

# Classification at the Top

---

**Ing. Václav Mácha**

September 20, 2023

**Supervisor:** doc. Ing. Václav Šmíd, Ph.D.

**Supervisor specialist:** Mgr. Lukáš Adam, Ph.D.

# Motivation

---

# Binary Classification

- General form of binary classification

$$\begin{aligned} \underset{\mathbf{w}, t}{\text{minimize}} \quad & C_1 \sum_{i \in \mathcal{I}_-} \mathbb{1}_{[s_i \geq t]} + C_2 \sum_{i \in \mathcal{I}_+} \mathbb{1}_{[s_i < t]} \\ \text{subject to} \quad & s_i = f(\mathbf{x}_i; \mathbf{w}), \quad i \in \mathcal{I} \end{aligned}$$

- $\mathbf{x}_i \in \mathbb{R}^d$  is a sample and  $y_i \in \{0, 1\}$  its corresponding label,  $C_1, C_2 \in \mathbb{R}$  are constants
- $\mathcal{I} = \mathcal{I}_- \cup \mathcal{I}_+$  is a set of indices of all sample where

$$\mathcal{I}_- = \{i \mid i \in \{1, 2, \dots, n\} \wedge y_i = 0\}$$

$$\mathcal{I}_+ = \{i \mid i \in \{1, 2, \dots, n\} \wedge y_i = 1\}$$

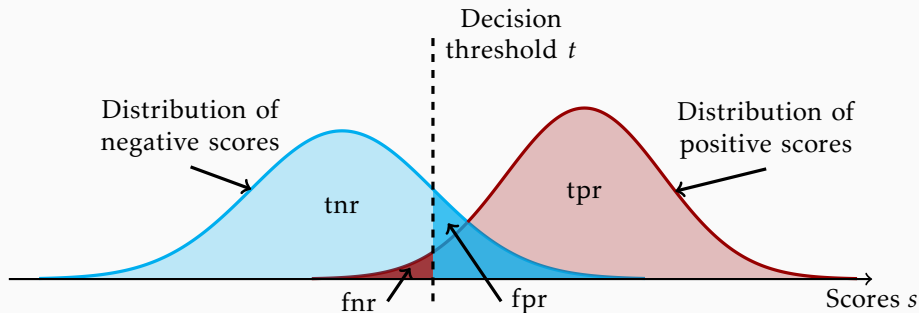
- $\mathbb{1}_{[\cdot]}$  is Iverson function defined by

$$\mathbb{1}_{[x]} = \begin{cases} 0 & \text{if } x \text{ is false} \\ 1 & \text{if } x \text{ is true} \end{cases}$$

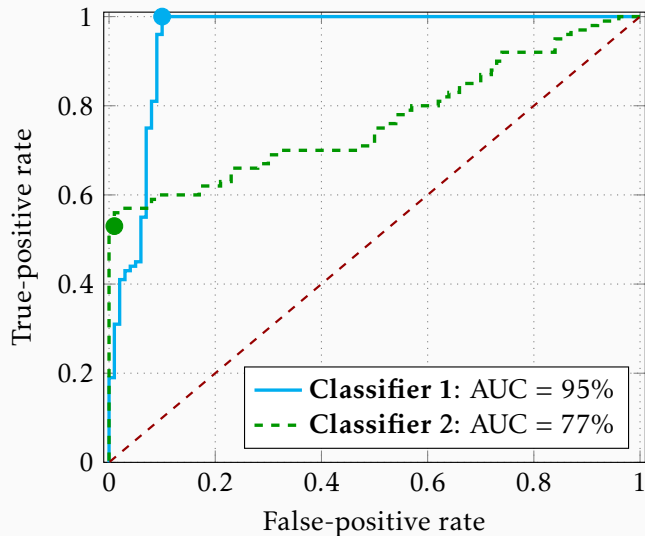
- Classifier consists of two parts: model  $f : \mathbb{R}^d \mapsto \mathbb{R}$  with trainable parameters  $\mathbf{w}$  that maps samples  $\mathbf{x}$  to scores  $s$  and decision threshold  $t \in \mathbb{R}$

# False rates

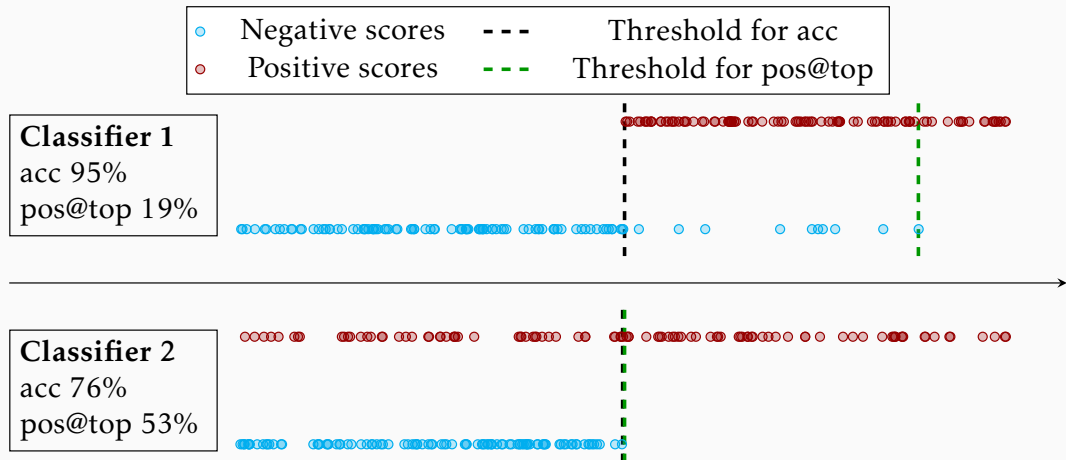
- Inference: Sample  $\mathbf{x}$  is classified as positive if  $s = f(\mathbf{x}; \mathbf{w}) \geq t$



## Classifier 1 is better ... or not?



## Sometimes Classifier 2 is the better one...



## Classification at the Top

---

# General problem formulation

- **Goal:** classify correctly only the most relevant samples. The most relevant samples are samples with the highest scores
- General formulation

$$\begin{aligned} \underset{\mathbf{w}}{\text{minimize}} \quad & C_1 \sum_{i \in \mathcal{I}_-} \mathbb{1}_{[s_i \geq t]} + C_2 \sum_{i \in \mathcal{I}_+} \mathbb{1}_{[s_i < t]} \\ \text{subject to} \quad & s_i = f(\mathbf{x}_i; \mathbf{w}), \quad i \in \mathcal{I}, \\ & t = G(\mathbf{s}, \mathbf{y}) \end{aligned}$$

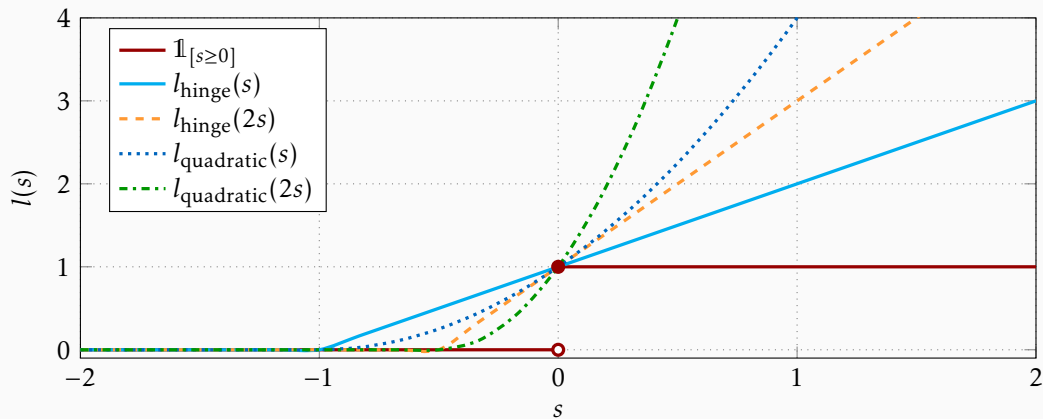
where threshold  $t$  is a function of all scores

- Difficult problem: constrained, discontinuous, non-convex, and non-decomposable



# How to get continuous objective function?

- By replacing  $\mathbb{1}_{[\cdot]}$  Iverson function with its surrogate approximation



- Using the surrogate approximation to replace  $\mathbb{1}_{[\cdot]}$  lead to general surrogate formulation

$$\begin{aligned} & \underset{\mathbf{w}}{\text{minimize}} && C_1 \sum_{i \in \mathcal{I}_-} l(s_i - t) + C_2 \sum_{i \in \mathcal{I}_+} l(t - s_i) \\ & \text{subject to} && s_i = f(\mathbf{x}_i; \mathbf{w}), \quad i \in \mathcal{I}, \\ & && t = G(\mathbf{s}, \mathbf{y}) \end{aligned}$$

## How to choose the decision threshold?

- For simplicity, we focus only on formulations that minimize false-negative rate and compute the threshold only from negative samples

$$\begin{aligned} \underset{\mathbf{w}}{\text{minimize}} \quad & \frac{1}{2} \|\mathbf{w}\|^2 + \frac{1}{n_+} \sum_{i \in \mathcal{I}_+} l(t - s_i) \\ \text{subject to} \quad & s_i = f(\mathbf{x}_i; \mathbf{w}), \quad i \in \mathcal{I}, \\ & t = G(\mathbf{s}, \mathbf{y}) \end{aligned}$$

where we use  $C_1 = 0$  and  $C_2 = \frac{1}{n_+}$ . Regularization is added for numerical stability.

## How to choose the decision threshold?

- For simplicity, we focus only on formulations that minimize false-negative rate and compute the threshold only from negative samples

$$\begin{aligned} \underset{\mathbf{w}}{\text{minimize}} \quad & \frac{1}{2} \|\mathbf{w}\|^2 + \frac{1}{n_+} \sum_{i \in \mathcal{I}_+} l(t - s_i) \\ \text{subject to} \quad & s_i = f(\mathbf{x}_i; \mathbf{w}), \quad i \in \mathcal{I}, \\ & t = G(\mathbf{s}, \mathbf{y}) \end{aligned}$$

where we use  $C_1 = 0$  and  $C_2 = \frac{1}{n_+}$ . Regularization is added for numerical stability.

- TopPush* maximizes the number of positive samples at the top

$$t = G_{\text{TopPush}}(\mathbf{s}, \mathbf{y}) = \max_{j \in \mathcal{I}_-} s_j$$

# How to choose the decision threshold?

- For simplicity, we focus only on formulations that minimize false-negative rate and compute the threshold only from negative samples

$$\underset{\mathbf{w}}{\text{minimize}} \quad \frac{1}{2} \|\mathbf{w}\|^2 + \frac{1}{n_+} \sum_{i \in \mathcal{I}_+} l(t - s_i)$$

$$\text{subject to} \quad s_i = f(\mathbf{x}_i; \mathbf{w}), \quad i \in \mathcal{I},$$
$$t = G(\mathbf{s}, \mathbf{y})$$

where we use  $C_1 = 0$  and  $C_2 = \frac{1}{n_+}$ . Regularization is added for numerical stability.

- TopPush* maximizes the number of positive samples at the top

$$t = G_{\text{TopPush}}(\mathbf{s}, \mathbf{y}) = \max_{j \in \mathcal{I}_-} s_j$$

- Pat&Mat-NP* maximizes true-positive rate with fixed false-positive rate

$$t = G_{\text{Pat\&Mat-NP}}(\mathbf{s}, \mathbf{y}) \iff t \text{ solves } \frac{1}{n_-} \sum_{i \in \mathcal{I}_-} l(s_i - t) = \tau$$

## Classification at the Top: Linear Model

---

- General surrogate formulation with linear model  $f(\mathbf{x}; \mathbf{w}) = \mathbf{w}^\top \mathbf{x}$

$$\begin{aligned} \underset{\mathbf{w}}{\text{minimize}} \quad & \frac{1}{2} \|\mathbf{w}\|^2 + \frac{1}{n_+} \sum_{i \in \mathcal{I}_+} l(t - s_i) \\ \text{subject to} \quad & s_i = \mathbf{w}^\top \mathbf{x}_i, \quad i \in \mathcal{I}, \\ & t = G(\mathbf{s}, \mathbf{y}) \end{aligned}$$

- Properties that we are interested in:
  - Convexity of the objective function
  - Robustness to outliers

# Convexity of the objective function

## Theorem

If the threshold  $t$  is a convex function of the weights  $\mathbf{w}$ , then function

$$L(\mathbf{w}) = \frac{1}{2} \|\mathbf{w}\|^2 + \frac{1}{n_+} \sum_{i \in \mathcal{I}_+} l(t(\mathbf{w}) - \mathbf{w}^\top \mathbf{x}_i)$$

is convex.

- What does it mean?
  - Both formulations *TopPush* and *Pat&Mat-NP* have convex thresholds
  - Both formulations are convex and continuous
  - We can solve both formulations using gradient descent algorithm



# How to solve it?

- Using gradient descent

$$\mathbf{w}^{k+1} \leftarrow \mathbf{w}^k - \alpha^k \cdot \nabla L(\mathbf{w}^k),$$

where  $\alpha^k > 0$  is a learning rate, and  $\nabla L(\mathbf{w}^k)$  is a gradient of the objective function

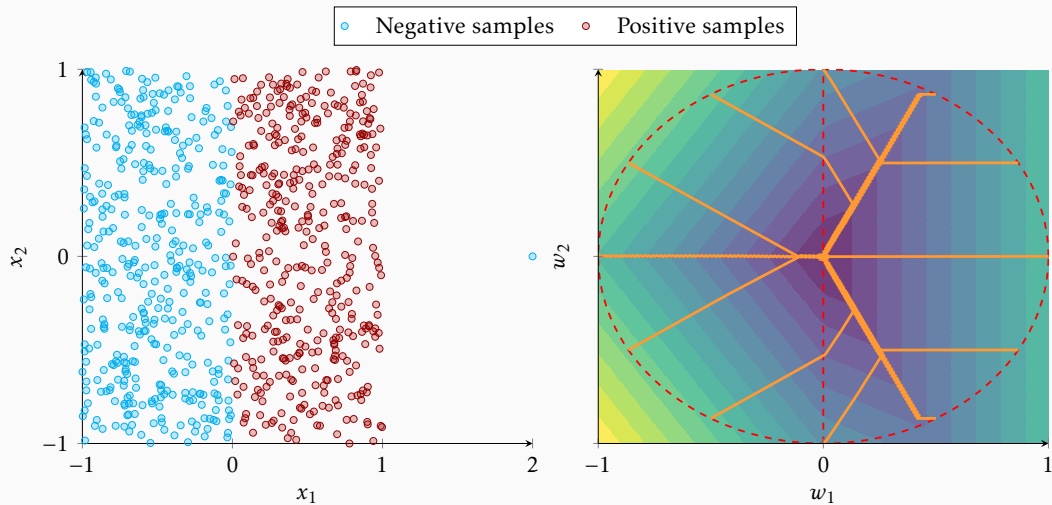
$$\nabla L(\mathbf{w}) = \mathbf{w} + \frac{1}{n_+} \sum_{i \in \mathcal{I}_+} l'(\mathbf{t}(\mathbf{w}) - \mathbf{w}^\top \mathbf{x}_i) (\nabla \mathbf{t}(\mathbf{w}) - \mathbf{x}_i)$$

- How to compute gradient of the threshold  $\nabla \mathbf{t}(\mathbf{w})$ ?
  - For *TopPush* it is easy

$$j^* = \arg \max_{j \in \mathcal{I}_-} s_j \quad \rightarrow \quad t = s_{j^*} \quad \rightarrow \quad \nabla \mathbf{t}(\mathbf{w}) = \nabla f(\mathbf{x}_{j^*}; \mathbf{w}) = \mathbf{x}_{j^*}$$

- For *Pat&Mat-NP* we have to use **implicit function theorem**.

# When convexity is not enough...



## **Classification at the Top: Non-linear Model**

---

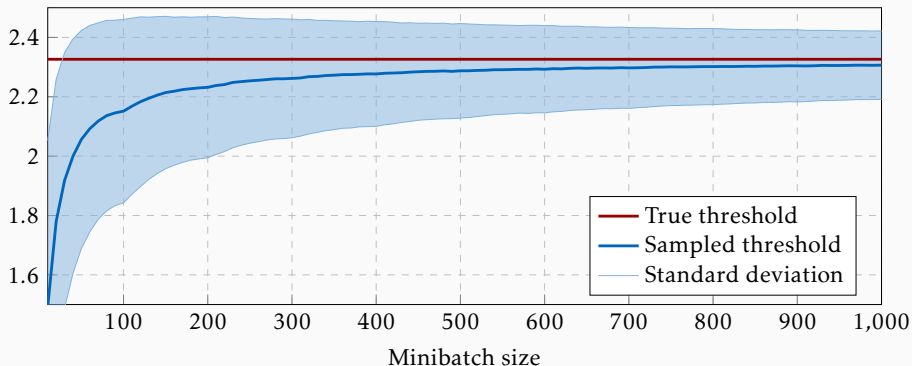
- General surrogate formulation with non-linear model  $f(\mathbf{x}; \mathbf{w})$

$$\begin{aligned} \underset{\mathbf{w}}{\text{minimize}} \quad & \frac{1}{2} \|\mathbf{w}\|^2 + \frac{1}{n_+} \sum_{i \in \mathcal{I}_+} l(t - s_i) \\ \text{subject to} \quad & s_i = f(\mathbf{x}_i; \mathbf{w}), \quad i \in \mathcal{I}, \\ & t = G(\mathbf{s}, \mathbf{y}) \end{aligned}$$

- Disadvantages:
  - Objective function is not convex
  - Non-linear models are usually large and expensive to train
- What to do if the dataset is too large to fit in memory? Stochastic gradient descent.

# Issues with stochastic gradient descent

- The threshold is a function of all scores  $\rightarrow$  the loss function is non-decomposable
- As a result, stochastic gradient descent provides a biased gradient estimate

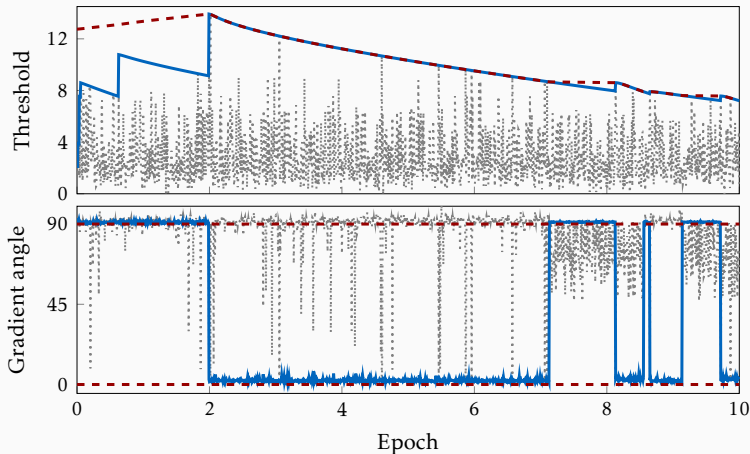


- How to reduce bias? Increase size of minibatch ...

# Is there a better way to reduce bias?

- *DeepTopPush*: Add threshold from last minibatch

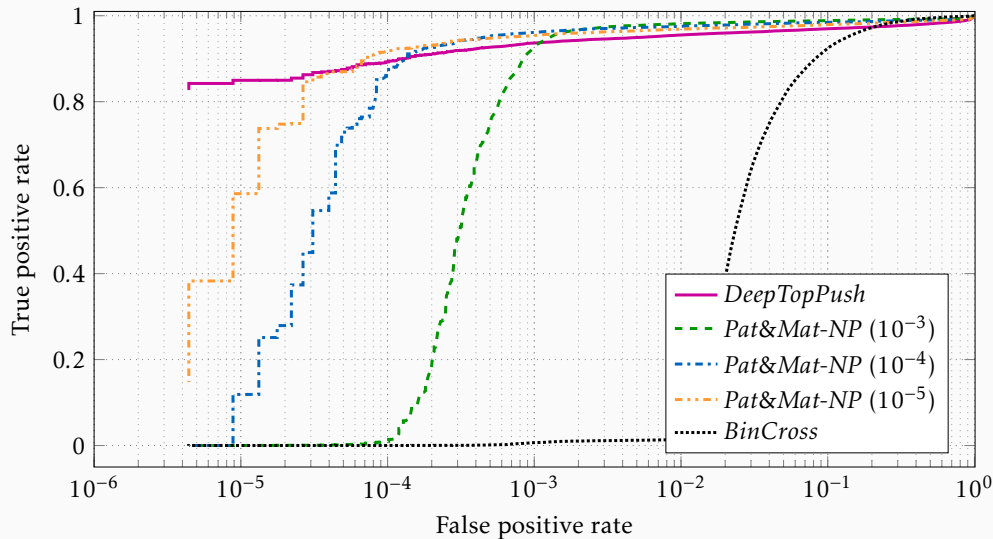
$$j^* = \arg \max_{j \in \mathcal{I}_-} s_j \rightarrow t = s_{j^*} \rightarrow \nabla t(\mathbf{w}) = \nabla f(\mathbf{x}_{j^*}; \mathbf{w})$$



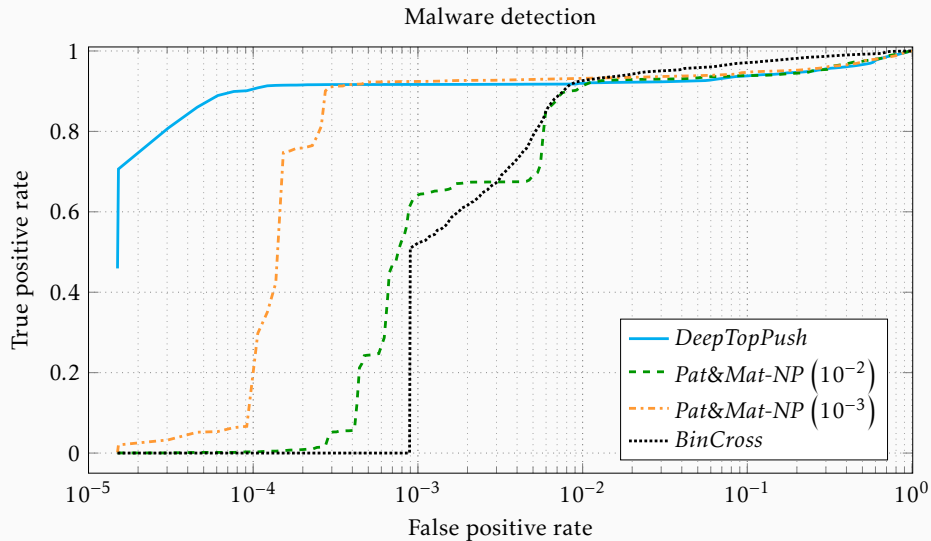
**How does it work?**

---

# Steganalysis







## Contributions

---

- **Unification Contributions:**

- Introduction of a unified framework for classification at the top
- Showed that problems such as Ranking or Accuracy at the Top fall into the framework
- Introduction of *Pat&Mat* and *Pat&Mat-NP* formulations

- **Theoretical Contributions:**

- Derivation of theoretical properties of formulations from the framework with linear model
- Derivation of dual forms and use of non-linear kernels

- **Algorithmic Contributions:**

- Derivation of an efficient algorithm for solving dual forms
- Introduction of a modified stochastic gradient descent
- Introduction of *DeepTopPush* formulation

**Thank you for your attention.**

---