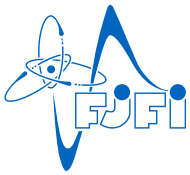
Czech Technical University in Prague
Faculty of Nuclear Sciences and
Physical Engineering

General Framework for Classicifcation at the Top

Dissertation



Author: Ing. Václav Mácha
Academic year: 2021/2022

Poděkování:

Thanks thanks

Čestné prohlášení:

Prohlašuji na tomto místě, že jsem předloženou práci vypracoval samostatně, a že jsem uvedl veškerou použitou literaturu.

V Praze dne 1. prosince 2021

.....
Ing. Václav Mácha

Název: **Title title title title title title**

Autor: Ing. Václav Mácha

Obor: Matematické inženýrství

Druh práce: Disertační práce

Školitel: doc. Ing Václav Šmídl, Ph.D.

Školitel specialista: Mgr. Lukáš Adam, Ph.D.

Abstrakt: Abstract abstract

Klíčová slova: Keywords keywords keywords keywords keywords keywords keywords keywords keywords keywords keywords keywords keywords

Title: **Title title title title title title**

Abstract: Abstract abstract

Keywords: Keywords keywords keywords keywords keywords keywords keywords keywords keywords keywords keywords keywords keywords

Contents

1 How to

Theorem 1.1: Goldbach's conjecture

Every even integer greater than 2 can be expressed as the sum of two primes.

We recall 1.1:

Theorem 1.1: Goldbach's conjecture

Every even integer greater than 2 can be expressed as the sum of two primes.

2 Introduction

Many binary classification problems focus on separating the dataset by a linear hyperplane $\boldsymbol{w}^\top \boldsymbol{x} - t$. A sample $\boldsymbol{x}$ is deemed to be positive or relevant (depending on the application) if its score $\boldsymbol{w}^\top \boldsymbol{x}$ is above a threshold t. Multiple problem categories belong to this framework:

- *Ranking problems* 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* 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* 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.

Examples of this type can be found in search engines, where the user is interested only in the first few queries. These queries need to be of high quality. Other examples include cybersecurity [1], where a low false-negative rate is crucial as a high number of false alarms would result in the software being uninstalled, or drug development, where potentially useful drugs need to be preselected and manually investigated. All these three applications may be written (possibly after a reformulation) in a similar form as a minimization of the false-negatives (misclassified positives) above 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 this paper is to provide a unified framework for these three applications and perform its theoretical and numerical analysis.

The goal of the ranking problems is to rank the relevant samples higher than the non-relevant ones. A prototypical example is the RankBoost [2] maximizing the area under the ROC curve, the Infinite Push [3] or the p-norm push [4] 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 [5], 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 with the threshold equal to the largest score computed from negative samples.

Accuracy at the Top (τ-quantile) was formally defined in [6] 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, [7] 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. [6] 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. [8] derived upper approximations, their error bounds and solved these approximations. [1] proposed the projected gradient descent method where after each gradient step, the quantile is recomputed. [9] suggested new formulations for various criteria and argued that they keep desired properties such as convexity. [10] showed that accuracy at the top is maximized by thresholding the posterior probability of the relevant class. The closest approach to our framework is [11, 12], where the authors considered multi-class classification problems, and their goal was to optimize the performance on the top few classes and [13], where the authors implicitly removed some variables and derived an efficient algorithm.

3 Linear Classification at the Top

3.1 Framework for Minimizing Missclassification Above a Threshold

Many important binary classification problems minimize the number of misclassified samples below (or above) certain threshold. Since these problems are usually considered separately, in this section, we provide a unified framework for their handling and present several classification problems falling into this framework.

For samples $\boldsymbol{x}$, we consider the linear classifier $f(\boldsymbol{w}) = \boldsymbol{w}^\top \boldsymbol{x} - t$, where $\boldsymbol{w}$ is the normal vector to the separating hyperplane and t is a threshold. The most well-known example is the support vector machines, where t is an optimization variable. In many cases the threshold t is computed from the scores $s = \boldsymbol{w}^\top \boldsymbol{x}$. For example, *TopPush* from [5] sets the threshold t to the largest score s^- corresponding to negative samples and [1] sets it to the quantile of all scores.

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

$$\begin{aligned} \mathrm{tp}(\boldsymbol{w}, t) &= \sum_{\boldsymbol{x} \in \mathcal{X}^+} \left[\boldsymbol{w}^\top \boldsymbol{x} - t \geq 0\right], \quad \mathrm{fn}(\boldsymbol{w}, t) = \sum_{\boldsymbol{x} \in \mathcal{X}^+} \left[\boldsymbol{w}^\top \boldsymbol{x} - t < 0\right], \\ \mathrm{tn}(\boldsymbol{w}, t) &= \sum_{\boldsymbol{x} \in \mathcal{X}^-} \left[\boldsymbol{w}^\top \boldsymbol{x} - t < 0\right], \quad \mathrm{fp}(\boldsymbol{w}, t) = \sum_{\boldsymbol{x} \in \mathcal{X}^-} \left[\boldsymbol{w}^\top \boldsymbol{x} - t \geq 0\right]. \end{aligned} \tag{3.1}$$

Here $[\cdot]$ is the 0-1 loss (Iverson bracket, characteristic function) which is equal to 1 if the argument is true and to 0 otherwise. Moreover, $\mathcal{X}/\mathcal{X}^+/\mathcal{X}^-$ denotes the sets of all/positive/negative samples and by $n/n^+/n^-$ their respective sizes.

Since the misclassified samples below the threshold are the false-negatives, we arrive at the following problem

$$\begin{aligned} &\text{minimize} && \frac{1}{n^+}\mathrm{fn}(\boldsymbol{w}, t) \\ &\text{subject to} && \text{threshold } t \text{ is a function of } \{\boldsymbol{w}^\top \boldsymbol{x}_i\}_{i=1}^n. \end{aligned} \tag{3.2}$$

As the 0-1 loss in (3.1) is discontinuous, problem (3.2) is difficult to handle. The usual approach is to employ a surrogate function such as the hinge loss function defined by

$$l_{\mathrm{hinge}}(s) = \max\{0, 1 + s\}. \tag{3.3}$$

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 (3.1) may be approximated by

their surrogate counterparts

$$\begin{aligned} \overline{\mathrm{tp}}(\boldsymbol{w},t) &= \sum_{\boldsymbol{x}\in\mathcal{X}^+} l(\boldsymbol{w}^\top\boldsymbol{x}-t), \quad \overline{\mathrm{fn}}(\boldsymbol{w},t) = \sum_{\boldsymbol{x}\in\mathcal{X}^+} l(t-\boldsymbol{w}^\top\boldsymbol{x}), \\ \overline{\mathrm{tn}}(\boldsymbol{w},t) &= \sum_{\boldsymbol{x}\in\mathcal{X}^-} l(t-\boldsymbol{w}^\top\boldsymbol{x}), \quad \overline{\mathrm{fp}}(\boldsymbol{w},t) = \sum_{\boldsymbol{x}\in\mathcal{X}^-} l(\boldsymbol{w}^\top\boldsymbol{x}-t). \end{aligned} \tag{3.4}$$

Since $l(\cdot) \geq [\cdot]$, the surrogate counts (3.4) provide upper approximations of the true counts (3.1). Replacing the counts in (3.2) by their surrogate counterparts and adding a regularization results in

$$\begin{aligned} \text{minimize} \quad & \frac{1}{n^+}\overline{\mathrm{fn}}(\boldsymbol{w},t) + \frac{\lambda}{2}\|\boldsymbol{w}\|^2 \\ \text{subject to} \quad & \text{threshold } t \text{ is a function of } \{\boldsymbol{w}^\top\boldsymbol{x}_i\}_{i=1}^n. \end{aligned} \tag{3.5}$$

In the rest of this section, we list formulations which fall into the framework of (3.2) and (3.5).

3.1.1 Methods based on pushing positives to the top

The first category of formulations falling into our framework (3.2) and (3.5) are ranking methods which attempt to put as many positives (relevant samples) to the top as possible. Specifically, for each sample $\boldsymbol{x}$, they compute the score $s = \boldsymbol{w}^\top\boldsymbol{x}$ and then sort the vector $\boldsymbol{s}$ into $\boldsymbol{s}_{[\cdot]}$ with decreasing components $s_{[1]} \geq s_{[2]} \geq \cdots \geq s_{[n]}$. The number of positives on top equals to the number of positives above the highest negative. This amounts to maximizing true-positives or, equivalently, minimizing false-negatives, which may be written as

$$\begin{aligned} \text{minimize} \quad & \frac{1}{n^+}\mathrm{fn}(\boldsymbol{w},t) \\ \text{subject to} \quad & t = s^-_{[1]}, \\ & \text{components of } \boldsymbol{s}^- \text{ equal to } s^- = \boldsymbol{w}^\top\boldsymbol{x}^- \text{ for } \boldsymbol{x}^- \in \mathcal{X}^-. \end{aligned} \tag{3.6}$$

As t is a function of the scores $s = \boldsymbol{w}^\top\boldsymbol{x}$, problem (3.6) is a special case of (3.2).

TopPush from [5] replaces the false-negatives in (3.6) by their surrogate and adds a regularization term to arrive at

$$\begin{aligned} \text{minimize} \quad & \frac{1}{n^+}\overline{\mathrm{fn}}(\boldsymbol{w},t) + \frac{\lambda}{2}\|\boldsymbol{w}\|^2 \\ \text{subject to} \quad & t = s^-_{[1]}, \\ & \text{components of } \boldsymbol{s}^- \text{ equal to } s^- = \boldsymbol{w}^\top\boldsymbol{x}^- \text{ for } \boldsymbol{x}^- \in \mathcal{X}^-. \end{aligned} \tag{3.7}$$

Note that this falls into the framework of (3.5).

As we will show in Section 3.2.4, *TopPush* is sensitive to outliers and mislabelled data. To robustify it, we follow the idea from [11] and propose to replace the largest negative score by the mean of k largest negative scores. This results in

$$\begin{aligned} \text{minimize} \quad & \frac{1}{n^+}\overline{\mathrm{fn}}(\boldsymbol{w},t) + \frac{\lambda}{2}\|\boldsymbol{w}\|^2 \\ \text{subject to} \quad & t = \frac{1}{k}(s^-_{[1]} + \cdots + s^-_{[k]}), \\ & \text{components of } \boldsymbol{s}^- \text{ equal to } s^- = \boldsymbol{w}^\top\boldsymbol{x}^- \text{ for } \boldsymbol{x}^- \in \mathcal{X}^-. \end{aligned} \tag{3.8}$$

We used the mean of highest k negative scores instead of the value of the k-th negative score to preserve convexity as shown in Section 3.2.2.

3.1.2 Accuracy at the Top

The previous category considers formulations which minimize the false-negatives below the highest-ranked negative. Accuracy at the Top [6] takes a different approach and minimizes false-positives above the top τ-quantile defined by

$$t_1(\boldsymbol{w}) = \max\{t \mid \mathrm{tp}(\boldsymbol{w}, t) + \mathrm{fp}(\boldsymbol{w}, t) \geq n\tau\}. \tag{3.9}$$

Then the Accuracy at the Top problem is defined by

$$\begin{aligned} \text{minimize} \quad & \frac{1}{n^-}\mathrm{fp}(\boldsymbol{w}, t) \\ \text{subject to} \quad & t \text{ is the top } \tau\text{-quantile: it solves (3.9).} \end{aligned} \tag{3.10}$$

Due to Lemma **??** in the Appendix, the previous problem (3.10) is equivalent (up to a small theoretical issue) to

$$\begin{aligned} \text{minimize} \quad & \mu\mathrm{fn}(\boldsymbol{w}, t) + (1-\mu)\mathrm{fp}(\boldsymbol{w}, t) + \frac{\lambda}{2}\|\boldsymbol{w}\|^2 \\ \text{subject to} \quad & t \text{ is the top } \tau\text{-quantile: it solves (3.9)} \end{aligned} \tag{3.11}$$

for any $\mu \in [0, 1]$. This problem with $\mu = 0$ equals to (3.10), with $\mu = 1$ it falls into our framework (3.2), while with $\mu = \frac{n^-}{n}$ it corresponds to the original definition from [6].

Apart from the quantile (3.9), there are two other possible choices of the threshold

$$t_2(\boldsymbol{w}) = \frac{1}{n\tau}\sum_{i=1}^{n\tau} s_{[i]}, \tag{3.12}$$

$$t_3(\boldsymbol{w}) \quad \text{solves} \quad \frac{1}{n}\sum_{i=1}^{n} l(\beta(s_i - t)) = \tau. \tag{3.13}$$

We again use the vector of scores $\boldsymbol{s}$ with components $s_i = \boldsymbol{w}^\top \boldsymbol{x}_i$ and for the rest of the paper we assume, for simplicity, that $n\tau$ is an integer. The quantile (3.9) is sometimes denoted as VaR (value at risk) and (3.12) as CVaR (conditional value of risk). It is known is that the latter is the tightest convex approximation of the former. We will sometimes denote (3.13) as surrogate top τ-quantile. We will investigate the relations between these three objects as well as their properties such as convexity, differentiability or stability in Section 3.2.

Paper [1] builds on the Accuracy at the Top problem (3.11), where it replaces $\mathrm{fn}(\boldsymbol{w}, t)$ and $\mathrm{fp}(\boldsymbol{w}, t)$ in the objective by their surrogate counterparts $\overline{\mathrm{fn}}(\boldsymbol{w}, t)$ and $\overline{\mathrm{fp}}(\boldsymbol{w}, t)$. This leads to

$$\begin{aligned} \text{minimize} \quad & \frac{1}{n^+}\overline{\mathrm{fn}}(\boldsymbol{w}, t) + \frac{1}{n^-}\overline{\mathrm{fp}}(\boldsymbol{w}, t) + \frac{\lambda}{2}\|\boldsymbol{w}\|^2 \\ \text{subject to} \quad & t \text{ is the top } \tau\text{-quantile: it solves (3.9).} \end{aligned} \tag{3.14}$$

Based on the first author, we name this formulation *Grill*. The main purpose of (3.12) is to provide a convex approximation of the non-convex quantile (3.9). Putting it into the constraint results in a convex approximation problem, which we call *TopMeanK*

$$\begin{aligned} \text{minimize} \quad & \frac{1}{n^+}\overline{\mathrm{fn}}(\boldsymbol{w}, t) + \frac{\lambda}{2}\|\boldsymbol{w}\|^2 \\ \text{subject to} \quad & t = \frac{1}{n\tau}(s_{[1]} + \cdots + s_{[n\tau]}), \\ & \text{components of } \boldsymbol{s} \text{ equal to } s = \boldsymbol{w}^\top \boldsymbol{x} \text{ for } \boldsymbol{x} \in \mathcal{X}. \end{aligned} \tag{3.15}$$

Similarly, we can use the surrogate top quantile in the constraint to arrive at

$$\begin{aligned} \text{minimize} \quad & \frac{1}{n^+}\overline{\text{fn}}(\boldsymbol{w}, t) + \frac{\lambda}{2}\|\boldsymbol{w}\|^2 \\ \text{subject to} \quad & t \text{ is the surrogate top } \tau\text{-quantile: it solves (3.13).} \end{aligned} \tag{3.16}$$

Note that *Grill* minimizes the convex combination of false-positives and false-negatives while (3.15) and (3.16) minimize only the false-negatives. The reason for this will be evident in Section 3.2.2 and amounts to preservation of convexity. Moreover, as will see later, problem (3.16) provides a good approximation to the Accuracy at the Top problem, it is easily solvable due to convexity and requires almost no tuning, we named it *Pat&Mat* (Precision At the Top & Mostly Automated Tuning).

3.1.3 Methods optimizing the Neyman-Pearson criterion

Another category falling into the framework of (3.2) and (3.5) is the Neyman-Pearson problem which is closely related to hypothesis testing, where null H_0 and alternative H_1 hypotheses are given. Type I error occurs when H_0 is true but is rejected, and type II error happens when H_0 is false, but it fails to be rejected. The standard technique is to minimize Type II error while a bound for Type I error is given.

In the Neyman-Pearson problem, the null hypothesis H_0 states that a sample $\boldsymbol{x}$ has the negative label. Then Type I error corresponds to false-positives while Type II error to false-negatives. If the bound on Type I error equals τ, we may write this as

$$t_1^{\text{NP}}(\boldsymbol{w}) = \max\Big\{t \mid \text{fp}(\boldsymbol{w}, t) \geq n^-\tau\Big\}. \tag{3.17}$$

Then, we may write the Neyman-Pearson problem as

$$\begin{aligned} \text{minimize} \quad & \frac{1}{n^+}\text{fn}(\boldsymbol{w}, t) \\ \text{subject to} \quad & t \text{ is Type I error at level } \tau\text{: it solves (3.17).} \end{aligned} \tag{3.18}$$

Since (3.18) differs from (3.11) only by counting only the false-positives in (3.17) instead of counting all positives in (3.9), we can derive its approximations in exactly the same way as in Section 3.1.2. We therefore provide only their brief description and start with approximations of (3.17)

$$t_2^{\text{NP}}(\boldsymbol{w}) = \frac{1}{n^-\tau}\sum_{i=1}^{n^-\tau} s^-_{[i]}, \tag{3.19}$$

$$t_3^{\text{NP}}(\boldsymbol{w}) \quad \text{solves} \quad \frac{1}{n}\sum_{i=1}^{n^-} l(\beta(s_i^- - t)) = \tau. \tag{3.20}$$

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

$$\begin{aligned} \text{minimize} \quad & \frac{1}{n^+}\overline{\text{fn}}(\boldsymbol{w}, t) + \frac{1}{n^-}\overline{\text{fp}}(\boldsymbol{w}, t) + \frac{\lambda}{2}\|\boldsymbol{w}\|^2 \\ \text{subject to} \quad & t \text{ is the Neyman-Pearson threshold: it solves (3.17).} \end{aligned} \tag{3.21}$$

Similarly, the Neyman-Pearson alternative to *TopMeanK* reads

$$\begin{aligned} \text{minimize} \quad & \frac{1}{n^+}\overline{\text{fn}}(\boldsymbol{w}, t) + \frac{\lambda}{2}\|\boldsymbol{w}\|^2 \\ \text{subject to} \quad & t = \frac{1}{n^-\tau}(s^-_{[1]} + \cdots + s^-_{[n^-\tau]}), \\ & \text{components of } \boldsymbol{s}^- \text{ equal to } s^- = \boldsymbol{w}^\top \boldsymbol{x}^- \text{ for } \boldsymbol{x}^- \in \mathcal{X}. \end{aligned} \tag{3.22}$$

This problem already appeared in [14] under the name τ-*FPL*. Finally, *Pat&Mat-NP* reads

$$\begin{aligned} \text{minimize} \quad & \frac{1}{n^+}\overline{\text{fn}}(\boldsymbol{w}, t) + \frac{\lambda}{2}\|\boldsymbol{w}\|^2 \\ \text{subject to} \quad & t \text{ is the surrogate Neyman-Pearson threshold: it solves (3.20).} \end{aligned} \tag{3.23}$$

We may see (3.22) from two different viewpoints. First, τ-*FPL* provide convex approximations of *Grill-NP*. Second, τ-*FPL* has the same form as *TopPushK*. The only difference is that for τ-*FPL* 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 problems.

3.2 Theoretical Analysis of the Framework

In this section, we provide a theoretical analysis of the unified framework from Section 3.1. We consider purely the problem *formulations* and not individual *algorithms* which specify how to solve these formulations. We focus mainly on the following desirable properties:

- *Convexity* implies a guaranteed convergence for many optimization algorithms or their better convergence rates [15].
- *Differentiability* increases the speed of convergence.
- *Stability* is a general term, by which we mean that the global minimum is not at $\boldsymbol{w} = \boldsymbol{0}$. This actually happens for many formulations from Section 3.1 and results in the situation where the separating hyperplane is degenerate and does not actually exist.

For a nicer flow of text, we show the results only for formulations from Section 3.1.2. The results for methods from Section 3.1.3 are identical. For the same reason, we postpone the proofs to Appendix **??**.

3.2.1 Threshold value comparison

We start with the following proposition, which compares the threshold approximation quality.

Proposition 3.1: [14]

We always have

$$t_1(\boldsymbol{w}) \leq t_2(\boldsymbol{w}) \leq t_3(\boldsymbol{w}).$$

Whenever the objective contains only false-negatives, a lower threshold t means a lower objective function. Therefore, a lower threshold is preferred.

3.2.2 Convexity

Convexity is one of the most important properties in numerical optimization. It ensures that the optimization problem has neither stationary points nor local minima. All points of interest are global minima. Moreover, it allows for faster convergence rates. We present the following two results.

Proposition 3.2

Thresholds t_2 and t_3 are convex functions of the weights $\boldsymbol{w}$. The threshold function t_1 is non-convex.

Theorem 3.3

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.

While the proof of Theorem 3.3 is simple, it points to the necessity of considering only false-negatives in the objective of the problems in Section 3.1. In such a case, *TopPush*, *TopPushK*, *TopMeanK*, τ*-FPL*, *Pat&Mat* and *Pat&Mat-NP* are convex problems. At the same time, *Grill* and *Grill-NP* are not convex problems.

3.2.3 Differentiability

Similarly to convexity, differentiability allows for faster convergence rate and in some algorithms, better termination criteria. The next theorem shows which formulations are differentiable.

Theorem 3.4

If the surrogate function l is differentiable, then threshold t_3 is a differentiable function of the weights $\boldsymbol{w}$ and its derivative equals to

$$\nabla t_3(\boldsymbol{w}) = \frac{\sum_{\boldsymbol{x}\in\mathcal{X}} l'(\beta(\boldsymbol{w}^\top \boldsymbol{x} - t_3(\boldsymbol{w})))\boldsymbol{x}}{\sum_{\boldsymbol{x}\in\mathcal{X}} l'(\beta(\boldsymbol{w}^\top \boldsymbol{x} - t_3(\boldsymbol{w})))}.$$

The threshold functions t_1 and t_2 are non-differentiable.

This theorem shows that the objective functions of *Pat&Mat* and *Pat&Mat-NP* are differentiable. This allows us to prove the convergence of the stochastic gradient descent for these two formulations in Section 3.3.

3.2.4 Stability

We first provide a simple example and show that many formulations from the previous section are degenerate for it. Then we analyze general conditions under which this degenerate behaviour happens.

Example of a Degenerate Behavior

We consider 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 Figure 3.1 (left). We consider the hinge loss and no regularization. If n is large, the point at $(2, 0)$ is an outlier and the dataset is separable and the separating hyperplane has the normal vector $\boldsymbol{w} = (1, 0)$.

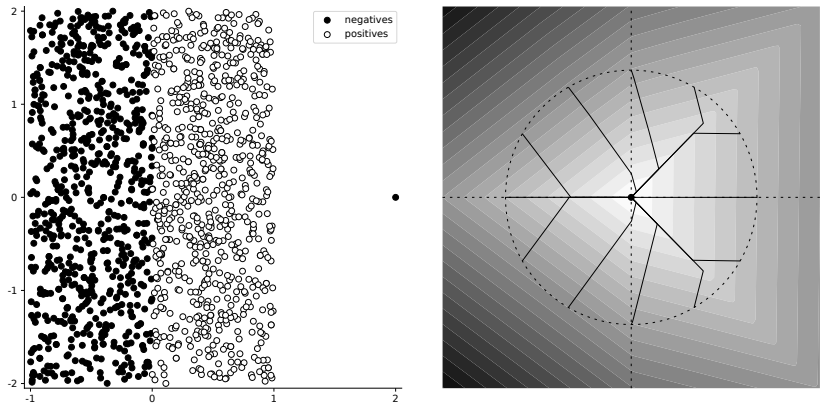


Figure 3.1: Left: distribution of positive (empty circle) and negative samples (full circles) for the example from Section 3.2.4. Right: contour plot for *TopPush* and its convergence to the zero vector from 12 initial points.

Table 3.1 shows the threshold t and the objective value f for two points $\boldsymbol{w}_1 = (0, 0)$ and $\boldsymbol{w}_2 = (1, 0)$. These two points are both important: $\boldsymbol{w}_1$ does not generate any separating hyperplane, while $\boldsymbol{w}_2$ generates the optimal separating hyperplane. We show the precise computation in Appendix **??**. Since the dataset is perfectly separable by $\boldsymbol{w}_2$, we expect that $\boldsymbol{w}_2$ provides a lower objective than $\boldsymbol{w}_1$. By shading the better objective in Table 3.1 by grey, we see that this did not happen for *TopPush* and *TopMeanK*.

It can be shown that $\boldsymbol{w}_1 = (0, 0)$ is even the global minimum for *TopPush* and *TopMeanK*. 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 Figure 3.1 (right). Here, we run *TopPush* 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.

Stability and Global minimum at zero

The convexity derived in the previous section guarantees that there are no local minima. However, as we showed in the example above, the global minimum may be at $\boldsymbol{w} = \boldsymbol{0}$. This is highly undesirable 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. The first result states that if the threshold $t(\boldsymbol{w})$ is above a certain value, then

Table 3.1: Comparison of formulations on the very simple problem from Section 3.2.4. Two formulations have the global minimum (denoted by grey color) at $\boldsymbol{w}_1 = (0, 0)$ which does not generate any separating hyperplane. The optimal separating hyperplane is generated by $\boldsymbol{w}_2 = (1, 0)$.

		$\boldsymbol{w}_1 = (0,0)$		$\boldsymbol{w}_2 = (1,0)$	
Name	Label	t	f	t	f
TopPush	(3.7)	0	1	2	2.5
TopPushK	(3.8)	0	1	$\frac{2}{k}$	$0.5 + \frac{2}{k}$
Grill	(3.14)	0	2	$1 - 2\tau$	$1.5 + 2\tau(1 - \tau)$
TopMeanK	(3.15)	0	1	$1 - \tau$	$1.5 - \tau$
Pat&Mat	(3.16)	$\frac{1}{\beta}(1 - \tau)$	$1 + \frac{1}{\beta}(1 - \tau)$	$\frac{1}{\beta}(1 - \tau)$	$0.5 + \frac{1}{\beta}(1 - \tau)$

zero has a better objective that $\boldsymbol{w}$. If this happens for all $\boldsymbol{w}$, then zero is the global minimum.

Theorem 3.5

Consider any of these formulations: *TopPush*, *TopPushK*, *TopMeanK* or τ-*FPL*. Fix any $\boldsymbol{w}$ and denote the corresponding threshold $t(\boldsymbol{w})$. If we have

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

then $f(\mathbf{0}) \leq f(\boldsymbol{w})$. Specifically, denote the scores $z^+ = \boldsymbol{w}^\top \boldsymbol{x}^+$ for $\boldsymbol{x}^+ \in \mathcal{X}^+$ and $z^- = \boldsymbol{w}^\top \boldsymbol{x}^-$ for $\boldsymbol{x}^- \in \mathcal{X}^-$ and the ordered variants with decreasing components of $\boldsymbol{s}^-$ by $\boldsymbol{s}^-_{[\cdot]}$. Then

$$\begin{aligned}
s^-_{[1]} \geq \frac{1}{n^+} \sum_{i=1}^{n^+} s^+_i \quad &\Longrightarrow \quad f(\mathbf{0}) \leq f(\boldsymbol{w}) \text{ for } \textit{TopPush}, \\
\frac{1}{k} \sum_{i=1}^{k} s^-_{[i]} \geq \frac{1}{n^+} \sum_{i=1}^{n^+} s^+_i \quad &\Longrightarrow \quad f(\mathbf{0}) \leq f(\boldsymbol{w}) \text{ for } \textit{TopPushK}, \\
\frac{1}{n^-\tau} \sum_{i=1}^{n^-\tau} s^-_{[i]} \geq \frac{1}{n^+} \sum_{i=1}^{n^+} s^+_i \quad &\Longrightarrow \quad f(\mathbf{0}) \leq f(\boldsymbol{w}) \text{ for } \tau\textit{-FPL}.
\end{aligned} \tag{3.24}$$

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

Corollary 3.6

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

Corollary 3.7

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

The proof of Theorem 3.5 employs the fact that all formulations in the theorem statement have only false-negatives in the objective. If $\boldsymbol{w}_0 = \mathbf{0}$, then $\boldsymbol{w}_0^\top \boldsymbol{x} = 0$ for all samples $\boldsymbol{x}$, the threshold equals to $t = 0$ and the objective equals to one. If the threshold is large for $\boldsymbol{w}$, many positives are below the threshold, and the false-negatives have the average surrogate value larger than one. In such a case, $\boldsymbol{w}_0 = \mathbf{0}$ becomes the global minimum. There are two fixes to this situation:

- Include false-positives to the objective. This approach is taken by *Grill* and *Grill-NP* and necessarily results in the loss of convexity.
- Move the threshold away from zero even when all scores $\boldsymbol{w}^\top \boldsymbol{x}$ are zero. This approach is taken by our formulations *Pat&Mat* and *Pat&Mat-NP* and keeps convexity.

The next theorem shows the advantage of the second approach.

Theorem 3.8

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

$$\frac{1}{n^+} \sum_{\boldsymbol{x}^+ \in \mathcal{X}^+} \boldsymbol{w}^\top \boldsymbol{x}^+ > \frac{1}{n^-} \sum_{\boldsymbol{x}^- \in \mathcal{X}^-} \boldsymbol{w}^\top \boldsymbol{x}^-. \tag{3.25}$$

Then there is a scaling parameter β_0 from (3.13) such that $f(\boldsymbol{w}) < f(\mathbf{0})$ for all $\beta \in (0, \beta_0)$.

These theorem shed some light on the behaviour of the formulations. Theorem 3.5 states that the stability of τ-*FPL* requires

$$\frac{1}{n^- \tau} \sum_{i=1}^{n^- \tau} z_{[i]}^- < \frac{1}{n^+} \sum_{i=1}^{n^+} z_i^+, \tag{3.26}$$

while Theorem 3.8 states that the stability of *Pat&Mat-NP* is ensured by

$$\frac{1}{n^-} \sum_{i=1}^{n^-} z_{[i]}^- < \frac{1}{n^+} \sum_{i=1}^{n^+} z_i^+. \tag{3.27}$$

The right-hand sides of (3.26) and (3.27) are the same, while the left-hand side of (3.27) is always smaller than the left-hand side of (3.26). This implies that if τ-*FPL* is stable, then *Pat&Mat-NP* is stable as well.

At the same time, there may be a huge difference in the stability of both formulations. Since the scores of positive samples should be above the scores of negative samples, the scores z may be interpreted as performance. Then formula (3.26) states that if the mean performance of a *small number of the best* negative samples is larger than the average

performance of *all* positive samples, then τ-*FPL* fails. On the other hand, formula (3.27) states that if the average performance of *all* positive samples is better than the average performance of *all* negative samples, then *Pat&Mat-NP* is stable. The former may well happen as accuracy at the top is interested in a good performance of only a small number of positive samples.

3.2.5 Method comparison

We provide a summary of the obtained results in Table 3.2. There we give basic characterizations of the formulations such as their definition label, their source, the hyperparameters, whether the formulation is differentiable and convex, and whether it has stability problems with $\boldsymbol{w} = \mathbf{0}$ being the global minimum.

Table 3.2: Summary of the formulations from Section 3.1. The table shows their definition label, the source or the source they are based on, the hyperparameters, whether the formulation is differentiable, convex and stable (in the sense of having problems with $\boldsymbol{w} = \mathbf{0}$).

Name	Source	Definition	Hyperpars	Convex	Differentiable	Stable
TopPush	[5]	(3.7)	λ	✓	✗	✗
TopPushK	ours	(3.8)	λ, k	✓	✗	✗
Grill	[1]	(3.14)	λ	✗	✗	✓
Pat&Mat	ours	(3.16)	β, λ	✓	✓	✓
TopMeanK	-	(3.15)	λ	✓	✗	✗
Grill-NP	-	(3.21)	λ	✗	✗	✓
Pat&Mat-NP	ours	(3.23)	β, λ	✓	✓	✓
τ-*FPL*	[14]	(3.22)	λ	✓	✗	✗

A similar comparison is performed in Figure 3.2. Methods in green and grey are convex, while formulations in white are non-convex. Based on Theorem 3.5, four formulations in grey are vulnerable to have the global minimum at $\boldsymbol{w} = 0$. This theorem states that the higher the threshold, the more vulnerable the formulation is. The full arrows depict this dependence. If it points from one formulation 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 usually but not always, the precise formulation is provided in Appendix **??**. This complies with Corollaries 3.6 and 3.7 which state that *TopPush* and *TopMeanK* are most vulnerable. At the same time, it says that τ-*FPL* is the best one from the grey-coloured formulations. Finally, even though *Pat&Mat-NP* has a worse approximation of the true threshold than τ-*FPL* due to Theorem 3.5, it is more stable due to the discussion after Theorem 3.8.

3.3 Convergence of stochastic gradient descent

The previous section analyzed the formulations from Section 3.1 but did not consider any optimization algorithms. In this section, we show a basic version of the stochastic gradient descent and then show its convergent version. Since due to considering the threshold, gradient computed on a minibatch is a biased estimate of the true gradient, we need to use variance reduction techniques, and the proof is rather complex.

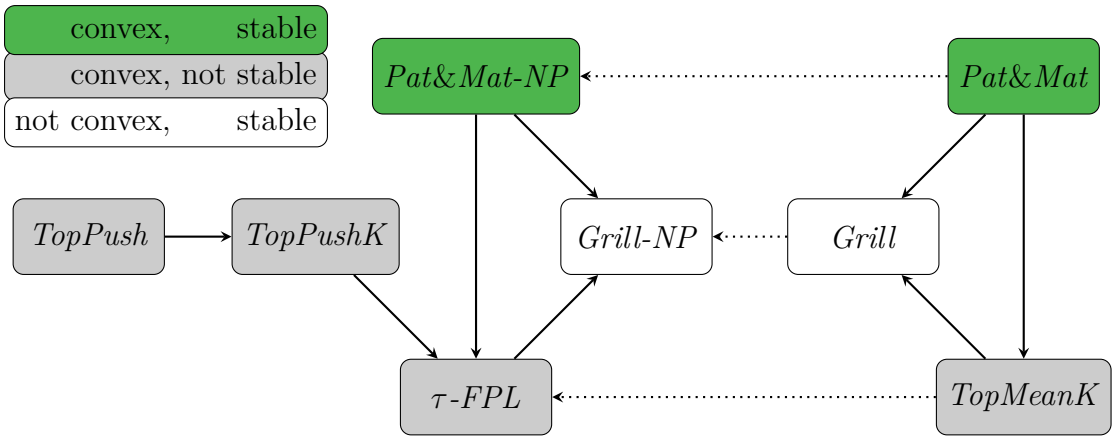


Figure 3.2: Summary of the formulations from Section 3.1. Methods in green and grey are convex, while formulations in white are non-convex. Methods in grey are vulnerable to have the global minimum at $\boldsymbol{w} = 0$. Full (dotted) arrow pointing from one formulation to another show that the latter formulation has always (usually) smaller threshold.

3.3.1 Stochastic gradient descent: Basic

Many optimization algorithms for solving the formulations from Section 3.1 use primal-dual or purely dual formulations. [9] introduced dual variables and used alternating optimization to the resulting min-max problem. [5] and [14] dualized the problem and solved it with the steepest gradient ascent. [16] followed the same path but added kernels to handle non-linearity. We follow the ideas of [13] and [17] and solve the problems directly in their primal formulations. Therefore, even though we use the same formulation for *TopPush* as [5] or for τ-*FPL* as [14], our solution process is different. However, due to convexity, both algorithms should converge to the same point.

The decision variables in (3.5) 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 [1] is to compute the gradient only with respect to $\boldsymbol{w}$ and then recompute t. A more sophisticated way is based on the chain rule. 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

$$\begin{aligned} f(\boldsymbol{w}) &= \frac{1}{n^+} \sum_{\boldsymbol{x} \in \mathcal{X}^+} l(t(\boldsymbol{w}) - \boldsymbol{w}^\top \boldsymbol{x}) + \frac{\lambda}{2} \|\boldsymbol{w}\|^2, \\ \nabla f(\boldsymbol{w}) &= \frac{1}{n^+} \sum_{\boldsymbol{x} \in \mathcal{X}^+} l'(t(\boldsymbol{w}) - \boldsymbol{w}^\top \boldsymbol{x})(\nabla t(\boldsymbol{w}) - \boldsymbol{x}) + \lambda \boldsymbol{w}. \end{aligned} \tag{3.28}$$

The only remaining part is the computation of $\nabla t(\boldsymbol{w})$. It is simple for $\nabla t_1(\boldsymbol{w})$ and $\nabla t_2(\boldsymbol{w})$ and Theorem 3.4 shows the computation for $\nabla t_3(\boldsymbol{w})$. Appendix **??** provides an efficient computation method for $t_3(\boldsymbol{w})$.

Having derivative (3.28), deriving the stochastic gradient is simple. It partitions the dataset into minibatches and provides an update of the weights $\boldsymbol{w}$ based only on a mini-batch, namely by replacing the mean over the whole dataset in (3.28) by a mean over the minibatch.

3.3.2 Stochastic gradient descent: Convergent for *Pat&Mat* and *Pat&Mat-NP*

For the convergence proof, we need differentiability which is due to Theorem 3.4 possessed only by *Pat&Mat* and *Pat&Mat-NP*. Therefore, we consider only these two formulations and for simplicity, show it only for *Pat&Mat*. We apply a variance reduction technique based on delayed values similar to SAG [18].

At iteration k we have the decision variable $\boldsymbol{w}^k$ and the active minibatch I^k. First, we update the score vector $\boldsymbol{s}^k$ only on the active minibatch by setting

$$s_i^k = \begin{cases} \boldsymbol{x}_i^\top \boldsymbol{w}^k & \text{for all } i \in I^k, \\ s_i^{k-1} & \text{for all } i \notin I^k. \end{cases} \tag{3.29}$$

We keep scores from previous minibatches intact. We use Appendix **??** to compute the surrogate quantile t^k as the unique solution of

$$\sum_{i \in X} l(\beta(s_i^k - t^k)) = n\tau. \tag{3.30}$$

This is an approximation of the surrogate quantile $t(\boldsymbol{w}^k)$ from (3.13). The only difference from the true value $t(\boldsymbol{w}^k)$ is that we use delayed scores. Then we introduce artificial variable

$$\boldsymbol{a}^k = \sum_{i \in I^k} l'(\beta(s_i^k - t^k))\boldsymbol{x}_i. \tag{3.31}$$

Finally, we approximate the derivative $\nabla f(\boldsymbol{w}^k)$ from (3.28) by

$$g(\boldsymbol{w}^k) = \frac{1}{n_+^k} \sum_{i \in I_+^k} l'(t^k - s_i^k)(\nabla t^k - \boldsymbol{x}_i), \tag{3.32}$$

where ∇t^k is an approximation of $\nabla t(\boldsymbol{w}^k)$ from Theorem 3.4 defined by

$$\nabla t^k = \frac{\boldsymbol{a}^k + \boldsymbol{a}^{k-1} + \cdots + \boldsymbol{a}^{k-m+1}}{\sum_{i \in X} l'(\beta(s_i^k - t^k))}. \tag{3.33}$$

A perhaps more straightforward possibility would be to consider only $\boldsymbol{a}^k$ in the numerator of (3.33). However, choice (3.33) enables us to prove the convergence and it adds stability to the algorithm for small minibatches.

The whole procedure does not perform any vector operations outside of the current minibatch I^k. We summarize it in Algorithm 3.1. Note that a proper initialization for the first m iterations is needed. We finish the theoretical part by the convergence proof.

Theorem 3.9

Consider the *Pat&Mat* or *Pat&Mat-NP* formulation, stepsizes $\alpha^k = \frac{\alpha^0}{k+1}$ and piecewise disjoint minibatches $I^1, \ldots, I^m$ which cycle periodically $I^{k+m} = I^k$. If l is the smoothened (Huberized) hinge function, then Algorithm 3.1 converges to the global minimum of (3.16).

Algorithm 3.1 Stochastic gradient descent for maximizing accuracy at the top

Require: Dataset X, Minibatches $I^1, \ldots, I^m$, Stepsize α^k

Initialize weights $\boldsymbol{w}^0$
for $k = 0, 1, \ldots$ **do**
    Select a minibatch I^k
    Compute s_i^k for all $i \in I^k$ according to (3.29)
    Compute t^k according to (3.30)
    Compute $\boldsymbol{a}^k$ according to (3.31)
    Compute ∇t^k according to (3.33)
    Compute $g(\boldsymbol{w}^k)$ according to (3.32)
    Set $\boldsymbol{w}^{k+1} \leftarrow \boldsymbol{w}^k - \alpha^k g(\boldsymbol{w}^k)$
end for

4
Non-Linear Classification at the Top

Bibliography

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

[2] Yoav Freund, Raj Iyer, Robert E Schapire, and Yoram Singer. An efficient boosting algorithm for combining preferences. *The Journal of machine learning research*, 4:933–969, 2003.

[3] Shivani 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.

[4] Cynthia Rudin. The p-norm push: A simple convex ranking algorithm that concentrates at the top of the list. *J. Mach. Learn. Res.*, 10:2233–2271, December 2009.

[5] Nan Li, Rong Jin, and Zhi-Hua Zhou. Top rank optimization in linear time. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'14, pages 1502–1510, Cambridge, MA, USA, 2014. MIT Press.

[6] Stephen Boyd, Corinna Cortes, Mehryar Mohri, and Ana Radovanovic. Accuracy at the top. In *Advances in neural information processing systems*, pages 953–961, 2012.

[7] Thorsten 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.

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

[9] Elad Eban, Mariano Schain, Alan Mackey, Ariel Gordon, Ryan Rifkin, and Gal Elidan. Scalable learning of non-decomposable objectives. In *Artificial Intelligence and Statistics*, pages 832–840, 2017.

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

[11] Maksim Lapin, Matthias Hein, and Bernt Schiele. Top-k multiclass svm. In *Advances in Neural Information Processing Systems*, pages 325–333, 2015.

[12] Maksim Lapin, Matthias Hein, and Bernt Schiele. Analysis and optimization of loss functions for multiclass, top-k, and multilabel classification. *IEEE transactions on pattern analysis and machine intelligence*, 40(7):1533–1554, 2018.

[13] Alan Mackey, Xiyang Luo, and Elad Eban. Constrained classification and ranking via quantiles. *arXiv preprint arXiv:1803.00067*, 2018.

[14] Ao Zhang, Nan Li, Jian Pu, Jun Wang, Junchi Yan, and Hongyuan Zha. *tau*-fpl: Tolerance-constrained learning in linear time. *arXiv preprint arXiv:1801.04701*, 2018.

[15] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[16] Václav Mácha, Lukáš Adam, and Václav Šmídl. Nonlinear classifiers for ranking problems based on kernelized svm. *arXiv preprint arXiv:2002.11436*, 2020.

[17] Lukáš Adam and Martin Branda. Machine learning approach to chance-constrained problems: An algorithm based on the stochastic gradient descent. *arXiv preprint arXiv:1905.10986*, 2019.

[18] Mark Schmidt, Nicolas Le Roux, and Francis Bach. Minimizing finite sums with the stochastic average gradient. *Mathematical Programming*, 162(1-2):83–112, 2017.