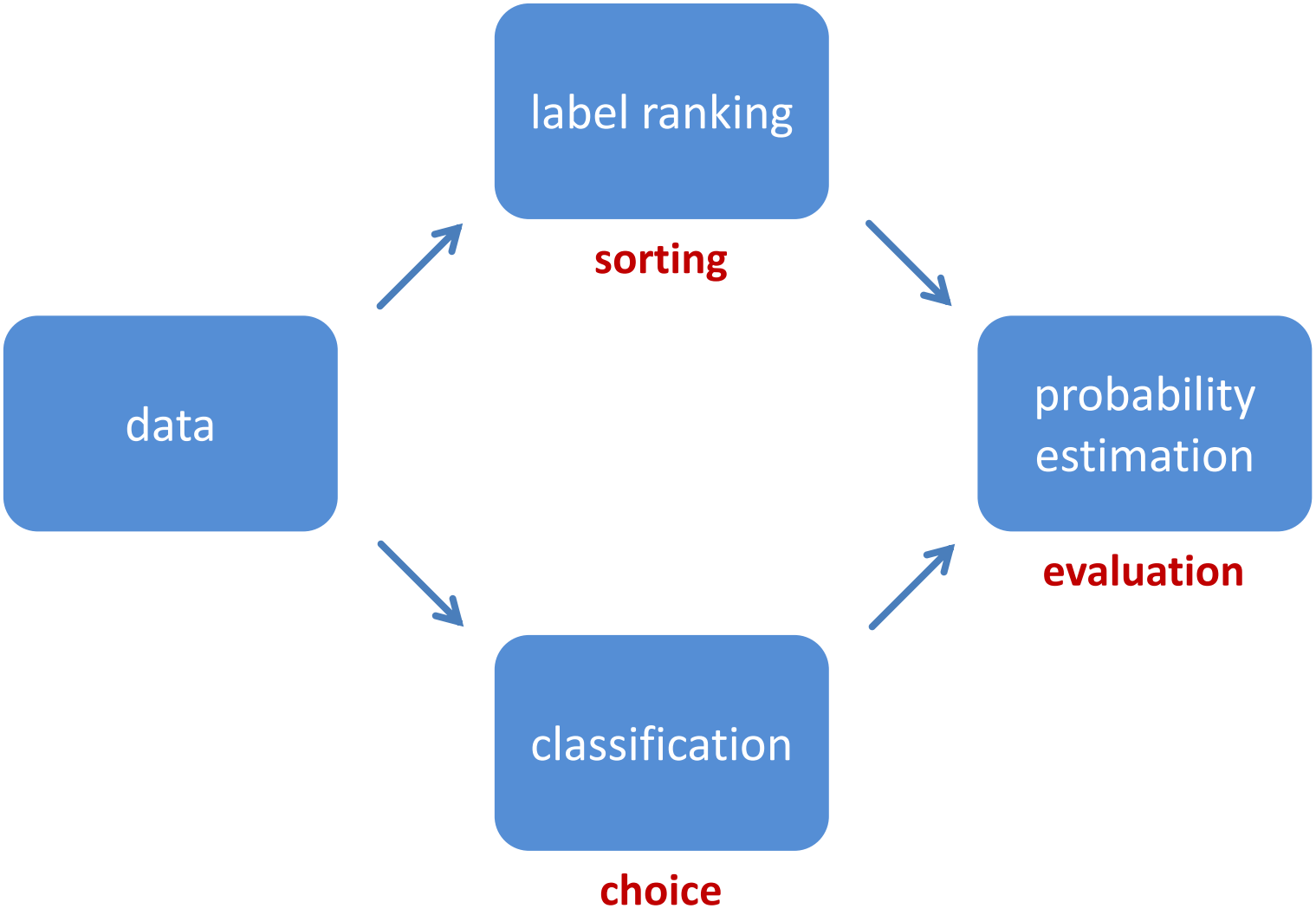
Probability Estimation for Multi-Class Classification based on Label Ranking

Weiwei Cheng, Eyke Hüllermeier
Mathematics and Computer Science
University of Marburg, Germany

ECMLPKDD 2012, Bristol


data
classification
probability estimation


label ranking
sorting
data
probability estimation
evaluation
classification
choice

Multi-Class Probability Estimation

Giving the standard multi-class classification setting with an instance space X and labels (i.e., classes) $\mathrm{Y} = \{y_1, \ldots, y_n\}$, learn a model that estimates the conditional probabilities of a given instance $\boldsymbol{x} \in \mathrm{X}$:

$$(p_1, \ldots, p_n) = (\mathrm{P}_\mathrm{Y}(y_1|\boldsymbol{x}), \ldots, \mathrm{P}_\mathrm{Y}(y_n|\boldsymbol{x}))$$

Notes

- Bayes decision can be taken to minimize any loss in expectation.
- A classification can be obtained by $\hat{y} = \mathrm{argmax}_{y_i \in \mathrm{Y}} \widehat{\mathrm{P}}(y_i|\boldsymbol{x})$ and is correct as long as the estimated probability is highest for the true class.

Pairwise Coupling

Pairwise Coupling is a binary decomposition technique, tackling multi-class problems with binary classifiers.

- At training time, a separate model M_{ij} is learned for each pair of labels (y_i, y_j).
- At prediction time, a query instance $\boldsymbol{x}$ is submitted to all models M_{ij} and

Bradley-Terry model

$$p_{ij} = M_{ij}(\boldsymbol{x}) = \mathrm{P}_{\mathrm{Y}}\left(y_i \mid \{y_i, y_j\}\right) = \frac{p_i}{p_i + p_j}.$$

- The unconditional probabilities p_i are derived from the conditional pairwise probabilities p_{ij}. (more details in Wu et al., JMLR 2004)

Pairwise Coupling

	p_{12}	p_{13}	p_{14}
p_{21}		p_{23}	p_{24}
p_{31}	p_{32}		p_{34}
p_{41}	p_{42}	p_{43}	

p_1
p_2
p_3
p_4

$$p_1 = (p_1+p_2)p_{12}$$
$$p_1 = (p_1+p_3)p_{13}$$
$$p_1 = (p_1+p_4)p_{14}$$
$$p_2 = (p_2+p_3)p_{23}$$
$$p_2 = (p_2+p_4)p_{24}$$
$$p_3 = (p_3+p_4)p_{34}$$

- Pairwise coupling tries to solve a system of equations, that is over-constrained.
- Values of p_{ij} can be inconsistent, because M_{ij} are trained independently.

From Bradley-Terry to Plackett-Luce Model

- Pairwise coupling, based on Bradley-Terry model (BT), is restricted to the comparison between pairs of classes, thus requiring a non-trivial combination step.

- Loss of information. Selecting y_i among the set of candidates is not the same as selecting y_i in the pairwise comparisons independently:

$$\frac{p_i}{p_1 + \cdots + p_n} \neq \prod_{j \neq i} \frac{p_i}{p_i + p_j}$$

- Plackett-Luce model is an extension of BT, which is able to model
  (1) 1-of-n choices: $p_i/(p_1 + \ \ldots \ + \ p_n)$
  (2) top-K rankings (a generalization of (1))
  (3) incomplete rankings (i.e., rankings of a subset of classes)

The Plackett-Luce Model

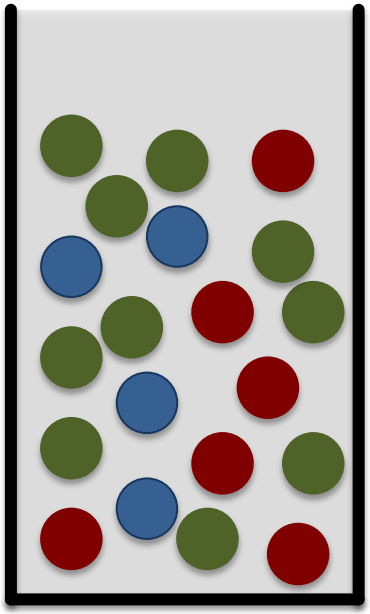
… is a **multistage** model specified by a vector $\boldsymbol{v} = (v_1, \ldots, v_n) \in \mathrm{R}_+^n$:

$$\mathbf{P}(\pi \mid \boldsymbol{v}) = \prod_{i=1}^{n} \frac{v_{\pi^{-1}(i)}}{v_{\pi^{-1}(i)} + v_{\pi^{-1}(i+1)} + \cdots + v_{\pi^{-1}(n)}}$$

where $\pi^{-1}(i)$ is the index of the label ranked at position i.

A ranking is produced by choosing labels one by one, with a probability proportional to their respective “skills”.

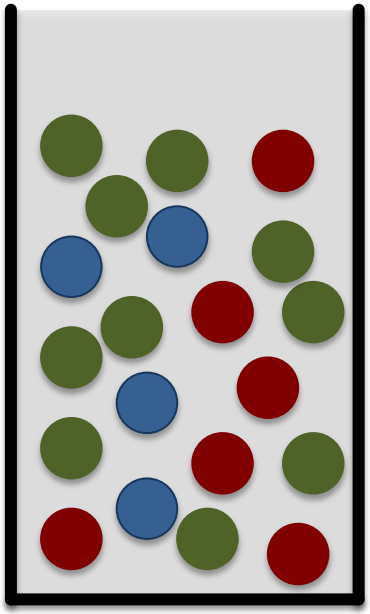
The Plackett-Luce Model



$v_{\bullet} = 10, \quad v_{\bullet} = 6, \quad v_{\bullet} = 4$

$\mathbf{P}(\ \bullet\ \bullet\ \bullet\ )$

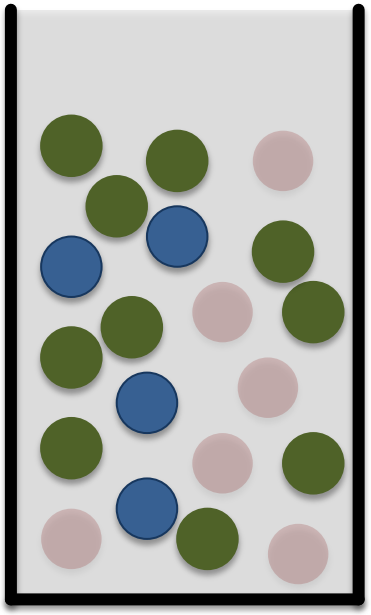
The Plackett-Luce Model



$v_{\bullet} = 10, \qquad v_{\bullet} = 6, \qquad v_{\bullet} = 4$

$\mathbf{P}(\ \bullet\ \bullet\ \bullet\ ) = \frac{6}{20}$

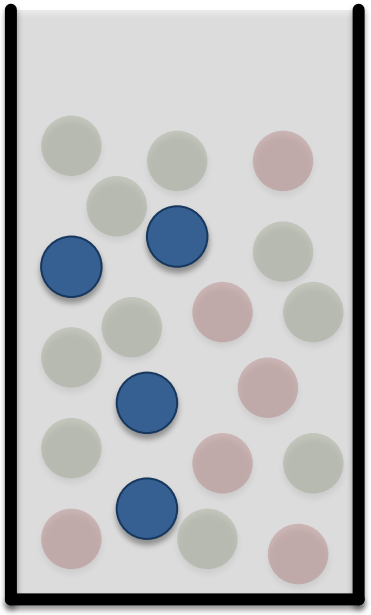
The Plackett-Luce Model



$v_{\bullet} = 10, \qquad v_{\bullet} = 6, \qquad v_{\bullet} = 4$

$\mathbf{P}(\ \bullet\ \bullet\ \bullet\ ) = \frac{6}{20} \times \frac{10}{14}$

The Plackett-Luce Model

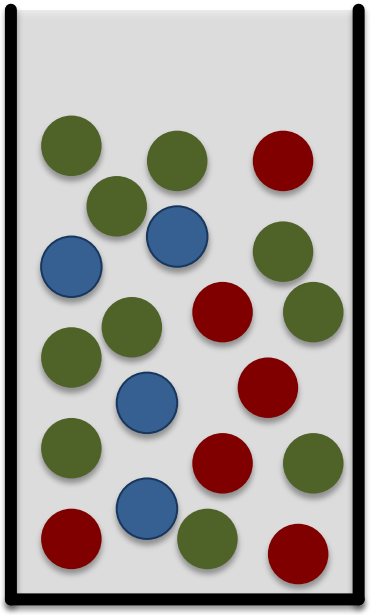


$v_{\bullet} = 10, \qquad v_{\bullet} = 6, \qquad v_{\bullet} = 4$

$$\mathbf{P}(\ \bullet\ \bullet\ \bullet\ ) = \frac{6}{20} \times \frac{10}{14} \times \frac{4}{4}$$

$$= \frac{3}{14}$$

The Plackett-Luce Model



$v_{\bullet} = 10, \qquad v_{\bullet} = 6, \qquad v_{\bullet} = 4$

$$\mathbf{P}(\ \bullet\ \bullet\ ) = \frac{6}{16} \times \frac{10}{10}$$

$$= \frac{3}{8}$$

Label Ranking

Given:

- a set of training instances $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m\} \subseteq \mathrm{X}$
- a set of labels $\mathrm{Y} = \{y_1, \ldots, y_n\}$
- for each training instance $\boldsymbol{x}_k$: an associated ranking of Y (possibly incomplete)

Find:

- a ranking function that maps each $\boldsymbol{x} \in \mathrm{X}$ to a ranking $\succ_{\boldsymbol{x}}$ of Y (permutation $\pi_{\boldsymbol{x}} \in \Omega$) and generalizes well in terms of a loss function on rankings

Label Ranking with Probabilistic Models

The output (label ranking) of an instance $\boldsymbol{x}$ is generated according to a distribution $\mathrm{P}_{\Omega}(\cdot \mid \boldsymbol{x})$.

π	$\mathrm{P}_{\Omega}(\cdot \mid \boldsymbol{x})$
$y_1 \succ y_2 \succ y_3$	0.05
$y_1 \succ y_3 \succ y_2$	0.30
$y_2 \succ y_1 \succ y_3$	0.20
$y_2 \succ y_3 \succ y_1$	0.20
$y_3 \succ y_1 \succ y_2$	0.25
$y_3 \succ y_2 \succ y_1$	0

Label Ranking based on PL

[Cheng et al., ICML 2010]

Recall the PL model:

$$\mathbf{P}(\pi \mid \boldsymbol{v}) = \prod_{i=1}^{n} \frac{v_{\pi^{-1}(i)}}{v_{\pi^{-1}(i)} + v_{\pi^{-1}(i+1)} + \cdots + v_{\pi^{-1}(n)}}$$

We model the parameter v_i as a linear function of the features describing the instance:

$$v_i = \exp\left(\sum_{j=1}^{d} \alpha_j^{(i)} \cdot x_j\right), 1 \leq i \leq n, 1 \leq j \leq d$$

Maximum Likelihood Estimation

[Cheng et al., ICML 2010]

Given training data $D = \left\{\left(\boldsymbol{x}^{(k)}, \pi^{(k)}\right)\right\}_{k=1}^{m}$ with $\boldsymbol{x}^{(k)} = \left(x_1^{(k)}, \ldots, x_d^{(k)}\right)$, the log-likelihood function is

$$\mathbf{P}(D \mid \boldsymbol{\alpha}) = \sum_{k=1}^{m} \sum_{i=1}^{n_k} \left[\log v\left(\tilde{\pi}^{(k)}(i), k\right) - \log \sum_{j=i}^{n_k} v\left(\tilde{\pi}^{(k)}(j), k\right) \right]$$

where $\tilde{\pi}(i) = \pi^{-1}(i)$ is the index of the label ranked at position i, n_k is the number of labels in the ranking $\pi^{(k)}$, and

$$v(i,k) = \exp\left(\sum_{j=1}^{d} \alpha_j^{(i)} \cdot x_j^{(k)}\right).$$

It is convex!

PELARA: Probability Estimation via Label Ranking

- The training data consist of instance $\boldsymbol{x} \in \mathrm{X}$ together with label ranking of varying length.
- A label ranker M' is trained with the procedure outlined earlier.
- At prediction time, M' assigns a vector of PL parameters to each test instance $\boldsymbol{x}$:

$$M' : \boldsymbol{x} \mapsto \boldsymbol{v} = \boldsymbol{v}(\boldsymbol{x}) \in \mathrm{R}_+^n$$

- The probability estimate is obtained by normalization, i.e.

$$M(\boldsymbol{x}) = \boldsymbol{p}(\boldsymbol{x}) = (p_1(\boldsymbol{x}), \ldots, p_n(\boldsymbol{x})) \propto \boldsymbol{v}(\boldsymbol{x}),$$

such that $\|\boldsymbol{p}(\boldsymbol{x})\| = 1$.

The model M defines a probability estimator.

Comparison with Decomposition Schemes

- The estimation is solved in one go. No decomposition needed.
- Loss of information is avoided. No complicated aggregation is needned for recovering p_i from p_{ij}.
- In a one-vs-rest decomposition, a linear number of models are trained, but each individual model is much more complex. PELARA, on the other hand, trains these models simultaneously, without building negative meta-classes.
- It is capable to deal with top-K rankings, incomplete rankings, etc.

Experiments – Brier Score

$$L(\boldsymbol{p}, y) = \sum_{i=1\ldots n} (p_i - [\![y = y_i]\!])^2$$

data set	#inst.	#att.	#cls.	PC	PC-HT	OVR	PELARA
iris	150	4	3	0.044	0.044	0.087	0.043
glass	214	9	6	0.439	0.434	0.442	0.432
wine	178	13	3	0.044	0.044	0.037	0.044
vowel	528	10	11	0.246	0.241	0.555	0.389
vehicle	846	18	4	0.241	0.240	0.270	0.240
segment	2310	19	7	0.060	0.070	0.134	0.068
dna	2000	180	3	0.140	0.141	0.124	0.157
pendigits	7494	16	10	0.028	0.043	0.094	0.053
poker	25010	10	10	0.566	0.566	0.567	0.565
satimage	4435	36	6	0.189	0.190	0.246	0.198
svmguide4	300	10	6	0.642	0.716	0.715	0.737
svmguide2	391	20	3	0.275	0.259	0.277	0.266
letter	15000	16	26	0.228	0.291	0.473	0.336
shuttle	43500	9	7	0.068	0.067	0.135	0.061

The two-tailed sign test at significant level $\alpha = 0.05$:
PELARA ≈ PC ≈ PC-HT ≻ OVR

Experiments – Run Time

data set	PC	PC-HT	OVR	PELARA
iris	0.19(1.63x)	0.23(2.00x)	0.13(1.14x)	0.12(1x)
glass	2.37(1.73x)	2.18(1.59x)	1.75(1.28x)	1.37(1x)
wine	0.24(1.88x)	0.35(2.70x)	0.33(2.51x)	0.13(1x)
vowel	6.08(1.04x)	6.99(1.19x)	0.74(0.13x)	5.86(1x)
vehicle	7.37(2.45x)	5.51(1.83x)	6.14(2.04x)	3.01(1x)
segment	18.80(1.77x)	14.73(1.39x)	17.84(1.68x)	10.63(1x)
dna	161.57(0.96x)	166.18(0.99x)	336.30(2.00x)	168.54(1x)
pendigits	25.87(1.30x)	39.52(1.99x)	46.09(2.32x)	19.91(1x)
poker	10.98(0.32x)	62.70(1.83x)	7.30(0.21x)	34.29(1x)
satimage	38.52(2.24x)	44.13(2.57x)	10.08(0.59x)	17.16(1x)
svmguide4	11.23(6.72x)	5.42(3.25x)	2.69(1.62x)	1.67(1x)
svmguide2	8.58(5.55x)	3.07(1.98x)	3.54(2.29x)	1.55(1x)
letter	179.76(0.33x)	264.75(0.49x)	25.13(0.05x)	538.56(1x)
shuttle	39.16(0.63x)	90.00(1.44x)	22.93(0.37x)	62.32(1x)

PELARA is the most efficient on average.

Conclusions

- We propose a new framework for class probability estimation via the probabilistic label ranking methods.
- The PELARA method, based on the Plackett-Luce model, requires no aggregation mechanism.
- While competitive in terms of prediction performance, this method is highly efficient.
- It is capable to deal with incomplete rankings, top-K rankings, etc. (*to be exploited in the future work*)