Label Ranking through Nonparametric Regression

ICML 2022

Alkis Kalavasis w/ Dimitris Fotakis and Eleni Psaroudaki

National Technical University of Athens

June 23, 2022

Menu

- Introduction
- Motivation
- Our Contributions
- Computational Label Ranking
- Experimental Results
- Statistical Label Ranking (in the paper)

Multiclass Learning

Set of labels-alternatives $[k] = \{1, \ldots, k\}$

The learning problem is an unknown distribution $\mathcal{D}$ over $\mathbb{X} \times [k]$

Learning Algorithm:

Input: $(x_1, y_1), \ldots, (x_n, y_n)$ i.i.d. from $\mathcal{D}$

Output: Classifier $h : \mathbb{X} \to [k]$

Goal: Small misclassification error on future examples from $\mathcal{D}$

$$\mathop{\mathbf{E}}_{(x,y)\sim\mathcal{D}} [1\{h(x) \neq y\}]$$

Label Ranking: Labels are rankings of $[k]$

Multiclass:

- Set of labels $[k]$
- Observe $(x, y) \sim \mathcal{D}$
- Output $h : \mathbb{X} \to [k]$
- Loss $\mathbf{E}_{(x,y)\sim\mathcal{D}}[1\{h(x) \neq y\}]$

Label Ranking:

- Set of labels $\mathbb{S}_k$
- Observe $(x, \sigma) \sim \mathcal{D}$
- Output $h : \mathbb{X} \to \mathbb{S}_k$
- Loss $\mathbf{E}_{(x,\sigma)\sim\mathcal{D}}[\Delta(h(x), \sigma)]$

Motivation

LR has various practical applications such as

- pattern recognition
- web advertisement
- sentiment analysis
- document categorization
- bio-informatics

Example: Adapt the ranking of the observed products according to user preferences

Motivation

Several approaches for tackling LR from the applied CS community [Vembu and Gartner, 2010], [Zhou et al., 2014].

These solutions come with experimental evaluation and no theoretical guarantees; e.g., algorithms based on decision trees and random forests are a workhorse for practical LR and lack formal theoretical guarantees.

Can we obtain theoretical guarantees and increase our understanding for algorithms based on DTs and RFs in view of their practical success?

Summary of Contributions

- We give a **formal theoretical generative model for LR**, motivated by existing applied CS previous works, through the lens of **nonparametric regression**.
- We provide the **first theoretical performance guarantees for algorithms based on decision trees and random forests** for LR, under mild conditions.
- We **experimentally** study the robustness of our algorithms to various **noise models**.
- We also study statistical LR with incomplete rankings building on the results of [Korba et al., 2017], [Clemencon et al., 2018], [Clemencon and Korba, 2018], [Clemencon and Vogel, 2020].

Distribution-free Nonparametric LR

[Hullermeier et al. 2008] evaluate individual alternatives through a real-valued score function. We assume that there exists such an underlying nonparametric score function $m^\star : \mathbb{X} \to [0,1]^k$, mapping features to score values.

Let $\mathbb{X} \subseteq \mathbb{R}^d$, $\mathcal{C}$ be a class of functions from $\mathbb{X}$ to $[0,1]^k$ and $\mathcal{D}_x$ be an arbitrary distribution over $\mathbb{X}$. Consider a noise distribution $\mathcal{E}$ over $\mathbb{R}^k$. Let $m^\star$ be an unknown target function in $\mathcal{C}$.

Distribution-free Nonparametric LR

[Hullermeier et al. 2008] evaluate individual alternatives through a real-valued score function. We assume that there exists such an underlying nonparametric score function $m^\star : \mathbb{X} \to [0,1]^k$, mapping features to score values.

Let $\mathbb{X} \subseteq \mathbb{R}^d$, $\mathcal{C}$ be a class of functions from $\mathbb{X}$ to $[0,1]^k$ and $\mathcal{D}_x$ be an arbitrary distribution over $\mathbb{X}$. Consider a noise distribution $\mathcal{E}$ over $\mathbb{R}^k$. Let $m^\star$ be an unknown target function in $\mathcal{C}$.

An example oracle $\mathrm{Ex}(m^\star, \mathcal{E})$ with complete rankings, works as follows:

- $x \sim \mathcal{D}_x$ and $\xi \sim \mathcal{E}$ independently,
- $\sigma = \mathbf{argsort}(m^\star(x) + \xi)$ $\mathbf{argsort}([0.2, 1.7, -0.7]) = (2, 1, 3)$
- it returns a labeled example $(x, \sigma) \in \mathbb{X} \times \mathbb{S}_k$.

In the noiseless case ($\xi = 0$ almost surely), we simply write $\mathrm{Ex}(m^\star)$.

Computational LR in ranking metric Δ

The learner is given i.i.d. samples from the oracle $\mathrm{Ex}(m^\star, \mathcal{E})$ and its goal is to *efficiently* output a hypothesis $h : \mathbb{R}^d \to \mathbb{S}_k$ such that, with high probability, the error $\mathbb{E}_{x \sim \mathcal{D}_x}[\Delta(h(x), h^\star(x))]$ is small.

$h^\star(x) := \mathrm{argsort}(m^\star(x))$

Efficiency: runtime is $\mathrm{poly}(d, k, 1/\epsilon, \log(1/\delta))$

In practice This problem is mainly solved using decision trees and random forests; however, theoretical guarantees were not known for this task.

Computational LR Conditions for Theoretical Guarantees

Feature space $\mathbb{X} = \{0,1\}^d$.
Regression vector-valued function $m^\star : \{0,1\}^d \to [0,1]^k$ with $m^\star = (m_1^\star, \dots, m_k^\star)$.
We assume that the following hold for any $j \in [k]$.

Computational LR Conditions for Theoretical Guarantees

Feature space $\mathbb{X} = \{0,1\}^d$.
Regression vector-valued function $m^\star : \{0,1\}^d \to [0,1]^k$ with $m^\star = (m_1^\star, \ldots, m_k^\star)$.
We assume that the following hold for any $j \in [k]$.

1. (Sparsity) $m_j^\star : \{0,1\}^d \to [0,1]$ is r-sparse, i.e., it depends on r out of d coordinates.

Computational LR Conditions for Theoretical Guarantees

Feature space $\mathbb{X} = \{0,1\}^d$.
Regression vector-valued function $m^\star : \{0,1\}^d \to [0,1]^k$ with $m^\star = (m_1^\star, \ldots, m_k^\star)$.
We assume that the following hold for any $j \in [k]$.

1. (Sparsity) $m_j^\star : \{0,1\}^d \to [0,1]$ is r-sparse, i.e., it depends on r out of d coordinates.
2. (Approximate Submodularity) The mean squared error L_j of $m_j^\star$ is C-approximate-submodular, i.e., for any $S \subseteq T \subseteq [d]$, $i \in [d]$, it holds that

$$L_j(T) - L_j(T \cup \{i\}) \leq C \cdot (L_j(S) - L_j(S \cup \{i\}))\ .$$

The **Mean Squared Error** (MSE) of a function $f : \{0,1\}^d \to [0,1]$ is equal to

$$L(f,S) = \mathbb{E}_{x \sim \mathcal{D}_x}\left[\left(f(x) - \mathbb{E}_{w \sim \mathcal{D}_x}[f(w) | w_S = x_S]\right)^2\right]\ .$$

Computational LR Contributions

We give the first theoretical guarantees for the Label Ranking problem for algorithms based on decision trees and random forests in the noiseless setting $\mathrm{Ex}(m^\star)$ and extensive experimental evidence for robustness of random forests and shallow decision trees in the noisy case $\mathrm{Ex}(m^\star, \mathcal{E})$.

The algorithms we study [Syrgkanis and Zampetakis, 2020]:

- Decision Trees with Level Splits
- Decision Trees with Breiman
- Random Forests with Level Splits
- Random Forests with Breiman

Computational LR Result

We choose the ranking metric $\Delta = d_{\mathrm{Spearman}}$

$d_{\mathrm{Spearman}}(\pi, \sigma) = \sum_i (\pi(i) - \sigma(i))^2$

$h^\star(x) = \mathbf{argsort}(m^\star(x))$

Theorem: Under r-sparsity and C-approximate submodularity conditions, there exists an algorithm based on Decision Trees via Level-Splits that draws $\widetilde{O}\left(\log(d) \cdot \mathrm{poly}_{C,r}(k/\epsilon)\right)$ i.i.d. samples from $\mathrm{Ex}(m^\star)$ and, in $\mathrm{poly}_{C,r}(d, k, 1/\epsilon)$ time, computes an estimate $h : \{0,1\}^d \to \mathbb{S}_k$ which, with probability 99%, satisfies $\mathbf{E}_{x \sim \mathcal{D}_x}\left[d_{\mathrm{Spearman}}(h(x), h^\star(x))\right] \leq \epsilon$.

Level-Splits: Every node at the same level of the tree has to split in the same coordinate, by using the next greedy criterion: at every level, we choose the coordinate that minimizes the total empirical mean squared error.

Computational LR

In practice, sparsity of the instance's "score function" is one of the reasons why such algorithms work well and efficiently in real world.

We can obtain similar results for

- Decision Trees with Breiman
- Random Forests with Level Splits and Random Forests with Breiman

Research Direction 1: Establish similar theoretical guarantees for the noisy oracle $\mathrm{Ex}(m^\star, \mathcal{E})$

Research Direction 2: Obtain theoretical results for LR for other practical algorithms, e.g., based on Neural Networks.

Computational LR with Noise and Experiments

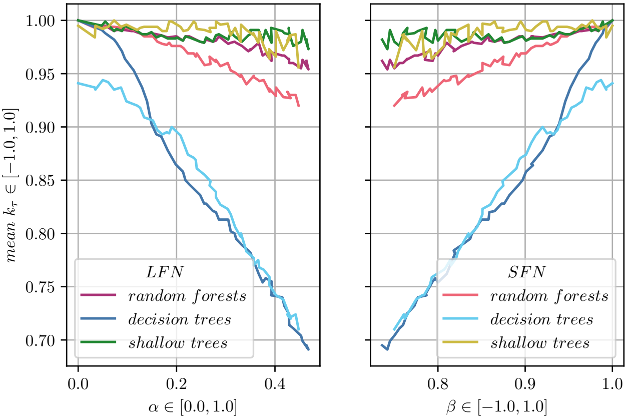
An example oracle $\mathrm{Ex}(m^\star, \mathcal{E})$ with complete rankings, works as follows:

- $x \sim \mathcal{D}_x$ and $\xi \sim \mathcal{E}$ independently,
- $\sigma = \mathbf{argsort}(m^\star(x) + \xi)$ // $h^\star(x) = \mathbf{argsort}(m^\star(x))$
- it returns a labeled example $(x, \sigma) \in \mathbb{X} \times \mathbb{S}_k$.

The noise distribution $\mathcal{E}$ is α-inconsistent for some $\alpha \in [0, 1]$ if

$$\mathop{\mathbf{E}}_{x \sim \mathcal{D}_x}\left[\mathop{\mathbf{Pr}}_{\xi \sim \mathcal{E}}[h^\star(x) \neq \sigma]\right] = \alpha\,.$$

Experimental Results k_τ large if rankings are similar

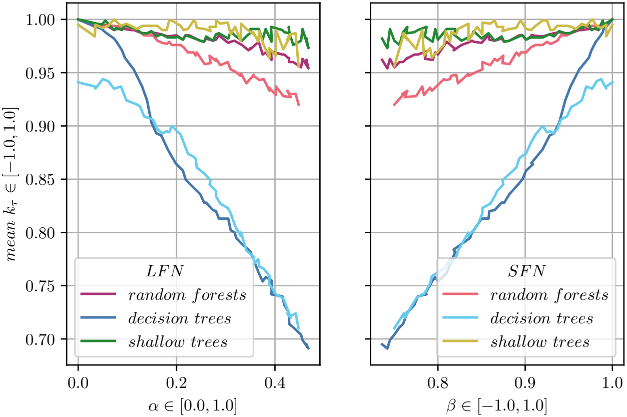


A second output inconsistency index

The noise distribution $\mathcal{E}$ satisfies the β-k_τ gap property for some $\beta \in [-1, 1]$ if

$$\mathop{\mathbf{E}}_{x \sim \mathcal{D}_x} \mathop{\mathbf{E}}_{\xi \sim \mathcal{E}} [k_\tau(h^\star(x), \sigma)] = \beta\,.$$

Experimental Results k_τ large if rankings are similar



Conclusion

- We give a formal theoretical generative model for LR.
- We provide the first theoretical performance guarantees for algorithms based on decision trees and random forests for LR, under mild conditions.
- We experimentally study robustness to noise.
- We study statistical LR with incomplete rankings (see paper).

Research Directions.

1. Obtain theoretical results for $\mathrm{Ex}(m^\star, \mathcal{E})$.
2. Obtain theoretical guarantees for other practical LR algorithms.

Conclusion

- We give a formal theoretical generative model for LR.
- We provide the first theoretical performance guarantees for algorithms based on decision trees and random forests for LR, under mild conditions.
- We experimentally study robustness to noise.
- We study statistical LR with incomplete rankings (see paper).

Research Directions.

1. Obtain theoretical results for $\mathrm{Ex}(m^\star, \mathcal{E})$.
2. Obtain theoretical guarantees for other practical LR algorithms.

Thank You!