and sets $t = \frac{2}{k}$. *Grill* selects the τ-top quantile, which for the uniform distribution of $\boldsymbol{w}^\top \boldsymbol{x}^+$ on the interval $[-1, 1]$ equals to $t = 1 - 2\tau$. For the *Pat&Mat* method and β small enough we have

$$\begin{aligned}\tau &= \frac{1}{n}\sum_{i\in\mathcal{I}} l(\boldsymbol{w}_2^\top \boldsymbol{x}_i - t) \approx \int_{-1}^{1} l(s-t)\,\mathrm{d}s = \int_{-1}^{1} \max\{0, 1+\beta(s-t)\}\,\mathrm{d}s \\ &= \int_{-1}^{1} (1+\beta(s-t))\,\mathrm{d}s = 1 - \beta t + \beta\int_{-1}^{1} s\,\mathrm{d}s = 1-\beta t,\end{aligned} \tag{17}$$

and thus again $t = \frac{1}{\beta}(1-\tau)$. Note that

$$1+\beta(s-t) \geq 1+\beta(-1-t) = 1-\beta-1+\tau = -\beta+\tau$$

and if $\beta \leq \tau$, then we may indeed ignore the max operator in (17). Finally *TopMean* computes the average between the true quantile $1-2\tau$ and the upper bound 1 and thus $t = 1-\tau$. The objective for $t \geq 0$ equals to

$$\frac{1}{n^+}\mathrm{fn}(\boldsymbol{w}_2, t) \approx \int_0^1 l(t-s)\,\mathrm{d}s = \int_0^1 (1+t-s)\,\mathrm{d}s = \frac{1}{2}+t.$$

Then we have $f(\boldsymbol{w}_2) = \frac{1}{2}+t$ for all methods with the exception of *Grill* where we have to add the false-positives to get $f(\boldsymbol{w}_2) = \frac{1}{2}+t+\frac{1}{2}(1-t)^2$.

These results are summarized in Table 1. We chose these two points because both are important: $\boldsymbol{w}_1$ does not generate any separating hyperplane while $\boldsymbol{w}_2$ is the normal vector to the optimal separating hyperplane. Since the dataset is perfectly separable by $\boldsymbol{w}_2$, we expect that $\boldsymbol{w}_2$ provides a lower objective than $\boldsymbol{w}_1$. The lower objective function in Table 1 is highlighted by green color if corresponds to the optimal separating hyperplane $\boldsymbol{w}_2$ and by red color otherwise. We see that *TopPush* and *TopMean* degenerate to the minimum at $\boldsymbol{w}_1$.

It can be shown that $\boldsymbol{w}_1 = (0, 0)$ is even the global minimum for *TopPush* and *TopMean*. This raises the question of whether some tricks, such as early stopping or excluding a small ball around zero, cannot overcome this difficulty. The answer is negative as shown in the right part of Figure 1. Here, we run the *TopPush* method from several starting points and it always converges to zero from one of the three possible directions; all of them far from the normal vector to the separating hyperplane. This shows the need for a formal analysis of the framework proposed in Section 2.

Table 1: Comparison of methods on the very simple problem from Section 3.1. Two methods have the global minimum at $\boldsymbol{w}_1 = (0, 0)$ which does not determine any separating hyperplane. The perfect separating hyperplane is generated by $\boldsymbol{w}_2 = (1, 0)$.

		$\boldsymbol{w}_1 = (0,0)$		$\boldsymbol{w}_2 = (1,0)$	
Method	**Page**	t	f	t	f
TopPush	4	0	1	2	2.5
TopPushK	4	0	1	$\frac{2}{k}$	$0.5 + \frac{2}{k}$
Grill	6	0	2	$1 - 2\tau$	$1.5 + 2\tau(1 - \tau)$
Pat&Mat	6	$\frac{1}{\beta}(1 - \tau)$	$1 + \frac{1}{\beta}(1 - \tau)$	$\frac{1}{\beta}(1 - \tau)$	$0.5 + \frac{1}{\beta}(1 - \tau)$
TopMean	6	0	1	$1 - \tau$	$1.5 - \tau$

3.2 Robustness and global minimum at zero

The convexity derived in Section 2 guarantees that there are no local minima. However, as we have shown in Section 3.1, it may happen that the global minimum is at $\boldsymbol{w} = \boldsymbol{0}$. This is a highly undesirable situation since $\boldsymbol{w}$ is the normal vector to the separating hyperplane and the zero vector provides no information. In this section, we analyze when this situation happens. Recall that the threshold t depends on the weights $\boldsymbol{w}$; sometimes we stress this by writing $t(\boldsymbol{w})$.

The first result states that if the threshold $t(\boldsymbol{w})$ is above a certain value, then zero has a better objective that $\boldsymbol{w}$. If this happens for all $\boldsymbol{w}$, then zero is the global minimum.

Theorem 3.1. *Consider any of these methods: TopPush, TopPushK, TopMean or TopMean-NP. Fix any $\boldsymbol{w}$ and denote the corresponding threshold $t(\boldsymbol{w})$. If we have*

$$t(\boldsymbol{w}) \geq \frac{1}{n^+} \sum_{i \in \mathcal{I}^+} \boldsymbol{w}^\top \boldsymbol{x}_i,$$

then $f(\boldsymbol{0}) \leq f(\boldsymbol{w})$.

We can use this result immediately to deduce that some methods may have the global minimum at $\boldsymbol{w} = \boldsymbol{0}$. More specifically, *TopPush* fails if there are outliers, *TopPushK* fails if there are many outliers and *TopMean* fails whenever there are many positive samples.

Corollary 3.2. *Consider the TopPush method. If the positive samples lie in the convex hull of negative samples, then* $\boldsymbol{w} = \mathbf{0}$ *is the global minimum.*

Corollary 3.3. *Consider the TopMean method. If* $n^+ \geq n\tau$*, then* $\boldsymbol{w} = \mathbf{0}$ *is the global minimum.*

The proof of Theorem 3.1 employs the fact that all methods in the theorem statement have only false-negatives in the objective. If $\boldsymbol{w}_1 = 0$, then $\boldsymbol{w}_1^\top \boldsymbol{x} = 0$ for all samples $\boldsymbol{x}$, the threshold equals to $t = 0$ and the objective equals to one. If the threshold is large for some $\boldsymbol{w}$, many positives are below the threshold and the false-negatives have the average surrogate value larger than one. In such a case, $\boldsymbol{w} = \mathbf{0}$ becomes the global minimum. There are two fixes to this situation. The first one is to include false-positives in the objective. This approach was taken by *Grill* and *Grill-NP* and necessarily results in the loss of convexity. The alternative is to move the threshold from zero even when all scores $\boldsymbol{w}^\top \boldsymbol{x}$ equal to zero. This second approach was taken by our methods *Pat&Mat* and *Pat&Mat-NP*. It keeps the convexity and, as we derive in the next result, the global minimum is away from zero.

Theorem 3.4. *Consider the Pat&Mat or Pat&Mat-NP method with the hinge surrogate and no regularization. Assume that for some* $\boldsymbol{w}$ *we have*

$$\frac{1}{n^+} \sum_{i \in \mathcal{I}^+} \boldsymbol{w}^\top \boldsymbol{x}_i > \frac{1}{n^-} \sum_{j \in \mathcal{I}^-} \boldsymbol{w}^\top \boldsymbol{x}_j.$$

Then there exists scaling parameter β_0 *from* (12) *such that* $\forall \beta \in (0, \beta_0)$ *we have* $f(\boldsymbol{w}) < f(\mathbf{0})$.

We would like to compare the results of Theorems 3.1 and 3.4. The former states that if the average score $\boldsymbol{w}^\top \boldsymbol{x}_i$, $i \in \mathcal{I}^-$, for a small number of largest negative samples is larger than the average score $\boldsymbol{w}^\top \boldsymbol{x}_i$, $i \in \mathcal{I}^+$, for all positive samples, then the corresponding method fails. We show in Lemma A.4 that this number equals to 1 for *TopPush*, to K for *TopPushK* and to $n^-\tau$ for *TopMean-NP*. On the other hand, Theorem 3.4 states that if the average score of all positive samples is larger than the average score of all negative samples, then *Pat&Mat* and *Pat&Mat-NP* do not have problems with zero. Note that since we push positives to the top, for a good classifier scores of positive samples should be higher than scores of the negative samples.

3.3 Method comparison

We provide a visualization of the obtained results in Table 2 and Figure 2. Table 2 gives the basic characterization of the methods such as their source, the criterion they approximate, the hyperparameters, whether the method is convex and whether it has problems with $\boldsymbol{w} = \mathbf{0}$ based on Theorem 3.1.

A similar comparison is performed in Figure 2. Methods in green and light red are convex while methods in dark red are non-convex. Based on Theorem 3.1, four methods in light red are vulnerable to have the global minimum at $\boldsymbol{w} = \mathbf{0}$. This theorem states that the higher the threshold, the more vulnerable the method is. This dependence is depicted by the full arrows. If it points from one method to another, the latter one has a smaller threshold and thus is less vulnerable to this undesired global minima. The dotted arrows indicate that this holds true usually but not always. This complies with Corollaries 3.2 and 3.3 which state that *TopPush* and *TopMean* are most vulnerable.

Table 2: Summary of the methods from Section 2. The table shows their source or the source they are based on, the criterion they approximate, the hyperparameters, whether the method is convex and whether the method is robust (in the sense of having problems with $\boldsymbol{w} = \mathbf{0}$).

Method	Page	Source	Criterion	Hyperpars	Convex	Robust
TopPush	4	[21]	(5)	λ	✓	✗
TopPushK	4	Ours[a]	(5)	λ, K	✓	✗
Grill	6	[14]	(7)	λ	✗	✓
Pat&Mat	6	Ours[b]	(7)	β, λ	✓	✓
TopMean	6	Ours[c]	(7)	λ	✓	✗
Grill-NP	7	Ours[d]	(15)	λ	✗	✓
Pat&Mat-NP	7	Ours[b]	(15)	β, λ	✓	✓
TopMean-NP	8	Ours[c]	(15)	λ	✓	✗

[a] *TopPushK* is a is a modification of [21]
[b] *Pat&Mat* and *Pat&Mat-NP* are loosely connected to [26]
[c] *TopMean* and *TopMean-NP* are inspired by [20]
[d] *Grill-NP* is a modification of [14]

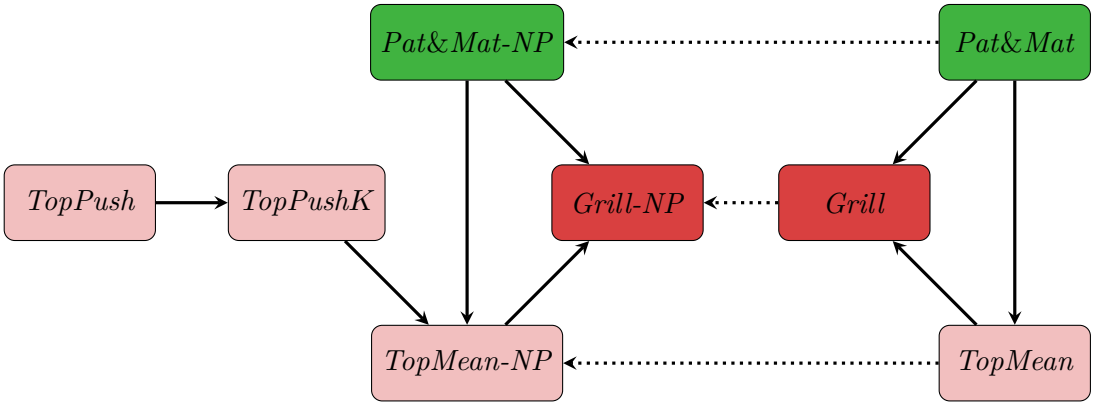

Figure 2: Summary of the methods from Section 2. Methods in green and light red are convex while methods in dark red are non-convex. Methods in light red are vulnerable to have the global minimum at $\boldsymbol{w} = \mathbf{0}$. Full (dotted) arrow pointing from one method to another show that the latter method has always (usually) smaller threshold.

4 Numerical considerations

So far, we have performed a theoretical analysis of the presented methods. In this section, we present some numerical considerations as well.

4.1 Gradient computation and variable reduction

The decision variables in (4) are the normal vector of the separating hyperplane $\boldsymbol{w}$ and the threshold t. To apply an efficient optimization method, we need to compute gradients. The simplest idea is to perform the gradient only with respect to $\boldsymbol{w}$ and then recompute t. This was followed in [14]. However, there is a more sophisticated way based on the implicit function

theorem. For each $\boldsymbol{w}$, the threshold t can be computed uniquely. We stress this dependence by writing $t(\boldsymbol{w})$ instead of t. By doing so, we effectively remove the threshold t from the decision variables and $\boldsymbol{w}$ remains the only decision variable. Note that the convexity is preserved. Then we can compute the derivative via the chain rule

$$\begin{aligned} f(\boldsymbol{w}) &= \frac{1}{n^+} \sum_{i \in \mathcal{I}^+} l(t(\boldsymbol{w}) - \boldsymbol{w}^\top \boldsymbol{x}_i) + \frac{\lambda}{2} \|\boldsymbol{w}\|^2, \\ \nabla f(\boldsymbol{w}) &= \frac{1}{n^+} \sum_{i \in \mathcal{I}^+} l'(t(\boldsymbol{w}) - \boldsymbol{w}^\top \boldsymbol{x}_i)(\nabla t(\boldsymbol{w}) - \boldsymbol{x}_i) + \lambda \boldsymbol{w}. \end{aligned} \tag{18}$$

The only remaining part is the computation of $\nabla t(w)$. We show here an efficient computation for *TopPushK* and *Pat&Mat*. For other methods, it can be performed in a similar manner.

For *TopPushK*, we recall that the threshold t equals to the mean of K largest scores $s = \boldsymbol{w}^\top \boldsymbol{x}_i$, for all negative samples $i \in \mathcal{I}^-$. The locally, this is a linear function for which it is simple to compute derivatives

$$\nabla t = \frac{1}{K} \sum_{j=1}^{K} \boldsymbol{x}_{l_j},$$

where $l_1, l_2, \ldots, l_K$ are indeces of K largest scores coresponding to the negative samples. To prevent unnecessary computations for (18), we summarize the computations of derivatives for *TopPushK* in Algorithm 4.1.

Algorithm 4.1 Efficient computation of (18) for *TopPushK*.

1: Compute the scores $s_i = \boldsymbol{w}^\top \boldsymbol{x}_i$ for all indices $i \in \mathcal{I}$.
2: Find K indices $l_1, \ldots, l_K$ where $\boldsymbol{s}$ has the greatest values on $\mathcal{I}^-$.
3: Compute the threshold $t = \frac{1}{k} \sum_{j=1}^{K} s_{l_j}$.
4: Compute the threshold derivative $\nabla t = \frac{1}{K} \sum_{j=1}^{K} \boldsymbol{x}_{l_j}$.
5: Compute the derivative as $\frac{1}{n_+} \sum_{i \in \mathcal{I}_+} l'(t - s_i)(\nabla t - \boldsymbol{x}_i) + \lambda \boldsymbol{w}$.

For the *Pat&Mat* method, the computation is slightly more difficult. The threshold t is defined through equation

$$\frac{1}{n} \sum_{i \in \mathcal{I}} l\Big(\beta\Big(\boldsymbol{w}^\top \boldsymbol{x}_i - t(\boldsymbol{w})\Big)\Big) = \tau,$$

i.e. there is no explicit relation between the threshold t and the weights $\boldsymbol{w}$. However, differentiating this equation with respect to $\boldsymbol{w}$ results in

$$\sum_{i \in \mathcal{I}} l'\Big(\beta\Big(\boldsymbol{w}^\top \boldsymbol{x}_i - t(\boldsymbol{w})\Big)\Big)(\boldsymbol{x}_i - \nabla t(\boldsymbol{w})) = 0,$$

and thus

$$\nabla t(\boldsymbol{w}) = \frac{\sum_{i \in \mathcal{I}} l'\Big(\beta\Big(\boldsymbol{w}^\top \boldsymbol{x}_i - t(\boldsymbol{w})\Big)\Big) \boldsymbol{x}_i}{\sum_{i \in \mathcal{I}} l'(\beta(\boldsymbol{w}^\top \boldsymbol{x}_i - t(\boldsymbol{w})))}.$$

Similarly to the previous case, we present the efficient computation of (18) in Algorithm 4.2.

Algorithm 4.2 Efficient computation of (18) for *Pat&Mat*.

1: Compute the scores $s_i = \boldsymbol{w}^\top \boldsymbol{x}_i$ for all indices $i \in \mathcal{I}$.
2: Solve the equation $\sum_{i\in\mathcal{I}} l(\beta(z_i - t)) = n\tau$ for the threshold t.
3: Compute the threshold derivative $\nabla t = \frac{\sum_{i\in\mathcal{I}} l'(\beta(s_i - t))\boldsymbol{x}_i}{\sum_{i\in\mathcal{I}} l'(\beta(s_i - t))}$.
4: Compute the derivative as $\frac{1}{n_+}\sum_{i\in\mathcal{I}_+} l'(t - s_i)(\nabla t - \boldsymbol{x}_i) + \lambda\boldsymbol{w}$.

4.2 Implementational details

We recall that all methods fall into the framework of either (3) or (4). Since the threshold t depends on the weights $\boldsymbol{w}$, we can consider the decision variable to be only $\boldsymbol{w}$. Then to apply a method, we implemented the iterative procedure described in Algorithm 4.3.

Algorithm 4.3 General gradient descent based procedure.

1: Initialization: $j_{\max}$, $\boldsymbol{w}^0$.
2: Set $j \leftarrow 0$.
3: **while** $j \leq j_{\max}$ **do**
4: Compute the threshold $t^j = t(\boldsymbol{w}^j)$.
5: Compute the gradient of the objectiv fuction according to (18).
6: Apply the ADAM scheme [19] to get the descent step $\boldsymbol{p}^j$.
7: Update the weights $\boldsymbol{w}^{j+1} \leftarrow \boldsymbol{w}^j - \boldsymbol{p}^j$.
8: Update $j \leftarrow j + 1$.
9: **end while**
10: Compute the threshold $t^{j_{\max}} = t(\boldsymbol{w}^{j_{\max}})$.
11: Return $\boldsymbol{w}^{j_{\max}}$, $t^{j_{\max}}$.

5 Numerical experiments

5.1 Dataset description

For the numerical results, we considered nine datasets summarized in Table 3. Five of them are standard and can be downloaded from the UCI repository. Datasets Ionosphere [29] and Spambase are small, Hepmass [3] contains a large number of samples while Gisette [15] contains a large number of features. To make the dataset more difficult, we created Hepmass 10% dataset which is a random subset of the Hepmass dataset and reduces the fraction of positive samples from 50% to 10%.

Besides these datasets, we also considered two real-world datasets CTA and NetFlow obtained from existing network intrusion detection systems. The first one, further called CTA, was created by the Cisco's Cognitive Threat Analytics engine [8] which analyzes HTTP proxy logs (typically produced by proxy servers located on a network perimeter). The second one, further called NetFlow, was collected by the NetFlow anomaly detection engine [25, 13] which processes NetFlow [2] records exported by routers and other network traffic shaping devices. All samples were manually labelled by experienced Cisco analysts. Therefore it is safe to assume that samples labelled as malicious are indeed malicious. However, it may happen that some samples labelled as legitimate can be actually malicious, which typically happens due to

Table 3: Structure of the used datasets. The training, validation and testing sets show the number of features m, minibatches $n_{\min}$, samples n and the fraction of positive samples $\frac{n^+}{n}$.

			Training		**Validation**		**Testing**	
	m	$n_{\min}$	n	$\frac{n^+}{n}$	n	$\frac{n^+}{n}$	n	$\frac{n^+}{n}$
Ionosphere	34	1	175	36.0%	88	35.2%	88	36.4%
Spambase	57	1	2 300	39.4%	1 150	39.4%	1 151	39.4%
Gisette	5 001	1	6 000	50.0%	500	50.0%	500	50.0%
Hepmass	28	40	5 250 000	50.0%	2 625 040	50.0%	2 624 960	50.0%
Hepmass 10	28	40	2 916 636	10.0%	1 458 320	10.0%	1 458 240	10.0%
CTA	34	26	263 796	2.1%	263 796	2.2%	527 488	2.1%
CTA MLT	34	9	91 314	0.3%	91 314	0.3%	182 592	2.1%
NetFlow	25	8	142 676	8.7%	142 674	8.7%	285 454	8.6%
NetFlow MLT	25	8	145 260	2.7%	145 261	2.8%	285 508	8.6%

the new type of attack (malware) with a different behaviour from what has been seen in the past. To study classifier robustness to this type of errors, we follow the experimental protocol of [14] and besides the original dataset, we inject artificial noise *MLT* which corresponds to the situation where security analysts have failed to identify 50% of attack types and partially mislabelled the other 50% attack types. Thus, some positive samples are either missing or have a wrong (negative) label in the training and validation set. However, they are present in the testing set. A more precise description of the used datasets and the data generating and labelling process is in [14].

5.2 Performance criteria

In Section 2, we described three classes of methods, each optimizing a different criterion. Utilizing the "Criterion" column in Table 2, we summarize these three criteria in Table 4. We recall that the precision and recall are for a threshold t defined by

$$\text{Precision} = \frac{\text{tp}(\boldsymbol{w}, t)}{\text{tp}(\boldsymbol{w}, t) + \text{fp}(\boldsymbol{w}, t)}, \quad \text{Recall} = \frac{\text{tp}(\boldsymbol{w}, t)}{\text{tp}(\boldsymbol{w}, t) + \text{fn}(\boldsymbol{w}, t)}.$$

We will show the Precision-Recall (PR) and the Precision-τ (Pτ) curves. The former is a well-accepted visualization for highly unbalanced data [10] while the latter better reflects that the methods concentrate only on the top of the scores.

Table 4: The criteria for the three classes from Section 2.

Name	**Description**
Positives@Top	Fraction of positives above the largest negative (5).
Positives@Quantile	Fraction of positives above the τ-quantile (7).
Positives@NP	Fraction of positives above the Neyman-Pearson threshold (15).

5.3 Hyperparameter choice

In Algorithm 4.3, we described the iterative procedure used for all of the methods from Section 2. Only for *Grill* and *Grill-NP* method we stick to the original paper [14] and apply the projection of weights $\boldsymbol{w}$ onto the l_2-unit ball after each gradient step (this is not advised for other methods as convexity would be lost). All methods used the hinge surrogate (2) and the maximal number of iterations were set to 1000 iterations. Moreover, for large datasets, we replaced the standard gradient descent by its stochastic counterpart [4] where the threshold t^j and the gradient are computed only on a part of the dataset called a minibatch.

In the experimental part, we use the following sets of hyperparameters for the methods

$$\tau \in \{0.01,\ 0.02,\ 0.03,\ 0.04,\ 0.05,\ 0.1\}, \qquad \beta \in \{0.0001,\ 0.001,\ 0.01,\ 0.1,\ 1,\ 10\},$$
$$\lambda \in \{0,\ 0.00001,\ 0.0001,\ 0.001,\ 0.01,\ 0.1\}, \quad K \in \{1,\ 3,\ 5,\ 10,\ 15,\ 20\}.$$

Since *TopPushK*, *Pat&Mat* and *Pat&Mat-NP* use two hyperparameters, we fixed $\lambda = 0.001$ for these methods to have only six hyperparameters for all methods. For all datasets, we choose the hyperparameter which minimized the corresponding criterion from Table 2 on the validation set. The results are computed on the testing set which was not used during training the methods.

Note that *TopPush* was originally implemented in the dual. However, to allow for the same framework and for the stochastic gradient descent, we implemented it in the primal. These two approaches are, at least theoretically, equivalent.

5.4 Results

In this section, we present the numerical results. Note that all results are computed on the testing set which was not available during training. We run the algorithms from $\boldsymbol{w}^0 = \mathbf{0}$ and from a randomly generated point in the interval $[-1, 1]$. Since the former showed a better performance in 58.7% cases, we show only the results starting from zero.

Having shown a good performance of the methods, we focus on their comparison. Since all methods optimize one of the criteria from Table 4, we base the comparison on these criteria. First, we consider 14 methods (we count different values of τ as a different method) as depicted in Table 5. Note that *TopPushK* for $K = 1$ reduces to the classical *TopPush*. For each dataset and each method, we evaluated criteria from Table 4. For each dataset and criterion, we computed the rank of all methods and averaged them with respect to all datasets. Rank 1 refers to the best performance for given criteria while rank 14 is the worst. Since criteria are in columns, the comparison among methods is column-wise. The best method is depicted in dark green and comparable methods (at most one rank away) are depicted in light green. Based on the results in the Table 5, we make several observations:

- *TopPushK* performs better than *TopPush*. We relate this to the greater stability given by considering K largest negatives in *TopPushK* instead of 1 in *TopPush*. This may have alleviated the problems with $\boldsymbol{w} = \mathbf{0}$ as suggested in Theorem 3.1.
- Neither *Grill* nor *Grill-NP* perform well. We believe that this is due to the lack of convexity as indicated in Theorem 2.1 and the discussion thereafter.
- *TopMean* does not perform well either. Since the thresholds τ are small, then $\boldsymbol{w} = \mathbf{0}$ is the global minimum as proved in Corollary 3.3.

Table 5: The average rank of all methods across all datasets on the criteria from Table 4. Rank 1 is the best and rank 14 is the worst. This evaluation is performed for each criterion. Dark green depicts the best method while light green depicts comparable methods.

		Positives@Top	**Positives@Quantile**		**Positives@NP**	
			$\tau = 0.01$	$\tau = 0.03$	$\tau = 0.01$	$\tau = 0.03$
TopPush		5.5	8.1	8.4	8.3	9.3
TopPushK		4.2	7.2	8.4	7.4	7.3
Grill	$\tau = 0.01$	10.1	9.7	11.2	11.4	10.9
	$\tau = 0.03$	8.9	10.1	10.3	11.1	10.1
Pat&Mat	$\tau = 0.01$	8.1	7.1	6.8	5.6	6.1
	$\tau = 0.03$	6.3	6.8	5.5	5.9	5.2
TopMean	$\tau = 0.01$	9.1	10.3	8.8	9.9	10.7
	$\tau = 0.03$	9.4	9.1	8.3	9.8	10.2
Grill-NP	$\tau = 0.01$	9.6	8.4	9.4	10.2	9.8
	$\tau = 0.03$	10.8	8.5	9.3	9.3	7.6
Pat&Mat-NP	$\tau = 0.01$	6.6	5.0	3.4	2.8	4.4
	$\tau = 0.03$	6.9	5.4	4.7	4.3	3.2
TopMean-NP	$\tau = 0.01$	4.9	5.3	6.3	5.9	6.0
	$\tau = 0.03$	4.6	4.1	4.0	2.9	4.1

- *TopPush* and *TopPushK* methods work better on the Positives@Top criterion. This complies with the theory as these methods were designed for this criterion.
- *Pat&Mat*, *Pat&Mat-NP* and *TopMean-NP* methods work better on the Positives@Quantile and Positives@NP criterion. This complies with the theory as these methods were designed for one of these criteria.

In Table 6 we investigate the impact of $\boldsymbol{w} = \mathbf{0}$ as a potential global minimum. Each method was optimized for six different values of hyperparameters. The table depicts the condition under which the final value has a lower objective that $\boldsymbol{w} = \mathbf{0}$. Thus, ✓ means that it is always better while ✗ means that the algorithm made no progress from the starting point $\boldsymbol{w} = 0$. The latter case implies that $\boldsymbol{w} = \mathbf{0}$ seems to be the global minimum. We make the following observations:

- *TopPushK* has a lower number of successes than *Pat&Mat-NP* which corresponds to Figure 2 showing that the latter method has a lower threshold.
- Similarly, Figure 2 states that the methods from Section 2.2 has a higher threshold than their Neyman-Pearson variants from Section 2.3. This is documented in the table as the latter have a higher number of successes.
- *Pat&Mat* and *Pat&Mat-NP* are the only methods which succeeded at every dataset for some hyperparameter. Moreover, for each dataset, there was some β_0 such that these methods were successful if and only if $\beta \in (0, \beta_0)$. This is in agreement with Theorem 3.4. The only exception was Spambase which we attribute to numerical errors.

Table 6: Necessary hyperparameter choice for the solution to have a better objective than zero. ✓ means that the solution was better than zero for all hyperparameters while ✗ means that it was worse for all hyperparameters.

		Ionosphere	**Spambase**	**Gisette**	**Hepmass**	**CTA**	**NetFlow**
TopPush		✓	✗	✓	✗	✓	✗
TopPushK		✓	✗	✓	✗	✓	✗
Grill	$\tau = 0.01$	✗	✗	✗	✗	✓	✓
	$\tau = 0.03$	✓	✗	✗	✗	✓	✓
Pat&Mat	$\tau = 0.01$	$\beta \leq 0.1$	$\beta = 0.001$	$\beta \leq 0.001$	$\beta \leq 0.1$	$\beta \leq 0.1$	$\beta \leq 0.1$
	$\tau = 0.03$	$\beta \leq 0.1$	$\beta = 0.01$	$\beta \leq 0.001$	$\beta \leq 0.1$	$\beta \leq 0.1$	$\beta \leq 0.1$
TopMean	$\tau = 0.01$	✗	✗	✗	✗	✗	✗
	$\tau = 0.03$	✗	✗	✗	✗	✓	✗
Grill-NP	$\tau = 0.01$	✓	✗	✗	✗	✓	✓
	$\tau = 0.03$	✓	✗	✗	✗	✓	✓
Pat&Mat-NP	$\tau = 0.01$	$\beta \leq 1$	$\beta = 0.01$	✓	$\beta \leq 0.1$	✓	$\beta \leq 0.1$
	$\tau = 0.03$	$\beta \leq 1$	$\beta = 0.1$	✓	$\beta \leq 0.1$	✓	$\beta \leq 0.1$
TopMean-NP	$\tau = 0.01$	✓	$\lambda = 0.001$	✓	✗	✓	✗
	$\tau = 0.03$	✓	✓	✓	✗	✓	✗

- *TopMean* fails everywhere which agrees with Corollary 3.3. The only exception is the case of $\tau = 3\%$ and CTA which has only 2.1% positive samples. This again agrees with Corollary 3.3.

In Figure 3 we present the Pτ curves which show the precision at the top τ-quantile. They are equivalent to the well-known PR curves. Each row corresponds to one dataset (NetFlow top row, Gisette middle row and Hepmass bottom row), the left column shows the whole interval for the quantile $\tau \in [0, 1]$ while the right one is its zoomed version to $\tau \in [0, 0.01]$. We show the *TopPushK*, *Pat&Mat*, *Pat&Mat-NP* and *TopMean-NP* methods which perform the best as can be seen from the previous results. Note that the right column is more important as all these method focus on maximizing the accuracy only on the top of the dataset, which precisely corresponds to small values of τ. The *Pat&Mat* and *Pat&Mat-NP* methods perform the best on larger quantiles. *TopPushK* performs reasonably well for large quantiles while its performance increases with decreasing τ. In most cases, the precision is 1 when τ is sufficiently small.

The final Table 7 depicts the time needed for one iteration in milliseconds. We show four selected representative methods. Note that the time is relatively stable and even for most of the datasets it is below one millisecond. Thus, the whole optimization for one hyperparameter (without loading and evaluation) can be usually performed within one second.

Table 7: Time in miliseconds needed for one iteration.

	Ionosphere	**Spambase**	**Gisette**	**Hepmass**	**CTA**	**NetFlow**
TopPushK	0.0	0.1	22.3	5.6	0.5	0.9
Grill	0.0	0.1	41.5	6.2	0.5	1.1
Pat&Mat	0.0	0.2	38.4	8.4	0.4	1.1
TopMean	0.0	0.2	43.9	7.1	0.4	0.9

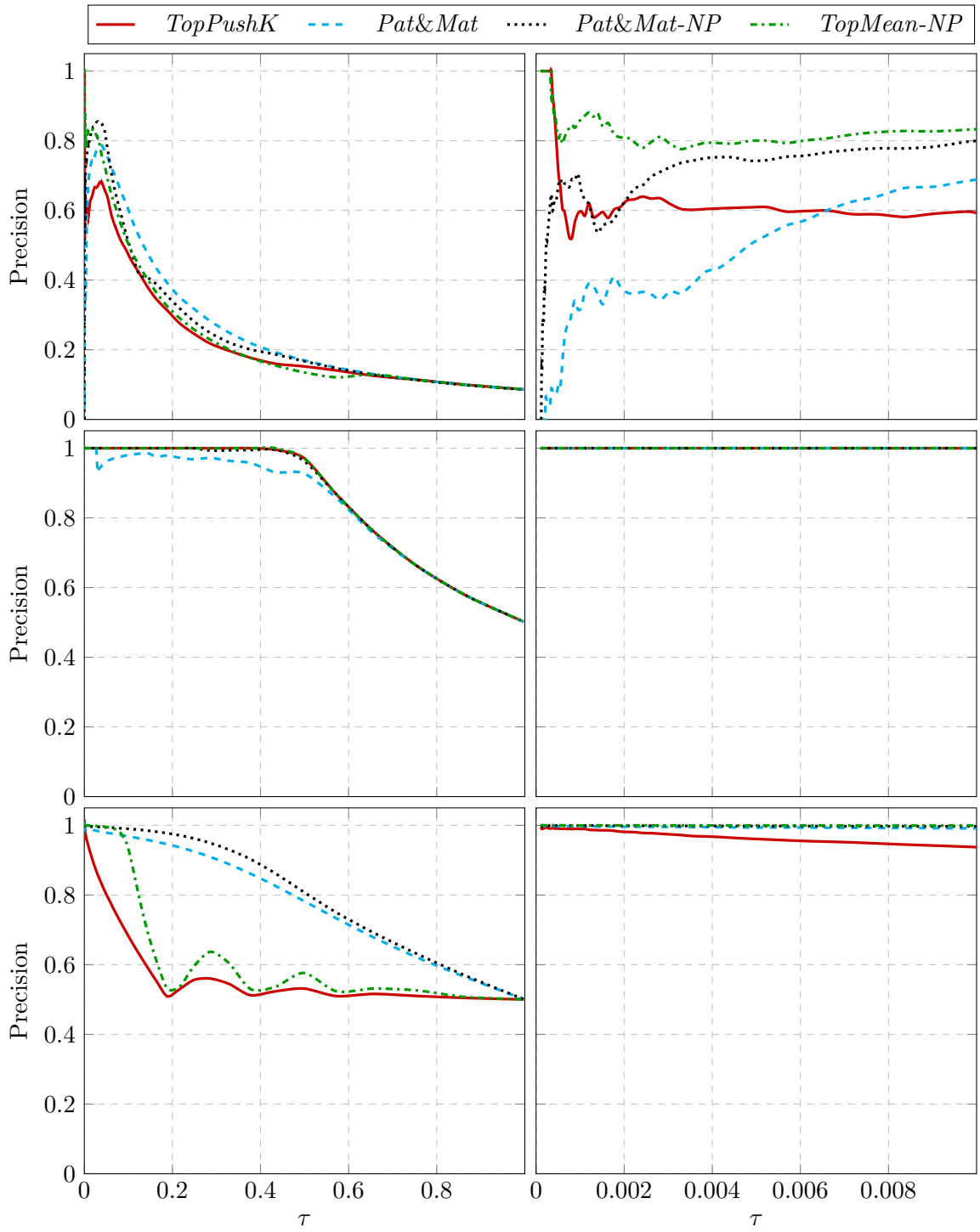

Figure 3: Pτ curves for datasets NetFlow (top row), Gisette (middle row) and Hepmass (bottom row) and two different zooms of $\tau \in [0, 1]$ and $\tau \in [0, 0.01]$ (columns).

6 Conclusion and future work

In this work we achieved the following results:

- We presented a unified framework for the three criteria from Section 2.
- We showed which known methods (*TopPush*, *Grill*) fall into our framework and derived both completely new methods (*Pat&Mat*, *Pat&Mat-NP*, *TopMean*) and modifications of known methods (*TopPushK*, *Grill-NP*, *TopMean-NP*).
- We performed a theoretical analysis of the methods. We showed that both known methods suffer from certain disadvantages. While *TopPush* is sensitive to outliers, *Grill* is non-convex.
- We performed a numerical comparison where we showed a good and fast performance. The methods converge within seconds on datasets with 5 million training samples.
- The extensive theoretical analysis is supported by the numerical analysis.

Based on the results, we recommend using *TopPushK* or *TopMean-NP* for extremely small τ. For larger τ, we recommend using *Pat&Mat* or *Pat&Mat-NP* as they are convex and do not suffer from problems at $\boldsymbol{w} = \mathbf{0}$.

6.1 Future work

All presented methods cover only the case where the linear classification is sufficient. However, many problems are not linearly separable and then the non-linear classifiers are needed. One could say that the easiest way how to incorporate non-linearity in the formulation of our framework (3) is to replace the linear classifier by the non-linear one. The problem is that it would only lead to the loss of the convexity and the task will be hardly solvable. Nevertheless, we can use existing theory to get some inspiration for our future work. The key point is to realize that our framework is really similar to the primal formulation of the famous support vector machines [9]. The classical way how to incorporate non-linearity into SVM is to derive the dual formulation [6] corresponding to the primal problem and use the kernels methods [28]. The main problem of this approach is that dimension of the dual formulation depends on the number of samples and as a consequence, it is hard to solve SVM in dual for large datasets. However, there are two main ways how to increase the scalability which are used for SVM.

- The first way is to reduce the number of data samples. This approach uses the well-known feature of SVM that the resulting classifier is determined only by the subset of all training samples called support vectors. Since the classifier is characterized only by support vectors and the rest of the samples has no effect on the classification result, it is possible to reduce the training data set to a set of support vectors. Data reduction can easily be used in the testing phase since a set of support vectors is known at this stage. On the other hand, it is hard to use this property in the training phase because it is generally difficult to determine which samples are support vectors during this phase. However, there are for example screening methods [24, 31] which solve this problem and try to identify non-support vectors to reduce the number of samples during the training phase.

- The second way is to improve the training method. Such an improvement can be made, for example, by the coordinate descent method [7, 16]. The key idea of this approach is to update only one variable (or a few) instead of updating all variables at once. Solving the subproblem for just one variable is much easier than solving the whole dual task. In addition, the subproblem of one variable sometimes has an analytical solution and can be solved at a constant time.

Our main object of interest for the future is to properly derive and solve dual formulations of all methods described in this work.

A Additional results and proofs

Here, we provide additional results and proofs of results mentioned in the main body. For convenience, we repeat the result statements.

A.1 Equivalence of (7), (10) and (11)

To show this equivalence, we will start with an auxiliary lemma.

Lemma A.1. *Denote by t the exact quantile from (8). Then for all $\alpha \in [0, 1]$ we have*

$$\mathrm{fp}(\boldsymbol{w}, t) = \alpha\mathrm{fp}(\boldsymbol{w}, t) + (1-\alpha)\mathrm{fn}(\boldsymbol{w}, t) + (1-\alpha)(n\tau - n^+) + (1-\alpha)(q-1),$$

where $q := \#\{\boldsymbol{x} \in \mathcal{X} \mid \boldsymbol{w}^\top \boldsymbol{x} = t\}$.

Proof. By the definition of the quantile we have

$$\mathrm{tp}(\boldsymbol{w}, t) + \mathrm{fp}(\boldsymbol{w}, t) = n\tau + q - 1.$$

This implies

$$\mathrm{fp}(\boldsymbol{w}, t) = n\tau + q - 1 - \mathrm{tp}(\boldsymbol{w}, t) = n\tau + q - 1 - n^+ + \mathrm{fn}(\boldsymbol{w}, t).$$

From this relation we deduce

$$\begin{aligned}\mathrm{fp}(\boldsymbol{w}, t) &= \alpha\mathrm{fp}(\boldsymbol{w}, t) + (1-\alpha)\mathrm{fp}(\boldsymbol{w}, t) = \alpha\mathrm{fp}(\boldsymbol{w}, t) + (1-\alpha)\Big(\mathrm{fn}(\boldsymbol{w}, t) + n\tau - n^+ + q - 1\Big) \\ &= \alpha\mathrm{fp}(\boldsymbol{w}, t) + (1-\alpha)\mathrm{fn}(\boldsymbol{w}, t) + (1-\alpha)\Big(n\tau - n^+\Big) + (1-\alpha)(q-1),\end{aligned}$$

which is precisely the lemma statement. □

A.2 Results related to convexity and continuity

Theorem 2.1 (page 3) *If the threshold t is a convex function of the weights $\boldsymbol{w}$, then function $f(\boldsymbol{w}) = \overline{\mathrm{fn}}(\boldsymbol{w}, t(\boldsymbol{w}))$ is convex.*

Proof. Due to the definition of the surrogate counts (2), the objective of (4) equals to

$$\frac{1}{n^+} \sum_{i \in \mathcal{I}^+} l\Big(t(\boldsymbol{w}) - \boldsymbol{w}^\top \boldsymbol{x}_i\Big).$$

Here we write $t(\boldsymbol{w})$ to stress the dependence of t on $\boldsymbol{w}$. Since $\boldsymbol{w} \mapsto t(\boldsymbol{w})$ is a convex function, we also have that $\boldsymbol{w} \mapsto t(\boldsymbol{w}) - \boldsymbol{w}^\top \boldsymbol{x}$ is a convex function. From its definition, the surrogate function l is convex and non-decreasing. Since a composition of a convex function with a non-decreasing convex function is a convex function, this finishes the proof. □

Lemma A.2. *Functions $\boldsymbol{w} \mapsto \bar{t}_{\mathrm{Q}}(\boldsymbol{w})$ defined in (13) and $\boldsymbol{w} \mapsto \bar{t}_{\mathrm{NP}}(\boldsymbol{w})$ defined in (16) are convex.*

Proof. We will show this result only for the function $\boldsymbol{w} \mapsto \bar{t}_{\mathrm{Q}}(\boldsymbol{w})$. For the second function, it can be shown in a identical way. This former function is defined via the implicit equation

$$g(\boldsymbol{w}, t) := \frac{1}{n} \sum_{i \in \mathcal{I}} l\Big(\boldsymbol{w}^\top \boldsymbol{x}_i - t\Big) - \tau = 0.$$

Since l is convex, we immediately obtain that g is jointly convex in both variables.

To show the convexity, consider $\boldsymbol{w}_1$, $\boldsymbol{w}_2$ and the corresponding $t_1 = \bar{t}_{\mathrm{Q}}(\boldsymbol{w}_1)$, $t_2 = \bar{t}_{\mathrm{Q}}(\boldsymbol{w}_2)$. Note that this implies

$$g(\boldsymbol{w}_1, t_1) = g(\boldsymbol{w}_2, t_2) = 0. \tag{19}$$

Then for any $\lambda \in [0, 1]$ we have

$$g(\lambda \boldsymbol{w}_1 + (1-\lambda)\boldsymbol{w}_2,\ \lambda t_1 + (1-\lambda) t_2) \leq \lambda g(\boldsymbol{w}_1, t_1) + (1-\lambda) g(\boldsymbol{w}_2, t_2) = 0, \tag{20}$$

where the inequality follows from the convexity of g and the equality (19). From the definition of the surrogate quantile we have

$$g(\lambda \boldsymbol{w}_1 + (1-\lambda)\boldsymbol{w}_2, \bar{t}_{\mathrm{Q}}(\lambda \boldsymbol{w}_1 + (1-\lambda)\boldsymbol{w}_2)) = 0. \tag{21}$$

Since g is non-increasing in the second variable, from (20) and (21) we deduce

$$\bar{t}_{\mathrm{Q}}(\lambda \boldsymbol{w}_1 + (1-\lambda)\boldsymbol{w}_2) \leq \lambda t_1 + (1-\lambda) t_2 = \lambda \bar{t}_{\mathrm{Q}}(\boldsymbol{w}_1) + (1-\lambda) \bar{t}_{\mathrm{Q}}(\boldsymbol{w}_2),$$

which implies that function $\boldsymbol{w} \mapsto \bar{t}_{\mathrm{Q}}(\boldsymbol{w})$ is convex. □

Lemma A.3. *Functions $\boldsymbol{w} \mapsto t_{\mathrm{Q}}(\boldsymbol{w})$ defined in (8) and $\boldsymbol{w} \mapsto t_{\mathrm{NP}}(\boldsymbol{w})$ defined in (14) are Lipschitz continuous.*

Proof. The quantile $t_{\mathrm{Q}}(\boldsymbol{w})$ equals to one or more scores $z = \boldsymbol{w}^\top \boldsymbol{x}$. This implies that function $\boldsymbol{w} \mapsto t_{\mathrm{Q}}(\boldsymbol{w})$ is piecewise linear which implies that it is Lipschitz continuous. □

A.3 Results related to the global minima at zero

Theorem 3.1 (page 10) *Consider any of these methods: TopPush, TopPushK, TopMean or TopMean-NP. Fix any $\boldsymbol{w}$ and denote the corresponding threshold t. If we have*

$$t \geq \frac{1}{n^+} \sum_{i^+ \in \mathcal{I}^+} \boldsymbol{w}^\top \boldsymbol{x}_i, \tag{22}$$

then $f(\mathbf{0}) \leq f(\boldsymbol{w})$.

Proof. First note that due to $l(0) = 1$ and the convexity of l we have $l(s) \geq 1 + cs$, where c equals to the derivative of l at 0. Then we have

$$f(\boldsymbol{w}) \geq \frac{1}{n^+}\overline{\mathrm{fn}}(\boldsymbol{w}, t) = \frac{1}{n^+}\sum_{i\in\mathcal{I}^+} l(t - \boldsymbol{w}^\top \boldsymbol{x}_i) \geq \frac{1}{n^+}\sum_{i\in\mathcal{I}^+}(1 + c(t - \boldsymbol{w}^\top \boldsymbol{x}_i)) \tag{23}$$

$$= 1 + \frac{c}{n^+}\sum_{i\in\mathcal{I}^+}(t - \boldsymbol{w}^\top \boldsymbol{x}_i) = 1 + ct - \frac{c}{n^+}\sum_{i\in\mathcal{I}^+}\boldsymbol{w}^\top \boldsymbol{x}_i \geq 1, \tag{24}$$

where the last inequality follows from (22). Now we realize that for any method from the statement, the corresponding threshold for $\boldsymbol{w} = 0$ equals to $t = 0$, and thus $f(\mathbf{0}) = 1$. But then $f(\mathbf{0}) \leq f(\boldsymbol{w})$, which finishes the proof. □

Theorem 3.4 (page 11) *Consider the Pat&Mat or Pat&Mat-NP method with the hinge surrogate and no regularization. Assume that for some $\boldsymbol{w}$ we have*

$$\frac{1}{n^+}\sum_{i\in\mathcal{I}^+}\boldsymbol{w}^\top \boldsymbol{x}_i > \frac{1}{n^-}\sum_{j\in\mathcal{I}^-}\boldsymbol{w}^\top \boldsymbol{x}_j. \tag{25}$$

Then there exists some $\beta > 0$ such that $f(\boldsymbol{w}) < f(\mathbf{0})$.

Proof. Define

$$s_{\min} = \min_{i\in\mathcal{I}}\boldsymbol{w}^\top \boldsymbol{x}_i, \qquad \bar{s} = \frac{1}{n}\sum_{i\in\mathcal{I}}\boldsymbol{w}^\top \boldsymbol{x}_i, \qquad s_{\max} = \max_{i\in\mathcal{I}}\boldsymbol{w}^\top \boldsymbol{x}.$$

Then we have the following chain of relations

$$\begin{aligned}\bar{s} &= \frac{1}{n}\sum_{i\in\mathcal{I}}\boldsymbol{w}^\top \boldsymbol{x}_i = \frac{1}{n}\sum_{i\in\mathcal{I}^+}\boldsymbol{w}^\top \boldsymbol{x}_i + \frac{1}{n}\sum_{i\in\mathcal{I}^-}\boldsymbol{w}^\top \boldsymbol{x}_i < \frac{1}{n}\sum_{i\in\mathcal{I}^+}\boldsymbol{w}^\top \boldsymbol{x}_i + \frac{n^-}{nn^+}\sum_{i\in\mathcal{I}^+}\boldsymbol{w}^\top \boldsymbol{x}_i \\ &= \frac{1}{n}\left(1 + \frac{n^-}{n^+}\right)\sum_{i\in\mathcal{I}^+}\boldsymbol{w}^\top \boldsymbol{x}_i = \frac{1}{n}\frac{n^+ + n^-}{n^+}\sum_{i\in\mathcal{I}^+}\boldsymbol{w}^\top \boldsymbol{x}_i = \frac{1}{n^+}\sum_{i\in\mathcal{I}^+}\boldsymbol{w}^\top \boldsymbol{x}_i.\end{aligned} \tag{26}$$

The only inequality follows from (25) and the last equality follows from $n^+ + n^- = n$.

Due to (25) we observe $s_{\min} < \bar{s} < s_{\max}$. Then we can define

$$\beta = \min\left\{\frac{\tau}{\bar{s} - s_{\min}}, \frac{1 - \tau}{s_{\max} - \bar{s}}\right\}, \qquad t = \frac{1 - \tau}{\beta} + \bar{s}.$$

We note that $\beta > 0$. At the same time we obtain

$$1 + \beta(\boldsymbol{w}^\top \boldsymbol{x} - t) \geq 1 + \beta(s_{\min} - t) = 1 + \beta s_{\min} - 1 + \tau - \beta\bar{s} = \beta(s_{\min} - \bar{s}) + \tau \geq 0. \tag{27}$$

Here, the first equality follows from the definition of t and the last inequality from the definition of β. Then we have

$$\begin{aligned}\frac{1}{n}\sum_{i\in\mathcal{I}} l(\boldsymbol{w}^\top \boldsymbol{x}_i - t) &= \frac{1}{n}\sum_{i\in\mathcal{I}}\max\{1 + \beta(\boldsymbol{w}^\top \boldsymbol{x}_i - t), 0\} = \frac{1}{n}\sum_{i\in\mathcal{I}}\Big(1 + \beta(\boldsymbol{w}^\top \boldsymbol{x}_i - t)\Big) \\ &= 1 - \beta t + \frac{\beta}{n}\sum_{i\in\mathcal{I}}\boldsymbol{w}^\top \boldsymbol{x}_i = 1 - \beta t + \beta\bar{s} = \tau,\end{aligned}$$

where the second equality employs (27), the third one the definition of $\bar{s}$ and the last one the definition of t. But this means that t is the threshold corresponding to $\boldsymbol{w}$.

Similarly to (27) we get

$$1 + t - \boldsymbol{w}^\top \boldsymbol{x} \geq 1 + t - s_{\max} = 1 + \frac{1-\tau}{\beta} + \bar{s} - s_{\max} \geq \frac{1-\tau}{\beta} + \bar{s} - s_{\max} \geq 0,$$

where the last inequality follows from the definition of β. Then for the objective we have

$$\begin{aligned} f(\boldsymbol{w}) &= \frac{1}{n^+} \sum_{i \in \mathcal{I}^+} l(t - \boldsymbol{w}^\top \boldsymbol{x}_i) = \frac{1}{n^+} \sum_{i \in \mathcal{I}^+} \max\{1 + t - \boldsymbol{w}^\top \boldsymbol{x}_i, 0\} = \\ &= \frac{1}{n^+} \sum_{i \in \mathcal{I}^+} \left(1 + t - \boldsymbol{w}^\top \boldsymbol{x}_i\right) = 1 + t - \frac{1}{n^+} \sum_{i \in \mathcal{I}^+} \boldsymbol{w}^\top \boldsymbol{x}_i < 1 + t - \bar{s} \qquad (28) \\ &= 1 + \frac{1-\tau}{\beta} + \bar{s} - \bar{s} = 1 + \frac{1-\tau}{\beta} = f(\boldsymbol{0}), \end{aligned}$$

where we used (28) and (26) and the results for $\boldsymbol{w} = \boldsymbol{0}$ from Section 3.1. Thus, we finished the proof for *Pat&Mat*. The proof for *Pat&Mat-NP* can be performed in a identical way by replacing in the definition of $\bar{s}$ the mean with respect to all samples by the mean with respect to all negative samples. □

A.4 Results related to threshold comparison

Lemma A.4. *Denote the scores $s^+ = \boldsymbol{w}^\top \boldsymbol{x}_i$ for $i \in \mathcal{I}^+$ and $s^- = \boldsymbol{w}^\top \boldsymbol{x}_j$ for $j \in \mathcal{I}^-$ and the ordered variants with decreasing components of $\boldsymbol{s}^-$ by $\boldsymbol{s}^-_{[\cdot]}$. Then we have the following implications*

$$\begin{aligned} s^-_{[1]} \geq \frac{1}{n^+} \sum_{i=1}^{n^+} s^+_i \implies f(\boldsymbol{0}) \leq f(\boldsymbol{w}) \text{ for TopPush,} \\ \frac{1}{K} \sum_{i=1}^{K} s^-_{[i]} \geq \frac{1}{n^+} \sum_{i=1}^{n^+} s^+_i \implies f(\boldsymbol{0}) \leq f(\boldsymbol{w}) \text{ for TopPushK,} \\ \frac{1}{n^-\tau} \sum_{i=1}^{n^-\tau} s^-_{[i]} \geq \frac{1}{n^+} \sum_{i=1}^{n^+} s^+_i \implies f(\boldsymbol{0}) \leq f(\boldsymbol{w}) \text{ for TopMean-NP.} \end{aligned}$$

Proof. Since the left-hand side in the equality is the threshold for the corresponding methods, the result follows immediately from Theorem 3.1. □

Lemma A.5. *The threshold for the Pat&Mat method is greater or equal than the threshold for the TopMean method. Similarly, the threshold for the Pat&Mat-NP method is greater or equal than the threshold for the TopMean-NP method.*

Proof. Define the scores $s = \boldsymbol{w}^\top \boldsymbol{x}$ and define $\mathcal{J}$ to be the set of $n\tau$ indices where the scores have the largest values. Due to $l(0) = 1$ and the convexity of l we have $l(s) \geq 1 + cs$, where c is the derivative of l at 0. From the non-negativity of l we have

$$n\tau = \sum_{i=1}^{n} l(s_i - t) \geq \sum_{i \in \mathcal{J}} l(s_i - t) \geq \sum_{i \in \mathcal{J}} (1 + c(s_i - t)) = n\tau - n\tau ct + c \sum_{i \in \mathcal{J}} s_i,$$

which implies

$$t \geq \frac{1}{n\tau} \sum_{i \in \mathcal{J}} s_i.$$

But this finishes the proof for the *Pat&Mat* method. The same result can be shown for the *Pat&Mat-NP* method by considering only indices corresponding to negative samples. □

Lemma A.6. *Define vector* $\boldsymbol{s}^+$ *with components* $s^+ = \boldsymbol{w}^\top \boldsymbol{x}_i$ *for* $i \in \mathcal{I}^+$ *and similarly define vector* $\boldsymbol{s}^-$ *with components* $s^- = \boldsymbol{w}^\top \boldsymbol{x}_j$ *for* $j \in \mathcal{I}^-$. *Denote by* $\boldsymbol{s}^+_{[\cdot]}$ *and* $\boldsymbol{s}^-_{[\cdot]}$ *the sorted versions of* $\boldsymbol{s}^+$ *and* $\boldsymbol{s}^-$, *respectively. Then we have the following statements:*

$$s^+_{[n^+\tau]} > s^-_{[n^-\tau]} \implies \textit{Grill has larger threshold than Grill-NP},$$

$$\frac{1}{n^+\tau} \sum_{i=1}^{n^+\tau} s^+_{[i]} > \frac{1}{n^-\tau} \sum_{i=1}^{n^-\tau} s^-_{[i]} \implies \textit{TopMean has larger threshold than TopMean-NP}.$$

Proof. Since $\boldsymbol{s}^+$ and $\boldsymbol{s}^-$ are computed on disjunctive indices, we have

$$s_{[n\tau]} \geq \min\{s^+_{[n^+\tau]}, s^-_{[n^-\tau]}\}.$$

Since $s_{[n\tau]}$ is the threshold for the *Grill* method and $s^-_{[n^-\tau]}$ is the threshold for the *Grill-NP* method, the first statement follows. The second part can be shown in a similar way. □

Since the goal of the presented methods is to push s^+ above s^-, we may expect that the conditions in Lemma A.6 hold true.

References

[1] S. Agarwal. The infinite push: A new support vector ranking algorithm that directly optimizes accuracy at the absolute top of the list. In *Proceedings of the 2011 SIAM International Conference on Data Mining*, pages 839–850. SIAM, 2011.

[2] E. B. Claise. Cisco Systems NetFlow services export version 9. 2004.

[3] P. Baldi, K. Cranmer, T. Faucett, P. Sadowski, and D. Whiteson. Parameterized neural networks for high-energy physics. *The European Physical Journal C*, 76(5):235, 2016.

[4] L. Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, pages 177–186. Springer, 2010.

[5] S. Boyd, C. Cortes, M. Mohri, and A. Radovanovic. Accuracy at the top. In *Advances in neural information processing systems*, pages 953–961, 2012.

[6] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[7] K.-W. Chang, C.-J. Hsieh, and C.-J. Lin. Coordinate descent method for large-scale l2-loss linear support vector machines. *Journal of Machine Learning Research*, 9(Jul):1369–1398, 2008.

[8] Cisco Systems. CTA Cisco cognitive threat analytics on Cisco cloud web security. http://www.cisco.com/c/en/us/solutions/enterprise-networks/cognitive-threat-analytics, 2014–2015.

[9] C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[10] J. Davis and M. Goadrich. The relationship between precision-recall and roc curves. In *Proceedings of the 23rd international conference on Machine learning*, pages 233–240. ACM, 2006.

[11] E. E. Eban, M. Schain, A. Mackey, A. Gordon, R. A. Saurous, and G. Elidan. Scalable learning of non-decomposable objectives. In *Artificial Intelligence and Statistics*, pages 832–840, 2017.

[12] Y. Freund, R. Iyer, R. E. Schapire, and Y. Singer. An efficient boosting algorithm for combining preferences. *The Journal of machine learning research*, 4:933–969, 2003.

[13] S. Garcia, M. Grill, J. Stiborek, and A. Zunino. An empirical comparison of botnet detection methods. *Computers & Security*, 45:100–123, 2014.

[14] M. Grill and T. Pevný. Learning combination of anomaly detectors for security domain. *Computer Networks*, 107:55–63, 2016.

[15] I. Guyon, S. Gunn, A. Ben-Hur, and G. Dror. Result analysis of the nips 2003 feature selection challenge. In *Advances in neural information processing systems*, pages 545–552, 2005.

[16] C.-J. Hsieh, K.-W. Chang, C.-J. Lin, S. S. Keerthi, and S. Sundararajan. A dual coordinate descent method for large-scale linear svm. In *Proceedings of the 25th international conference on Machine learning*, pages 408–415. ACM, 2008.

[17] T. Joachims. A support vector method for multivariate performance measures. In *Proceedings of the 22nd international conference on Machine learning*, ICML '05, pages 377–384, New York, NY, USA, 2005. ACM, ACM.

[18] P. Kar, H. Narasimhan, and P. Jain. Surrogate functions for maximizing precision at the top. In *International Conference on Machine Learning*, pages 189–198, 2015.

[19] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[20] M. Lapin, M. Hein, and B. Schiele. Top-k multiclass svm. In *Advances in Neural Information Processing Systems*, pages 325–333, 2015.

[21] N. Li, R. Jin, and Z.-H. Zhou. Top rank optimization in linear time. In *Advances in neural information processing systems*, NIPS'14, pages 1502–1510, Cambridge, MA, USA, 2014. MIT Press.

[22] A. Mackey, X. Luo, and E. Eban. Constrained classification and ranking via quantiles. *arXiv preprint arXiv:1803.00067*, 2018.

[23] J. Neyman and E. S. Pearson. Ix. on the problem of the most efficient tests of statistical hypotheses. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 231(694-706):289–337, 1933.

[24] K. Ogawa, Y. Suzuki, and I. Takeuchi. Safe screening of non-support vectors in pathwise svm computation. In *International conference on machine learning*, pages 1382–1390, 2013.

[25] M. Rehák, M. Pěchouček, M. Grill, J. Stiborek, K. Bartoš, and P. Čeleda. Adaptive multiagent system for network traffic monitoring. *IEEE Intelligent Systems*, (3):16–25, 2009.

[26] R. T. Rockafellar, S. Uryasev, et al. Optimization of conditional value-at-risk. *Journal of risk*, 2:21–42, 2000.

[27] C. Rudin. The p-norm push: A simple convex ranking algorithm that concentrates at the top of the list. *Journal of Machine Learning Research*, 10(Oct):2233–2271, 2009.

[28] B. Scholkopf and A. J. Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2001.

[29] V. G. Sigillito, S. P. Wing, L. V. Hutton, and K. B. Baker. Classification of radar returns from the ionosphere using neural networks. *Johns Hopkins APL Technical Digest*, 10(3):262–266, 1989.

[30] D. Tasche. A plug-in approach to maximising precision at the top and recall at the top. *arXiv preprint arXiv:1804.03077*, 2018.

[31] J. Wang, P. Wonka, and J. Ye. Scaling svm and least absolute deviations via exact data reduction. In *International conference on machine learning*, pages 523–531, 2014.