Classification with Costly Features using Deep Reinforcement Learning

Jaromír Janisch • Tomáš Pevný • Viliam Lisý

Artificial Intelligence Center, Dept. of Computer Science, FEE, CTU in Prague

Abstract

TL;DR: With approximated Q-learning, we train a classifier that maximizes accuracy while minimizing required information.

We study a classification problem where each feature can be acquired for a cost and the goal is to optimize a trade-off between the expected classification error and the feature cost. We revisit a former approach that has framed the problem as a sequential decision-making problem and solved it by Q-learning with a linear approximation, where individual actions are either requests for feature values or terminate the episode by providing a classification decision. On a set of eight problems, we demonstrate that by replacing the linear approximation with neural networks the approach becomes comparable to the state-of-the-art algorithms developed specifically for this problem. The approach is flexible, as it can be improved with any new reinforcement learning enhancement, it allows inclusion of pre-trained high-performance classifier, and unlike prior art, its performance is robust across all evaluated datasets.

Costly Features

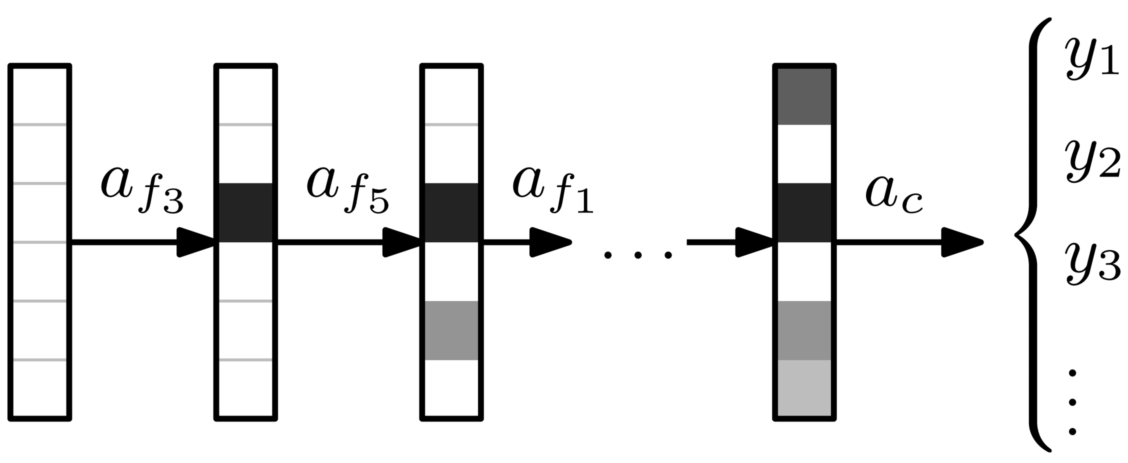
Features can only be retrieved only after paying some cost, which can be in form of *money*, *time* or *other resources*.

Feature group	Cost
Common information (weight, age, ...)	\$ 0
Simple examinations (heart rate, blood pressure)	\$ 5
Blood screeing	\$ 20
Radiograph	\$ 50
Magnetic resonance	\$ 200

Table 1. Example features and their costs from medical domain.

The goal is to find optimal solution for *any given sample* that maximizes accuracy and minimizes cost.

Sequential Decision-Making Problem



One sample at a time, we *sequentially* acquire features, until enough information is gathered and classification is performed.

Problem definition

The goal is to maximize expected accuracy while minimizing cost. The model is a pair of functions: $y_\theta(x) \to$ class, $z_\theta(x) \to$ cost.

Objective:

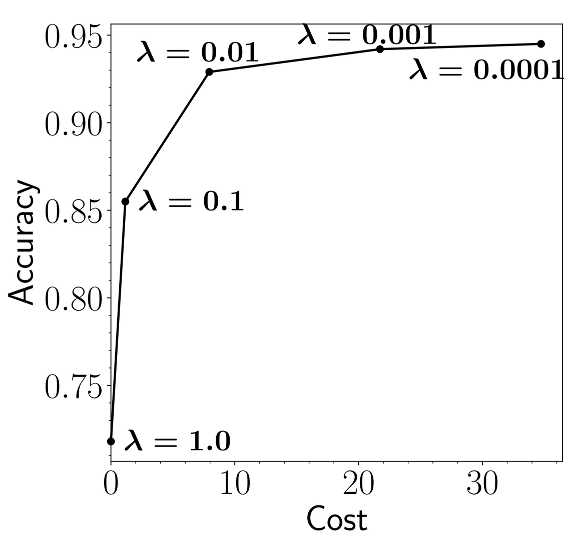
$$\min_\theta \mathop{\mathbb{E}}_{(x,y)\in\mathcal{D}} \left[\ell(y_\theta(x), y) + \lambda z_\theta(x)\right]$$

We use binary loss and fixed cost vector c.

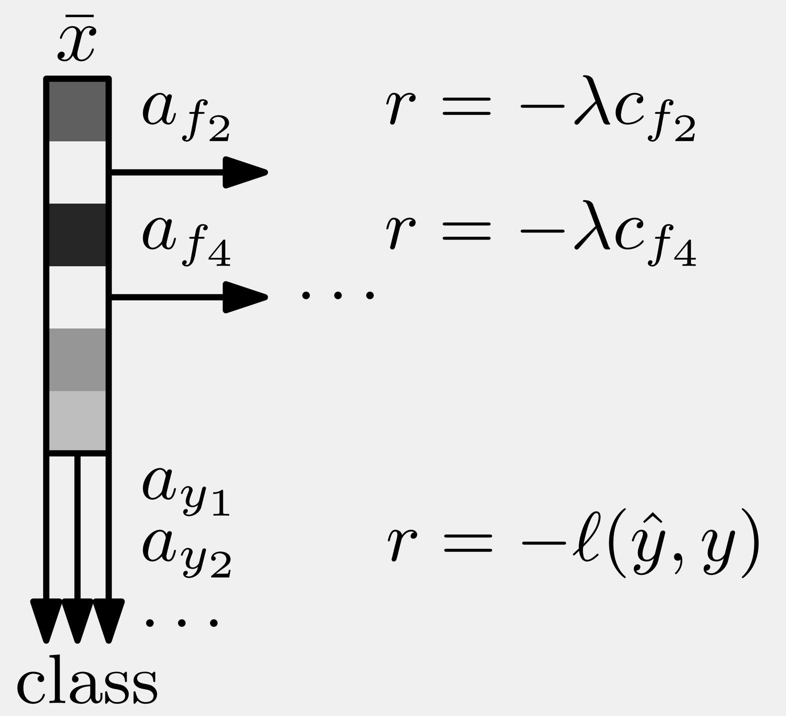
$$\ell(\hat{y}, y) = \begin{cases} 0 & \text{if } \hat{y} \neq y \\ 1 & \text{if } \hat{y} = y \end{cases} \qquad z_\theta(x) = c \cdot f_\theta(x)$$

Effect of different λ

The λ controls the weight of the feature costs in the objective. Low labda translates to cheap features, high lambda to expensive features.



Markov Decision Process



The problem is modeled as an MDP with full information. The agent has only partial-visibility of states.

$$s = (x, y, \mathcal{F}) \in \mathcal{S} \qquad \mathcal{A} = \mathcal{A}_c \cup \mathcal{A}_f$$

$$r(s,a) = \begin{cases} -\lambda c(a) \\ -\ell(a, y) \end{cases} \qquad t(s,a) = \begin{cases} (x, y, \mathcal{F} \cup a) & \text{if } a \in \mathcal{A}_f \\ \mathcal{T} & \text{if } a \in \mathcal{A}_c \end{cases}$$

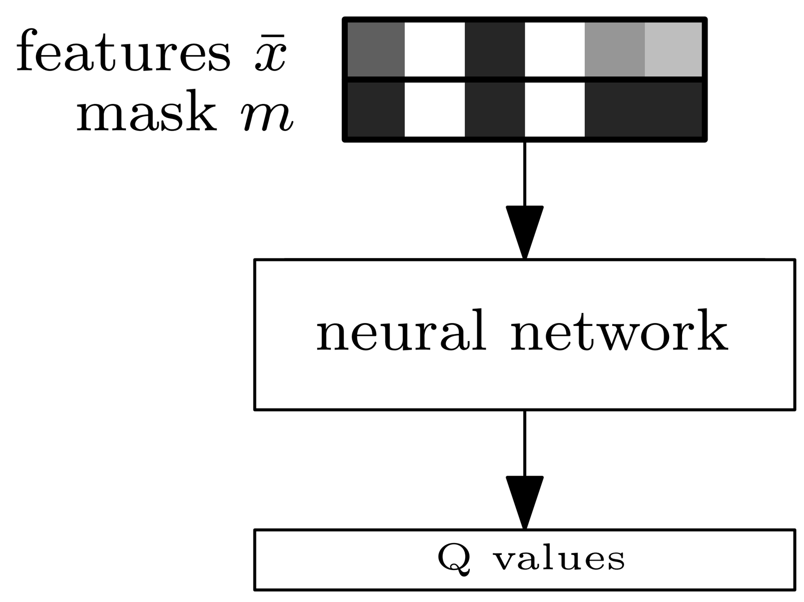
We treat each sample as a separate episode. The total reward per episode is:

$$R(x,y) = -\left[\ell(y_\theta(x), y) + \lambda z_\theta(x)\right]$$

So, by maximizing the expected reward, we are solving the problem objective.

Model

We use Q-learning with function approximation in various implementations: linear approximation, approximation with neural networks (DQN) and with recent deep RL techniques (double Q-learning, dueling architecture and Retrace).



Extensions

Pretraining: Since the Q-values for $\mathcal{A}_c$ actions are terminal, this part of the model can be pretrained supervisedly.

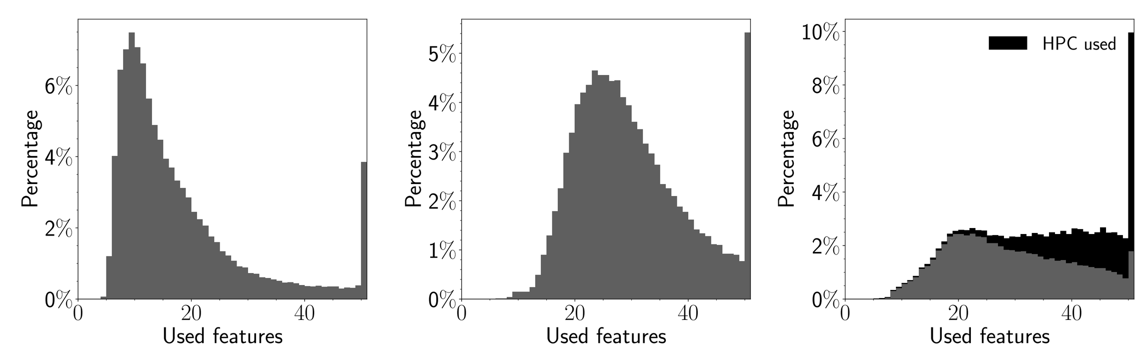
High-Performance Classifier: We can include another classifier as a terminal action to our model.

Deep RL extensions: Any new techniques from Deep RL can be directly used. We implemented Double DQN with Dueling architecture and Retrace importance sampling.

Other possibilities: Cost-sensitive classification, more HPCs, feature grouping, hard budget, etc.

Analysis of behaviour

We analyzed the behaviour of the agent on miniboone dataset, plotting a histogram of amount of used features for different settings.

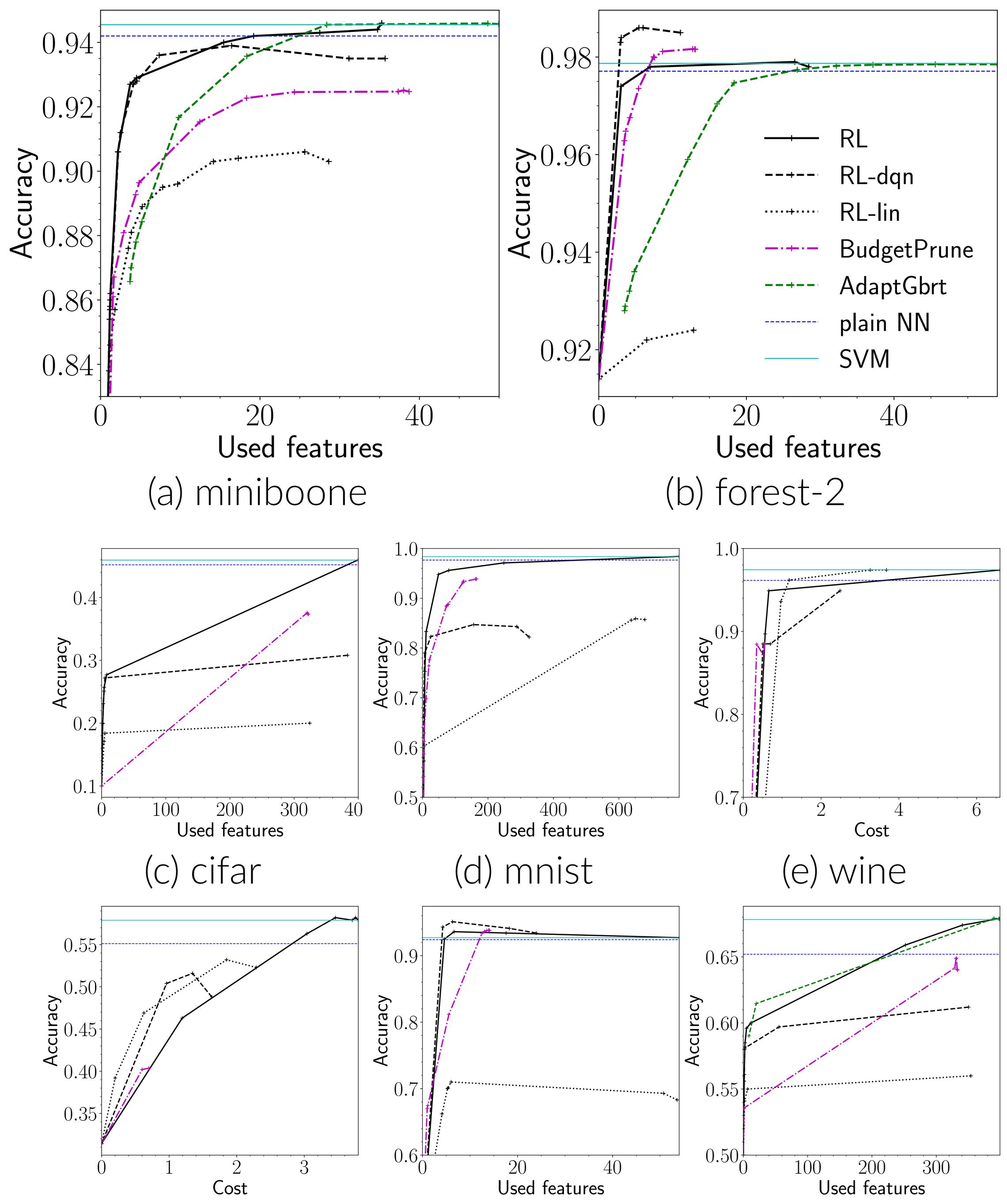


(a) $\lambda = 0.003$ (b) $\lambda = 0.0003$ (c) HPC, $\lambda = 0.0003$

Acknowledgments: This research was supported by the European Office of Aerospace Research and Development (grant no. FA9550-18-1-7008) and by The Czech Science Foundation (grants no. 18-21409S and 18-27483Y). The GPU used for this research was donated by the NVIDIA Corporation. Computational resources were provided by the CESNET LM2015042 and the CERIT Scientific Cloud LM2015085, provided under the program Projects of Large Research, Development, and Innovations Infrastructures.

Results

We compare to *AdaptGbrt* [2], *BudgetPrune* [3], Q-learning with linear approximation [1], DQN implementation and baseline neural network and SVM (with all features).



(a) miniboone (b) forest-2 (c) cifar (d) mnist (e) wine (f) yeast (g) forest (h) cifar-2

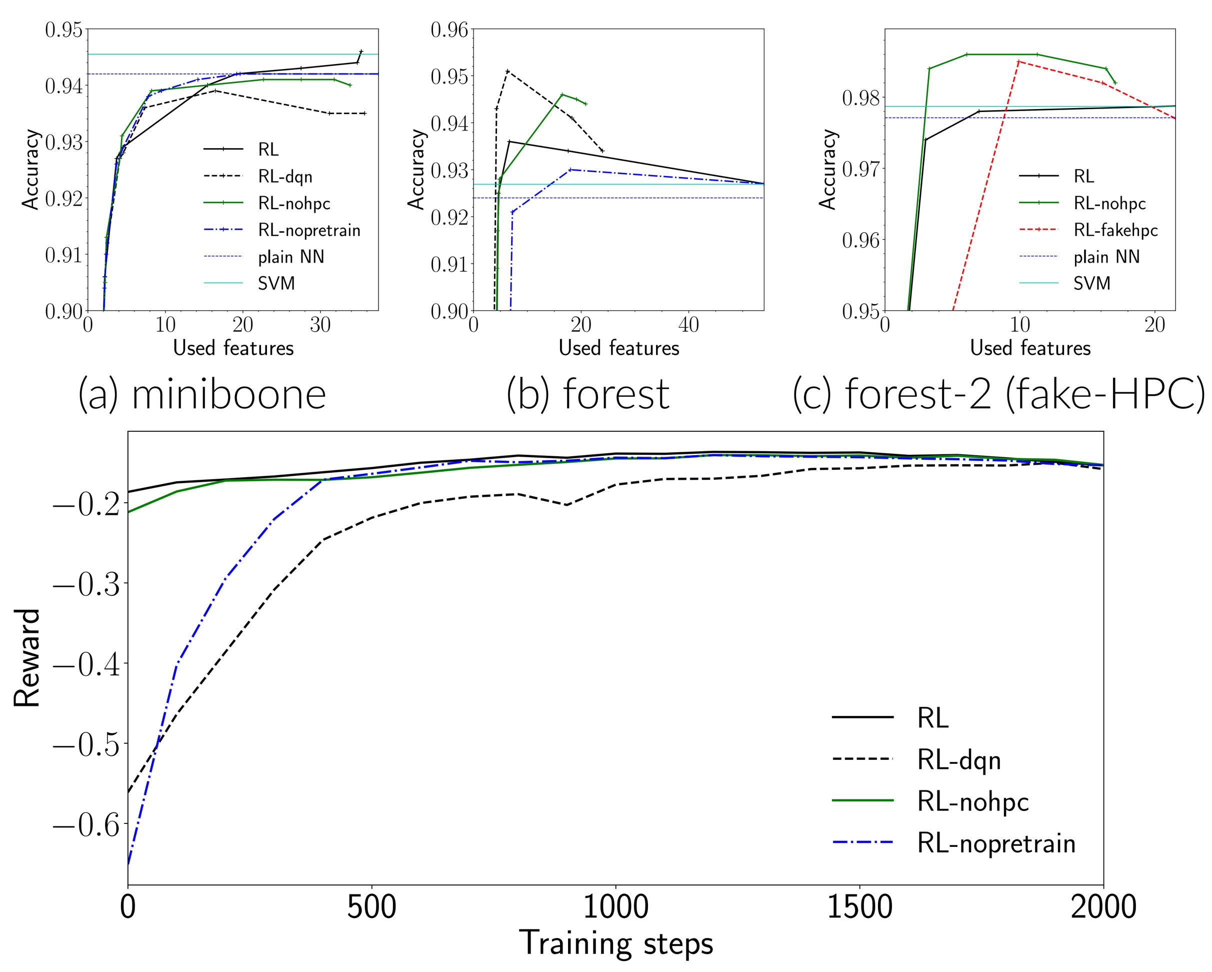
Figure 1. Comparison with prior-art algorithms.

Dataset	feats.	cls.	#trn	#val	#tst	costs
mnist	784	10	50k	10k	10k	U
cifar	400	10	40k	10k	10k	U
cifar-2	400	2	40k	10k	10k	U
forest	54	7	200k	81k	300k	U
forest-2	54	2	200k	81k	300k	U
miniboone	50	2	45k	19k	65k	U
wine	13	3	70	30	78	V
yeast	8	10	600	200	684	V

Table 2. Used datasets. The cost is either uniform (U) or variable (V).

Using HPC and pretraining

We studied using HPC and pretraining and noted that using HPC is not always helpful (see Figure 3c). However pretraining always bootstrap the training so that it converges quicker.



(a) miniboone (b) forest (c) forest-2 (fake-HPC)

Figure 2. Do HPC and pretraining bring any value?

References

[1] Gabriel Dulac-Arnold, Ludovic Denoyer, Philippe Preux, and Patrick Gallinari. Datum-wise classification: a sequential approach to sparsity. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 375–390. Springer, 2011.

[2] Feng Nan and Venkatesh Saligrama. Adaptive classification for prediction under a budget. In *Advances in Neural Information Processing Systems*, pages 4730–4740, 2017.

[3] Feng Nan, Joseph Wang, and Venkatesh Saligrama. Pruning random forests for prediction on a budget. In *Advances in Neural Information Processing Systems*, pages 2334–2342, 2016.