Extreme Multi-label Loss Functions for Recommendation, Tagging, Ranking & Other Missing Label Applications – Supplementary Material

Himanshu Jain
IIT Delhi
himanshu.j689@gmail.com

Yashoteja Prabhu
IIT Delhi
yashoteja.prabhu@gmail.com

Manik Varma
Microsoft Research
manik@microsoft.com

1. PROPENSITY SCORED LOSSES

This section provides proofs for all the theorems stated in Section 4 of the paper.

THEOREM 4.1. *The loss function $\mathcal{L}(\mathbf{y},\hat{\mathbf{y}})$ evaluated on the observed ground truth $\mathbf{y}$ is an unbiased estimator of the true loss function $\mathcal{L}^*(\mathbf{y}^*,\hat{\mathbf{y}})$ evaluated on complete ground truth $\mathbf{y}^*$. Thus, $\mathbb{E}_{\mathbf{y}}[\mathcal{L}(\mathbf{y},\hat{\mathbf{y}})] = \mathbb{E}_{\mathbf{y}^*}[\mathcal{L}^*(\mathbf{y}^*,\hat{\mathbf{y}})]$, for any $P(\mathbf{y}^*)$ and $P(\mathbf{y})$ related through propensities p_l and any fixed $\hat{\mathbf{y}}$.*

PROOF.

$$\mathbb{E}_{\mathbf{y}}[\mathcal{L}(\mathbf{y},\hat{\mathbf{y}})] = \sum_{\mathbf{y}\in\{0,1\}^L} \mathcal{L}(\mathbf{y},\hat{\mathbf{y}})P(\mathbf{y}) \tag{1}$$

$$= \sum_{l=1}^{L}\sum_{\mathbf{y}\in\{0,1\}^L} \frac{\mathcal{L}_l^*(y_l,\hat{y}_l)}{p_l} P(y_1\dots y_L) \tag{2}$$

Since the loss function $\mathcal{L}(\mathbf{y},\hat{\mathbf{y}})$ decomposes over labels, $P(\mathbf{y})$ also decomposes. Assuming $\mathcal{S} = \{y_1\dots y_L\}\setminus\{y_l\}$

$$\mathbb{E}_{\mathbf{y}}[\mathcal{L}(\mathbf{y},\hat{\mathbf{y}})] = \sum_{l=1}^{L}\sum_{\mathbf{y}\in\{0,1\}^L} \frac{\mathcal{L}_l^*(y_l,\hat{y}_l)}{p_l} P(\mathcal{S}|y_l)P(y_l) \tag{3}$$

$$= \sum_{l=1}^{L}\sum_{y_l\in\{0,1\}} \frac{\mathcal{L}_l^*(y_l,\hat{y}_l)}{p_l} P(y_l)\sum_{\mathcal{S}} P(\mathcal{S}|y_l) \tag{4}$$

$$= \sum_{l=1}^{L}\sum_{y_l\in\{0,1\}} \frac{\mathcal{L}_l^*(y_l,\hat{y}_l)}{p_l} P(y_l) \tag{5}$$

Since $\mathcal{L}_l^*(y_l,\hat{y}_l) = 0$ if $y_l = 0$

$$= \sum_{l=1}^{L} \frac{\mathcal{L}_l^*(y_l=1,\hat{y}_l)}{p_l} P(y_l=1) \tag{6}$$

$$= \sum_{l=1}^{L} \frac{\mathcal{L}_l^*(y_l=1,\hat{y}_l)}{p_l} \Big(P(y_l=1|y_l^*=1)P(y_l^*=1) + \cancelto{0}{P(y_l=1|y_l^*=0)}P(y_l^*=0)\Big) \tag{7}$$

(Label noise is assumed to be one sided)

$$\mathbb{E}_{\mathbf{y}}[\mathcal{L}(\mathbf{y},\hat{\mathbf{y}})] = \sum_{l=1}^{L} \mathcal{L}_l^*(1,\hat{y}_l)P(y_l^*=1) \tag{8}$$

$$= \sum_{l=1}^{L} \big(\mathcal{L}_l^*(1,\hat{y}_l)P(y_l^*=1) + \mathcal{L}_l^*(0,\hat{y}_l)P(y_l^*=0)\big) \tag{9}$$

$$= \sum_{l=1}^{L}\sum_{y_l^*\in\{0,1\}} \mathcal{L}_l^*(y_l^*,\hat{y}_l)P(y_l^*) \tag{10}$$

$$= \sum_{\mathbf{y}^*} \mathcal{L}^*(\mathbf{y}^*,\hat{\mathbf{y}})P(\mathbf{y}^*) \tag{11}$$

$$= \mathbb{E}_{\mathbf{y}^*}[\mathcal{L}^*(\mathbf{y}^*,\hat{\mathbf{y}})] \tag{12}$$

□

THEOREM 4.2. *If $P(\mathbf{y}^*)$ is a delta function then $\mathbb{E}_{\mathbf{y}}[\mathcal{L}(\mathbf{y},\hat{\mathbf{y}})] = \mathbb{E}_{\mathbf{y}^*}[\mathcal{L}^*(\mathbf{y}^*,\hat{\mathbf{y}})]$ for non-decomposable loss functions of the form $\mathcal{L}^*(\mathbf{y}^*,\hat{\mathbf{y}}) = \sum_{l:y_l^*=1}^{L} \frac{\mathcal{L}_l^*(1,\hat{y}_l)}{g^*(\mathbf{y}^*,\hat{\mathbf{y}})}$ and $\mathcal{L}(\mathbf{y},\hat{\mathbf{y}}) = \sum_{l:y_l=1}^{L} \frac{\mathcal{L}_l^*(1,\hat{y}_l)}{g^*(\mathbf{y}^*,\hat{\mathbf{y}})p_l}$ with arbitrary propensities p_l.*

PROOF.

$$\mathbb{E}_{\mathbf{y}}[\mathcal{L}(\mathbf{y},\hat{\mathbf{y}})] = \sum_{\mathbf{y}\in\{0,1\}^L}\sum_{l=1}^{L} \frac{\mathcal{L}_l^*(y_l,\hat{y}_l)}{g^*(\mathbf{y}^*,\hat{\mathbf{y}})p_l} P(\mathbf{y}) \tag{13}$$

Since $g^*(\mathbf{y}^*,\hat{\mathbf{y}})$ is not dependent on $\mathbf{y}$, following can be written

$$\mathbb{E}_{\mathbf{y}}[\mathcal{L}(\mathbf{y},\hat{\mathbf{y}})] = \frac{1}{g(\mathbf{y}^*,\hat{\mathbf{y}})}\sum_{\mathbf{y}\in\{0,1\}^L}\sum_{l=1}^{L} \frac{\mathcal{L}_l^*(y_l,\hat{y}_l)}{p_l} P(\mathbf{y}) \tag{14}$$

Following steps 3-8 from proof of Theorem 4.1

$$= \frac{1}{g(\mathbf{y}^*,\hat{\mathbf{y}})}\sum_{l=1}^{L} \mathcal{L}_l^*(1,\hat{y}_l)P(y_l^*=1) \tag{15}$$

Since $P(\mathbf{y}^*)$ is a delta function, $P(y_l^*=1) = 1$ if $y_l^* = 1$ and 0 otherwise. Also it is assumed that if $y_l^* = 0$, $\mathcal{L}_l^*(y_l^*,\hat{y}_l) = 0$

$$\mathbb{E}_{\mathbf{y}}[\mathcal{L}(\mathbf{y},\hat{\mathbf{y}})] = \frac{1}{g^*(\mathbf{y}^*,\hat{\mathbf{y}})}\sum_{l=1}^{L} \mathcal{L}_l^*(y_l^*,\hat{y}_l) \tag{16}$$

$$= \mathcal{L}^*(\mathbf{y}^*,\hat{\mathbf{y}}) \tag{17}$$

□

COROLLARY 4.2.1. *If $P(\mathbf{y}^*)$ is a delta function and labels are retained with propensities $p_l = g_l/g^*(\mathbf{y}^*)$, then $\mathbb{E}_{\mathbf{y}}[\mathcal{L}(\mathbf{y},\hat{\mathbf{y}})] = \mathbb{E}_{\mathbf{y}^*}[\mathcal{L}^*(\mathbf{y}^*,\hat{\mathbf{y}})]$ for non-decomposable loss functions of the form $\mathcal{L}^*(\mathbf{y}^*,\hat{\mathbf{y}}) = \sum_{l:y_l^*=1}^{L} \frac{\mathcal{L}_l^*(1,\hat{y}_l)}{g^*(\mathbf{y}^*)}$ and $\mathcal{L}(\mathbf{y},\hat{\mathbf{y}}) = \sum_{l:y_l=1}^{L} \frac{\mathcal{L}_l^*(1,\hat{y}_l)}{g_l}$.*

PROOF. From Theorem 4.2

$$\mathbb{E}_{\mathbf{y}^*}\left[\sum_{l:y_l^*=1}^{L}\frac{\mathcal{L}_l^*(1,\hat{y}_l)}{g^*(\mathbf{y}^*)}\right] = \mathbb{E}_{\mathbf{y}}\left[\sum_{l:y_l=1}^{L}\frac{\mathcal{L}_l^*(1,\hat{y}_l)}{g^*(\mathbf{y}^*)p_l}\right] \tag{18}$$

Putting $p_l = g_l/g^*(\mathbf{y}^*)$

$$= \mathbb{E}_{\mathbf{y}}\left[\sum_{l:y_l=1}^{L}\frac{\mathcal{L}_l^*(1,\hat{y}_l)}{g_l}\right]$$

□

THEOREM 4.3. ***(Concentration bound)*** *Let $\mathbf{Y} = \{\mathbf{y}_i \in \{0,1\}^L\}_{i=1}^N$ be a set of N independent observed ground truth random variables. Then with probability at least $1-\delta$ $\left|\mathbb{E}_{\mathbf{Y}}\left[\frac{1}{N}\sum_{i=1}^N \mathcal{L}(\mathbf{y}_i,\hat{\mathbf{y}}_i)\right] - \frac{1}{N}\sum_{i=1}^N \mathcal{L}(\mathbf{y}_i,\hat{\mathbf{y}}_i)\right| \leq \rho\bar{L}\sqrt{\frac{1}{2N}\log\left(\frac{2}{\delta}\right)}$ where $\rho = max_{il}\left|\frac{1}{p_{il}}\frac{\mathcal{L}_l^*(y_{il},\hat{y}_{il})}{g(\mathbf{y}_i^*,\hat{\mathbf{y}}_i)}\right|$, $\bar{L} = \sqrt{\frac{1}{N}\sum_{i=1}^N {L_i^*}^2}$ and L_i^* is the maximum number of labels that can be relevant to a data point i in the complete ground truth.*

PROOF. Change c_i, in the average loss function value when one of the N random variables ($\{\mathbf{y}_i\}_{i=1}^N$) is changed is:

$$c_i = \frac{1}{N}\sum_{l=1}^{L}\left(\frac{\mathcal{L}_l^*(y_{il},\hat{y}_{il})}{g(\mathbf{y}_i^*,\hat{\mathbf{y}}_i)p_{il}} - \frac{\mathcal{L}_l^*(y'_{il},\hat{y}_{il})}{g(\mathbf{y}_i^*,\hat{\mathbf{y}}_i)p_{il}}\right) \tag{19}$$

Since either of y_{il}, y'_{il} has to be zero, correspondingly the value of function $\mathcal{L}_l^*$ will also be zero.

$$c_i \leq \frac{1}{N}\sum_{l=1}^{L}\left(\frac{\mathcal{L}_l^*(y_{il},\hat{y}_{il})}{g(\mathbf{y}_i^*,\hat{\mathbf{y}}_i)p_{il}}\right) \tag{20}$$

Note that for a given instance i, not all random variables $\{y_{il}\}_{l=1}^L$ can be changed because of one sided nature of noise i. e. random variables corresponding to only those instance-label pairs can be changed for which $y_{il}^* = 1$. So assuming that L_i^* is the maximum number of labels relevant to an instance i then for that instance at max L_i^* random variables can be changed

$$c_i \leq \frac{L_i^*}{N}\rho \qquad \text{where } \rho = max_{il}\left|\frac{1}{p_{il}}\frac{\mathcal{L}_l^*(y_{il},\hat{y}_{il})}{g(\mathbf{y}_i^*,\hat{\mathbf{y}}_i)}\right|$$

Now using McDiarmid's Theorem, with probability at least $1-\delta$

$$\left|\mathbb{E}_{\mathbf{Y}}\left[\frac{1}{N}\sum_{i=1}^N \mathcal{L}(\mathbf{y}_i,\hat{\mathbf{y}}_i)\right] - \frac{1}{N}\sum_{i=1}^N \mathcal{L}(\mathbf{y}_i,\hat{\mathbf{y}}_i)\right| \leq \sqrt{\frac{1}{2}\sum_{i=1}^N c_i^2\log\left(\frac{2}{\delta}\right)} \tag{21}$$

$$\leq \sqrt{\frac{\rho^2}{2N^2}\sum_{i=1}^N {L_i^*}^2\log\left(\frac{2}{\delta}\right)} \tag{22}$$

$$= \rho\bar{L}\sqrt{\frac{1}{2N}\log\left(\frac{2}{\delta}\right)} \tag{23}$$

□

THEOREM 4.4. *For any $P(\mathbf{y}^*)$ and $P(\mathbf{y})$ related through propensities p_l and any fixed $\hat{\mathbf{y}}$, $\mathbb{E}_{\mathbf{y}}[\mathcal{L}(\mathbf{y},\hat{\mathbf{y}})] = \mathbb{E}_{\mathbf{y}^*}[\mathcal{L}^*(\mathbf{y}^*,\hat{\mathbf{y}})]$ where $\mathcal{L}(\mathbf{y},\hat{\mathbf{y}}) = \sum_{l=1}^L\left(\frac{1}{p_l}(1-2\hat{y}_l)\right)y_l + \hat{y}_l^2$ is an unbiased estimator of the Hamming loss $\mathcal{L}^*(\mathbf{y}^*,\hat{\mathbf{y}}) = \sum_l \|y_l^* - y_l\|^2$ with concentration bound $\rho\bar{L}\sqrt{\frac{1}{2N}\log(2/\delta)}$ where $\rho = max_{il}(1/p_{il})$.*

PROOF.

$$\mathbb{E}_{\mathbf{y}}[\mathcal{L}(\mathbf{y},\hat{\mathbf{y}})] = \mathbb{E}_{\mathbf{y}}\left[\sum_{l=1}^{L}\left(\frac{1}{p_l}(1-2\hat{y}_l)\right)y_l + \hat{y}_l^2\right] \tag{24}$$

Using steps 1-5 from Theorem 4.1, this can we written as

$$= \sum_{l=1}^{L}\sum_{y_l\in\{0,1\}}\left(\frac{1}{p_l}(1-2\hat{y}_l)y_l + \hat{y}_l^2\right)P(y_l) \tag{25}$$

$$= \sum_{l=1}^{L}\left(\frac{1}{p_l}(1-2\hat{y}_l) + \hat{y}_l^2\right)P(y_l=1) + \hat{y}_l^2 P(y_l=0) \tag{26}$$

$$= \sum_{l=1}^{L}\frac{1}{p_l}(1-2\hat{y}_l)P(y_l=1) + \hat{y}_l^2 \tag{27}$$

$$= \sum_{l=1}^{L}\frac{1}{p_l}(1-2\hat{y}_l)P(y_l=1|y_l^*=1)P(y_l^*=1) + \hat{y}_l^2 \tag{28}$$

$$= \sum_{l=1}^{L}(1-2\hat{y}_l)P(y_l^*=1) + \hat{y}_l^2(P(y_l^*=1)+P(y_l^*=0)) \tag{29}$$

$$= \sum_{l=1}^{L}(1-2\hat{y}_l+\hat{y}_l^2)P(y_l^*=1) + \hat{y}_l^2 P(y_l^*=0) \tag{30}$$

$$= \sum_{l=1}^{L}(y_l^*-\hat{y}_l)^2 P(y_l^*=1) + \hat{y}_l^2 P(y_l^*=0) \tag{31}$$

$$= \sum_{l=1}^{L}\sum_{y_l\in\{0,1\}}(y_l^*-\hat{y}_l)^2 P(y_l^*) \tag{32}$$

$$= \mathbb{E}_{\mathbf{y}^*}\left[\sum_{l=1}^{L}(y_l^*-\hat{y}_l)^2\right] \tag{33}$$

Concentration bound

Change c_i, in the average hamming loss value when one of the N random variables ($\{\mathbf{y}_i\}_{i=1}^N$) is changed is

$$c_i = \frac{1}{N}\sum_{l=1}^{L}\left(\frac{1}{p_{il}}(1-2\hat{y}_{il})y_{il} - \frac{1}{p_{il}}(1-2\hat{y}_{il})y'_{il}\right) \tag{34}$$

Since either of y_{il}, y'_{il} has to be zero.

$$c_i \leq \frac{1}{N}\sum_{l=1}^{L}\frac{1}{p_{il}}(1-2\hat{y}_{il})y_{il} \tag{35}$$

$$c_i \leq \frac{L_i^*}{N}\rho \tag{36}$$

where $\rho = max_{il}\frac{1}{p_{il}}$ and L_i^* is the maximum number of labels relevant to an instance i

Now using McDiarmid's Theorem, with probability at least $1-\delta$

$$\left|\mathbb{E}_{\mathbf{Y}}\left[\frac{1}{N}\sum_{i=1}^{N}\mathcal{L}(\mathbf{y}_i,\hat{\mathbf{y}}_i)\right]-\frac{1}{N}\sum_{i=1}^{N}\mathcal{L}(\mathbf{y}_i,\hat{\mathbf{y}}_i)\right|\le\rho\bar{L}\sqrt{\frac{1}{2N}\log\left(\frac{2}{\delta}\right)} \tag{37}$$

□

Algorithm 1 FastXML-PREDICT($\{\mathcal{T}_1,..\mathcal{T}_T\},\mathbf{x}$)

```
for i = 1,..,T do
    n ← T_i.root
    while n is not a leaf do
        w ← n.w
        if w^T x > 0 then
            n ← n.left_child
        else
            n ← n.right_child
        end if
    end while
    P_i^leaf(x) ← n.P            #Label probabilities in leaf node n
end for
Q = 1/T Σ_{i=1}^T P_i^leaf(x)
return Q
```

Algorithm 2 PfastreXML-TRAIN($\{\mathbf{x}_i,\mathbf{y}_i\}_{i=1}^N,\mathbf{p},T$)

```
Require:
    {x_i, y_i}_{i=1}^N: Training set
    p: Propensities
    T: Number of trees

    for i = 1,...,N do
        for l = 1,...,L do
            y_il^p = y_il / p_il
        end for
    end for
    {T_1,..,T_T} = FastXML-TRAIN({x_i, y_i^p}_{i=1}^N, T)
        # Call Algorithm 1 in (?)
    for l = 1,..,L do
        μ_l = (Σ_{i=1}^N y_il x_i) / (Σ_{i=1}^N y_il)
    end for
    return {T_1,..,T_T}, {μ_1,..,μ_L}
```

Algorithm 3 PfastreXML-PREDICT($\{\mathcal{T}_1..\mathcal{T}_T\},\{\boldsymbol{\mu}_1..\boldsymbol{\mu}_L\},\mathbf{x},\alpha,\gamma$)

```
Q = FastXML-PREDICT({T_1,..T_T}, x)
P = 0
for l ∈ {l' : Q_l' > 0} do
    P_l = 1 / (1 + exp(γ/2 ||x − μ_l||^2))
    s_l = α log(Q_l) + (1 − α) log(P_l)
end for
r = rank_k (s)
return r,s
```

2. PfastreXML DERIVATIONS

Let N, D, L be the number of training points, features and labels respectively in the training set. Let $\mathbf{x}_i \in \mathcal{R}^D, \mathbf{y}_i \in \{0,1\}^L, \mathbf{y}_i^* \in \{0,1\}^L$ denote the feature vector; incomplete, observed label vector; and complete, unobserved label vector respectively of the ith point.

2.1 Tail label classifiers

We model the decision boundary for each label as a compact hyperspherical surface. Next, we assume conditional independence of labels given a feature vector, thus simplifying the parameter estimation problem into L independent and much smaller maximum likelihood estimation (MLE) problems. Finally, we assume $y_{il} \perp\!\!\!\perp \mathbf{x}_i | y_{il}^*$ and the previously stated hyperspherical models to derive the final expressions for MLE.

Maximum likelihood estimation:

Let $\{\boldsymbol{\mu}_j\} = \{\boldsymbol{\mu}_1,..,\boldsymbol{\mu}_L\}$ be the parameters of our model, whose values need to be estimated.

The MLE objective can be stated and simplified as follows:

$$\begin{aligned}\{\boldsymbol{\mu}_j^*\} &= \arg\max_{\{\boldsymbol{\mu}_j\}} \prod_{i=1}^{N} P(\mathbf{y}_i|\mathbf{x}_i;\{\boldsymbol{\mu}_j\})\\ &= \arg\max_{\{\boldsymbol{\mu}_j\}} \prod_{i=1}^{N}\prod_{l=1}^{L} P(y_{il}|\mathbf{x}_i;\boldsymbol{\mu}_l)\\ \boldsymbol{\mu}_l^* &= \arg\max_{\boldsymbol{\mu}_l} \prod_{i=1}^{N} P(y_{il}|\mathbf{x}_i;\boldsymbol{\mu}_l)\quad \forall l\in\{1,..,L\}\end{aligned} \tag{38}$$

where, we have used the assumption of conditional independence over labels to arrive at L smaller and independent problems.

By marginalizing y_{il}^* from the joint distribution over y_{il}, y_{il}^*, we get the following:

$$\begin{aligned}P(y_{il}|\mathbf{x}_i;\boldsymbol{\mu}_l) &= \sum_{y_{il}^*=0}^{1} P(y_{il},y_{il}^*|\mathbf{x}_i;\boldsymbol{\mu}_l)\\ &= \sum_{y_{il}^*=0}^{1} P(y_{il}|y_{il}^*,\mathbf{x}_i)P(y_{il}^*|\mathbf{x}_i;\boldsymbol{\mu}_l)\quad (\because \text{chain rule})\\ &= \sum_{y_{il}^*=0}^{1} P(y_{il}|y_{il}^*)P(y_{il}^*|\mathbf{x}_i;\boldsymbol{\mu}_l)\quad (\because y_{il}\perp\!\!\!\perp\mathbf{x}_i|y_{il}^*)\end{aligned} \tag{39}$$

Let $p_{il} = P(y_{il}=1|y_{il}^*=1)$ denote the propensity of label l for point i. Due to one-sided label noise, $(y_{il}=1) \implies (y_{il}^*=1)$. Using these observations:

$$\begin{aligned}P(y_{il}|y_{il}^*) &= \mathbb{1}(y_{il}^*=1)\Big(p_{il}\mathbb{1}(y_{il}=1)+(1-p_{il})\mathbb{1}(y_{il}=0)\Big)\\ &\quad+\mathbb{1}(y_{il}^*=0)\Big(0\mathbb{1}(y_{il}=1)+1\mathbb{1}(y_{il}=0)\Big)\\ &= y_{il}^*\Big(p_{il}y_{il}+(1-p_{il})(1-y_{il})\Big)+(1-y_{il}^*)(1-y_{il})\\ &= (1-y_{il})+p_{il}y_{il}^*(2y_{il}-1)\end{aligned} \tag{40}$$

We learn compact hyperspherical decision boundaries for each label independently, according to:

$$P(y^*_{il}|\mathbf{x}_i;\boldsymbol{\mu}_i) = 1/(1+v_{il}^{2y^*_{il}-1}) \tag{41}$$
$$\text{where} \quad v_{il} = \beta e^{\frac{\gamma}{2}\|\mathbf{x}_i-\boldsymbol{\mu}_l\|^2}$$

Substituting the results 40, 41 into 39 followed by some simplification, we get:

$$P(y_{il}|\mathbf{x}_i;\boldsymbol{\mu}_l) = (1-y_{il}) + \frac{p_{il}(2y_{il}-1)}{1+v_{il}} \tag{42}$$

We use 42 in 38, and take logarithm of probabilities as follows:

$$\begin{aligned}
\boldsymbol{\mu}_l^* &= \arg\max_{\boldsymbol{\mu}_l} \sum_{i=1}^{N} \log\left(P(y_{il}|\mathbf{x}_i;\boldsymbol{\mu}_l)\right)\\
&= \arg\max_{\boldsymbol{\mu}_l} \sum_{i=1}^{N} \log\left((1-y_{il}) + \frac{p_{il}(2y_{il}-1)}{1+v_{il}}\right)\\
&= \arg\max_{\boldsymbol{\mu}_l} O_l\\
\text{where,}\ \ O_l &= \sum_{i=1}^{N} \log\left((1-y_{il}) + \frac{p_{il}(2y_{il}-1)}{1+v_{il}}\right) \qquad (43)
\end{aligned}$$

2.2 Optimization

43 can be solved by usual gradient descent techniques. In this section, we derive the expression for gradient of 43.

Taking derivative of O_l w.r.t $\boldsymbol{\mu}_l$:

$$\begin{aligned}
\nabla_{\boldsymbol{\mu}_l} O_l &= \sum_{i=1}^{N} \nabla_{\boldsymbol{\mu}_l} \log((1-y_{il})(1+v_{il}) + p_l(2y_{il}-1))\\
&\quad - \nabla_{\boldsymbol{\mu}_l} \log(1+v_{il})\\
&= \sum_{i=1}^{N} \left(\frac{1-y_{il}}{(1-y_{il})(1+v_{il}) + p_l(2y_{il}-1)} - \frac{1}{1+v_{il}}\right)\nabla_{\boldsymbol{\mu}_l} v_{il}\\
&= \sum_{i=1}^{N} \left(\frac{1-y_{il}}{(1-y_{il})(1+v_{il}) + p_l(2y_{il}-1)}\right.\\
&\qquad \left. - \frac{1}{1+v_{il}}\right)(-\gamma v_{il}(\mathbf{x}_i - \boldsymbol{\mu}_l)) \qquad (44)
\end{aligned}$$

Since the derivative at the optimum must vanish:

$$\nabla_{\boldsymbol{\mu}_l^*} O_l = \mathbf{0}$$
$$\sum_{i=1}^{N} \gamma u_{il}(\mathbf{x}_i - \boldsymbol{\mu}_l^*) = \mathbf{0}$$

where,

$$u_{il} = \left(\frac{1-y_{il}}{(1-y_{il})(1+v_{il}) + p_l(2y_{il}-1)} - \frac{1}{1+v_{il}}\right) v_{il} \tag{45}$$

$$\implies \quad \boldsymbol{\mu}_l^* = \frac{\sum_{i=1}^{N} u_{il}\mathbf{x}_i}{\sum_{i=1}^{N} u_{il}} \tag{46}$$

2.3 Approximation

Gradient descent techniques do not scale to millions of label, and hence in this section we present an approximate but much faster solution to 43. We assume the following:

$$\exists \Delta \in \mathcal{R}, \quad \|\mathbf{x}_i - \boldsymbol{\mu}_l\| \geq \Delta > 0 \quad \forall i \in \{1,..,N\} \tag{47}$$

and

$$\gamma \gg \frac{-2\log(\beta)}{\Delta^2} \tag{48}$$

Above assumptions imply that:

$$\begin{aligned}
&\gamma\|\mathbf{x}_i - \boldsymbol{\mu}_l\|^2 \geq \Delta^2\lambda\\
&\gg -2\log(\beta)
\end{aligned}$$

Table 1: In terms of vanilla nDCG@k (Nk) and Precision@k (Pk) also, PfastreXML outperforms all the competing methods, except on two datasets namely EUR-Lex and Wiki10-31K, where SLEEC outperforms PfastreXML.

(a) EUR-Lex $N = 15K, D = 5K, L = 4K$

Algorithm	N1(%)	N3(%)	N5(%)	P1(%)	P3(%)	P5(%)
Popularity	6.69	6.10	5.94	6.69	5.88	5.48
1-vs-All	79.89	**69.62**	63.04	79.89	**66.01**	53.80
SLEEC	**79.94**	69.40	**63.16**	**79.94**	65.84	**54.19**
LEML	63.40	53.56	48.47	63.40	50.35	41.28
WSABIE	68.55	58.44	53.03	68.55	55.11	45.12
CPLST	72.28	61.64	55.92	72.28	58.16	47.73
CS	58.52	48.67	40.79	58.52	45.51	32.47
ML-CSSP	62.09	51.63	47.11	62.09	48.39	40.11
FastXML	72.35	64.03	58.93	72.35	61.19	51.24
LPSR	76.37	66.63	60.61	76.37	63.36	52.03
PfastreXML	76.11	66.99	61.48	76.11	63.92	53.24

(b) AmazonCat-13K $N = 1.18M, D = 203K, L = 13K$

Algorithm	N1(%)	N3(%)	N5(%)	P1(%)	P3(%)	P5(%)
Popularity	29.88	23.54	22.57	29.88	18.78	14.86
SLEEC	90.53	84.96	82.77	90.53	76.33	61.52
FastXML	**93.05**	87.02	85.11	**93.05**	78.16	63.37
PfastreXML	93.01	**87.03**	**85.14**	93.01	**78.19**	**63.42**

(c) Wiki10-31K $N = 14K, D = 101K, L = 31K$

Algorithm	N1(%)	N3(%)	N5(%)	P1(%)	P3(%)	P5(%)
Popularity	18.18	15.77	14.31	18.18	15.13	13.29
SLEEC	**80.18**	**67.84**	**59.60**	**80.18**	**64.25**	**53.68**
FastXML	69.70	58.53	52.01	69.70	55.27	47.06
PfastreXML	71.71	61.78	55.57	71.71	58.92	50.98

(d) WikiLSHTC-325K $N = 1.78M, D = 1.62M, L = 325K$

Algorithm	N1(%)	N3(%)	N5(%)	P1(%)	P3(%)	P5(%)
Popularity	15.88	8.40	7.04	15.88	6.03	3.80
SLEEC	54.84	47.25	46.16	54.84	33.43	23.86
FastXML	49.88	45.30	44.81	49.88	33.15	24.47
PfastreXML	**57.24**	**50.98**	**50.49**	**57.24**	**36.58**	**26.85**

(e) Amazon-670K $N = 490K, D = 136K, L = 670K$

Algorithm	N1(%)	N3(%)	N5(%)	P1(%)	P3(%)	P5(%)
Popularity	0.28	0.27	0.25	0.28	0.27	0.23
SLEEC	34.61	32.71	31.57	34.61	30.88	28.27
FastXML	36.90	35.09	33.87	36.90	33.27	30.54
PfastreXML	**38.86**	**37.45**	**36.51**	**38.86**	**35.52**	**32.93**

(f) Ads-9M $N = 70.45M, D = 2.08M, L = 8.84M$

Algorithm	N1(%)	N3(%)	N5(%)	P1(%)	P3(%)	P5(%)
Popularity	0.05	0.08	0.09	0.05	0.09	0.12
FastXML	15.11	15.58	16.01	15.11	9.10	6.62
PfastreXML	**15.57**	**17.24**	**18.29**	**15.57**	**10.15**	**7.73**

$$\implies v_{il} = \beta \exp(\frac{\lambda}{2}\|\mathbf{x}_i - \boldsymbol{\mu}_l\|^2) \gg 1 \quad \forall i \in \{1, .., N\} \tag{49}$$

Using the above result, we can simplify u_{il} in 45:

$$\begin{aligned} y_{il} = 1 \implies u_{il} &= \frac{v_{il}}{1 + v_{il}} \\ &\sim 1 \text{ (from 49)} \\ y_{il} = 0 \implies u_{il} &= \frac{v_{il}}{1 + v_{il}} - \frac{v_{il}}{1 + v_{il} - p_l} \\ &\sim 1 - 1 = 0 \text{ (from 49)} \end{aligned}$$

Hence,

$$\begin{aligned} u_{il} &\sim y_{il} \\ \implies \boldsymbol{\mu}_l^* &\sim \frac{\sum_{i=1}^{N} y_{il}\mathbf{x}_i}{\sum_{i=1}^{N} y_{il}} \end{aligned}$$

3. RESULTS

In paper, PfastreXML is compared against several state-of-the-art and baseline algorithms using propensity scored nDCG@k and propensity scored Precision@k. Here Table 1 provides comparison in terms of vanilla nDCG@k and Precision@k. Except for two datasets, viz. EUR-Lex and Wiki10-31K, PfastreXML outperforms state-of-the-art SLEEC and FastXML on all other datasets. This shows that PfastreXML not only predicts significantly more number of tail labels as compared to other methods, it performs fairly well in terms of vanilla precision and nDCG too.