On The Benefit of Logic-based Approach to Learn Pairwise Comparisons

Nunung Nurul Qomariyah[1], **Dimitar Kazakov**[2], **Ahmad Nurul Fajar**[3]
[1]Computer Science Department, Faculty of Computing and Media, Bina Nusantara University, Jakarta, Indonesia, 11480
[2]Computer Science Department, University of York, UK
[3]Information System Management Department, Binus Graduate Program - Master of Information Systems Management, Bina Nusantara University Jakarta, Indonesia 11480

Article Info

Article history:

Received Jan 29, 2020
Revised Feb 26, 2020
Accepted March 2, 2020

Keywords:

Inductive logic approach
Pairwise comparisons
Machine learning
Recommender system

ABSTRACT

In recent years, many daily processes such as internet web searching, e-mail filtering, social media services, e-commerce have benefited from Machine Learning (ML) techniques. The implementation of ML techniques has been largely focused on black box methods where the general conclusions are not easily interpretable. Hence, the elaboration with other declarative software models to identify the correctness and completeness of the models is not easy to perform. On the other hand, the emerge of some logic-based machine learning approaches that can overcome such limitations with their white box methods has been proven to be well-suited for many software engineering tasks. In this paper, we propose the use of a logic-based approach to learn user preference in the form of pairwise comparisons. APARELL as a novel approach of inductive learning is able to model the user's preferences in Description Logic(DL) and then build a model by generalising the concept for all examples given. This offers a rich, relational representation beyond the usual propositional domain, which is then can be used to produce a set of recommendations. A user study has been performed in our experiment to evaluate the implementation of pairwise preference recommender system when compared to a standard list interface. The result of the experiment shows that the pairwise interface was significantly better than the other interface in many ways.

Corresponding Author:

Nunung Nurul Qomariyah,
Computer Science Department,
Faculty of Computing and Media,
Bina Nusantara University, Jakarta, Indonesia, 11480
Email: nunung.qomariyah@binus.edu

1. INTRODUCTION

The two paradigms in Artificial Intelligence (AI), logical AI and statistical AI, have been around for years. Even though nowadays the industry has favoured the statistical based AI learning approach over the logic-based one due to the complexity of the implementation, but logic based approach has also shown its benefit on some research work. Logical AI uses first-order logic and related representations to describe complex relationships and knowledge about the world while statistical AI relies heavily on the data points representation. One concept in logical AI where the machine generalizes the hypothesis/model from facts/examples is called inductive concept learning. Induction itself means that we infer information from specific facts then build the generalised principle of the whole set of observations. In AI, the learning method which adopts the inductive strategy is known as inductive learning, or also known as supervised learning. While the statistical inductive learning is expressed by using numerical method, the generalised principle of logic-based inductive learning is expressed by using some logic language, e.g. Propositional Logic, First Order Logic (FOL), or Description Logics (DL). APARELL [1] is one of the inductive algorithm specialised in learning relation in DL language. It is a further implementation of Inductive Logic Programming (ILP) [2] which was developed to learn classification problem in FOL language and DL-Learner [3] which was developed to learn classes in DL language. Some of ILP advantages are simplicity, readability and efficiency to handle numerous data. In this paper, we aim to show the benefit of using logical representation to learn relation between two classes of a set of datapoint with specific characteristics.

Learning user preference itself has been studied by many researchers and it has grown to become a specific research field, which called Preference Learning (PL), introduced by Fürnkranz and Hüllermeier in 2010 [4]. The main purpose of this field is to generate a valid predictive hypothesis model from a set of known value of user preferences. The most common form of this learning output is in a full order ranking. For example, a user may give a preference that he/she likes item A better than item B, item B better than item C. Then the aim of the learning is to produce global ranking, such as, according to that particular user, the following hypothesis is valid: `item A` $\succ$ `item B` $\succ$ `item C`. It uses "$\succ$" symbol to express the higher order in the ranking position. The most well-known algorithm in this field called *label ranking* which was also introduced by the same group of the researcher in 2008 [5]. The latest work in this field has been proposed by Pandey, et. al. in [6] which introduced $Pref2Vec$ algorithm using a vector to learn pairwise item preferences.

We use pairwise comparisons problem as a case study to learn relations by using a logic-based approach. In this paper, we implemented the learning algorithm in recommender system area to make it easier for the reader to understand the problem. Recommender System (RS) is a subfield of Information Filtering where the main goal is to predict the customer preferences and then recommend the best suited items. The most common method for preference elicitation in recommender system is using ratings, reviews and feedback. There is also an alternative method for elicit user's taste, called pairwise preference elicitation, where the users can express their favour in the form of choices. In pairwise preference, the user is asked to rank the items based on their liking and it is shown in pair of items. Some researchers like Balakrishnan and Chopra [7] has shown that by using pairwise comparisons, the majority of users (74%) find that the recommendation produced by this preference elicitation method is better than using numeric rating system although only 45% of the participants like to answer the question in pairwise choices. The use of pairwise comparisons can also reduce the user inconsistency of giving numeric ratings. Pairwise recommender systems have been an interest in these recent years. This is because the use of pairwise comparisons as preference elicitation itself is still uncommon. Research in this area includes using collaborative filtering in pairwise item [8], Orthogonal Pairwise Queries [9], music recommender system based on Gaussian Process [10] and pairwise recommender system using Lazy Decision Tree [11].

On the other hand, most of the work in recommender systems use statistical machine learning to predict the most favourite items of the customers. While this approach has been proven to be statistically accurate, but there are still some disadvantages as they can only work well with a large number of data. In the case when there is not enough data or information given to the system, this approach has to make some approximate calculations to create the recommendation. In this paper, we introduce the use of logic based approach in recommender system which works well with limited data as well. Using a logic-based machine learning approach, the system can also produce a nice explanation of why the recommender system can show such a valid recommendation. Studies on the explanatory aspect of recommender systems which shows process transparency to users have been growing [12, 13, 14]. People tend to express more, as well as increasing trust and loyalty, if they are able to get the explanation on how the system can give them a recommendation. Some of the studies on explanation interfaces for collaborative filtering method are discussed in a number of other publications [15]. A logic based approach, which use more representative language to express data and model, has a capability to build white box machine learning which can easily be understood by the human. The advantage of this logic based approach has not been widely used in recommender systems. There is an opportunity which can be explored more in this area.

In this paper, we describe the implementation of a recommender system for used car sales domain by using our previous work on a logic based approach, called APARELL. From a set of pairwise comparisons, the system can learn the user's preferences to produce a set of recommendations and provide an explanation of why the system chooses the recommended items. This algorithm has been evaluated and outperformed the other baseline algorithms, which also shows higher accuracy against statistic-based learning algorithm [16]. The contribution of this paper includes: (1) implementing our novel logic based machine learning approach for a real-life recommender system, (2) introducing the use of pairwise preference in recommender system and (3) evaluating the combination of logic based machine learning with pairwise interface performance against the standard most common list interface. The rest of the paper is organised as follows: in Section 2, we explain the proposed main modules, which consist of the learning algorithm for generating the rules and the recommendation algorithm. We then discuss the research method in Section 3 which discuss about the usability technique and questionnaire, followed by the result analysis in Section 4. Finally, we conclude our work and provide our plan for further work in Section 5.

2. THE PROPOSED METHOD

We built the personalised recommender system based on pairwise choices. The system learned from a set of answers given by the users when the choices shown to them. The users could also use filtering system where they could narrow down the options closest to their needs.

The system consisted of two main modules: (1) learner and (2) recommender. The learner module was implemented by using a logic-based machine learning algorithm, APARELL and the recommender system was built based on the rules produced by the first module.

2.1. Learning Algorithm

We have built a general purpose learning algorithm called APARELL (Active Pairwise Relations Learner). The complete description about this algorithm and the technical aspect can be found in our other article in [1]. It is a machine learning algorithm which uses Description Logics (DL) representation to describe the problem. By using the terms from Inductive Logic Programming (ILP) [17], we explain the representation of APARELL in three parts: (1) hypothesis language, (2) set of examples and (3) background knowledge.

Hypothesis Language. Hypothesis language is the representation language used to describe the outputs of the machine learning. In APARELL, the hypotheses are described in the forms of specific relation between two individuals or group of individuals which has a set of particular properties (defined as a class or subclass). For example, a car that belongs to Manual and SUV class always better than a car that belong to Automatic class. APARELL will search the best hypothesis from the hypothesis space described as in the previous examples. The hypothesis language used in here is the **role** itself which is the relation *betterthan*. The representation of hypothesis language by using RDF format is written as:

```
carsontology:betterthan rdf:type owl:ObjectProperty.
```

Set of Examples. We consider the transitivity in the *betterthan* relation as its natural behaviour. The transitivity in this context means that whenever a user prefer item A than item B, and prefer item B than item C, we can conclude that this particular user will prefer item A over item C. So we can add this new fact to the examples. Since APARELL uses DL as the representation language, inferring a new fact as transitivity closure can easily be performed by using the ontology reasoners, such as Pellet, FaCT++, HerMiT, ELK, etc. These reasoners have the capability to automatically detect the inferred facts or relations from the given knowledge bases.

In addition to that, the *betterthan* relation satisfies the anti-symmetric property, which means that there is no counter-example exists in the dataset. For instance, if a user likes item A better than item B, there is no single evidence that says this particular user likes item B better than item A (the other way around). Observing this behaviour, APARELL put the additional reverse order of the *betterthan* relation as negative examples (incorrect data). From here, the algorithm will have a set of positive (correct order) and a set of negative (reverse order) examples to be learned. The representation of examples in RDF format is shown below:

```
    carsontology:car1 carsontology:betterthan carsontology:car3,
                                              carsontology:car4,
                                              carsontology:car5,
                                              carsontology:car10.
```

Background Knowledge. In ILP, background knowledge is important information that the learning algorithm can use to build a set of hypotheses [17]. This important information can be a description of the properties that each individual holds. In DL, the background knowledge is expressed through the classes and their membership. For example, "car 1 belongs to a class: Manual and SUV". In RDF format, it representation is shown below:

```
carsontology:Hatchback rdf:type owl:Class ;
               rdfs:subClassOf carsontology:Car;
               owl:disjointWith carsontology:Suv.

carsontology:car1 rdf:type owl:NamedIndividual,
                       carsontology:Car,
                       carsontology:Automatic,
                       carsontology:Petrol,
                       carsontology:Hatchback.
```

Learning Step. APARELL as an inductive learner learns from a set of given facts and background knowledge to find the best general hypothesis which correctly explain the information provided. The learning step is shown in Figure 1. In order to find the valid hypotheses, APARELL uses these basic steps as follows:

1. Start from one positive example from the given input.
2. Generate the botom clause for the chosen positive example in the previous step.
3. Start building a possible hypothesis space. In this step, APARELL builds a simple combination of each class available as the bottom clause in the previous step. The top hypothesis will be: Thing is-better-than Thing.
4. Start searching from the top, capped by the bottom clause. In this step, APARELL is finding the generalisations of the bottom clause. It aims to find the most general hypothesis.
5. Calculate the score. In this step, APARELL only considers the hypothesis which covered none of the negative examples. The scoring function is using the number of positive examples covered. It adds one point for each positive examples covered by the hypothesis. The best hypothesis is the one which has the highest score which means covered more positive examples.
6. Build the set of possible best hypothesis based on the best score. This process is shown as the greyed box in Figure 1. A set of the valid hypothesis will be the output of the algorithm.
7. Eliminate the positive examples which have already been covered in the previous hypothesis search.

2.2. Recommendation algorithm

The output of APARELL is a set of hypothesis which consistent with the preferences given by the users. Some of the possible APARELL outputs are shown below:

```
(Manual)    betterthan (MediumCar)
(Hatchback) betterthan (MediumCar)
(MediumCar) betterthan (Automatic)
(Hatchback) betterthan (Automatic)
(Hatchback) betterthan (Estate)
(Petrol)    betterthan (Diesel)
```

In this implementation, we use a rule-based algorithm which implies that every item that dominated the preferences will always better than the other ones. A directed graph is build to better illustrate the process. Once we have a list of cars family preferred by the users, the system can build the list of recommendation system based on the characteristics found in the APARELL's rules. For example, in the above output we can draw a directed graph as shown in Figure 2a. From the given output, we can conclude that the users' most preferred cars are described as: Manual, Hatchback and Petrol. The recommendation module will search the item match with that attributes. This best cars attributes will then be shown in the explanation part of the system. The explanation side in the system is shown in Figure 2b.

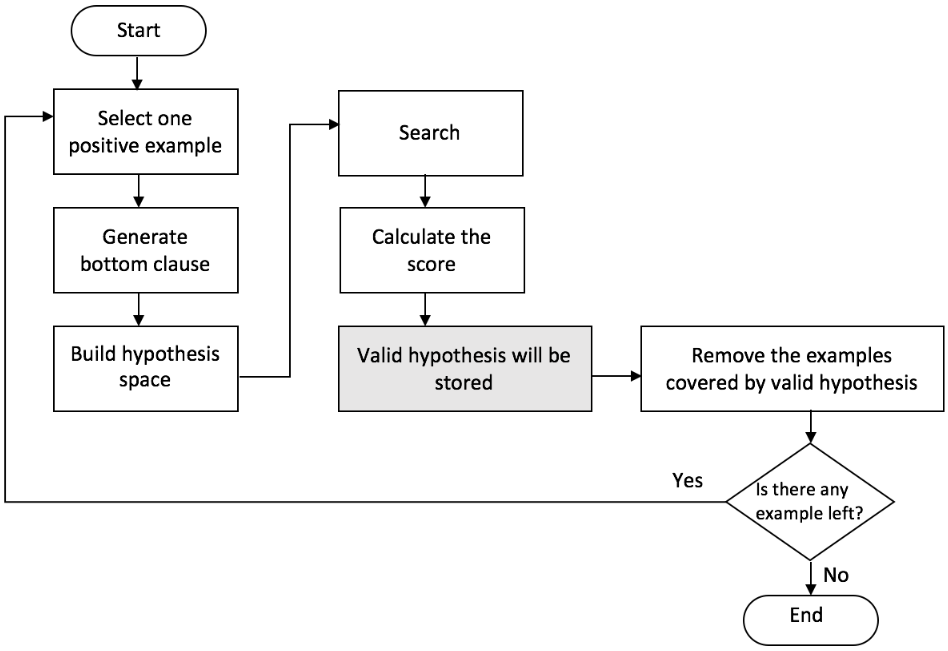
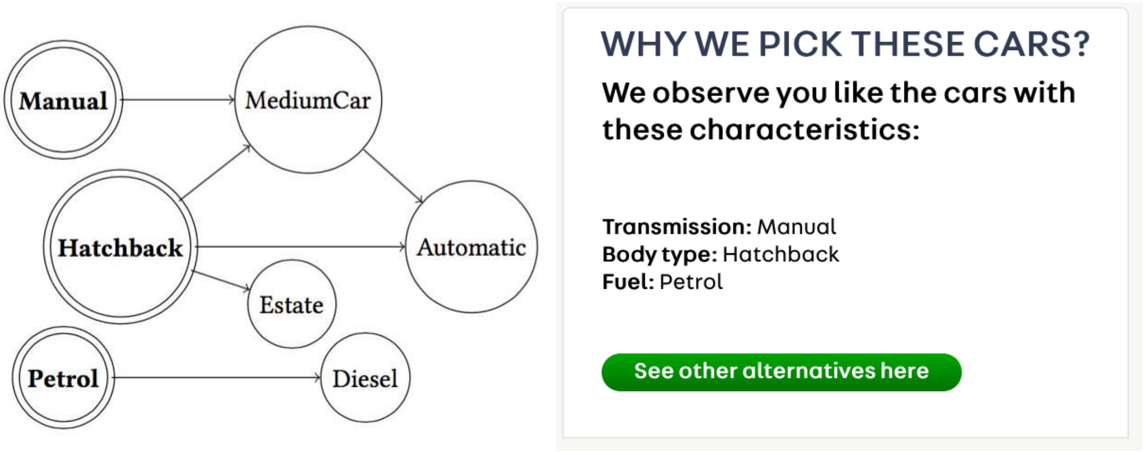

Start
Select one positive example
Generate bottom clause
Build hypothesis space
Search
Calculate the score
Valid hypothesis will be stored
Remove the examples covered by valid hypothesis
Yes
Is there any example left?
No
End

Figure 1. APARELL Learning Step
(the greyed box shows the process where the valid rules produced as the output of the algorithm)



WHY WE PICK THESE CARS?

We observe you like the cars with these characteristics:

Transmission: Manual
Body type: Hatchback
Fuel: Petrol

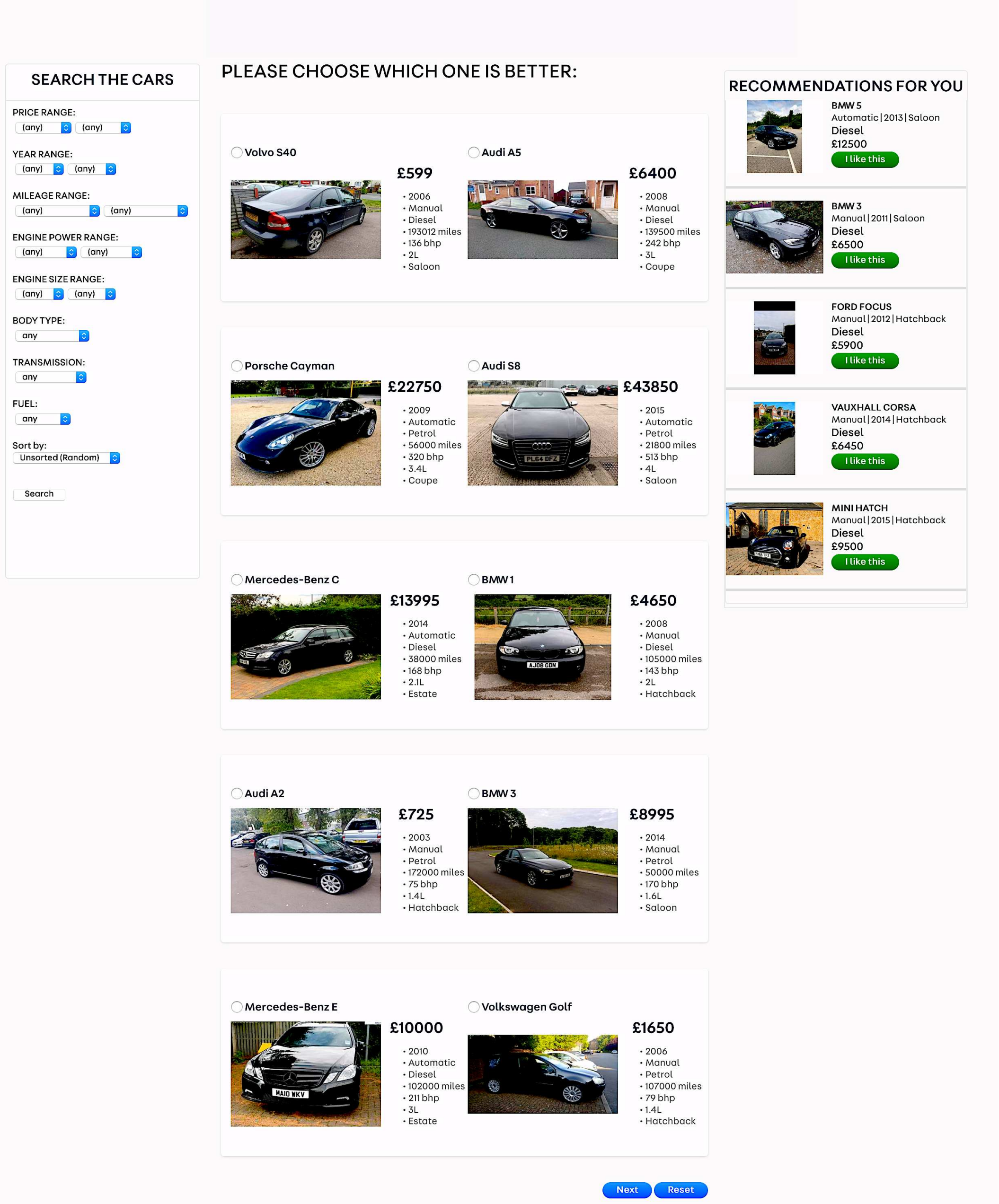
(a) Directed graph of user preferences

(b) Recommendation explanation

Figure 2. Explanation of the Recommendation System

3. RESEARCH METHOD

Users were interacted with the system by using a web interface shown in Figure 3. Through the interface, users were allowed to use the filtering system in order to narrow down the choices shown to them. For example, if a user wants to find a medium-sized car for commuting purposes, then the users could limit the choices by body type, by choosing “SUV” and engine size ranges from 1.5 to 2 Litre. By using the filtering system, the users had control of their choices, so that the ranges of options were still within their searching criteria. As shown in Figure 3, the users were given 5 pairs of cars that he/she needs to pick which one was better than the other one. After giving the preferences to the system, they could proceed to the next page and giving more preferences. The more preferences are given to the system, the more accurate the recommendation would be, as the learning data increased. We also built a standard list interface which shown in Figure 4, which allowed users to compare their experience when using the pairwise interface to the standard interface which they often find it in the most e-commerce systems. After the users expressed their preferences to the system, they could click the Finish button and the Top Pick page were shown which contains the list of recommended cars. The top pick page is shown in Figure 5.



SEARCH THE CARS

PRICE RANGE: (any) (any)

YEAR RANGE: (any) (any)

MILEAGE RANGE: (any) (any)

ENGINE POWER RANGE: (any) (any)

ENGINE SIZE RANGE: (any) (any)

BODY TYPE: any

TRANSMISSION: any

FUEL: any

Sort by: Unsorted (Random)

PLEASE CHOOSE WHICH ONE IS BETTER:

○ Volvo S40 £599
- 2006
- Manual
- Diesel
- 193012 miles
- 136 bhp
- 2L
- Saloon

○ Audi A5 £6400
- 2008
- Manual
- Diesel
- 139500 miles
- 242 bhp
- 3L
- Coupe

○ Porsche Cayman £22750
- 2009
- Automatic
- Petrol
- 56000 miles
- 320 bhp
- 3.4L
- Coupe

○ Audi S8 £43850
- 2015
- Automatic
- Petrol
- 21800 miles
- 513 bhp
- 4L
- Saloon

○ Mercedes-Benz C £13995
- 2014
- Automatic
- Diesel
- 38000 miles
- 168 bhp
- 2.1L
- Estate

○ BMW 1 £4650
- 2008
- Manual
- Diesel
- 105000 miles
- 143 bhp
- 2L
- Hatchback

○ Audi A2 £725
- 2003
- Manual
- Petrol
- 172000 miles
- 75 bhp
- 1.4L
- Hatchback

○ BMW 3 £8995
- 2014
- Manual
- Petrol
- 50000 miles
- 170 bhp
- 1.6L
- Saloon

○ Mercedes-Benz E £10000
- 2010
- Automatic
- Diesel
- 102000 miles
- 211 bhp
- 3L
- Estate

○ Volkswagen Golf £1650
- 2006
- Manual
- Petrol
- 107000 miles
- 79 bhp
- 1.4L
- Hatchback

RECOMMENDATIONS FOR YOU

BMW 5
Automatic | 2013 | Saloon
Diesel
£12500

BMW 3
Manual | 2011 | Saloon
Diesel
£6500

FORD FOCUS
Manual | 2012 | Hatchback
Diesel
£5900

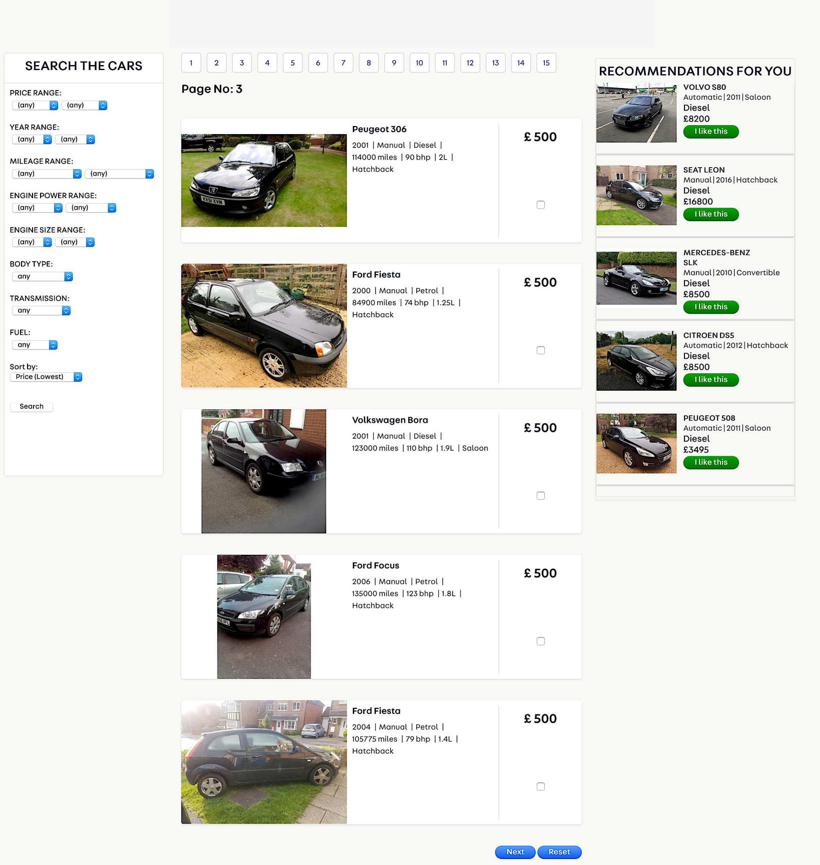
VAUXHALL CORSA
Manual | 2014 | Hatchback
Diesel
£6450

MINI HATCH
Manual | 2015 | Hatchback
Diesel
£9500

Figure 3. Pairwise comparisons user interface

3.1. User Study

The user study was taken place in York, United Kingdom. We asked the participants to complete two given tasks by using both interfaces, so they could have different experience and were able to evaluate them afterwards. The post-questionnaires used in this experiment were following **ResQue**, an evaluation framework of recommender systems proposed in [18]. This framework has been used in a similar study in [19]. We invited 24 participants in this study. We used a repeated-measures user study design in which the same individuals tested both user interfaces. We also designed a counter-balanced study in which we randomly divide the



SEARCH THE CARS

PRICE RANGE: (any) (any)

YEAR RANGE: (any) (any)

MILEAGE RANGE: (any) (any)

ENGINE POWER RANGE: (any) (any)

ENGINE SIZE RANGE: (any) (any)

BODY TYPE: any

TRANSMISSION: any

FUEL: any

Sort by: Price (Lowest)

Page No: 3

Peugeot 306
2001 | Manual | Diesel |
114000 miles | 90 bhp | 2L |
Hatchback
£ 500 □

Ford Fiesta
2000 | Manual | Petrol |
84900 miles | 74 bhp | 1.25L |
Hatchback
£ 500 □

Volkswagen Bora
2001 | Manual | Diesel |
123000 miles | 110 bhp | 1.9L | Saloon
£ 500 □

Ford Focus
2006 | Manual | Petrol |
135000 miles | 123 bhp | 1.8L |
Hatchback
£ 500 □

Ford Fiesta
2004 | Manual | Petrol |
105775 miles | 79 bhp | 1.4L |
Hatchback
£ 500 □

RECOMMENDATIONS FOR YOU

VOLVO S80
Automatic | 2011 | Saloon
Diesel
£8200

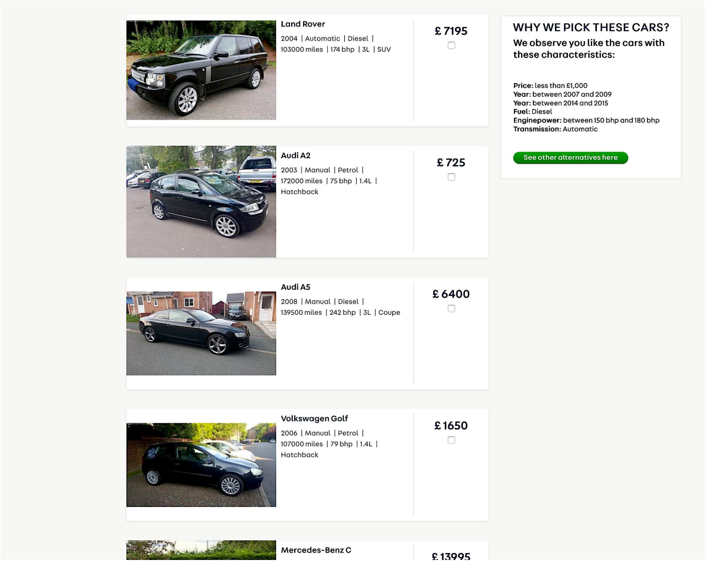
SEAT LEON
Manual | 2016 | Hatchback
Diesel
£16800

MERCEDES-BENZ SLK
Manual | 2010 | Convertible
Diesel
£8500

CITROEN DS5
Automatic | 2012 | Hatchback
Diesel
£8500

PEUGEOT 508
Automatic | 2011 | Saloon
Diesel
£3495

Figure 4. Standard list user interface



Land Rover
2004 | Automatic | Diesel |
103000 miles | 174 bhp | 3L | SUV
£ 7195 □

Audi A2
2003 | Manual | Petrol |
172000 miles | 75 bhp | 1.4L |
Hatchback
£ 725 □

Audi A5
2008 | Manual | Diesel |
139500 miles | 242 bhp | 3L | Coupe
£ 6400 □

Volkswagen Golf
2006 | Manual | Petrol |
107000 miles | 79 bhp | 1.4L |
Hatchback
£ 1650 □

Mercedes-Benz C
£ 13995

WHY WE PICK THESE CARS?

We observe you like the cars with these characteristics:

Price: less than £1,000
Year: between 2007 and 2009
Year: between 2014 and 2015
Fuel: Diesel
Enginepower: between 150 bhp and 180 bhp
Transmission: Automatic

Figure 5. Top picks page with explanation

participants into two different groups, the first group were asked to evaluate the pairwise interface first, while the other group were asked to evaluate the standard list first.

The user manual was provided to guide the users before they started the experiments. A series of demographic information was also collected beforehand. This experiment was conducted anonymously so that the users were not asked for giving their personal identity to comply with ethical practices designed to protect the privacy of the subjects. The dataset consists of 7,360 available used cars used in this experiment were crawled from Autotrader website. The cars have seven attributes, i.e. price, year, mileage, engine power, engine size, body type, transmission, fuel type.

The use of the pairwise preference interface consisted of the following steps:

1. Initially, the interface displayed the top five most popular car models to all users in the right-hand-side (RHS) margin of the page.

2. The users could choose to apply a search filter to narrow down the types of cars being displayed. (This was optional.) The filter used had the same appearance and functionality as the one provided in the list interface.

3. Consecutive pages, each containing 5 pairs of cars consistent with the filter options, were displayed and the user was given the option of selecting their preference for each pair. They could also skip any pair.

4. The APARELL learner was used to update the user model.

5. If at least 10 pairs had been annotated, the top 10 final recommendations were displayed and the user was invited to select up to three cars that matched their needs. Otherwise, the provisional recommendations were updated on the basis of the revised user model, and the top five displayed in the RHS margin.

The use of the list interface consisted of the following steps:

1. Initially, the interface displayed the top five most popular car models to all users in the right-hand-side (RHS) margin of the page.

2. The users had to apply a search filter to narrow down the types of cars being displayed. (This was mandatory.) The filter used had the same appearance and functionality as the one provided in the pairwise preference interface.

3. Consecutive pages, each containing 5 cars consistent with the filter options, were displayed and the user was given the option of flagging up cars of interest.

4. If at least 10 cars had been flagged up, the first 10 such cars were displayed and the user was invited to select up to three cars that matched their needs. Otherwise, the previous step was repeated, while another 5 cars for each of the top five most popular car models were displayed in the RHS margin.

3.2. User Study Tasks

The participants were given two different tasks to be completed using both interfaces. The tasks were designed to be as close as possible with their daily routines. The tasks were formulated as follows: (1) the participants were asked to find suitable cars for daily commute purposes between their home and office and (2) the participants were asked to find suitable cars for weekend shopping only. Each task needed to be done by using one of the interfaces.

In pairwise interface, for each page, the users were shown up to 5 pairs of cars at a time and they could give preference (pick the best one in each pair), they could choose to skip some pairs if they thought they did not like both.

3.3. Post-stage questionnaire

Following the evaluation procedure, the participants need to fill in a post-questionnaire after completing each task to measure how good is the recommender system. This questionnaire was using a five-point Likert scale to measure how much the participants agree or disagree on each question. This questionnaire can also be called usability and user satisfaction testing. The evaluation criteria and the questions of the post-stage questionnaire are shown below:

1. The first group of questions measured the quality of recommended items
   - **Q1.** Whether the items recommended by the system matched the users interests
   - **Q2.** Whether the list of recommended items were novel and interesting
   - **Q3.** Whether the recommended items were diverse
   - **Q4.** Whether the items recommended by the system were similar to each other (reverse scale)
2. The second group of questions measured the quality of the interactivity of the recommender system
   - **Q5.** The system explanation on why the recommended items were given
   - **Q6.** The attractiveness of the recommender interface layout
   - **Q7.** The easiness of finding an item to buy with the help of the recommender
   - **Q8.** Whether the users feel supported to find suitable items with the help of the recommender
   - **Q9.** Whether the users feel in control of expressing their preferences to the recommender
3. The third group of questions measured the attitude of the participants after evaluating the system
   - **Q10.** The overall satisfaction with the recommender
   - **Q11.** How convinced were they with the products recommended
   - **Q12.** How confident were they to like the items recommended
4. The last group of questions were measuring the behavioural intentions
   - **Q13.** Whether they would come back and use the system again
   - **Q14.** Whether they would purchase the items shown in the recommendation.

3.4. Interface preferences questionnaire

After evaluating both interfaces, the participants were given a final questionnaire from [19] to measure their preferences regarding the interfaces as below:
Q1. Measuring their general preferences by asking which interface did they prefer overall.
Q2. Measuring the informativeness between those two interfaces by asking which interfaces did they find more informative.
Q3. Measuring the usefulness by asking which interface did they find it gives more useful recommendation.
Q4. Measuring which interface was better at recommending.
Q5. Measuring the perceived diversity by asking which one of the interfaces showing more diverse choices. This questionnaire allowed the participants to vote either pairwise interface, standard list interface or both if they do not have any preference.

4. RESULTS AND DISCUSSION

The 24 participants were recruited from The University of York, UK. Their age were ranged from 20 to 45 years old and the population had a good balance on the gender distribution with 54% male, 42% female and 4% of another gender.

4.1. Post-stage questionnaire results

All the ordinal values received from the participants on each question we averaged. In order to understand the difference between what the participants feel after reviewing the pairwise interface and after evaluating the standard list interface, we used a paired sample t-test. We display the comparisons on their answer on both interfaces are shown in side-by-side in Figure 6. The significant differences were found for six questions mentioned below with $p - value$ from paired sample t-test is 0.05, they are: **[Q3]** p-value=0.0208, **[Q5:]** p-value=0.0019, **[Q7]** p-value=0.0359, **[Q11]** p-value=0.0499, **[Q13]** p-value=0.0359 and **[Q14]** p-value=0.0256. Even though there were only six questions showed significant differences, the overall average of pairwise interface was higher than the standard list interface.

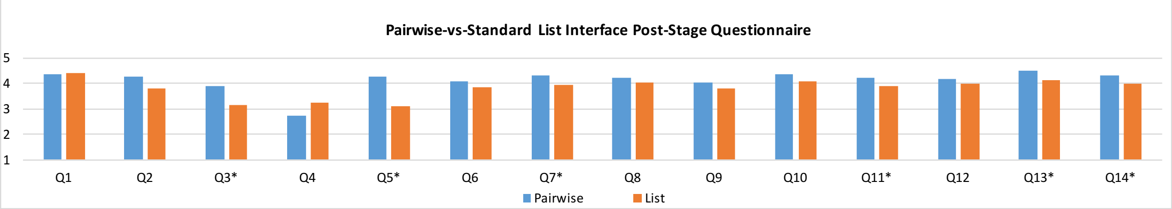


Figure 6. Usability and user satisfaction assessment results
Q with * mark is a question with the significant difference at $\alpha = 5\%$ $(p - value < 0.05)$

4.2. Interface preferences questionnaire results

The final questionnaires were measuring the general preferences of the recommender system. The result of this questionnaire also showed that the pairwise interface received more votes than the other interface in four factors: [**Q1.**] Measuring their general preferences by asking which interface did they prefer overall, [**Q2.**] Measuring the informativeness between those two interfaces by asking which interfaces did they find more informative, [**Q3.**] Measuring the usefulness by asking which interface did they find it gives more useful recommendation and [**Q5.**] Measuring the perceived diversity by asking which one of the interface which showing more diverse choices. While in [**Q4.**] Measuring which interface was better at recommending, both interfaces received the same number of votes. The result of the final preferences is shown in Figure 7.

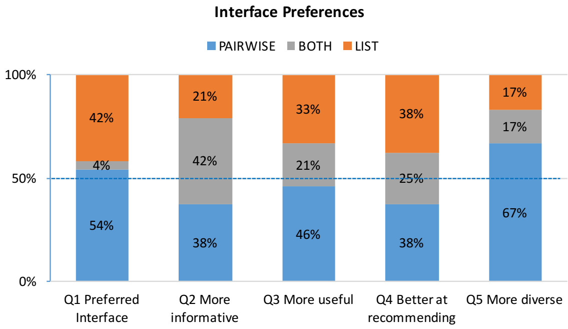


Figure 7. Interface preferences questionnaire results

5. CONCLUSION AND FUTURE WORK

In this paper, we show the implementation and evaluation of pairwise recommender system by using a logic-based approach machine learning, called APARELL. The experiment was performed by conducting a user study to compare between two interfaces and evaluating using a user-centric evaluation framework for recommender system, called ResQue. Result of the experiment shows that a logic-based learner has the benefit of better explaining why the system produces such a recommendation. The result of all questionnaires shows that the implementation of a logic-based learner in the pairwise interface, when compare to a usual standard list interface, is more preferred.

The use of logic-based learning is central to the design of this new interface, as it makes it possible to extract useful models from a very small number of examples, thus making user-centred models viable even when facing the ‘cold-start’ problem, where there is no previous data on the user. In addition, learning in logic produces an interpretable model, which can be used to provide explanations for the recommendations, which is a clear advantage. It will be an advantage to generalize the recommendation model and tested in many domains, such one concept we have proposed for e-learning pairwise recommender system in [20].

REFERENCES

[1] Nunung Nurul Qomariyah and Dimitar Kazakov. Learning from Ordinal Data with Inductive Logic Programming in Description Logic. *Late Breaking Papers of the 27th International Conference on Inductive Logic Programming*. CEUR Workshop Proceedings, 2018.

[2] Stephen Muggleton. Inductive Logic Programming. *New generation computing*, vol. 8, no. 4, pp. 295–318, 1991.

[3] Jens Lehmann. DL-Learner: Learning Concepts in Description Logics. *The Journal of Machine Learning Research*, vo. 10, pp. 2639–2642, 2009.

[4] Johannes Fürnkranz and Eyke Hüllermeier. Preference Learning: An Introduction. *Preference Learning*, pp. 1–17. Springer, 2010.

[5] Eyke Hüllermeier, Johannes Fürnkranz, Weiwei Cheng, and Klaus Brinker. Label Ranking by Learning Pairwise Preferences. *Artificial Intelligence*, vol. 172, no. 16, pp. 1897 – 1916, 2008.

[6] G. Pandey, S. Wang, Z. Ren, and Y. Chang. Vectors of Pairwise Item Preferences. In Proceedings of European Conference on Information Retrieval, pp. 323-336. Springer, Cham, 2019.

[7] S. Balakrishnan and S. Chopra. Two of A Kind or the Ratings Game? Adaptive Pairwise Preferences and Latent Factor Models, In Proceedings of the 10th IEEE International Conference on Data Mining (ICDM), pp. 725–730, 2010.

[8] W. Pan, L. Chen and Z. Ming. Personalized Recommendation with Implicit Feedback via Learning Pairwise Preferences over Item-sets. *Knowledge and Information Systems*. Springer, 2018.

[9] Li Qian, Jinyang Gao and H. V. Jagadish. Learning user preferences by adaptive pairwise comparison. In Proceedings of the VLDB Endowment, vol. 8, no. 11, pp. 1322–1333, 2015.

[10] B. S. Jensen, J. S. Gallego and J. Larsen. A predictive model of music preference using pairwise comparisons. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp 1977–1980. IEEE, 2012.

[11] Lior Rokach and Slava Kisilevich. Initial profile generation in recommender systems using pairwise comparison. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 6, pp. 1854–1859, 2012.

[12] Behnoush Abdollahi and Olfa Nasraoui. Using Explainability for Constrained Matrix Factorization. In Proceedings of the 11th ACM Conference on Recommender Systems, pp. 79–83. ACM, 2017.

[13] Jonathan L. Herlocker, Joseph A. Konstan, and John Riedl. Explaining Collaborative Filtering Recommendations. In Proceedings of the ACM Conference on Computer Supported Cooperative Work, pp. 241–250. ACM, 2000.

[14] Mustafa Bilgic and Raymond J. Mooney. Explaining Recommendations: Satisfaction vs. Promotion. In Proceedings of Beyond Personalization Workshop, International Conference on Intelligent User Interfaces (IUI), volume 5, page 153, 2005.

[15] Nava Tintarev and Judith Masthoff. Designing and Evaluating Explanations for Recommender Systems. *Recommender Systems Handbook*, pp. 479–510. Springer, 2011.

[16] N. N. Qomariyah, and D. Kazakov. Learning Binary Preference Relations. In Proceedings of the 4th Joint Workshop on Interfaces and Human Decision Making for Recommender Systems (IntRS), pp. 30, 2017.

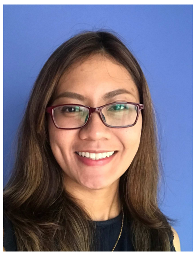
[17] Stephen Muggleton and Luc De Raedt. Inductive Logic Programming: Theory and Methods, *The Journal of Logic Programming*, vol. 19, pp. 629–679, 1994.

[18] Pearl Pu, Li Chen, and Rong Hu. A User-Centric Evaluation Framework for Recommender Systems. In Proceedings of the fifth ACM Conference on Recommender Systems, pages 157–164. ACM, 2011.

[19] Rong Hu and Pearl Pu. Helping Users Perceive Recommendation Diversity. In Proceedings of Workshop on Novelty and Diversity in Recommender Systems (DiveRS), pages 43–50, 2011.

[20] Nunung Nurul Qomariyah and Ahmad Nurul Fajar. Recommender System for e-Learning based on Personal Learning Style. *Proceedings of the 2nd International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*. IEEE, 2020.

BIOGRAPHIES OF AUTHORS



Dr. Nunung N. Qomariyah is a faculty member in Computer Science Department, Bina Nusantara University, Jakarta, Indonesia. Formerly, she was a member of Artificial Intelligence Research Group, Computer Science Department, University of York, UK. She was working under Dr. Dimitar Kazakov supervision during her PhD study. Her research interest is in the area of Recommender System, particularly in Preference Learning by using pairwise comparisons. She is currently affiliated with Indonesia Section IEEE computer society as professional member.
Further info on his homepage: http://international.binus.ac.id/computer-science/faculty/nunung-nurul-qomariyah/

Dr. Dimitar Kazakov is an associate professor (senior lecturer) at Computer Science Department, University of York, UK. He is currently appointed as the Head of Artificial Intelligence Research Group. He is also the chair of Departmental Teaching Committee. His research interest includes Machine Learning, Natural Language Processing; Machine Learning of Language.
Further info on his homepage: https://www-users.cs.york.ac.uk/ kazakov/

Dr. Ahmad Nurul Fajar is an associate professor in Computer Science, Graduate Program, Bina Nusantara University. In 2001, he was graduated from Gunadarma University majoring in Informatics. Master of Science Informatics was completed in 2004 at Bandung Institute of Technology (ITB). He hold PhD degree from Faculty of Computer Science University of Indonesia (UI) in 2014. His research interest is in the area of Software Engineering, Software Development, Information System Analysis and Design, Business Processes, and Service Oriented Architecture.