A Refined Margin Distribution Analysis for Forest Representation Learning

Shen-Huan Lyu, Liang Yang and Zhi-Hua Zhou

LAMDA Group, Nanjing University, Nanjing, China

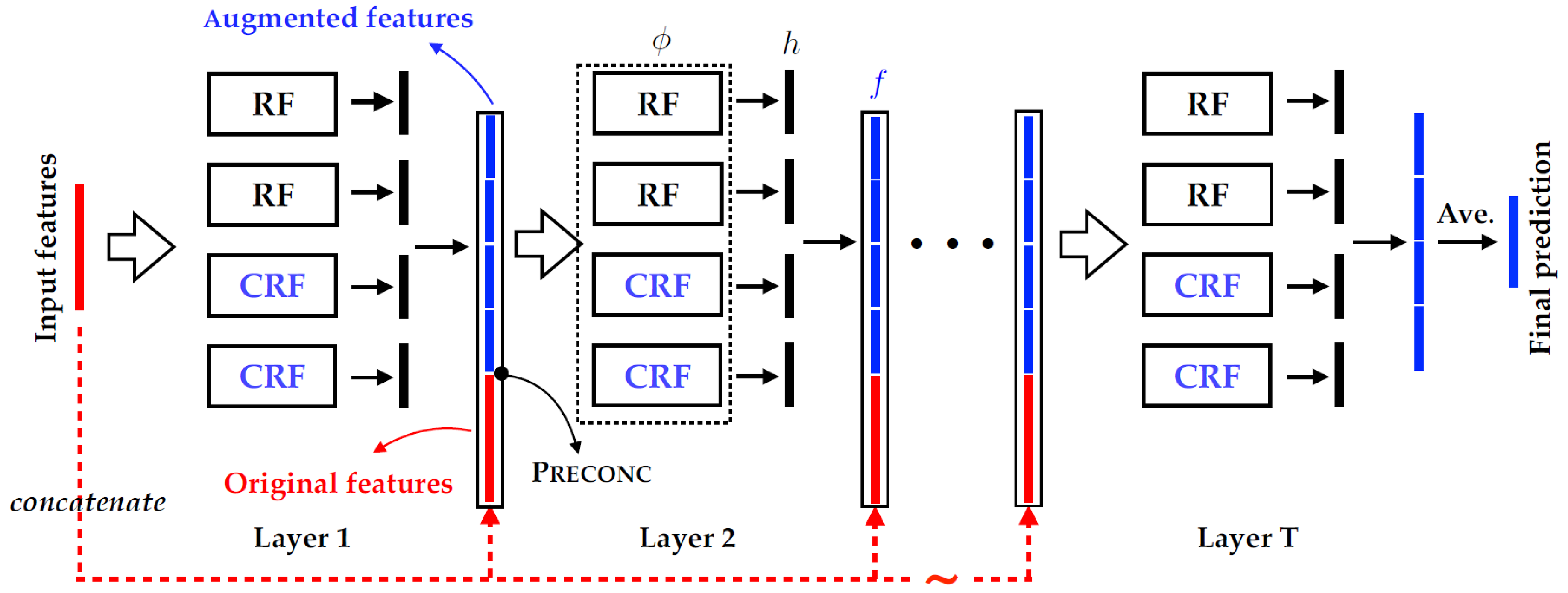
Learning And Mining from DatA
http://lamda.nju.edu.cn

Contact

lvsh@lamda.nju.edu.cn
yangl@lamda.nju.edu.cn
zhouzh@lamda.nju.edu.cn

Background

By realizing that the essence of deep learning lies in the *layer-by-layer processing*, *in-model feature transformation*, and *sufficient model complexity*, recently Zhou & Feng propose the deep forest model and the gcForest algorithm to achieve *forest representation learning*. It can achieve excellent performance on a broad range of tasks, and can even perform well on small or middle-scale of data.



Cascade Forest

The casForest model can be formalized as follows. We use a quadruple form

- **Forest block:** $\phi = (\phi_1, \phi_2, \ldots, \phi_T)$
- **casForest:** $\boldsymbol{h} = (h_1, h_2, \ldots, h_T)$
- **Augmented feature:** $\boldsymbol{f} = (f_1, f_2, \ldots, f_T)$
- **Sample distribution:** $\boldsymbol{\mathcal{D}} = (\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_T)$

ϕ_t is the function returned by the random forests block (Algorithm 1).

Algorithm 1 Random forests block $\mathcal{A}_{\text{rfb}}$ [33]

Input: A training set S drawn from $\mathcal{D}_t$ and the augmented feature $f_{t-1}(x_i), \forall i \in [m]$.
Output: The function computed by the random forests block in the t-th layer: ϕ_t.

```
Divide S to k-fold subsets {S_1, ..., S_k} randomly.
for S_i in {S_1, S_2, ..., S_k} do
    Using S/S_i to train two random forests and two completely random forests.
    Compute the prediction rate p_t^i(j) for the j-th leaf node generated by S/S_i.
    φ_t([x, f_{t-1}(x)]) ← E_j[p_t^i(j)], for any training sample (x, y) ∈ S_i.
end for
φ_t([x, f_{t-1}(x)]) ← E_{i,j}[p_t^i(j)], for any test sample (x, y) ∈ D.
return The function computed by the random forests block in the t-th layer: φ_t.
```

$$\phi_t = \begin{cases} \mathcal{A}_{\text{rfb}}\left([x_i; y_i]_{i=1}^m, \mathcal{D}_1\right) & t = 1, \\ \mathcal{A}_{\text{rfb}}\left([x_i, f_{t-1}(x_i); y_i]_{i=1}^m, \mathcal{D}_t\right) & t > 1. \end{cases}$$

$$f_t(x) = \begin{cases} \alpha_t h_t(x) & t = 1, \\ \alpha_t h_t(x) + f_{t-1}(x) & t > 1, \end{cases} \quad h_t(x) = \begin{cases} \phi_t(x) & t = 1, \\ \phi_t([x, f_{t-1}(x)]) & t > 1, \end{cases}$$

We find that the t-layer casForest model is defined by a *recursive formula*

$$h_t(x) = \phi_t([x, f_{t-1}(x)]) = \phi_t\left(\left[x, \sum_{l=1}^{t-1} \alpha_l h_l(x)\right]\right).$$

The entire additive cascade model is defined as follows

$$\tilde{F}(x) = \tilde{\sigma}(F(x)) = \underset{j \in \{1,2,\ldots,s\}}{\arg\max} \left[\sum_{t=1}^{T} \alpha_t h_t^j(x)\right]$$

Generalization Analysis

Theorem 1. *Let $\mathcal{D}$ be a distribution over $\mathcal{X} \times \mathcal{Y}$ and S be a training set of m samples drawn from $\mathcal{D}$. With probability at least $1-\delta$, for $r > 0$, the strong classifier $F(x)$ (the T-layer casDF model) satisfies that*

$$\Pr_{\mathcal{D}}[yF(x) < 0] \leq \inf_{r \in (0,1]} \left[\Pr_S[yF(x) < r] + \frac{1}{m^d} + \frac{3\sqrt{\mu}}{m^{3/2}} + \frac{7\mu}{3m} + \lambda\sqrt{\frac{3\mu}{m}}\right]$$

where

$$d = \frac{2}{1 - \mathbb{E}_S^2[yF(x)] + r/9} > 2, \ \mu = \ln m \ln(2\sum_{t=1}^{T} \alpha_t |\mathcal{H}_t|)/r^2 + \ln\frac{2}{\delta}, \lambda = \sqrt{\frac{\mathrm{Var}[yF(x)]}{\mathbb{E}_S^2[yF(x)]}}.$$

Remark 1. From Theorem 1, we know that the gap between the generalization error and empirical loss is generally bounded by the rate $\mathcal{O}(\lambda\sqrt{\ln m/m} + \ln m/m)$, which shows *minimizing the margin ratio* is the key to good generalization.
Remark 2. The hypothesis term $\ln\sum_{t=1}^{T} \alpha_t |\mathcal{H}_t|$ admits an explicit *dependency on the mixture coefficients*. Though some hypothesis sets used for learning could have large complexity, it will not be detrimental to generalization when the corresponding mixture weight is relatively small.

Optimization

Since we formulate casForest as an *additive model*, we utilize the reweighting approach to minimize the expected margin distribution loss

$$\mathbb{E}_S\left[\ell_{\text{md}}\left(\sum_{l=1}^{t} \alpha_l \gamma_l(x)\right)\right],$$

where the margin distribution loss function ℓ_{md} is designed to utilize the first- and second-order statistics of margins.

Algorithm 2 mdDF (margin distribution Deep Forest)

Input: Training set $S = \{(x_1, y_1), \ldots, (x_m, y_m)\}$ and random forests block algorithm $\mathcal{A}_{\text{rfb}}$.
Output: The final additive cascade model $\tilde{F}$.

Initialize $\alpha_0 \leftarrow 1$, $f_0 \leftarrow \emptyset$
Initialize sample weights: $\mathcal{D}_1(i) \leftarrow \frac{1}{m}, \forall i \in [m]$
for $t = 1, 2, \ldots, T$ **do**
  $\phi_t \leftarrow$ the random forests block returned by $\mathcal{A}_{\text{rfb}}([x_i, f_{t-1}(x_i); y_i]_{i=1}^m, \mathcal{D}_t)$.
  $h_t(x_i) \leftarrow \phi_t([x_i, f_{t-1}(x_i)]), \forall i \in [m]$.
  $\gamma_t(x_i) \leftarrow h_t^y(x_i) - \max_{j \neq y} h_t^j(x_i), \forall i \in [m]$.
  $\alpha_t \leftarrow \underset{\alpha_t}{\arg\min}\, \mathbb{E}_S[\ell_{\text{md}}(\sum_{l=1}^{t} \alpha_l \gamma_l(x))]$
  $f_t(x_i) \leftarrow \alpha_t h_t(x_i) + f_{t-1}(x_i), \forall i \in [m]$.
  $\mathcal{D}_{t+1}(i) \leftarrow \frac{\ell_{\text{md}}(\sum_{l=1}^{t} \alpha_l \gamma_l(x_i))}{\sum_{i=1}^{m} \ell_{\text{md}}(\sum_{l=1}^{t} \alpha_l \gamma_l(x_i))}, \forall i \in [m]$.
end for
return $\tilde{F} \leftarrow \underset{j \in \{1,2,\ldots,s\}}{\arg\max} \left[\sum_{t=1}^{T} \alpha_t h_t^j\right]$.

Results

Dataset	Attribute	MLP	RF	XGBoost	gcForest	mdDF	mdDF_{SF}	mdDF_{ST}	mdDF_{NP}
ADULT	Categorical	80.597	85.818	85.904	86.276 •	**86.560**	86.200	85.710	85.650
YEAST	Categorical	59.641	61.886	59.161	63.004 •	**63.340**	63.000	62.780	62.556
LETTER	Categorical	96.025	96.575	95.850	97.375 •	**97.500**	96.475	97.300	96.975
PROTEIN	Categorical	68.660	68.071	71.214 •	71.009	**71.247**	71.127	70.291	68.509
HAR	Mixed	94.231 •	92.569	93.112	94.224	**94.600**	93.926	94.290	94.060
SENSIT	Mixed	78.957	80.133	81.874	82.334 •	**82.534**	82.014	80.412	80.320
SATIMAGE	Numerical	91.125	91.200	90.450	91.700 •	**91.750**	91.600	91.300	90.800
MNIST	Numerical	98.621 •	96.831	97.730	98.252	**98.734**	98.254	98.101	98.240
Avg. Rank	-	3.650	4.000	3.750	2.375	1.000	-	-	-