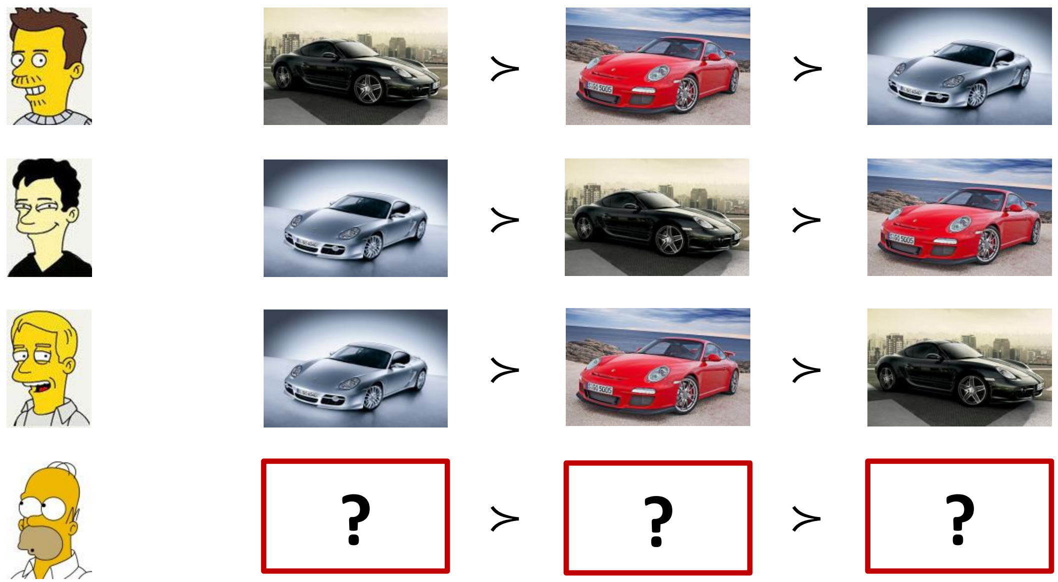
A Nearest Neighbor Approach to Label Ranking based on Generalized Labelwise Loss Minimization

Weiwei Cheng
Amazon
Germany

Eyke Hüllermeier
University of Marburg
Germany

Label Ranking – An Example



Instances are mapped to **total orders** over a fixed set of alternatives/labels.

Label Ranking: Training Data

TRAINING

X1	X2	X3	X4	Preferences
0.34	0	10	174	$A \succ B$, $C \succ D$
1.45	0	32	277	$B \succ C$
1.22	1	46	421	$B \succ D$, $A \succ D$, $C \succ D$, $A \succ C$
0.74	1	25	165	$C \succ A$, $C \succ D$, $A \succ B$
0.95	1	72	273	$B \succ D$, $A \succ D$
1.04	0	33	158	$D \succ A$, $A \succ B$, $C \succ B$, $A \succ C$

Instances are associated with pairwise preferences between labels.

... no demand for full rankings!

Label Ranking: Prediction

PREDICTION				A	B	C	D
0.92	1	81	382	?	?	?	?

new instance

ranking ?

Label Ranking: Prediction

PREDICTION				A	B	C	D
0.92	1	81	382	4	1	3	2

A ranking of all labels

new instance

$\hat{\pi}(i)$ = position of *i*-th label y_i

Label Ranking: Prediction

PREDICTION

0.92	1	81	382	4	1	3	2

A ranking of all labels

LOSS

GROUND TRUTH

0.92	1	81	382	2	1	3	4

SPEARMAN

$$D(\bar{\pi}, \hat{\pi}) = \sqrt{\sum_{i=1}^{K} \left(\bar{\pi}(i) - \hat{\pi}(j)\right)^2}$$

LOSS

$$\rho = 1 - \frac{6\, D^2(\bar{\pi}, \hat{\pi})}{K(K^2 - 1)}$$

RANK CORRELATION

Label Ranking: Prediction

PREDICTION

0.92	1	81	382	4	1	3	2

A ranking of all labels

LOSS

GROUND TRUTH

0.92	1	81	382	2	1	3	4

KENDALL

$$D(\bar{\pi}, \hat{\pi}) = \sum_{1 \leq i < j \leq K} [\![\left(\bar{\pi}(i) - \bar{\pi}(j)\right) \cdot \left(\hat{\pi}(i) - \hat{\pi}(j)\right) < 0]\!]$$

LOSS

$$\tau = 1 - \frac{4\, D(\bar{\pi}, \hat{\pi})}{K(K-1)}$$

RANK CORRELATION

Label Ranking: A Formal Setting

To learn a label ranker $\mathcal{M}^* : \mathbb{X} \to \mathbb{S}_K$, such that

$$\mathcal{M}^* \in \operatorname*{argmin}_{\mathcal{M} \in \mathbf{M}} \int_{\mathbb{X} \times \mathbb{S}_K} D(\mathcal{M}(x), \bar{\pi}) \, d\mathbf{P}(x, \bar{\pi})$$

NOTE In the training data, a ranking π can be incomplete, i.e., $y_{\sigma(1)} \succ y_{\sigma(2)} \succ \cdots \succ y_{\sigma(J)}$, where $J < K$ and $\{\sigma(1) \ldots \sigma(J)\} \subset \{1 \ldots K\}$. We denote, for example, the ranking $y_2 \succ y_1 \succ y_5$ as $\pi = (2,1,0,0,3)$.

Pairwise and Labelwise Decomposition

Pairwise decomposition

- e.g., [Hüllermeier et al., AI 08]
- **CON** quadratic number of models, higher computational cost
- **CON** non-trivial aggregation step
- **PRO** higher accuracy

Labelwise decomposition

- e.g., [Dekel et al., NIPS 03], [Cheng et al., ICML 10] and THIS WORK
- **PRO** linear number of models, lower computational cost
- **PRO** trivial or no aggregation step
- **CON** lower accuracy

Our Method LWD

- A meta-learning technique for label ranking directly uses the ranks of labels.

- When training data $\mathbb{D}$ consist of complete training information, we learn a model $\mathcal{M}_k : \mathbb{X} \to \{1 \dots K\}$ on the data

$$\mathbb{D}_k = \{ (x_n, r_n) \mid (x_n, \bar{\pi}_n) \in \mathbb{D}, r_n = \bar{\pi}_n(k) \}.$$

- Since the ranks have a natural order, it leads to K ordinal classification problems.

Our Method LWD cont.

- When training data $\mathbb{D}$ consist of incomplete training information, the previous setup is not directly applicable.

- Nevertheless, we can derive some information about the rank $\bar{\pi}(k)$:
  - **IF** $|\pi| = J$ and $\pi(k) = r > 0$, **THEN** $\bar{\pi}(k) \in \{r, r+1, \dots, r+K-J\}$.
  - If $\pi(k) = 0$, we can only derive $\bar{\pi}(k) \in \{1, \dots, K\}$.

- More information under additional assumptions. For example, if π is known to be the top of $\bar{\pi}$, then

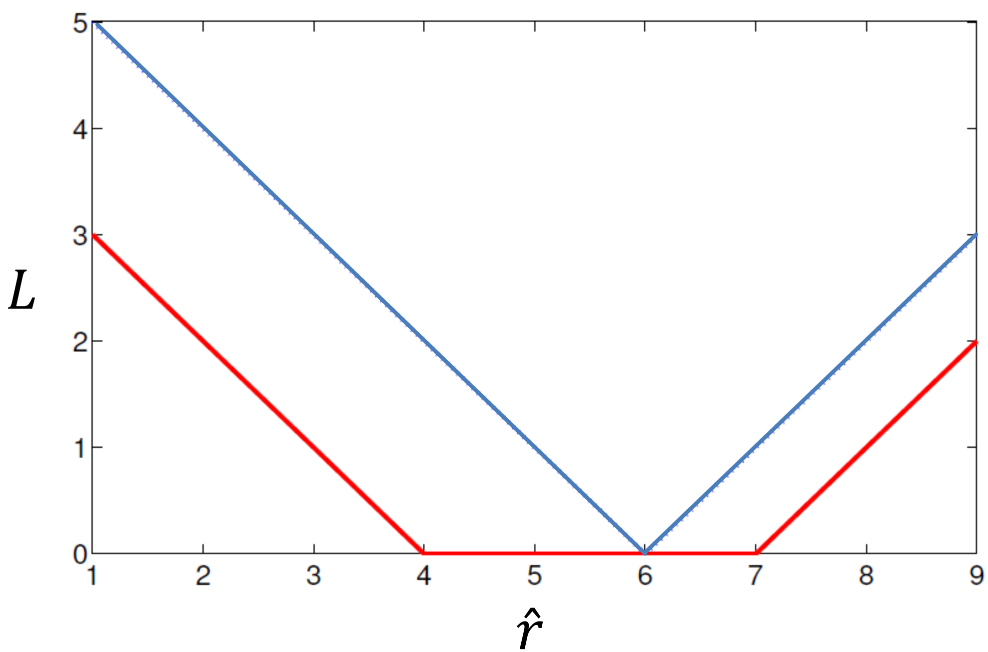
$$\begin{cases} \bar{\pi}(k) = \pi(k) & \text{if } \pi(k) > 0 \\ \bar{\pi}(k) \in \{J+1, \dots, K\} & \text{if } \pi(k) = 0 \end{cases}$$

A Generalized Loss Function

We make use of a generlized loss function, which compare a point predition with a set of possible "true" values:

$$L(\hat{r}, R) = \min_{r \in R} l(\hat{r}, r) \,,$$

where $R \in \{1 \dots K\}$ is a set of ranks, $\hat{r}$ is the predicted rank, and $l: \{1 \dots K\}^2 \to \mathbb{R}$ is the loss between predicted and true ranks.



In the figure on the left:

- $l(\hat{r}, r) = |\hat{r} - r|$
- blue line: $R = \{6\}$
- red line: $R = \{4,5,6,7\}$

Generalized Nearest Neighbor Estimation

Given a query instance x, a prediction $\hat{\pi}$ is obtained by combining the (possibly incomplete) rankings $\pi_1, \dots, \pi_Q$ from the Q nearest neighbors of x in the training data $\mathbb{D}$. Considering a loss function D on $\mathbb{S}_K$ that is labelwise decomposable, the empirical risk of $\hat{\pi}$ is given by

$$\sum_{n=1}^{Q} D(\pi_n, \hat{\pi}) = \sum_{n=1}^{Q} \sum_{k=1}^{K} L(R_{k,n}, \hat{\pi}(k))$$

where $R_{k,n}$ is the set of ranks π_n assigned to label y_k.

This leads to a straightforward procedure. Namely, for each label y_k, we select the rank $r \in \{1 \dots K\}$ that minimize $\sum_{n=1}^{Q} L\left(R_{k,n}, r\right)$.

But since each rank can only be assigned once, the procedure above is not valid.

Generalized Nearest Neighbor Estimation cont.

The minimization of $\sum_{n=1}^{Q} D(\pi_n, \hat{\pi})$ requires the solution of an optimal assignment problem:

- Label y_k must be uniquely assigned to rank $r = \hat{\pi}(k) \in \{1 \ldots K\}$;
- Assigning y_k to rank r has a cost of $L_k(r)$;
- The goal is to minimize the sum of all assignment costs.

This optimal assignment problem can be solved with the Hungarian algorithm. Its complexity for solving the problem above is $\mathcal{O}(K^3)$.

By solving the optimal assignment problem, we find the prediction $\hat{\pi}$ that minimizes $\sum_{k=1}^{K} L_k(\pi)$.

Experiments

- We empirically test our LWD framework with L1 loss, and compare it to another instance-based label ranking algorithm PL, which is based on the Plackett-Luce model for rankings. [Cheng et al., ICML 10]

- Both synthetic and real-world data are used.

data set	# instances	# attributes	# labels
authorship	841	70	4
glass	214	9	6
iris	150	4	3
pendigits	10992	16	10
segment	2310	18	7
vheicle	846	18	4
vowel	528	10	11
wine	178	13	3
sushi	5000	11	10
students	404	126	5

Kendall's tau on Synthetic Data

	complete ranking		30% missing labels		60% missing labels	
	LWD	PL	LWD	PL	LWD	PL
authorship	.933±.016	.936±.015	.925±.018	.833±.030	.891±.021	.601±.054
glass	.840±.075	.841±.067	.819±.078	.669±.064	.721±.072	.395±.068
iris	.960±.036	.960±.036	.932±.051	.896±.069	.876±.068	.787±.111
pendigits	.940±.002	.939±.002	.924±.002	.770±.004	.709±.005	.434±.007
segment	.953±.006	.950±.005	.914±.009	.710±.013	.624±.020	.381±.020
vehicle	.853±.031	.859±.028	.836±.032	.753±.032	.767±.037	.520±.050
vowel	.876±.021	.851±.020	.821±.022	.612±.027	.536±.034	.327±.033
wine	.938±.050	.947±.047	.933±.054	.919±.059	.921±.062	.863±.094
authorship	.933±.016	.936±.015	.932±.017	.927±.017	.923±.015	.886±.022
glass	.840±.075	.841±.067	.838±.074	.809±.066	.815±.075	.675±.069
iris	.960±.036	.960±.036	.956±.036	.926±.051	.932±.048	.868±.070
pendigits	.940±.002	.939±.002	.933±.002	.918±.002	.837±.004	.794±.004
segment	.953±.006	.950±.005	.943±.005	.874±.008	.844±.010	.674±.015
vehicle	.853±.031	.859±.028	.851±.033	.838±.030	.818±.032	.765±.035
vowel	.876±.021	.851±.020	.867±.021	.785±.020	.800±.021	.588±.024
wine	.938±.050	.947±.047	.936±.049	.926±.061	.930±.059	.907±.066

above: labels missing at random
bottom: top-rank setting

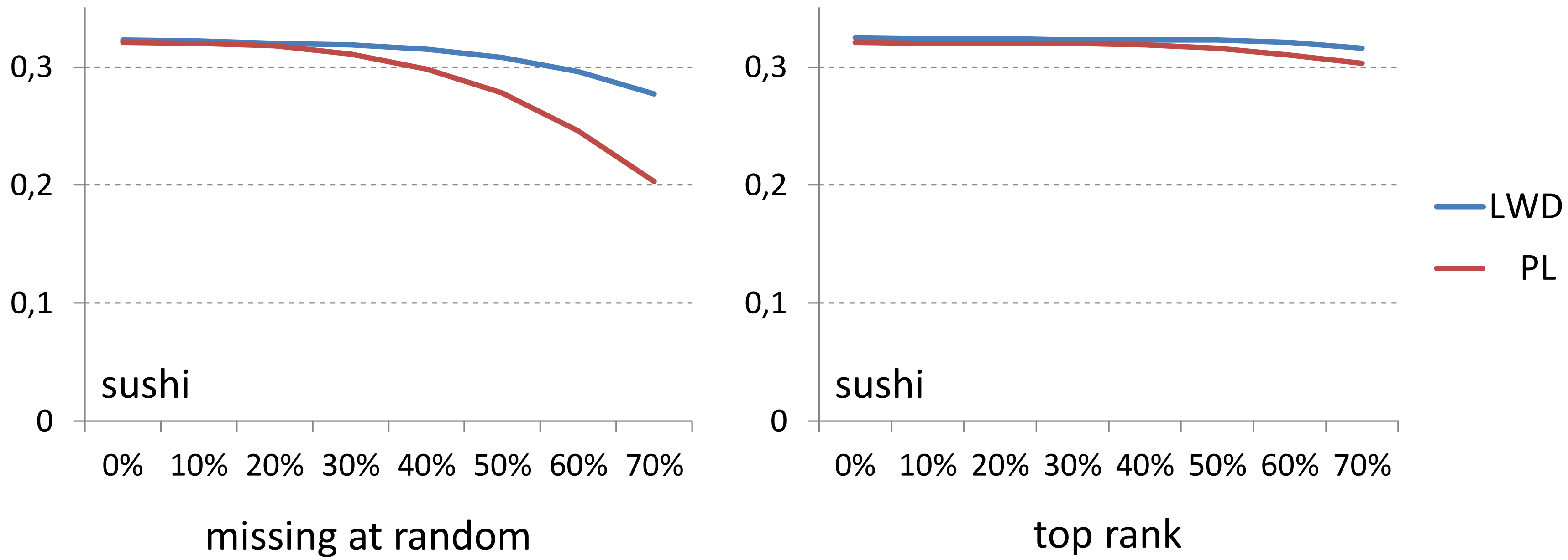
Kendall's tau on Real-World Data

sushi	0%	10%	20%	30%	40%	50%	60%	70%
LWD	.323±.012	.322±.011	.320±.011	.319±.010	.315±.011	.308±.011	.296±.011	.277±.010
PL	.321±.010	.320±.010	.318±.010	.311±.010	.298±.011	.278±.010	.246±.010	.203±.012
LWD	.325±.012	.324±.011	.324±.011	.323±.011	.323±.011	.323±.011	.321±.011	.316±.011
PL	.321±.010	.320±.010	.320±.011	.320±.011	.319±.010	.316±.010	.310±.010	.303±.011

students	0%	10%	20%	30%	40%	50%	60%	70%
LWD	.641±.051	.641±.051	.640±.050	.640±.051	.638±.052	.637±.051	.633±.054	.626±.055
PL	.386±.028	.384±.027	.382±.026	.377±.029	.365±.025	.350±.027	.327±.027	.274±.033
LWD	.641±.051	.641±.051	.641±.051	.641±.051	.640±.051	.640±.052	.638±.050	.628±.052
PL	.386±.028	.385±.028	.386±.028	.385±.027	.383±.029	.379±.026	.377±.026	.371±.028

above: labels missing at random
bottom: top-rank setting

Kendall's tau on Real-World Data cont.



- Top rank setting contains more information than missing at random setting.
- LWD is very robust against missing labels.

Summary

- We introduce labelwise decomposition as a new meta-learning technique for label ranking.
- It is realized for the specific case of nearest neighbor estimation.
- This approach is based on *absolute* preference informaiton in the form of ranks.
- The task of risk minimization is formulized as an optimal assignment problem.
- Empircal results indicate a very strong performance in the case of missing label information.