From Rankings to Ratings: Rank Scoring *via* Active Learning*

Jack O'Neill[1], Sarah Jane Delany[2], and Brian Mac Namee[3]

[1] Dublin Institute of Technology, Ireland jack.oneill1@mydit.ie
[2] sarahjane.delany@dit.ie
[3] University College Dublin, Ireland brian.macnamee@ucd.ie

Abstract. In this paper we present RaScAL, an active learning approach to predicting real-valued scores for items given access to an oracle and knowledge of the overall item-ranking. In an experiment on six different datasets, we find that RaScAL consistently outperforms the state-of-the-art. The RaScAL algorithm represents one step within a proposed overall system of preference elicitations of scores *via* pairwise comparisons.

1 Introduction

Supervised machine learning for regression problems, in which models are trained to learn the relationship between descriptive features and some continuous-valued response, is an important sub-field of machine learning. Data rating is the process of asking human subjects (*raters*) to provide real-valued labels (*ratings* or *scores*) for input data (*artifacts*); these labels are essential to training predictive models. Machine learning models for regression problems are typically trained on datasets using labels elicited *via* data rating. This scenario is particularly common in the domain of recommender systems, where the artifacts being rated are, for example, items from an online store, or films; and the labels provided are scores on a continuous scale, often $[1\,..\,10]$ or $[1\,..\,5]$. Data rating is not confined to the domain of recommender systems, however, and has also been used to train models to detect valence and activation of emotions in speech [10], and to make medical diagnoses [5], among other applications.

When compiling a labelled dataset for training a machine learning model for a regression problem, researchers typically face two major difficulties: acquiring sufficient labels for the task at hand (*data scarcity*), and ensuring the quality of labels supplied (avoidance of *noisy* data). The former is particularly problematic in the area of recommender systems, where models are usually employed to evaluate very large product sets, which in turn require a large number of labels to train an accurate model [17]. The latter is problematic in any scenario in

* This work will be published as part of the book "Emerging Topics in Semantic Technologies. ISWC 2018 Satellite Events. E. Demidova, A.J. Zaveri, E. Simperl (Eds.), ISBN: 978-3-89838-736-1, 2018, AKA Verlag Berlin"

which the reliability of raters is not guaranteed, as incorrect labels have the capacity to reduce the accuracy of any predictive model they are used to train.

Data rating is traditionally optimised by addressing the problem from either of two angles. *Active learning for data rating* overcomes issues of data scarcity by identifying items whose labels are likely to contribute most to improving the performance of the model. By issuing queries only for these most important items, it can make the most of a labelling budget, and train accurate models using fewer labels. Elahi *et al.* [7] have compiled a comprehensive survey of techniques falling into this category. The problem of noisy data is typically addressed using *rater reliability estimation* [19]. Rater reliability estimation, as its name suggests, seeks to identify reliable *raters* whose scores are more likely to be accurate. By directing queries only to reliable raters, it minimises the error in its labels which in turn improves the overall accuracy of any model trained on this data.

While the techniques described above improve data rating by optimising either who is asked for labels, or the choice of items whose labels are requested, we aim to deliver further performance gains by improving the way we ask the question. This study forms part of a wider investigation into the viability of a data rating system which, instead of asking labellers to provide scores for individual items in isolation, requests pair-wise comparisons between items. These comparisons can then be used to build an overall ranking among items. By employing active learning techniques, we can learn to map these rankings to item scores. Figure 1 depicts a high-level overview of the process. This paper focuses on Step 3 in the diagram above; taking an overall ranking among items, and using active learning techniques to efficiently query for these items' scores.

In a previous study [13], we showed that items which are ranked comparatively show a higher inter-rater reliability than items which are rated individually. In order to complete the rating process, however, we need to infer concrete scores from the overall item-ranking, which remains a non-trivial problem. In this paper we present Rank Scoring *via* Active Learning (**RaScAL**), a system which combines isotonic regression modelling with active learning techniques to infer a set of scores given access to an oracle and an overall item-ranking.

The rest of this paper is structured as follows. Section 2 discusses related work in the field of active learning for data rating. Section 3 describes the RaScAL algorithm in detail. Section 4 outlines the datasets and describes the methods and evaluation metrics used in our experiment. Section 5 reports the results of the experiment, while Section 6 discusses conclusions and considers possible directions of future research which will build on these findings.

2 Related Work

Active learning techniques can be divided broadly into two sub-fields based on the type of labels sought: active learning for classification, in which labels take the form of a class identifier, and active learning for regression, which deals with questions having real-valued (numeric) labels. There has been a wide range of studies dealing with the former, (see Settles [15] for a comprehensive treatment

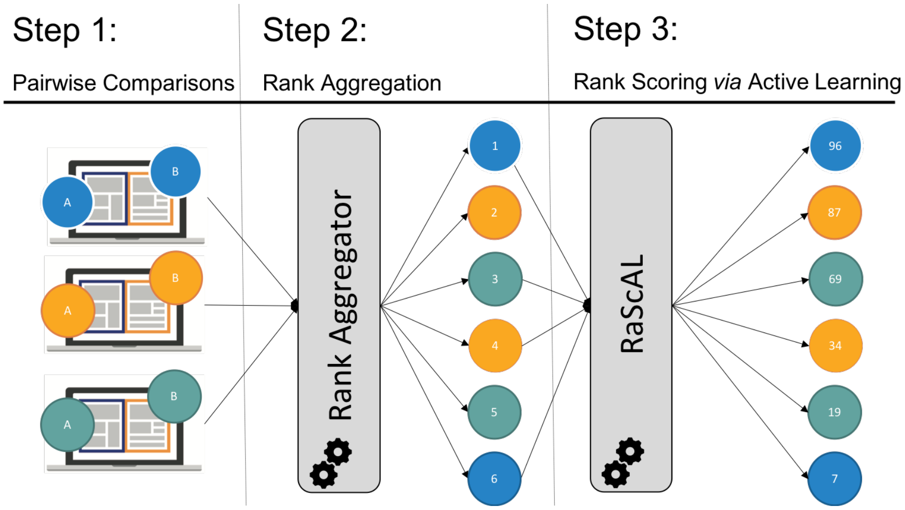


Fig. 1: Research Context. We begin by issuing queries for pairwise comparisons, and use this information to build up an overall item-ranking, which is then converted into a set of scores. This paper deals with Step 3 of the workflow; using active learning to predict scores given an overall ranking among items.

of the recent state-of-the art), but research on the latter is less common. Active learning for regression has its roots in the statistical field of Optimal Experimental Design [8], however, in recent years there has been increasing interest from researchers in the area of machine learning [12,18,4].

The idea that subjective judgements are prone to systematic rater biases first gained widespread acceptance through the work of psychologists Tversky and Kahnemann [20]. This idea has had consequences for any field of research in which judgement-based data is collected; and has been applied to sound quality evaluation [22], crowdsourcing [6], and collaborative filtering [1] among others.

The study described in this paper explores the possibility of using active learning techniques to efficiently infer a set of scores given an overall ranking among items and access to an oracle which can provide a score for any item on request. To the best of our knowledge, this particular problem has not previously been addressed in any great detail in the literature. However, the broader scenario of preference elicitation *via* rankings, rather than scores, is not new. Raykar *et al.* [14], were motivated by (among other reasons) the realisation that "*in many scenarios, it is more natural to obtain training data for pair-wise preference relations rather than the actual labels for individual examples*", to discard raw scores in datasets originally used for regression, and instead train a model to learn the ranking function over items.

The transformation of rankings to ratings has been successfully employed in an industry setting. Bockhorst *et al.* [3], reporting on their experience implementing a model to predict customer satisfaction scores, realised that self-reported scores showed high variability. They recognised that a more accurate model could be trained by collecting rankings from customers, rather than scores, and then transforming those rankings to real-valued scores using an isotonic regression. The work described in this paper extends the work of Bockhorst *et al.* by adding active learning to the rank transformation process, which, we hypothesize, can significantly increase the learning rate.

3 RaScAL

The RaScAL algorithm is an active learning approach to predicting real-valued scores for items given an overall item-ranking. Previously, we have shown that data collected *via* pairwise comparisons is more reliable than that collected through queries for absolute item scores. Pairwise comparisons allow us to build an overall ranking among items but do not allow us to infer scores. RaScAL enables us to bridge this gap. By issuing a small number of queries for absolute scores for selected items, we can make predictions for the remaining items, assigning scores in such a way that the rank ordering is preserved.

For example, consider three films, F_1, F_2 and F_3, with corresponding scores Sc_1, Sc_2 and Sc_3 where the rank order of the scores is known *i.e.* $Sc_1 \leq Sc_2 \leq Sc_3$. After issuing queries for the scores of F_1 and F_3, imagine we get $Sc_1 = 3$ and $Sc_3 = 5$ We then know that the score for F_2 will be between 3 and 5, inclusive. Technically, we achieve this by using the queried points to fit an isotonic regression [2], which we then use to predict the scores of the remaining items.

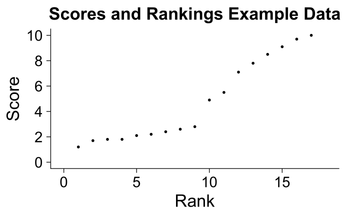
RaScAL differs from previous research in how it selects items to be queried. When faced with the problem of transforming rankings to a set of scores, Bockhorst *et al.* fit an isotonic regression model using training examples (scores) sampled uniformly from the set of labels [3]. For example, given 101 items and a labelling budget of 11 queries, the uniform sampling approach would query the first item, and every tenth item thereafter. This approach works well when the *distances* between scores are relatively uniform; however, if these scores are not uniformly distributed, this method is prone to underfit the data. RaScAL improves the robustness of this approach by selecting queries so as to minimise the expected error of the predicted scores. This is best illustrated with an example.

Table 2 (a) describes an artificial dataset which we will use to illustrate the RaScAL query selection strategy. This data is visualised in Figure 2 (b). We assume that the ranks of each item in the dataset are known before the labelling process begins, and the aim of the exercise is to accurately predict the scores of each item (which would be unknown) using as few queries as possible. We refer to each item in the dataset as I_x where x is the rank position of the item in question.

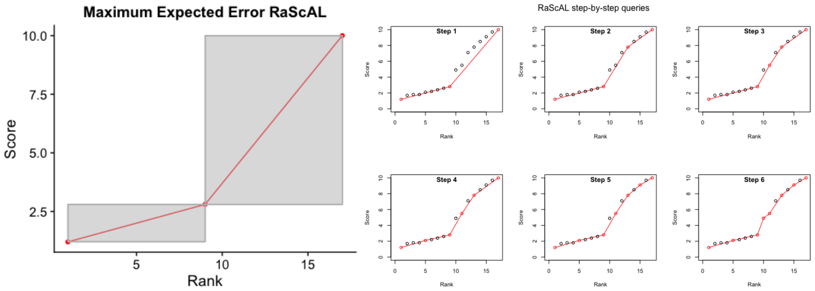
The first 3 queries in the RaScAL algorithm are always the same. We begin by establishing the upper and lower limits of the scores by querying for the lowest and highest ranked items, and then query for the item in middle of the

Rank	Score	Rank	Score
1	1.20	10	4.90
2	1.70	11	5.50
3	1.80	12	7.10
4	1.80	13	7.80
5	2.10	14	8.50
6	2.20	15	9.10
7	2.40	16	9.70
8	2.60	17	10.00

(a)



(b)



(c) (d)

Fig. 2: RaScAL illustrative example. (a) Example data values. (b) Example data visualised as scatter plot. (c) Calculating the next query for the RaScAL algorithm. The grey boxes represent the maximum possible error. (d) Step-by-step illustration of RaScAL query selection strategy

ranking[4]. In this case we issue a query for $\{I_1, I_9, I_{17}\}$. This query splits the data into two sequences of items. The first sequence, S_1, consists of the items $I_2 \ldots I_8$. Assuming that the oracle returns perfect scores, the potential scores for items in this sequence are bounded by the values of I_1 and I_9, meaning all scores fall within the range $\{1.2\,..\,2.8\}$. The second sequence, S_2, consists of the items $I_9 \ldots I_{17}$. The potential scores for items in this sequence are bounded by the values of I_9 and I_{17}, meaning all scores for this sequence fall within the range of $\{2.8\,..\,10\}$. By performing a simple linear interpolation between the labelled points, we fit an isotonic regression model which can then be used to predict the scores of the remaining items. The result of the first iteration of the RASCAL algorithm for this example are shown in Figure 2 (c). The red dots represent queried items; while the red line joining these dots depicts the fitted isotonic regression function which allows us to make predictions for each of the

[4] When there is an even number of items in the set, it is not possible to split it into two equally sized subsets, and the 'middle' rank must be rounded either up or down. Given that we have no prior knowledge of the distribution of scores there is no theoretical reason for preferring one over the other. In this study, however, we chose to round down when confronted with this problem

remaining items. The grey boxes show the bounds within which all remaining labels must fall. These bounds can be used to select the next items for which to query the oracle as they also bound the error of scores inferred using the isotonic regression.

If we predicted the value of 1.2 for each of I_2 to I_8 and each of the items I_2 to I_8 had a score of 2.8, we could expect a maximum error of $(2.8-1.2)$ for each of the unrated items. This expected error is approximated by the area of the grey box in Figure 2 (c). However, the isotonic regression diagonally bisects this box, reducing the maximum error by half. In the worst case scenario, the total error for S_1 (assuming all items have a score of 1.2, or all items have a score of 2.8) is $\frac{1}{2} \times (2.8-1.2) \times (9-1-1) = 5.2$. In the worst case scenario, the total error for S_2 (assuming all items have a score of 2.8, or all items have a score of 10) is $\frac{1}{2} \times (10-2.8) \times (17-9-1) = 25.2$. As S_2 has a greater potential error than S_1 we next query the oracle for the score for the item in the middle of S_2.

The calculation of maximum expected error can be formalised as:

$$\hat{E} = \frac{(Rank_j - Rank_i - 1) * (Y_i - Y_j)}{2} \tag{1}$$

where $\hat{E}$ is the maximum expected error for an unlabelled segment, $Rank_i$ and $Rank_j$ are the rank positions of the items bounding the segment, and Y_i and Y_j are the labelled scores for these items. The numerator represents the bounding box between labelled items i and j, corresponding to the shaded grey areas in Figure 2 (c). The isotonic regression bisects this rectangle, effectively halving the maximum expected error for the segment.

After querying the oracle for the score for the middle item in S_2 and bisecting this segment, we are left with three segments. We repeat the process, finding the segment with the maximum possible error and querying the item which bisects that segment, until no labelling budget remains. Algorithm 1 formalises the description of the RaScAL process. Figure 2 (d) shows the sequence of queries which would be made on the example data described above with a labelling budget of 8 queries. Figure 3 compares the isotonic functions fitted by uniform sampling *vs* that fitted by RaScAL, with a labelling budget of 4 queries. It is evident from these graphs that the RaScAL approach fits the data more closely.

4 Experimental Framework

In order to investigate the performance of RaScAL in accurately mapping rankings to scores we performed an experiment using both synthetically generated and real-world datasets. In Section 4.1 we describe the datasets used for this experiment, while Section 4.2 discusses the experimental evaluation process.

4.1 Datasets

The MovieLens 100k dataset [11] consists of 100,000 scores (on a scale of [1 . . 5]) for over 1,500 films. We aggregated all scores on a per-item basis, using the

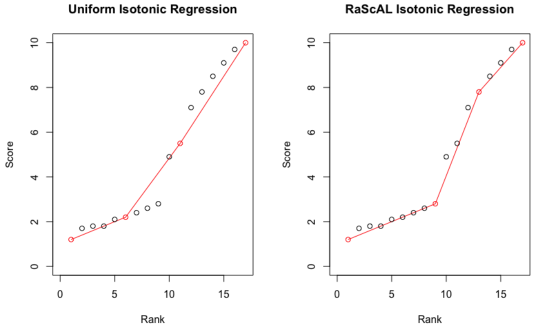


Fig. 3: Comparison of uniform sampling and RaScAL approaches to query selection.

Dataset	# Items	Scale	Mean	IQR	Min	Max
MovieLens	1,682	$[1\,..\,5]$	3.076	0.782	1	5
Jester	100	$[-10\,..\,10]$	0.824	2.239	-3.834	3.665
Boredom Videos	125	$[1\,..\,10]$	6.484	0.694	4.903	7.750
Book Crossing	675	$[1\,..\,10]$	7.401	3.000	1	10
Bi-Modal	100	$[0\,..\,100]$	49.670	48.821	0.077	96.850
Multi-Modal	100	$[0\,..\,100]$	42.149	68.974	1.927	99.164

Table 1: High-level distributional features of the datasets used in this evaluation

mean over all raters as the item's final score[5]. Individual scores were provided in integer format; however, after averaging scores many items ended up with a real-valued label.

The Jester dataset, originally provided by Ken Goldberg [9], is a dataset of scores for jokes provided by users on a $[-10, 10]$ scale. Unlike many collaborative filtering datasets, in which users provide scores on an integer scale, the jester dataset collected scores using a slider, allowing users to provide real-valued scores. Overall item scores were calculated as the mean score across all users.

The Boredom Videos corpus was gathered by Soleymani *et al.* [16] using the crowdsourcing platform *Amazon Mechanical Turk*[6]. Respondents were asked to rate videos on a scale of $[1\,..\,10]$ based on how boring they found them to be. As with the Jester dataset, overall items scores were calculated as the mean score across all users.

The Book Crossing dataset was collected by Ziegler *et al.* [21] from the Book Crossing online community. It contains a mixture of implicit and explicit scores. An implicit score indicates that a book has been read, while explicit scores are provided as a value on an integer scale of $[1\,..\,10]$. For this experiment, we only

[5] A superior algorithm which takes a probabilistic approach to aggregating scores has been proposed by Raykar *et al.* [14]. Although this approach has been shown to more accurately approximate the *true* rating; this aspect of rating elicitation is outside the scope of the current work as it would add unnecessary complexity to the experiment.

[6] `https://www.mturk.com`

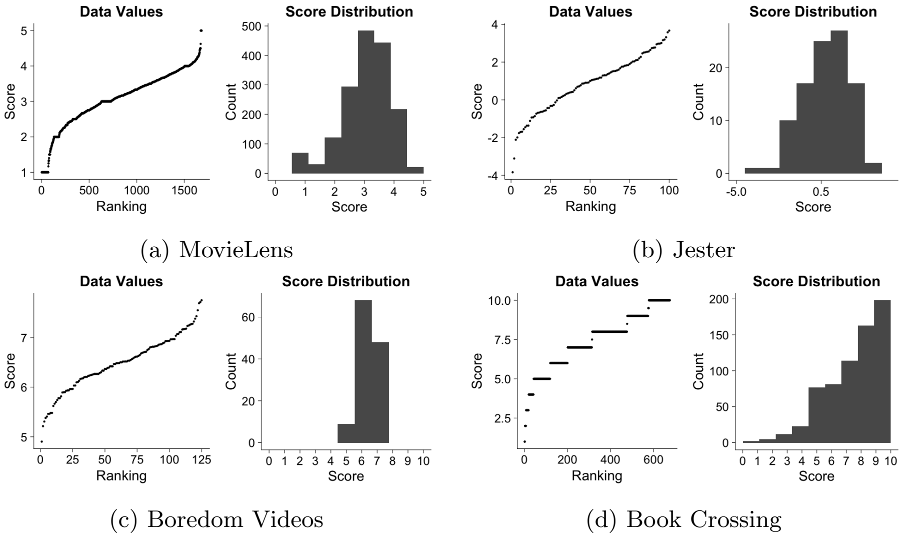
Algorithm 1 RaScAL algorithm

```
    struct Segment:
        property $first$
        property $last$
        property $max\_error$

        Segment(first, last, scores)
            $max\_error \leftarrow \textsc{max\_error}(first, last, scores[first], scores[last])$    ▷ see Equation 1

Input:
    $Items$    ▷ Rank ordered list of items
    $N$    ▷ The number of items in $Items$
Output:
    $Fitted\_Isotonic\_Regression$    ▷ A model which can predict scores for all items
Require:
    isotonic, a function which fits an isotonic regression given a set of items and corresponding scores
    sort, a function which sorts the segment vector by maximum expected error
    query, a function which requests a score for a given item from an oracle

$Scores \leftarrow [\ ]$
$Scores[0] \leftarrow \textsc{query}(Items[0])$
$Scores[N-1] \leftarrow \textsc{query}(Items[N-1])$
$new\_seg \leftarrow \textsc{segment}(0, N-1, Scores)$
$segments \leftarrow [new\_seg]$
repeat
    $segments.\textsc{sort}()$    ▷ Sort segments on $s.max\_error$ $descending$
    $s \leftarrow segments.\textsc{pop}()$
    $mid\_point \leftarrow s.first + (s.last - s.first)/2$
    $score \leftarrow \textsc{query}(Items[mid\_point])$
    $Scores[mid\_point] \leftarrow score$
    $seg\_low \leftarrow \textsc{segment}(s.first,\ mid\_point,\ Scores)$
    $seg\_high \leftarrow \textsc{segment}(mid\_point,\ s.last,\ Scores)$
    $segments.\textsc{append}([seg\_low,\ seg\_high])$
until Label budget exhausted OR all items ranked
return $\textsc{isotonic}(Items, Scores)$
```



(a) MovieLens

(b) Jester

(c) Boredom Videos

(d) Book Crossing

Fig. 4: Visual representation of the label values from real-world datasets

used explicit scores. We selected only the first 675 books, giving us 684 explicit scores in total. This dataset was unique in our experiment in that most items were rated by only one user. As scores were provided on an integer scale, this resulted in a large number of ties among items, as can be seen in Figure 4.

In addition to the real-world datasets described above, we created two artificial datasets using random sampling from known distributions. The Bi-Modal dataset consists of 100 scores in total. 50 scores were drawn from a normal distribution with mean 25 and standard deviation 10, with the remaining scores drawn from a normal distribution with mean 75 and standard deviation 10. The Multi-Modal dataset also consists of 100 scores, though these scores are drawn from 3 uniform distributions; 50 scores from a uniform distribution with range $[1 \ldots 20]$, 15 scores from a uniform distribution with range $[21 \ldots 70]$ and the remaining 35 scores drawn from a uniform distribution with range $[71 \ldots 100]$.

In their original formats, the target variable of each dataset is a numeric score. For each dataset we convert these scores to ranks. These ranks are then used as the input data for the RaScAL algorithm. Where ties were encountered, distinct ranks were assigned based on the order in which they occurred in the dataset. This means that an item with a rank value of 2 and an item with a rank value of 3 may have the same actual score.

Table 1 summarises the high-level distributional features of each of the datasets used, after pre-processing, where described above, was carried out. The distribution of scores for the Bi-Modal and Mult-iModal datasets are visualised in Figure 5. The distribution of labels for each of the real-world datasets are visualised in Figure 4.

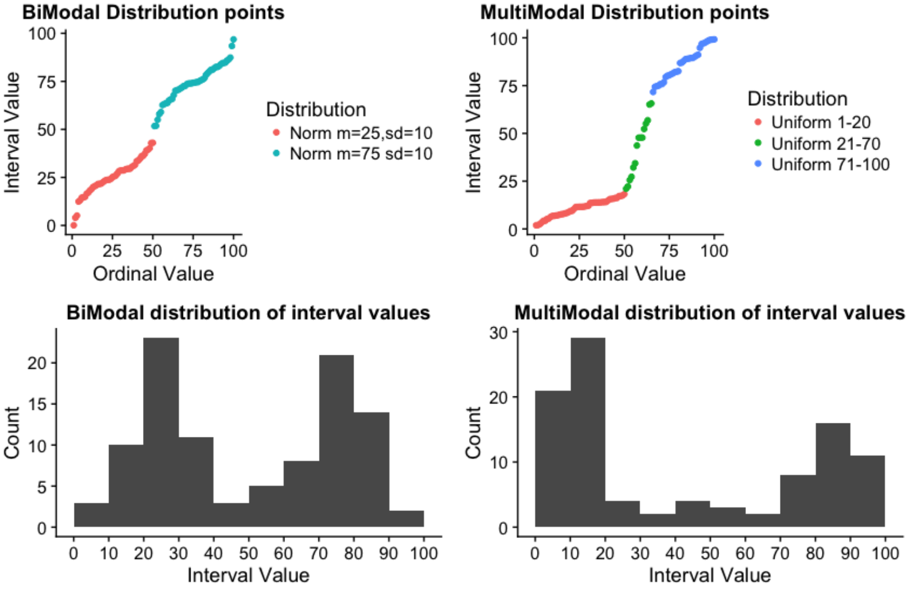


Fig. 5: Showing the distribution of scores for the Bi-Modal and Multi-Modal artificial datasets

4.2 Evaluation Metrics

We compare the RaScAL algorithm, described in Section 3, to the uniform-sampling approach employed by Bockhorst *et al.* [3][7] . We begin the evaluation by allowing three queries (the minimum required for the RaScAL algorithm). The evaluation then proceeds in single-instance batches, with each batch affording one additional query to the query budget of each algorithm. After each batch is complete, the returned scores are used to fit an isotonic regression which in turn is used to predict the scores of the remaining unlabelled items. We calculate the Root Mean Squared Error (RMSE) after each stage is complete and use the results to construct a learning curve plotting the RMSE of each algorithm's predictions against the number of labels requested. We use the trapezoidal rule[8] to approximate the *Area Under the Learning Curve* (AULC) which serves as an overall indicator of the learning rate, or accuracy of each approach. A lower AULC indicates a faster learning rate, and hence overall algorithm efficiency.

5 Results

Table 2 shows the AULC for RaScAL and uniform sampling on each of the datasets under examination. The RaScAL algorithm consistently outperformed

[7] Source code available at `https://github.com/joneill87/RaScAL`

[8] *pracma* `https://cran.r-project.org/web/packages/pracma/pracma.pdf`

the uniform sampling baseline on all datasets. The improvement is particularly pronounced on the Book Crossing and Movie Lens datasets, where the scarcity of raters and the integer-valued scores led to a significant number of ties.

Dataset	RaScAL	Uniform Sampling
Movie Lens	**5.00**	13.63
Jester	**4.87**	7.96
Boredom Videos	**2.10**	3.88
Book Crossing	**10.05**	55.40
Bi-Modal	**69.15**	98.23
Multi-Modal	**56.76**	91.58

Table 2: AULC for the RaScAL and Uniform Sampling algorithms on all datasets

Figure 6 depicts the learning curves for the RaScAL and baseline algorithms on each of the datasets under investigation. Figure 4 (d) shows that the Book Crossing dataset consists of densely packed scores, where there are far fewer items than there are distinct scores. It is clear from Figure 6 (d) that the RaScAL algorithm quickly identifies the key points in the ranking where the item scores increase, and reduces the error to 0 using only 10% of the labels (64 labels in total).

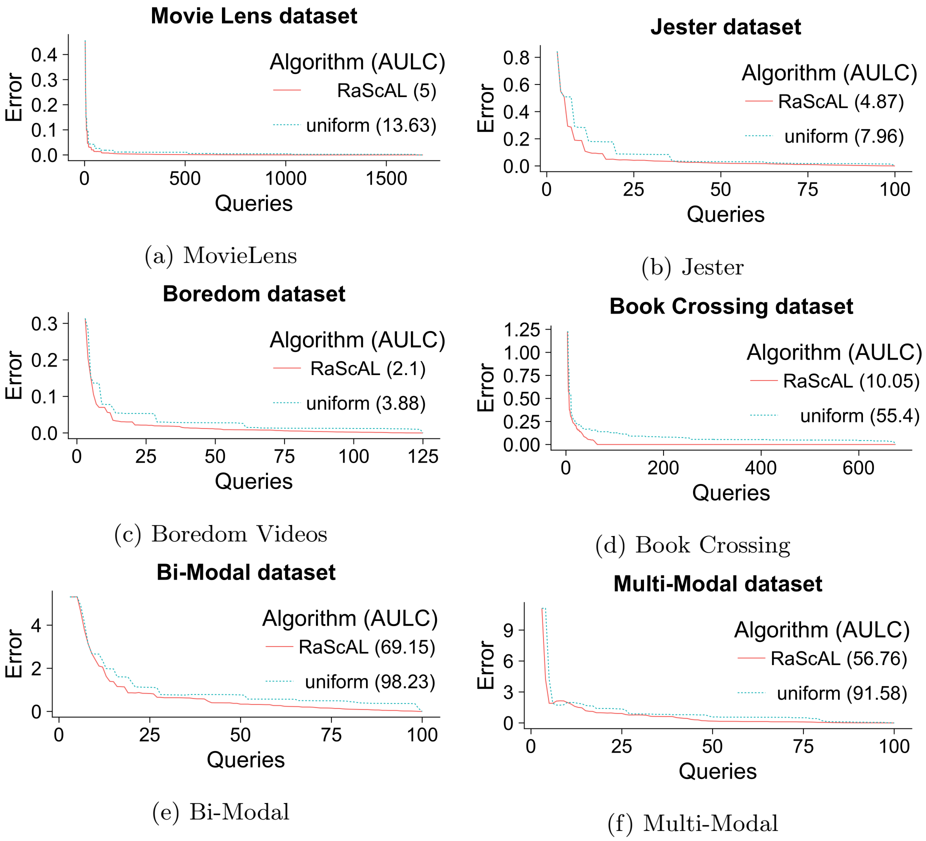
The RaScAL algorithm also outperforms the uniform sampling baseline in cases where there are few ties, but the distribution of scores is not uniform. Figure 6 (f) shows the learning curve decreasing rapidly and consistently for the RaScAL algorithm on the Multi-Modal dataset. This is due to its ability to issue more queries in areas of greater uncertainty.

The less uniform the distribution, the greater the improvement in learning rate. In the case of a perfectly uniform distribution of scores, the behaviour of RaScAL will mimic exactly the behaviour of the uniform sampling algorithm, so this approach should be preferred to uniform sampling in all cases.

6 Conclusions and Future Work

In this paper we presented RaScAL, an active learning approach to transforming rankings into scores. By simulating a data rating exercise we demonstrated that RaScAL performs at least well as, and often better than, a non active-learning baseline.

The task of recovering latent scores given a total rank ordering is one step within an overall system of preference elicitation of scores *via* pairwise comparisons. In the current study we have assumed knowledge of the overall ranking among items. In the final system, this ranking will need to be constructed by efficiently selecting of pairs of items for comparison and using rank aggregation to combine multiple 2-item rankings into an overall ranking. We anticipate that this will be achieved using a variant on the Bradley-Terry model.



(a) MovieLens

(b) Jester

(c) Boredom Videos

(d) Book Crossing

(e) Bi-Modal

(f) Multi-Modal

Fig. 6: Learning curve for each dataset used in this experiment.

Once this system has undergone end-to-end validation we aim to verify our findings using actual labellers labelling real data in a crowd-sourced environment. We hypothesize that if we elicit labels using pairwise comparisons as opposed to direct scores, the increased reliability of the resulting data will allow us to train more effective models using fewer labels.

References

1. Adomavicius, G., Bockstedt, J.C., Curley, S.P., Zhang, J.: Do recommender systems manipulate consumer preferences? a study of anchoring effects. Information Systems Research **24**(4), 956–975 (2013)
2. Barlow, R.E., Brunk, H.D.: The isotonic regression problem and its dual. Journal of the American Statistical Association **67**(337), 140–147 (1972)
3. Bockhorst, J., Yu, S., Polania, L., Fung, G.: Predicting Self-reported Customer Satisfaction of Interactions with a Corporate Call Center. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) **10536 LNAI**, 179–190 (2017)

4. Cai, W., Zhang, M., Zhang, Y.: Batch mode active learning for regression with expected model change. IEEE transactions on neural networks and learning systems **28**(7), 1668–1681 (2017)
5. Cholleti, S.R., Don, S., Goldman, S.A., Politte, D.G.: Veritas : Combining Expert Opinions without Labeled Data. International Journal on Artificial Intelligence Tools **18**(05), 633–651 (2009)
6. Eickhoff, C.: Cognitive biases in crowdsourcing. In: Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining. pp. 162–170. ACM (2018)
7. Elahi, M., Ricci, F., Rubens, N.: A survey of active learning in collaborative filtering recommender systems. Computer Science Review **20**, 29–50 (2016)
8. Fedorov, V.V.: Theory of optimal experiments. Elsevier (1972)
9. Goldberg, K., Roeder, T., Gupta, D., Perkins, C.: Eigentaste: A constant time collaborative filtering algorithm. information retrieval **4**(2), 133–151 (2001)
10. Grimm, M., Kroschel, K., Narayanan, S.: The Vera am Mittag German audio-visual emotional speech database. 2008 IEEE International Conference on Multimedia and Expo, ICME 2008 - Proceedings pp. 865–868 (2008)
11. Harper, F.M., Konstan, J.A.: The movielens datasets: History and context. ACM Transactions on Interactive Intelligent Systems (TiiS) **5**(4), 19 (2016)
12. Nie, R., Wiens, D.P., Zhai, Z.: Minimax robust active learning for approximately specified regression models. Canadian Journal of Statistics **46**(1), 104–122 (2018)
13. ONeill, J., Delany, S.J., Mac Namee, B.: Rating by ranking: An improved scale for judgement-based labels. In: 4 th Joint Workshop on Interfaces and Human Decision Making for Recommender Systems (IntRS) 2017. p. 24 (2017)
14. Raykar, V.C., Duraiswami, R., Krishnapuram, B.: A fast algorithm for learning a ranking function from large scale data sets. IEEE Transactions on Pattern Analysis and Machine Intelligence **30**(7), 1158–1170 (2008)
15. Settles, B.: Active learning. Synthesis Lectures on Artificial Intelligence and Machine Learning **6**(1), 1–114 (2012)
16. Soleymani, M., Larson, M.: Crowdsourcing for Affective Annotation of Video : Development of a Viewer-reported Boredom Corpus. Proceedings of the ACM SIGIR 2010 workshop on crowdsourcing for search evaluation (CSE 2010) pp. 4–8 (2010)
17. Su, X., Khoshgoftaar, T.M.: A Survey of Collaborative Filtering Techniques. Advances in Artificial Intelligence **2009**(Section 3), 1–19 (2009)
18. Sugiyama, M., Nakajima, S.: Pool-based active learning in approximate linear regression. Machine Learning **75**(3), 249–274 (2009)
19. Tarasov, A.: Dynamic Estimation of Rater Reliability using Multi-Armed Bandits. Doctoral thesis, Dublin Institute of Technology (2014)
20. Tversky, A., Kahneman, D.: Judgment under uncertainty: Heuristics and biases. science **185**(4157), 1124–1131 (1974)
21. Ziegler, C.N.C., McNee, S.M.S., Konstan, J.a.J., Lausen, G.: Improving recommendation lists through topic diversification. In: Proceedings of the 14th international conference on World Wide Web WWW 05. pp. 22–32 (2005)
22. Zielinski, S., Rumsey, F., Bech, S.: On some biases encountered in modern audio quality listening tests - a review. Journal of the Audio Engineering Society **56**(6), 427–451 (2008)