On Tree-based Methods for Similarity Learning

Stéphan Clémençon[1] and Robin Vogel[1,2] [1] Télécom Paris, [2] IDEMIA

Motivation

Biometric identification =
checks correspondance of two measurements (x, x').

Given a similarity s and a threshold t,

$$(x, x') \text{ is a match } \Leftrightarrow s(x, x') > t. \quad (1)$$

The ROC curve of s gives the true positive rate (TPR) given the false positive rate (FPR) associated to eq. (1) for all thresholds $t \in \mathbb{R}$.

An usual approach is to optimize the Area under the ROC_s curve (AUC) of the similarity function s.

Biometric systems are deployed for a fixed, low FPR, which is hard to optimize in practice (see [1]).

Contributions

An extension of the algorithm TreeRank (see [2]) that learns a symmetric similarity function which optimizes the ROC curve for the similarity ranking problem (see [1]).

Statistical guarantees in $\|\cdot\|_\infty$ in the ROC space.

Empirical illustration on synthetic data.

Trials on real data.

Preliminaries

Classification setting. Assume $(X, Y) \sim P$, with:

- $Y \in \{1, \dots, K\}$ the output label,
- $X \in \mathcal{X} \subset \mathbb{R}^d$ input random variable.

Similarity learning. Select similarity function S s.t.

the larger $S(X, X')$ the higher $\mathbb{P}\{Y = Y' \mid X, X'\}$,

with $(X, Y) \perp (X', Y') \sim P$.

Optimal similarities $\mathcal{S}^*$ are increasing transforms of:

$$\eta(x, x') = \mathbb{P}\{Y = Y' \mid (X, X') = (x, x')\}.$$

The accuracy of any $s \in \mathcal{S}$ can be measured by:

$$d_p(s, s^*) = \|\mathrm{ROC}_s - \mathrm{ROC}^*\|_p,$$

where $s^* \in \mathcal{S}^*$ and $p \in [1, +\infty]$.

Given i.i.d. copies $\mathcal{D}_n = \{(X_i, Y_i)\}_{i=1}^n$ of (X, Y), one wants to rank the dependent pairs

$$\{((X_i, X_j), Z_{i,j}) : 1 \le i < j \le n\},$$

TreeRank (see [2].) Recursive technique that builds a piecewise constant score s_{D_n} for bipartite ranking.

Bipartite ranking considers data $\{(X_i, Y_i)\}_{i=1}^n$, i.i.d. copies of $(X, Y) \in \mathcal{X} \times \{-1, +1\}$.

TreeRank splits the input space recursively for weighted classification with a class $\mathcal{A} \subset \mathcal{P}(\mathcal{X})$.

Under assumptions on the distribution and $\mathcal{A}$, [2] (Corollary 16 therein), state that, given a tree of depth $D = D_n \sim \log(\sqrt{n})$ as $n \to \infty$, $\forall \delta > 0, \exists \lambda$ s. t., w.p. $\ge 1 - \delta, \forall n \in \mathbb{N}$, for $i \in \{1, +\infty\}$,

$$d_i(s_{D_n}, s^*) \le \exp(-\lambda \sqrt{\log n}). \quad (2)$$

Similarity TreeRank

For $F_\sigma(\mathcal{C})$, $\sigma \in \{-, +\}$ and $\mathcal{C} \subset \mathcal{X} \times \mathcal{X}$, introduce :

$$\widehat{F}_{\sigma,n}(\mathcal{C}) = \frac{1}{n_\sigma} \sum_{i<j} \mathbb{I}\{(X_i, X_j) \in \mathcal{C},\ Z_{i,j} = \sigma 1\},$$

with $n_\sigma = (2/(n(n-1))) \sum_{i<j} \mathbb{I}\{Z_{i,j} = \sigma 1\}$, which are not i.i.d. averages, but ratios of U-statistics of degree two (*i.e.* averages over pairs of observations, see [1]).

Input. Maximal depth $D \ge 1$, class $\mathcal{A}$ of measurable symmetric subsets of $\mathcal{X} \times \mathcal{X}$, training dataset $\mathcal{D}_n$.

1. (Initialization.) Set $\mathcal{C}_{0,0} = \mathcal{X} \times \mathcal{X}$, $\alpha_{d,0} = \beta_{d,0} = 0$ and $\alpha_{d,2^d} = \beta_{d,2^d} = 1$ for $d \ge 0$.

2. (Iterations.)
For $d = 0, \dots, D-1$ and $k = 0, \dots, 2^d - 1$:

a) (Optimization step.)
Set the entropic measure:

$$\begin{aligned}\Lambda_{d,k+1}(\mathcal{C}) =& (\alpha_{d,k+1} - \alpha_{d,k})\widehat{F}_{+,n}(\mathcal{C}) \\ & - (\beta_{d,k+1} - \beta_{d,k})\widehat{F}_{-,n}(\mathcal{C})\ .\end{aligned}$$

Find the best subset $\mathcal{C}_{d+1,2k}$ of the cell $\mathcal{C}_{d,k}$ in the AUC sense:

$$\mathcal{C}_{d+1,2k} = \mathrm{argmax}_{\mathcal{C} \in \mathcal{A},\ \mathcal{C} \subset \mathcal{C}_{d,k}} \widehat{\Lambda}_{d,k+1}(\mathcal{C})\ .$$

Then, set $\mathcal{C}_{d+1,2k+1} = \mathcal{C}_{d,k} \setminus \mathcal{C}_{d+1,2k}$.

b) (Update.) Set

$\alpha_{d+1,2k+1} = \alpha_{d,k} + \widehat{F}_{-,n}(\mathcal{C}_{d+1,2k})$,

$\beta_{d+1,2k+1} = \beta_{d,k} + \widehat{F}_{+,n}(\mathcal{C}_{d+1,2k})$,

and $\alpha_{d+1,2k+2} = \alpha_{d,k+1}$, $\beta_{d+1,2k+2} = \beta_{d,k+1}$.

3. (Output.) After D iterations, get the piecewise constant similarity function:

$$s_D(x, x') = \sum_{k=0}^{2^D - 1} (2^D - k)\, \mathbb{I}\{(x, x') \in \mathcal{C}_{D,k}\}.$$

Guarantees

The theoretical guarantees formulated in eq. (2) for the standard bipartite ranking setup remain valid in the similarity learning framework.
The following technical assumptions are involved:

- the feature space $\mathcal{X}$ is bounded;
- $\alpha \mapsto \mathrm{ROC}^*(\alpha)$ is twice differentiable with a bounded first order derivative;
- the class $\mathcal{A}$ is intersection stable, *i.e.* $\forall (\mathcal{C}, \mathcal{C}') \in \mathcal{A}^2, \mathcal{C} \cap \mathcal{C}' \in \mathcal{A}$;
- the class $\mathcal{A}$ has finite VC dimension $V < +\infty$;
- we have $\{(x, x') \in \mathcal{X}^2 :\ \eta(x, x') \ge q\} \in \mathcal{A}$ for any $q \in [0, 1]$;

Let s_D denote the output of the Similarity TreeRank algorithm with depth D.

Theorem 1. *Assume that the conditions above are fulfilled. Choose $D = D_n$ so that $D_n \sim \sqrt{\log n}$, as $n \to \infty$. Then, for all $\delta > 0$, there exists a constant λ s.t., with probability at least $1 - \delta$, we have for all $n \ge 2$:*

$$d_\infty(s_{D_n}, s^*) \le \exp(-\lambda \sqrt{\log n}).$$

Synthetic experiments

We generate data with a random tree of depth D_{gt} and fix $\mathcal{X} = \mathbb{R}^3$, $\delta = 0.01$, $n_{\text{test}} = 100{,}000$, $n_{\text{train}} = 150 \cdot (5/4)^{D_{gt}^2}$. TreeRank outputs s_D, results feature 95% CI's based on 400 runs.

Class asymmetry		
p_+	$D_1(s_D, s^*)$	$D_\infty(s_D, s^*)$
0.5	$0.07(\pm0.07)$	$0.30(\pm0.07)$
10^{-1}	$0.08(\pm0.08)$	$0.31(\pm0.08)$
10^{-3}	$0.42(\pm0.17)$	$0.75(\pm0.17)$
$2 \cdot 10^{-4}$	$0.45(\pm0.08)$	$0.81(\pm0.08)$
Parameters: $D = D_{gt} = 3$.		

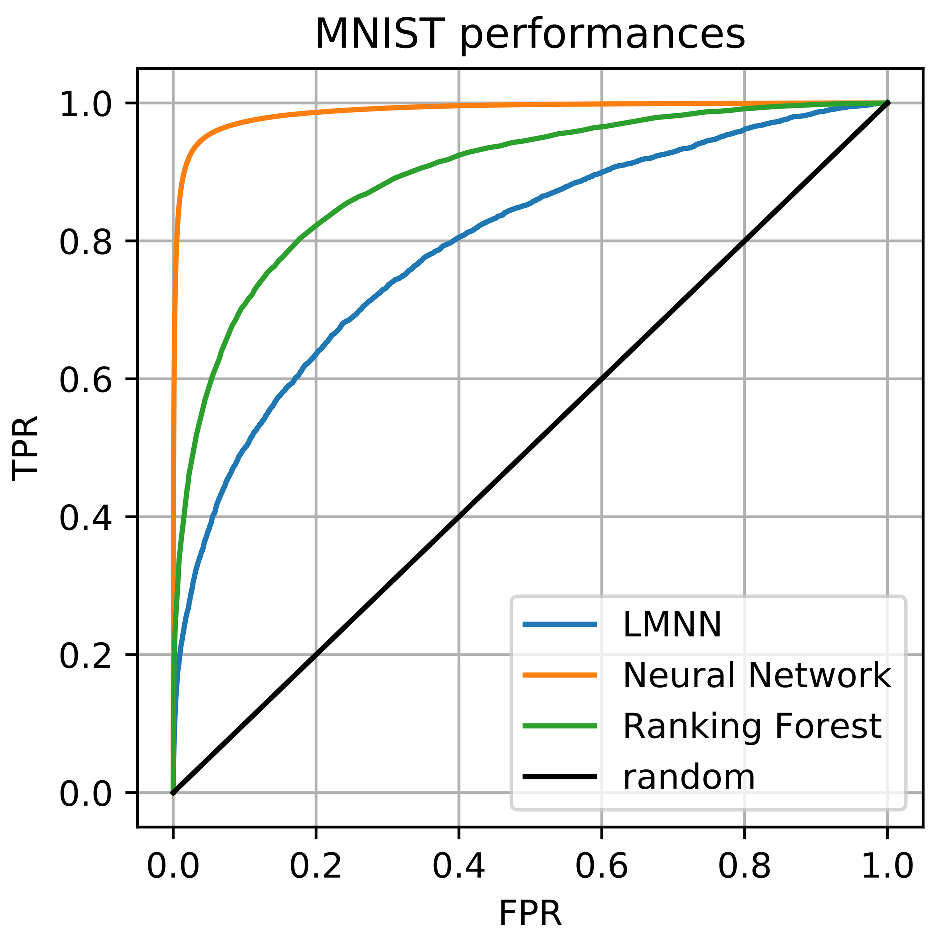
Model bias		
D	$D_1(s_D, s^*)$	$D_\infty(s_D, s^*)$
1	$0.21(\pm0.13)$	$0.65(\pm0.13)$
2	$0.11(\pm0.10)$	$0.43(\pm0.10)$
3	$0.07(\pm0.07)$	$0.30(\pm0.07)$
8	$0.06(\pm0.06)$	$0.28(\pm0.06)$
Parameters: $D_{gt} = 3$, $p = 0.5$.		

This table illustrate two parameters that impair generalization: pronounced **class asymmetry** and **under-specified models**.

Real data experiments

We try to validate our model by learning a similarity on MNIST, reduced by PCA. We compare three models:

· A linear metric learning algorithm: LMNN [3],
· A simple metric on a neural network encoding, trained for classification,
· Similarity TreeRank with decision stumps as $\mathcal{A}$.



Conclusions: Limited competitiveness of TreeRank. $\mathcal{A}$ is not expressive enough ?
Future work will explore Treerank approaches with neural networks for $\mathcal{A}$.

References

[1] Robin Vogel, Aurélien Bellet, and Stéphan Clémençon. A probabilistic theory of supervised similarity learning for pointwise ROC curve optimization. In *ICML*, 2018.

[2] Stéphan Clémençon and Nicolas Vayatis. Tree-based ranking methods. *IEEE Transactions on Information Theory*, 55(9):4316–4336, 2009.

[3] Kilian Q. Weinberger and Lawrence K. Saul. Distance Metric Learning for Large Margin Nearest Neighbor Classification. *JMLR*, 10:207–244, 2009.