Schedae Informaticae Vol. 25 (2016): 73–83
doi: 10.4467/20838476SI.16.006.6187

tfml 2017
theoretical foundations
of machine learning, Kraków

Pairwise versus Pointwise Ranking: A Case Study

Vitalik Melnikov[1], Pritha Gupta[1], Bernd Frick[2],
Daniel Kaimann[2], Eyke Hüllermeier[1]
[1]Department of Computer Science
[2]Faculty of Business Administration and Economics
Paderborn University
Warburger Str. 100, 33098 Paderborn
e-mail: {*melnikov,prithag,eyke*}*@mail.upb.de,* {*bernd.frick,daniel.kaimann*}*@upb.de*

Abstract. Object ranking is one of the most relevant problems in the realm of preference learning and ranking. It is mostly tackled by means of two different techniques, often referred to as pairwise and pointwise ranking. In this paper, we present a case study in which we systematically compare two representatives of these techniques, a method based on the reduction of ranking to binary classification and so-called expected rank regression (ERR). Our experiments are meant to complement existing studies in this field, especially previous evaluations of ERR. And indeed, our results are not fully in agreement with previous findings and partly support different conclusions.

Keywords: Preference learning, object ranking, linear regression, logistic regression, hotel rating, TripAdvisor

1. Introduction

Preference learning is an emerging subfield of machine learning that has received increasing attention in recent years [1]. Roughly speaking, the goal in preference learning is to induce preference models from observed data that reveals information

Received: 11 December 2016 / Accepted: 30 December 2016

about the preferences of an individual or a group of individuals in a direct or indirect way; these models are then used to predict the preferences in a new situation.

In general, a preference learning system is provided with a set of items (e.g., products) for which preferences are known, and the task is to learn a function that predicts preferences for a new set of items (e.g., new products not seen so far), or for the same set of items in a different context (e.g., the same products but for a different user). Frequently, the predicted preference relation is required to form a total order, in which case we also speak of a *ranking problem.* In fact, among the problems in the realm of preference learning, the task of "learning to rank" has probably received the most attention in the literature so far, and a number of different ranking problems have already been introduced. Based on the type of training data and the required predictions, Fürnkranz and Hüllermeier [1] distinguish between the problems of object ranking [2,3], label ranking [4–6] and instance ranking [7].

The focus of this paper is on object ranking. What we present is an empirical study in which we compare the two most common approaches to this problem: pairwise ranking and pointwise ranking, with the latter being represented by a method called *expected rank regression* [3,8,9]. Although we are not the first to conduct experiments of that kind, our study sheds new light on the comparison of these two techniques and helps to better understand their advantages and disadvantages.

The rest of the paper is organized as follows. In the next section, we recall the problem of object ranking as well as the techniques of pairwise and pointwise ranking. The ranking data used for the purpose of our case study is described in Section 3. The design of the experiments and the results obtained are then presented in Section 4, prior to concluding the paper in Section 5.

2. Object ranking

Consider a reference set of objects or items $\mathcal{X}$, and assume each item $\boldsymbol{x} \in \mathcal{X}$ to be described in terms of a feature vector; thus, an item is a vector $\boldsymbol{x} = (x_1, \ldots, x_d) \in \mathbb{R}^d$ and $\mathcal{X} \subseteq \mathbb{R}^d$. Training data consists of a set of rankings $\{O_1, \ldots, O_N\}$, where each ranking O_j is a total order of a subset of $n_j = |O_j|$ items $\boldsymbol{x}_{j_i} \in \mathbb{X}$:

$$O_j : \boldsymbol{x}_{j_1} \succ \boldsymbol{x}_{j_2} \succ \ldots \succ \boldsymbol{x}_{j_{n_j}} \tag{1}$$

The order relation $\succ$ is typically (though not necessarily) interpreted in terms of preferences, i.e., $\boldsymbol{x} \succ \boldsymbol{x}'$ suggests that $\boldsymbol{x}$ is preferred to $\boldsymbol{x}'$.

The goal in object ranking is to learn a *ranking function* that accepts any (query) subset $Q \subseteq \mathcal{X}$ of $n = |Q|$ items as input. As output, the function produces a ranking (total order) O of these items. This prediction is evaluated in terms of a suitable loss function or performance metric; a common choice is the Kendall τ correlation, which counts the number of item pairs $\boldsymbol{x}, \boldsymbol{x}' \in Q$ that are incorrectly ordered by O and normalizes this number (which is between 0 and $n(n-1)/2$) to the range $[-1, +1]$.

2.1. Representation and learning

The ranking function sought in object ranking is a complex mapping from $2^{\mathcal{X}}$ to the set of all total orders over subsets of $\mathcal{X}$. A first question, therefore, is how to represent a "ranking-valued" function of that kind, and a second one is how it can be learned efficiently.

As for the question of representation, a ranking function is typically implemented by means of a scoring function $U : \mathcal{X} \longrightarrow \mathbb{R}$, so that $\boldsymbol{x} \succ \boldsymbol{x}'$ if $U(\boldsymbol{x}) > U(\boldsymbol{x}')$ for all $\boldsymbol{x}, \boldsymbol{x}' \in \mathcal{X}$. In other words, a ranking-valued function is implicitly represented by a real-valued function. Obviously, U can be considered as a kind of utility function, and $U(\boldsymbol{x})$ as a latent utility degree assigned to an item $\boldsymbol{x}$. Seen from this point of view, the goal in object ranking is to learn a latent utility function on a reference set $\mathcal{X}$. In the following, we shall also refer to U itself as a ranking function.

The representation of a ranking function in terms of a real-valued (utility) function also suggests natural approaches to learning. In particular, two such approaches are prevailing in the literature. The first one reduces the original ranking problem to *regression*; as it seeks a model that assigns appropriate scores to individual items $\boldsymbol{x}$, it is referred to as the *pointwise* approach. The second idea is to reduce the problem to *binary classification*; here, the focus is on pairs of items, which is why the approach is called the *pairwise* approach.

2.2. Pairwise ranking

Given a ranking (1) as training information, the pairwise approach extracts all pairwise preferences $\boldsymbol{x}_{j_i} \succ \boldsymbol{x}_{j_k}$, $1 \leq i < k \leq n_j$, and considers these preferences as examples for a binary classification task. This approach is especially simple if U is a linear function of the form $U(\boldsymbol{x}) = \boldsymbol{w}^\top \boldsymbol{x}$. In this case, $U(\boldsymbol{x}) > U(\boldsymbol{x}')$ if $\boldsymbol{w}^\top \boldsymbol{x} > \boldsymbol{w}^\top \boldsymbol{x}'$, which is equivalent to $\boldsymbol{w}^\top \boldsymbol{z} > 0$ for $\boldsymbol{z} = \boldsymbol{x} - \boldsymbol{x}' \in \mathbb{R}^d$. Thus, from the point of view of binary classification (with a linear threshold model), $\boldsymbol{z}$ can be considered as a positive and $-\boldsymbol{z}$ as a negative example.

In principle, any binary classification algorithm can be applied to learn the weight vector $\boldsymbol{w}$ from set of examples produced in this way. In the case of logistic regression, the resulting model has a specifically nice interpretation. Given two items $\boldsymbol{x}$ and $\boldsymbol{x}'$, the model produces a probability for the preference $\boldsymbol{x} \succ \boldsymbol{x}'$ and a complementary probability for $\boldsymbol{x}' \succ \boldsymbol{x}$. The former corresponds to the probability of a positive label $y = +1$ for the instance $\boldsymbol{z} = \boldsymbol{x} - \boldsymbol{x}'$, i.e.,

$$\begin{aligned}\mathbf{P}(\boldsymbol{x} \succ \boldsymbol{x}') = \mathbf{P}(y = +1 \,|\, \boldsymbol{z}) &= \frac{1}{1 + \exp(-\boldsymbol{w}^\top(\boldsymbol{x} - \boldsymbol{x}'))} \\ &= \frac{\exp(U(\boldsymbol{x}))}{\exp(U(\boldsymbol{x})) + \exp(U(\boldsymbol{x}'))}\end{aligned}$$

Thus, observed preferences are supposed to follow the Bradley-Terry model of discrete choice [10]: Having to choose between two options $\boldsymbol{x}$ and $\boldsymbol{x}'$, the probability for deciding in favor of either of them is proportional to the (exponential of the) corresponding utility. The maximum likelihood estimator $\boldsymbol{w}$ then simply maximizes the probability of the observed preferences under this choice model.

2.3. Pointwise ranking

Pointwise ranking methods induce a (utility) function $U : \mathcal{X} \longrightarrow \mathbb{R}$ as well. To do so, however, they fit a regression function to training examples of the form $(\boldsymbol{x}_i, y_i)$. Here, an obvious question concerns the definition of the target values y_i. In the setting of object ranking as introduced above, only *relative* information about the preference of items $\boldsymbol{x}_i$ in comparisons to others is given, but no *absolute* evaluations that could immediately be associated with a single $\boldsymbol{x}_i$.

Obviously, a reasonable target y_i for an item $\boldsymbol{x}_i$ is its (relative) position in $\mathcal{X}$, i.e., $y_i = \#\{\boldsymbol{x} \in \mathcal{X} \,|\, \boldsymbol{x}_i \succ \boldsymbol{x}\}$,[1] because sorting according to these scores yields perfect ranking performance. Again, however, since only rankings O_j of *subsets* of $\mathcal{X}$ are observed, these scores are not part of the training data.

In the method of expected rank regression (ERR), the scores are therefore approximated in terms of their expectation [3, 8, 9]. More specifically, given a ranking O_j of length n_j, an item $\boldsymbol{x}_i$ ranked on position r_i in O_j is assigned the score $y_i = r_i/(n_i+1)$. This is justified by taking an expectation over all (complete) rankings of $\mathcal{X}$ and assuming a uniform distribution. Roughly speaking, the items in O_j are assumed to be distributed uniformly among the whole spectrum of ranks.

2.4. Pairwise versus pointwise ranking

The pointwise approach solves a regression problem on $|O_1| + \ldots + |O_N|$ training examples in total; thus, if $|O_j| \approx K$, the size of the training data is of the order $\mathcal{O}(KN)$. The number of examples created by the pairwise approach for binary classification is of the order $\mathcal{O}(K^2N)$—although, at the cost of a slight loss of information, it could be reduced to $\mathcal{O}(KN)$ as well, namely by only extracting consecutive preferences $\boldsymbol{x}_{j_i} \succ \boldsymbol{x}_{j_{i+1}}$ from (1). In any case, linear regression is simpler and computationally less expensive than methods for binary classification, such as logistic regression. Thus, from the point of view of complexity, the pointwise approach seems to be preferable.

Also note that, while the pointwise approach leaves the rankings O_j intact, the pairwise approach splits a ranking O_j into pairwise comparisons. This necessarily comes with a loss of information, even when generating the full set of comparisons. This is because, statistically, the probability of observing the ranking as a whole is normally different from the probability of observing the set of pairwise preferences independently of each other.

That being said, the assignment of scores y_i in ERR is based on a rather strong and arguably unrealistic assumption. Moreover, these scores do not reflect an inherent

[1] Assuming $\mathcal{X}$ is finite.

property of an item $\boldsymbol{x}_i$, but instead depend on the context in which $\boldsymbol{x}_i$ is observed. Therefore, one may wonder whether predicting the y_i as a function of item-features is possible at all. Roughly speaking, the pointwise approach could be questioned because it adds information to the training data that is actually not present. This information is unreliable at best and misleading at worst. The pairwise approach, on the other hand, extracts only qualitative information. This information is weaker but indeed valid.

Finally, one may wonder whether a regression approach is suitable for fitting (relative) ranks, because ranks are only measured on an ordinal scale. In this regard, however, one should also note that the regression function U is not required to fit the data well in a numerical sense, i.e., in terms of the squared error loss. Instead, as it is only used for the purpose of ranking, any function that is comonotonic with the ranks is equally good.

Empirically, ERR has indeed been shown to be competitive and sometimes even superior to other ranking methods [3, 8, 9], albeit under experimental conditions that agree with the assumptions underlying this method. This paper is meant to complement these experiments by another case study, in which we systematically control certain properties of the training data in order to see to what extent they affect ERR. In particular, we are interested in scenarios that violate the assumptions of ERR. The hypotheses of this study are as follows:

H1: Due to the disputable way in which target values are produced for training in ERR, this method should in general be inferior to pairwise ranking.

H2: In particular, the shorter the training rankings O_j, the less accurate the approximation of target values in terms of expected ranks, and hence the worse the performance of ERR should be. Likewise, we suspect that the variance of the lengths $n_1, \ldots, n_N$ of the rankings in the training data has a negative influence.

H3: The performance of ERR will also drop due to a violation of the assumption of uniform sampling of positions.

3. TripAdvisor hotel dataset

Our case study deals with the ranking of hotels. The dataset used for performing the experiments was taken from the TripAdvisor website,[2] using a combination of web crawling and web scraping tools, on September 21 and 22, 2014. The dataset contains five rankings for hotels in five major German cities: Düsseldorf (110 hotels), Hamburg (170 hotels), Berlin (363 hotels), Frankfurt (149 hotels), and Munich (194 hotels). These rankings are referred to as *complete rankings* in the rest of this paper (they correspond to the reference set $\mathcal{X}$).

The position of each hotel in the ranking is determined by the so-called *popularity index*, which is computed by TripAdvisor based on the reviews for the hotels.

[2] www.tripadvisor.com

Although the true underlying computational formula (utility function) is unknown, several major factors contributing to this index are mentioned on the website.[3] These include the number of reviews, the age of reviews, and the overall quality of reviews. We used these attributes as features for each hotel in the dataset and complemented them by a set of additional attributes: distance to the city center (real number), number of hotel stars (ordinal), number of pictures on the TripAdvisor website (natural number), number of hotel rooms (natural number), average price per double room (real number), recommendation percent (percentage), number of reviews, five numerical features containing the number of ratings (from very good to very poor) given by the reviewers, and six real-valued features with average rating for different hotel attributes (location, sleep quality, room service, cost benefit, and cleanliness).

4. Experiments

We compare ERR with pairwise ranking based on logistic regression (LR). To guarantee a fair comparison, a linear utility function $U(\boldsymbol{x}) = \boldsymbol{w}^\top \boldsymbol{x}$ is used in both approaches. Moreover, the preprocessing of the data, including a standardization of all input attributes, was done in exactly the same way.

The general experimental design is as follows: We use one complete ranking (Berlin with 363 hotels) to generate training data in the form of incomplete rankings $O_1, \ldots, O_N$. LR and ERR are trained on this data as described in Sections 2.2. and 2.3., respectively. The models thus obtained are then evaluated on the remaining four cities: The four complete rankings are predicted, the Kendall correlation is determined for each of them individually, and finally the correlations are averaged. All experiments are repeated 100 times.

To test our hypotheses, the sampling procedure was controlled as follows:

- The lengths $n_j = |O_j|$ were sampled (independently) at random from a uniform distribution on $\{K-d, \ldots, K+d\}$. Thus, the average length of an observation is K, and the standard deviation is proportional to d. We chose $K \in \{5, 10, 20, 50, 100, 250\}$ and $d \in \{0, 1, 3, 4, 5, 7, 17, 47, 97\}$.[4]
- To make the results of different experiments comparable, regardless of the parameters K and d, we set $N = 1000/K$. Thus, the total number of hotels included in a sample is always 1000 (on average).
- After the length n_j of a ranking has been obtained, the ranking O_j itself is produced by randomly sampling n_j hotels in Berlin (and keeping their original order). For the sampling procedure, three different scenarios are considered:
  - *Uniform.* In this case, hotels are drawn uniformly at random (without replacement).

[3] https://www.tripadvisor.com/TripAdvisorInsights/n684/tripadvisor-popularity-ranking-key-factors-and-how-improve

[4] Since a length cannot be negative, not all (K, d) combinations are possible.

- *Top hotels.* This procedure has a bias in favor of hotels in the top of the list. First, we randomly pick one hotel from the first 50 in the complete ranking. This hotel is removed, and the next one is chosen among its neighbors, namely the three ones above and below. This procedure is continued until n_j hotels are collected.
- *Two groups.* The same sampling procedure as in the previous case is used (with a smaller neighborhood of 2 instead of 3), but the first hotel is randomly taken from first or the last 50 hotels.

The results of the experiments are summarized in Table 1 in terms of the average Kendall τ correlation and its standard deviation. Moreover, the following loss/gain of performance of ERR relative to LR is shown in the form of heatmaps in Figure 1:

$$\frac{\tau_{LR} - \tau_{ERR}}{\tau_{LR}} \tag{2}$$

The following conclusions can be drawn from these results:

- The pairwise approach (LR) consistently outperforms the pointwise approach (ERR) in all experiments. Moreover, pairwise ranking remains relatively stable across all settings, whereas the performance of ERR is much more sensitive toward the parameters. Thus, our hypothesis H1 is clearly confirmed.
- Our conjecture that ERR benefits from longer rankings is confirmed as well. Indeed, the performance of ERR clearly improves with increasing K. For the variance of the lengths, a clear trend is not visible. A problem here could be that the mean and variance of the length cannot be separated completely: Increasing d will always lead to producing a few rankings that are longer than K, which might be beneficial for ERR. Anyway, our conjecture H2 is confirmed only partially.
- Both approaches perform best in the case of *uniform* sampling. This was to be expected, since uniform sampling produces observations from the complete feature space. In the *top hotels* scenario, the pairwise approach remains rather stable, whereas ERR significantly drops in performance and seems to predict almost at random. A similar picture is obtained for the *two groups* scenario. These results clearly support our conjecture H3, namely that ERR is very sensitive toward deviations from the assumption of uniform sampling.

Table 1. Mean ± standard deviation of Kendall's tau for the *uniform* scenario (top), *top hotels* (middle), and *two groups* (bottom).

Approach	K/d	0	1	3	4	5	7	17	47	97
Pointwise	2	.248 ± .228	–	–	–	–	–	–	–	–
Pairwise	2	.832 ± .010	–	–	–	–	–	–	–	–
Pointwise	5	.390 ± .229	.302 ± .221	.154 ± .204	–	–	–	–	–	–
Pairwise	5	.840 ± .006	.840 ± .005	.839 ± .006	–	–	–	–	–	–
Pointwise	10	.377 ± .179	.432 ± .145	.536 ± .104	.555 ± .125	.567 ± .111	.615 ± .091	–	–	–
Pairwise	10	.841 ± .005	.841 ± .005	.840 ± .005	.840 ± .005	.839 ± .006	.839 ± .005	–	–	–
Pointwise	20	.707 ± .054	.703 ± .053	.700 ± .063	.712 ± .064	.713 ± .048	.719 ± .038	.743 ± .037	–	–
Pairwise	20	.841 ± .004	.841 ± .004	.841 ± .005	.841 ± .005	.840 ± .005	.841 ± .005	.840 ± .005	–	–
Pointwise	50	.786 ± .016	.785 ± .016	.784 ± .019	.786 ± .019	.785 ± .019	.785 ± .017	.788 ± .018	.794 ± .016	–
Pairwise	50	.841 ± .004	.841 ± .004	.841 ± .004	.841 ± .005	.841 ± .004	.842 ± .005	.841 ± .005	.841 ± .004	–
Pointwise	100	.803 ± .009	.804 ± .009	.802 ± .010	.805 ± .009	.804 ± .009	.803 ± .010	.801 ± .010	.802 ± .009	.806 ± .008
Pairwise	100	.841 ± .004	.841 ± .004	.841 ± .004	.841 ± .004	.841 ± .004	.842 ± .003	.841 ± .004	.842 ± .004	.842 ± .004
Pointwise	250	.812 ± .003	.811 ± .003	.812 ± .004	.812 ± .004	.811 ± .004	.812 ± .004	.811 ± .004	.811 ± .004	.812 ± .004
Pairwise	250	.843 ± .003	.842 ± .003	.843 ± .003	.843 ± .003	.842 ± .003	.843 ± .003	.843 ± .003	.842 ± .003	.843 ± .002

Approach	K/d	0	1	3	4	5	7	17	47	97
Pointwise	2	.085 ± .232	–	–	–	–	–	–	–	–
Pairwise	2	.749 ± .079	–	–	–	–	–	–	–	–
Pointwise	5	.088 ± .339	.010 ± .316	.021 ± .287	–	–	–	–	–	–
Pairwise	5	.722 ± .073	.735 ± .068	.734 ± .081	–	–	–	–	–	–
Pointwise	10	.037 ± .291	.028 ± .233	.083 ± .227	.137 ± .285	.067 ± .253	.115 ± .257	–	–	–
Pairwise	10	.704 ± .078	.708 ± .082	.711 ± .075	.675 ± .105	.700 ± .084	.676 ± .087	–	–	–
Pointwise	20	.321 ± .217	.332 ± .224	.327 ± .231	.299 ± .205	.367 ± .203	.304 ± .216	.412 ± .183	–	–
Pairwise	20	.670 ± .090	.673 ± .067	.676 ± .075	.673 ± .072	.683 ± .068	.707 ± .066	.707 ± .050	–	–
Pointwise	50	.587 ± .042	.589 ± .046	.586 ± .047	.585 ± .040	.586 ± .048	.581 ± .046	.579 ± .043	.596 ± .054	–
Pairwise	50	.777 ± .016	.781 ± .014	.776 ± .016	.776 ± .016	.773 ± .019	.774 ± .018	.764 ± .027	.790 ± .018	–
Pointwise	100	.675 ± .013	.677 ± .014	.674 ± .012	.675 ± .012	.676 ± .010	.678 ± .012	.674 ± .018	.654 ± .035	.690 ± .052
Pairwise	100	.762 ± .013	.763 ± .014	.759 ± .014	.765 ± .013	.767 ± .014	.767 ± .012	.782 ± .014	.766 ± .010	.781 ± .010
Pointwise	250	.792 ± .006	.792 ± .005	.794 ± .007	.793 ± .006	.794 ± .007	.794 ± .008	.794 ± .011	.789 ± .013	.782 ± .022
Pairwise	250	.811 ± .004	.811 ± .004	.812 ± .004	.811 ± .004	.811 ± .004	.811 ± .005	.814 ± .005	.816 ± .004	.822 ± .009

Approach	K/d	0	1	3	4	5	7	17	47	97
Pointwise	2	.006 ± .279	–	–	–	–	–	–	–	–
Pairwise	2	.402 ± .210	–	–	–	–	–	–	–	–
Pointwise	5	.015 ± .287	.008 ± .173	.030 ± .139	–	–	–	–	–	–
Pairwise	5	.533 ± .132	.569 ± .124	.543 ± .109	–	–	–	–	–	–
Pointwise	10	.019 ± .243	.004 ± .190	.021 ± .183	.010 ± .155	.023 ± .187	.018 ± .160	–	–	–
Pairwise	10	.595 ± .065	.575 ± .070	.587 ± .062	.588 ± .069	.589 ± .076	.602 ± .069	–	–	–
Pointwise	20	.172 ± .263	.061 ± .211	.076 ± .190	.030 ± .217	.045 ± .206	.048 ± .198	.021 ± .212	–	–
Pairwise	20	.658 ± .051	.646 ± .052	.655 ± .045	.670 ± .050	.663 ± .052	.667 ± .054	.696 ± .044	–	–
Pointwise	50	.350 ± .295	.260 ± .251	.160 ± .233	.153 ± .225	.177 ± .228	.116 ± .254	.110 ± .242	.158 ± .249	–
Pairwise	50	.761 ± .016	.762 ± .016	.760 ± .018	.764 ± .018	.764 ± .017	.762 ± .017	.765 ± .017	.789 ± .014	–
Pointwise	100	.482 ± .249	.423 ± .237	.406 ± .212	.420 ± .217	.342 ± .193	.385 ± .221	.259 ± .272	.270 ± .296	.408 ± .266
Pairwise	100	.803 ± .015	.804 ± .010	.805 ± .009	.805 ± .012	.805 ± .013	.806 ± .011	.807 ± .011	.819 ± .010	.834 ± .008
Pointwise	250	.802 ± .011	.803 ± .012	.802 ± .012	.802 ± .012	.802 ± .012	.805 ± .012	.799 ± .019	.798 ± .023	.789 ± .032
Pairwise	250	.836 ± .008	.836 ± .008	.836 ± .008	.836 ± .009	.836 ± .007	.837 ± .007	.838 ± .006	.839 ± .007	.838 ± .009

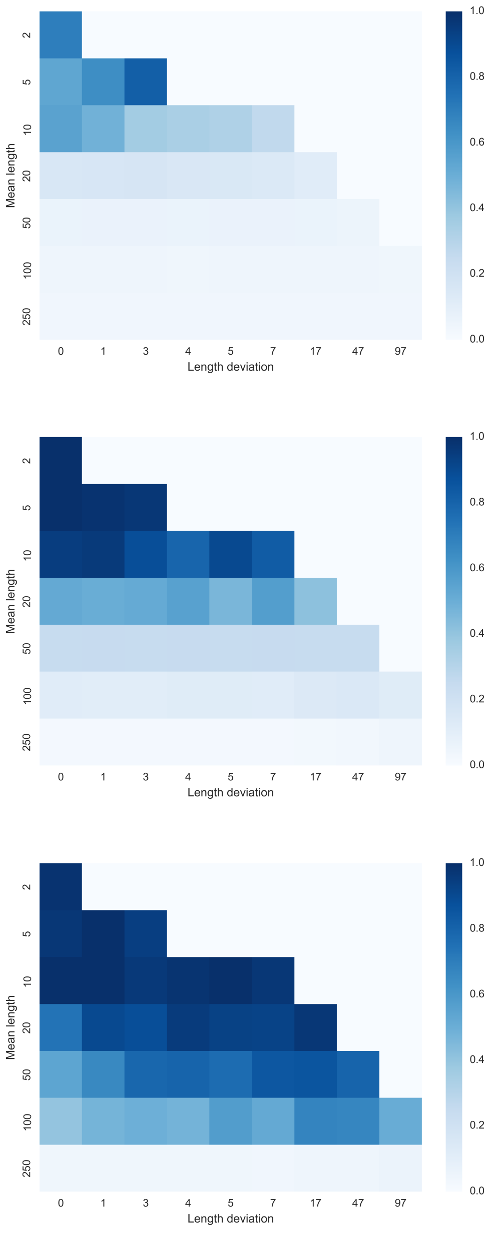

Figure 1. Relative improvement (2) for the *uniform* scenario (top), *top hotels* (middle), and *two groups* (bottom).

5. Conclusion

In summary, the results of our case study convey a picture that to some extent disagrees with previous experimental evaluations of expected rank regression, and which confirms our reservations regarding this approach. ERR seems to be competitive under ideal conditions, namely for sufficiently long rankings that are uniformly distributed across ranks. However, any deviation from these conditions leads to a significant drop in performance. As opposed to this, pairwise ranking shows a much more stable behavior and maintains a consistently strong performance across different experimental settings.

Needless to say, a single case study is necessarily limited in scope. Therefore, the conclusions drawn from the study should of course not be overgeneralized. Instead, we consider them as a starting point for further investigations that are needed to complete the picture and to gain a full understanding of the techniques of pairwise and pointwise ranking.

Acknowledgments

This work has been supported by the German Research Foundation (Deutsche Forschungsgesellschaft,DFG) within the Collaborative Research Centre "On-The-Fly Computing" (CRC 901).

6. References

[1] Fürnkranz J., Hüllermeier E., eds. *Preference Learning.* Springer, 2010.

[2] Cohen W., Schapire R., Singer Y., *Learning to order things.* Journal of Artificial Intelligence Research, 1999, 10 (1), pp. 243–270.

[3] Kamishima T., Kazawa H., Akaho S., A survey and empirical comparison of object ranking methods. In: Fürnkranz J., Hüllermeier E., eds.: *Preference Learning.* Springer 2010 pp. 181–202.

[4] Har-Peled, S., Roth, D., Zimak, D., *Constraint classification: a new approach to multiclass classification.* In: Cesa-Bianchi N., Numao M., Reischuk R., eds.: *Proceedings of the 13th International Conference on Algorithmic Learning Theory*, Springer, 2002, pp. 365–379.

[5] Cheng W., Hühn J., Hüllermeier E., *Decision tree and instance-based learning for label ranking.* In: *Proceedings of the 26th International Conference on Machine Learning*, Omnipress, 2009, pp. 161–168.

[6] Vembu S., Gärtner T., *Label ranking: a survey.* In: Fürnkranz J., Hüllermeier E., eds.: *Preference Learning.* Springer 2010 pp. 45–64.

[7] Fürnkranz J., Hüllermeier E., Vanderlooy S., *Binary decomposition methods for multipartite ranking.* In: *Proc. of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, Springer, 2009, pp. 359–374.

[8] Kamishima T., Kazawa H., Akaho S., *Supervised ordering – an empirical survey.* In: *Proc. ICDM, 5th IEEE International Conference on Data Mining*, Houston, Texas, 2005, pp. 673–676.

[9] Kamishima T., Akaho S., *Supervised ordering by regression combined with Thurstone's model.* Artificial Intelligence Review, 2006, 25 (3), pp. 231–246.

[10] Marden J., *Analyzing and Modeling Rank Data.* Chapman and Hall, London, New York, 1995.