Inf Retrieval J
DOI 10.1007/s10791-017-9293-y

# Which used product is more sellable? A time-aware approach

**Mengwen Liu[1] · Wanying Ding[1] · Dae Hoon Park[2] · Yi Fang[3] · Rui Yan[4] · Xiaohua Hu[1]**

Received: 13 September 2016 / Accepted: 1 January 2017

**Abstract** A number of online marketplaces enable customers to buy or sell *used* products, which raises the need for ranking tools to help them find desirable items among a huge pool of choices. To the best of our knowledge, no prior work in the literature has investigated the task of *used* product ranking which has its unique characteristics compared with regular product ranking. While there exist a few ranking metrics (e.g., price, conversion probability) that measure the "goodness" of a product, they do not consider the *time* factor, which is crucial in used product trading due to the fact that each used product is often unique while new products are usually abundant in supply or quantity. In this paper, we introduce a novel time-aware metric—"sellability", which is defined as the time duration for a used item to be traded, to quantify the value of it. In order to estimate the "sellability" values for newly generated used products and to present users with a ranked list of the most relevant results, we propose a combined Poisson regression and listwise ranking model.

✉ Mengwen Liu
ml943@drexel.edu

Wanying Ding
wd78@drexel.edu

Dae Hoon Park
daehoon@yahoo-inc.com

Yi Fang
yfang@scu.edu

Rui Yan
ruiyan@pku.edu.cn

Xiaohua Hu
xh29@drexel.edu

[1] College of Computing and Informatics, Drexel University, Philadelphia, PA, USA

[2] Yahoo! Inc., Sunnyvale, CA, USA

[3] Department of Computer Engineering, Santa Clara University, Santa Clara, CA, USA

[4] Institute of Computer Science and Technology, Peking University, Beijing, China

Published online: 02 February 2017

The model has a good property in fitting the distribution of “sellability”. In addition, the model is designed to optimize loss functions for regression and ranking simultaneously, which is different from previous approaches that are conventionally learned with a single cost function, i.e., regression or ranking. We evaluate our approach in the domain of used vehicles. Experimental results show that the proposed model can improve both regression and ranking performance compared with non-machine learning and machine learning baselines.

**Keywords** Used product ranking · Learning to rank · Implicit feedback

## 1 Introduction

Online E-commerce platforms, such as eBay,[1] Taobao,[2] Kijiji,[3] facilitate the creation of used product markets that feature a substantially wider selection and lower prices than their counterparts who sell brand new products. Thus, it is important to design ranking tools to help customers make purchase decisions. It is also not wise to directly use conventional ranking metric to measure the “goodness” of used products, due to their unique characteristics. Unlike a regular product whose value is often quantified by its price, the value of a used product cannot be measured simply using the price, since it is no longer in the same condition as it was brand new. Sales, which is usually used for regular product ranking on e-commerice sites (e.g. Amazon[4]), is not applicable to used product ranking either, as used products are dissimilar to each other in regards to their conditions even if they are the same product or from the same product category.

Time is of essence to used product trading. From the perspective of sellers, they want to craft suitable advertisements and avoid appearing to be a fraud, so that their advertisements receive much attention from buyers and their items can be sold quickly. From the perspective of buyers, due to the nature that each used product is unique, they are likely to be influenced by the thought that they might miss a used product, which is known as the *Principle of Scarcity* (Lynn 1989). Therefore, a used product channel can utilize the tactic, suggesting that certain used products might soon be off the market, to drive sales. For example, a used vehicle database engine displays “would be sold in X days” (see Table 1 for an example) would be very beneficial for creating a sense of urgency for buyers and stimulate them to pull the trigger. Some Customer-to-Customer (C2C) E-commerce platforms use conversion probability (Wu and Bolivar 2009), number of user clicks (Wang et al. 2010), etc., to rank products sold by third-parties. However, none of those metrics take into consideration of the *time* factor.

In this paper, we introduce the concept of a *time*-aware metric, “sellability”, which is defined as a time duration for a used product to be traded, to measure the “goodness” of a used product. The time duration could be counted from the time when a used item is published until the time when it is sold. The usages of “sellability” are beneficial for both parties of sellers and buyers who participate in used product trading: (1) a seller can check

[1] http://www.ebay.com/.

[2] http://www.taobao.com/.

[3] http://www.kijiji.ca/.

[4] http://www.amazon.com/Best-Sellers/zgbs.

**Table 1** Example of ranking used vehicles by sellability given the query "*price = $10,000*" and "*Transmission = Automatic*"

| Year | Make | Model | Odometer (miles) | Style | How soon it will be sold |
|---|---|---|---|---|---|
| 2007 | Honda | Civic | 77,000 | Si | Would be sold in 2 days |
| 2007 | Honda | Accord | 72,000 | LX | Would be sold in 5 days |
| 2008 | Toyota | Corolla | 42,800 | LE | Would be sold in 7 days |
| 2007 | Toyota | Camry | 96,718 | LE | Would not be sold very soon |

the estimated "sellability" of a draft advertisement of his/her product, and refine it (e,g., adding more photos or adjusting the price) until a satisfied "sellability" value is obtained; and (2) a buyer can submit a search query to a used product database engine and obtain a list of relevant used items ordered based on the estimated "sellability" values.

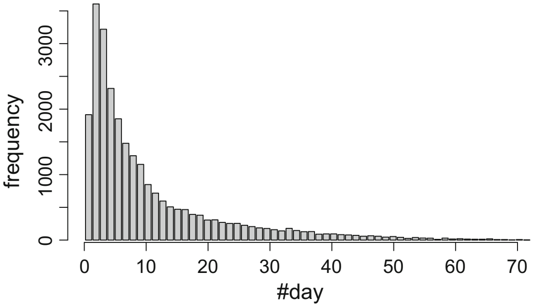
To enable the aforementioned two use cases, we develop a combined Poisson regression and listwise ranking model to estimate the "sellability" for newly generated used products (from sellers' perspective), and thus to rank them based on their predicted "sellability" values (from buyers' perspective). The model can be obtained through the training from a large amount of historical used products with their "sellability" values.

Our model is different from existing ranking models in two important ways. First, we adopt Poisson regression by assuming that the predicted values follow a Poisson distribution, which is more suitable for fitting the "sellability" of used products than ordinary linear regression (which assumes Gaussian distribution) or logistic regression (which assumes Bernoulli distribution) is. This assumption can be seen in Fig. 1, which plots the empirical distribution of "sellability" based on the dataset of used vehicles introduced in Sect. 4. Second, our model is optimized with regression and ranking loss functions simultaneously, while existing ranking models are conventionally optimized using a single cost function. A regression model that is optimized with a pointwise loss function can be used for prediction with minimal deviation from true values, but it does not consider the group structure of a ranked list. A ranking model optimized with a listwise loss function, on the contrary, can preserve the group structure of a ranked list, but it is incapable of predicting how many days it would take for a given used product to be sold.

The contributions of this paper are summarized as follows:

- We investigate the task of *used* product ranking which has its unique characteristics and challenges. To the best of our knowledge, the task has not been rigorously studied in the literature. We consider a novel time-aware metric—"sellability", which is defined as how quickly a used item is sold, to quantify the value of a *used* product.
- We propose a combined Poisson regression and listwise ranking model to predict the "sellability" of used products and to rank them. The model is optimized with both regression and ranking loss functions simultaneously, with the goal of predicting "sellability" accurately while preserving the ordering structure of products in a ranked list.
- We define a comprehensive set of regular product features and used product features to represent contents of used vehicle posts, and conduct thorough experiments in the domain of used vehicles. The experimental results demonstrate the effectiveness of our proposed approach. We also examine the most important features that contribute to a

Fig. 1 Sellability distribution of used vehicles



"sellable" used vehicle. We will make the data and code publicly available for the research community.

## 2 Related work

### 2.1 Product database search

The problem of retrieving records from product databases (Chang et al. 2004; Hristidis et al. 2003; Liu et al. 2006; Luo et al. 2007) has been studied by previous work. In particular, Chaudhuri et al. (2006) developed a Probabilistic Information Retrieval (PIR) algorithm incorporating global importance of attributes and correlations between attributes for database retrieval. They conducted experiments on the MSN HomeAdvisor database and the Internet Movie Database. Su et al. (2006) proposed a ranking approach—Query Result Ranking for E-commerce (QRRE)—that assigned different weights on different attributes, and tested it on Yahoo! Autos database and Yahoo! Real Estate database with five volunteers. Experiments showed that QRRE achieved better performance than PIR did. Telang et al. (2012) developed a similarity based ranking algorithm and tested it on two databases (a vehicle database and a real estate database provided by Google Base). Duan et al. (2013) developed a probabilistic approach to retrieve relevant products from databases leveraging user reviews in order to bridge the vocabulary gap between queries and product specifications. Recently, Park et al. (2015) proposed a mobile app retrieval model that effectively combines two different types of text data, which are app descriptions and user reviews. One limitation of the aforementioned studies is that there exists little user judgment on the relevance of those database records with respect to queries. Therefore, the proposed ranking functions had to be tested on small-scale datasets with a limited number of users. In our study, we rely on how quickly a used product is sold, namely "sellability" as implicit relevance feedback data to rank results. In this way, we can evaluate our algorithm on a large-scale dataset.

### 2.2 Product search and ranking on the web

Regular product search and ranking on the web has been studied in a few works. Guo and Agichtein (2010) mined user interactions with search results to detect user intents on products. Li et al. (2011) proposed a theory model for product search based on expected utility search from economics. Long et al. (2012) developed a ranking framework for enhancing product search based on best-selling prediction. Facet selection algorithm was proposed in Vandic et al. (2013) to minimize the number of steps needed to find a desirable product. Zhang et al. (2014) explored a Bayesian framework for modeling price preference. Pu et al. (2008) conducted two case studies to evaluate product search. To the best of our knowledge, none of the prior work targets on *used* product search and ranking.

With the rapid growth of World Wide Web, a large amount of implicit user feedback from end users is available from web search engine records. A number of studies (Wu and Bolivar 2009; Wang et al. 2010; Chung et al. 2012) make use of multiple types of implicit feedback (e.g. conversion probability, number of user clicks, etc.) to approximate the relevance labels for database results with respect to queries. The key idea is to build a classification/prediction model trained with a large amount of data. The model is usually optimized with pointwise loss functions, i.e., squared loss or logistic loss. Given a new query and its corresponding retrieved products, the results are ranked by the scores predicted by the model. Such an approach has been widely applied to product search and ranking on Customer-to-Customer (C2C) platforms such as eBay, Taobao, etc. Different from conventional implicit feedback, such as number of clicks or conversion probability, our study proposes a novel type of *time*-aware metric, which is more suitable for used products as discussed in Sect. 3.1.

### 2.3 User-generated content ranking

Our problem is similar to popularity prediction of user-generated content, e.g., tweet popularity prediction (Hong et al. 2011; Ma et al. 2012; Duan et al. 2010; Huang et al. 2011) and YouTube video popularity prediction (Borghol et al. 2012). As those tweets or videos are generated on the social web, users' information and their social network information are important resources for designing good ranking functions. Like prediction models for product ranking on C2C platforms, these ranking functions are also optimized with single loss functions, such as pointwise or pairwise loss functions. In our study, the proposed ranking model is learned with a combination of pointwise and listwise loss functions.

## 3 Methods

### 3.1 Time-aware metric: sellability

As the example shown in Table 1, all the four records match the user's query "*price = $10,000*" and "*Transmission = Automatic*". However, this kind of relevance alone is not sufficient, as there could be thousands of used products that are relevant to the user's need, which is known as the *Many-Answers* problem (Chaudhuri et al. 2004). In the domain of used products, time is essential for used product trading for both sellers and buyers due to the scarce nature of used products (i.e., used products are often unique or of limited quantities). If a used item posted by a seller cannot be traded quickly, its advertisement is likely to be buried by newly published advertisements on a used product channel. The seller then might have to re-post his/her item. Considering buyers are likely to worry about failing to capture a used product according to the *Principle of Scarcity* (Lynn 1989), a used product channel can utilize the tactic, suggesting that certain used products might soon be off the market, to drive sales.

However, existing metrics used for regular product ranking, e.g., price, sales, or conversion probability (Wu and Bolivar 2009), do not consider the *time* factor. Therefore, we introduce the concept of "sellability", which is defined as a time duration for a used product to be traded, to quantify the value of a used product. The time duration could be counted from the time when a used item is published until the time when it is traded, and could be represented by minutes, hours, or days.

The usages of "sellability" are beneficial for both parties of sellers and buyers who participate in used product trading: (1) when a seller is willing to trade his/her used products, the "sellability" can be used as a guide for publishing his/her products. Specifically, a prediction model, which is trained on historical dataset, can be used to estimate the "sellability" for newly generated advertisements of used products. If a seller is not satisfied with the predicted "sellability", he/she can revise the advertisement (e.g., price setting) to improve the estimated "sellability" value; and (2) when a buyer is browsing or searching for a used product that satisfies his/her need, which could be represented by a user query, a product or service that has high "sellability" is more likely to attract a buyer's attention, according to the scarcity heuristic (Lynn 1989), so they should be on the top of a ranked list. Specifically, the aforementioned prediction model can be used to rank used products based on their predicted "sellability" values. Therefore, our goal is to estimate the "sellability" of unseen used products (from sellers' perspective), and to rank a list of used products based on how soon they will be sold (from buyers' perspective).

### 3.2 Problem formulation

We consider the problem of predicting how quickly a newly published used product can be sold, namely "sellability". Hence, a list of used products could be ranked by their predicted "sellability" values. Considering the availability of a large volume of used products throughout different online channels, it is not wise to provide users with a global ranked list of used products. Instead, the ranked list should be relevant to user needs. Used product databases usually support search by either a simple Boolean query or a query conditioned on different schema attributes. Take an example shown in Table 1, when a user submits a query for a used vehicle, such as "*price = $10,000*" and "*Transmission = Automatic*", the database returns all tuples that satisfy the query. Our goal is to estimate the "sellability" values of corresponding used products and to rank them accordingly. It is noted that we treat all the tuples as relevant results that fulfill the user's need; and we aim to re-rank the results based on their estimated "sellability" values.

Formally, let $Q$ denotes the query space, and $P$ denotes the used product space. A dataset contains $N$ sets of data points: $\{q^{(i)}, \boldsymbol{p}^{(i)}, \boldsymbol{y}^{(i)}, \pi^{(i)}\}_{i=1}^{N}$, where $q^{(i)} \in Q$ is a query, $\boldsymbol{p}^{(i)} = \{\boldsymbol{x}_1^{(i)}, \ldots, \boldsymbol{x}_{m_i}^{(i)}\} \subseteq P$ is a set of used products that satisfy the query, $\boldsymbol{y}^{(i)} = \{y_1^{(i)}, \ldots, y_{m_i}^{(i)}\}$ is the label set representing "sellability", and $\pi^{(i)} = \{\pi_1^{(i)}, \ldots, \pi_{m_i}^{(i)}\}$ denote each used product's rank position in permutation that ranks in descending order of the product's "sellability". Each product associated with a query is represented by a group of features, so we have $\boldsymbol{x}_j^{(i)} = [x_{j0}^{(i)}, x_{j1}^{(i)}, \ldots, x_{jn}^{(i)}]^\top$. Here, we include a bias term $x_{j0}^{(i)} = 1$ in the feature set. The task is to learn a ranking function $f(\boldsymbol{x})$ that assigns a score to a newly published used product $\boldsymbol{x} \in P$ given a query $q \in Q$. $f(\boldsymbol{x})$ is expected to represent the distribution of "sellability" values (as shown in Fig. 1). We will discuss the choice of $f(\boldsymbol{x})$

in Sect. 3.3.2. The resultant ranking function can be applied to predict the “sellability” values of unseen used vehicles and to rank them accordingly.

## 3.3 Combined regression and listwise ranking model

Recall that our task is to predict “sellability” values for unseen used products (for sellers) as well as to rank a list of used products based on user’s query (for buyers). The ranking function $f(\boldsymbol{x})$ is expected to not only predict the “sellability” $y$ but also preserve the ordering structure of a ranked list of used products according to their “sellability” returned by a query. Formally, the expected risk $\sum_{i=1}^{N} L(f(\boldsymbol{p}^{(i)}); \boldsymbol{y}^{(i)}, \pi^{(i)})$ should be minimized with respect to both label set $\boldsymbol{y}^{(i)}$ and permutation $\pi^{(i)}$.

The prime motivation for our method is the need for the integration of regression models with ranking models. Regression models which are optimized with pointwise loss functions are effective for predicting exact values; while ranking models which are optimized with pairwise or listwise loss functions are effective for generating permutations (Cao et al. 2007). Ranking models that are obtained using listwise loss functions are more favored as they take into consideration the group structure of permutations (Xia et al. 2008). However, a perfect ranking model may not provide accurate regression values, as it might transform the order-preserving ground truth values. On the other hand, while a perfect regression model would yield perfect ranking results, in real-world cases, it is impossible to obtain perfect predictions by regression models, which are likely to cause ranking errors. Therefore, we aim to find a way to integrate pointwise regression and listwise ranking models such that the combined model could bring benefits from both methods.

In Sect. 3.3.1, we proposed a unified optimization framework that takes into consideration both regression and ranking losses. The combined loss aims to train a prediction model that can predict “sellability” values accurately, while preserving the group structure of a ranked list. In Sect. 3.3.2, we describe our choices of regression loss and ranking loss. In Sect. 3.3.3, we present the Stochastic Gradient Descent (SGD) algorithm to estimate model parameters.

### *3.3.1 Optimization framework*

Suppose the scoring function for predicting the “sellability” of a used product is a linear model, e.g., $f(\boldsymbol{x}) = \boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{x}$, where $\boldsymbol{x}$ represents a feature vector of a used product and $\boldsymbol{\theta}$ are model parameters. Recall that the ranking function is expected to predict the “sellability” of a used product accurately as well as to preserve the ordering structure of a ranked list of used products. Hence, our goal is to find the optimal parameters $\boldsymbol{\theta}$ that minimizes the following objective function:

$$L(\boldsymbol{\theta}) = L(\boldsymbol{\theta}; \boldsymbol{p}, \boldsymbol{y}) + \alpha L(\boldsymbol{\theta}; \boldsymbol{p}, \boldsymbol{\pi}) + \frac{1}{2}\lambda ||\boldsymbol{\theta}||_2^2 \tag{1}$$

where $L(\boldsymbol{\theta}; \boldsymbol{p}, \boldsymbol{y})$ is a loss function for regression, $L(\boldsymbol{\theta}; \boldsymbol{p}, \boldsymbol{\pi})$ is a loss function for ranking, $\alpha$ is a regularization constant for ranking loss, and $\lambda$ is a $L_2$ regularization constant for avoiding over-fitting of parameters $\boldsymbol{\theta}$. When $\alpha$ is set to 0, the cost function degenerates to one for regression. Here, the regression loss aims to optimize the errors between the predicted “sellability” values and ground truth values; and adding the ranking regularizer constraint aims to preserving the order of a ranked list of used products generated by user queries.

The above optimization framework is general. The regression loss $L(\boldsymbol{\theta};\boldsymbol{p},\boldsymbol{y})$ can be replaced by any pointwise loss function; and the ranking loss $L(\boldsymbol{\theta};\boldsymbol{p},\boldsymbol{\pi})$ can be fit with any pairwise or listwise loss function in the field of learning to rank (Liu 2009). Our method is similar to the combined regression and ranking framework proposed by Sculley (2010). The main difference is that his framework used a tradeoff parameter to sample pointwise examples or pairwise examples, which were used to optimize a linear ranking function using a pointwise loss function, e.g., squared loss or logistic loss. In our model, instead of training a ranking function using a single loss function, our model is based on the combination of pointwise and listwise loss functions.

### *3.3.2 Combined poisson regression and ListMLE ranking model*

As shown in Fig. 1, the "sellability" values for used products are non-negative and skewed to the left. In this case, Poisson distribution is a more reasonable assumption than Gaussian distribution. Thus, we choose to use Poisson regression to estimate the "sellability" value. For the ranking loss, we use the listwise loss function of ListMLE (Xia et al. 2008), which is desirable for order-preserving of a ranked list, and is one of the top performed listwise learning to rank algorithms (Lan et al. 2013).

The scoring function for poisson regression (PR) is:

$$f(\boldsymbol{x};\boldsymbol{\theta}) = \exp(\boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{x}) \tag{2}$$

Its loss function is defined as the negative log likelihood of the "sellability" values of used products:

$$LPR(\boldsymbol{\theta};\boldsymbol{p},\boldsymbol{y}) = -\sum_{i=1}^{N}\sum_{j=1}^{m_i}\left(y_j^{(i)}\left(\boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{x}_j^{(i)}\right) - \left(\boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{x}_j^{(i)}\right)\right) \tag{3}$$

The pointwise loss function considers residual errors with respect to each used product, but does not consider the group structure of a ranked list of used products. Hence, we incorporate a loss function of a ranking method, ListMLE (LM), which is defined over listwise difference of permutations between a perfect ranked list and a predicted ranked list:

$$LLM(\boldsymbol{\theta};\boldsymbol{p},\boldsymbol{\pi}) = -\sum_{i=1}^{N}\log\prod_{\mathrm{j=1}}^{\mathrm{m_i}}\frac{\exp\left(\mathrm{f}\left(\boldsymbol{x}_{\pi^{-1}(\mathrm{j})}^{(\mathrm{i})}\right)\right)}{\sum_{\mathrm{k=j}}^{\mathrm{m_i}}\exp\left(\mathrm{f}\left(\boldsymbol{x}_{\pi^{-1}(\mathrm{k})}^{(\mathrm{i})}\right)\right)} \tag{4}$$

where $\pi^{-1}(i)$ denotes the index of items in the $i$th position of $\pi$.

By instantiating the two loss functions in Eq. (1) by Eqs. (3) and (4), we have:

$$\min_{\boldsymbol{\theta}} \quad LPR(\boldsymbol{\theta};\boldsymbol{p},\boldsymbol{\pi}) + \alpha L_{\mathrm{LM}}(\boldsymbol{\theta};\boldsymbol{p},\boldsymbol{y}) + \frac{1}{2}\lambda||\boldsymbol{\theta}||_2^2 \tag{5}$$

### *3.3.3 Parameter estimation*

The objective function in Eq. (5) is a convex function, as it is the summation of three convex functions of parameters $\boldsymbol{\theta}$. It can be efficiently optimized by Stochastic Gradient Descent (SGD) algorithm (Algorithm 1) to obtain an approximate global optimal solution. The algorithm iterates over all the queries in the dataset, computes partial derivatives of $\boldsymbol{\theta}$

for the examples associated with each query, and updates $\boldsymbol{\theta}$ with respect to each query example after a maximum number of iterations $T$ or until the change of objective function $L(\boldsymbol{\theta})$ is small enough. To avoid any potential bias introduced by the arbitrary order queries in the training data, the queries are randomized before running SGD.

Given a training example $\{q^{(i)}, \boldsymbol{p}^{(i)}, \boldsymbol{y}^{(i)}, \pi^{(i)}\}$, the partial derivative of $\boldsymbol{\theta}$ is computed as follows:

$$\nabla_{\boldsymbol{\theta}}^{(i)} \ell(\boldsymbol{\theta}) = \frac{\partial \ell PR^{(i)}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} + \alpha \frac{\partial \ell LM^{(i)}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} + \lambda \boldsymbol{\theta},$$

where the derivatives of loss functions for Poisson regression and ListMLE are computed as follows:

$$\frac{\partial \ell_{\mathrm{PR}}^{(i)}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \sum_{j=1}^{m_i} \boldsymbol{x}_j^{(i)} \left( \exp(\boldsymbol{\theta}^{\mathrm{T}} \boldsymbol{x}_j^{(i)}) - y_j^{(i)} \right)$$

$$\frac{\partial \ell_{\mathrm{LM}}^{(i)}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \sum_{j=1}^{m_i} \left( \frac{\sum_{k=j}^{m_i} \boldsymbol{x}_{\pi^{-1}(k)}^{(i)} \exp(\boldsymbol{\theta}^{\mathrm{T}} \boldsymbol{x}_{\pi^{-1}(\mathrm{k})}^{(\mathrm{i})})}{\sum_{k=j}^{m_i} \exp(\boldsymbol{\theta}^{\mathrm{T}} \boldsymbol{x}_{\pi^{-1}(\mathrm{k})}^{(\mathrm{i})})} - \boldsymbol{x}_{\pi^{-1}(j)}^{(i)} \right)$$

**Algorithm 1:** The SGD Algorithm

```
input  : training data $\{q^{(i)}, \boldsymbol{p}^{(i)}, \boldsymbol{y}^{(i)}, \pi^{(i)}\}_{i=1}^{N}$, loss function trade off $\alpha$,
         regularization term $\lambda$, learning rate $\eta$, number of iterations $T$,
         threshold $\varepsilon$
output: model parameters $\boldsymbol{\theta}$
initialization $\boldsymbol{\theta} \leftarrow \mathbf{0}$, $L(\boldsymbol{\theta})^{(0)} \leftarrow \infty$
for $t = 1$ to $T$ do
    for $i = 1$ to $N$ do
        $\nabla_{\boldsymbol{\theta}}^{(i)} \ell(\boldsymbol{\theta}) = \frac{\partial \ell_{\mathrm{PR}}^{(i)}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} + \alpha \frac{\partial \ell_{\mathrm{PR}}^{(i)}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} + \lambda \boldsymbol{\theta}$
        $\boldsymbol{\theta}^{(t)} \leftarrow \boldsymbol{\theta}^{(t-1)} - \eta \nabla_{\boldsymbol{\theta}}^{(i)} \ell(\boldsymbol{\theta})$
    end
    $L(\boldsymbol{\theta})^{(t)} \leftarrow L_{\mathrm{PR}}^{(t)}(\boldsymbol{\theta}^{(t)}; \boldsymbol{p}, \boldsymbol{\pi}) +$
            $\alpha L_{\mathrm{LM}}^{(t)}(\boldsymbol{\theta}^{(t)}; \boldsymbol{p}, \boldsymbol{y}) + \frac{1}{2}\lambda ||\boldsymbol{\theta}^{(t)}||_2^2$
    if $L(\boldsymbol{\theta})^{(t-1)} - L(\boldsymbol{\theta})^{(t)} < \varepsilon$ then
        return $\boldsymbol{\theta}$
    end
end
return $\boldsymbol{\theta}$
```

# 4 Data collection

## 4.1 Gathering website data

There are many websites hosting second hand products, such as Amazon Used Product Store,[5] Used Products Vestigingen,[6] etc. In our study, we focus on used vehicle ranking as an example of used product ranking problem; and we rely on the analysis of Craigslist

[5] http://www.amazon.com/b?node=6943309011.

[6] http://www.usedproducts.nl/webshop/.

posts from San Francisco Bay Area[7] as a case study. Our methodology of collecting data can be applied to other websites hosting used vehicle information and other domains of used products.

We need to obtain the content of posts of used vehicles and their "sellability" values. However, Craigslist does not provide the information indicating how soon a used vehicle is sold. Therefore, we design a data crawl (see next section) which is used to obtain the number of days taken for a post to be removed to *approximate* the "sellability" values. The crawler runs everyday to obtain newly generated posts and to check if historical posts have been removed. Since there are thousands of used vehicle ads published on Craigslist everyday, it would make heavy requests to the server if the crawler checks all of their status every day, which is prohibited by Craigslist. Therefore, we select ten representative car manufacturers and models in our study: (1) manufacturers: *Benz*, *BMW*, *Chevrolet*, *Ford*, *Mazda*; and (2) models: *Accord*, *Civic*, *Camry*, *Corolla*, *Prius*, as they are popular American, German, or Japanese vehicles in the San Francisco Bay Area. These makes and models of cars are directly used as queries to retrieve posts from Craigslist. We gather data from April 16th, 2015 to July 10th, 2015, resulting in 60,943 records. Table 2 summarizes the number of posts in terms of different queries. It shows that the largest proportion (58.8%) of our data collection is Japanese vehicles (*Accord*, *Civic*, *Camry*, *Corolla*, *Mazda*, and *Prius*).

## 4.2 Obtaining "sellability"

In addition to the used vehicle posts, we need to have the ground truth indicating how a used vehicle is capable of being sold quickly, namely, "sellability". If a log server exists, such data could be easily obtained. Unfortunately, individual researchers usually have limited access to such information. Therefore, we utilize the number of days that a post of a used product is deleted by its seller as a *proxy* to its "sellability" value. The time duration taken for a post to be removed is thus used as implicit feedback to train the ranking model proposed in Sect. 3.3.1. The assumption behind using such a kind of implicit feedback needs to be validated in the future work. Figure 2 presents an overview of our methodology to crawl the dataset and obtain the number of days taken for a used vehicle post to be deleted. A data crawler is carried out everyday to gather posts and store them into a database.

For each day, the crawler first uses the ten car makes/models introduced in Sect. 4.1 as queries to retrieve posts from Craigslist. After obtaining a list of URLs of used vehicle posts, the crawler checks if each of the URLs has already been stored in the database. If it has not, the retrieved post will be added. Otherwise, the crawler checks the post content to decide the status of the post: (1) online post: if the content of a post remains unchanged, we assume the corresponding item has not been sold yet; (2) deleted post: we monitor 20 posts' status and contact their owners about their vehicles' availability. We observe that a seller tends to delete the post after he/she makes a deal with a buyer, otherwise potential buyers would continuously contact the seller. Therefore, we could use the number of days until a post is deleted to *approximate* the ground truth of how soon the vehicle is sold; (3) expired post: If a post has not been deleted by its author, it expires after seven days from the posting time; and (4) flagged post: some of the post content shows "flagged", which has been flagged by the website or customers. This category of posts could be useful for an

[7] https://sfbay.craigslist.org/search/cta.

**Table 2** Post distribution w.r.t queries

| | American vehicles | | Germany vehicles | |
|---|---|---|---|---|
| | Chevrolet | Ford | Benz | BMW |
| Number | 6064 | 5463 | 4270 | 9283 |
| Percentage | 9.95 | 8.96 | 7.01 | 15.23 |

| | Japanese vehicles | | | | | |
|---|---|---|---|---|---|---|
| | Accord | Camry | Civic | Corolla | Mazda | Prius |
| Number | 9552 | 6475 | 10,818 | 4427 | 2593 | 1998 |
| Percentage | 15.67 | 10.62 | 17.75 | 7.26 | 4.25 | 3.28 |

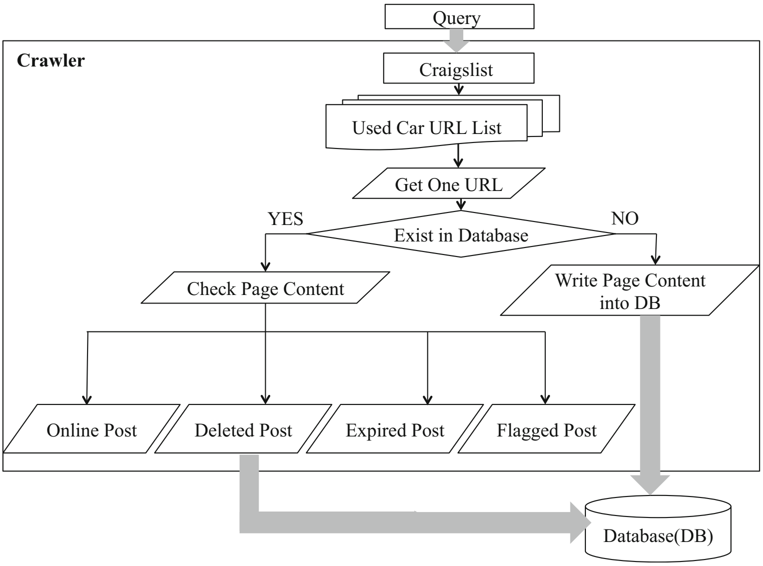


**Fig. 2** Overview of data crawler

interesting problem of identifying spams of used items, such as Park et al. (2014) and Wadleigh et al. (2015). We leave it for future work.

The post distribution of our dataset with respect to post status is shown in Table 3. In our study, we leverage only deleted posts for experiments, as we know the number of days taken for those posts to be deleted by their sellers. After removing invalid posts, which might be caused by network connection error during the data collection phase, the total number of resultant deleted posts is 25,410. As previously shown in Fig. 1, the“sellability”distribution seems to follow a Poisson distribution. Note that although sellers might repost their advertisements after they get expired, we use the day duration between the deletion time and original publishing time as the“sellability”value. That is the reason that the “sellability” values of nearly half (43.3%) of vehicles are greater than 7, which is the time before their corresponding posts get expired.

**Table 3** Post distribution w.r.t. post status

| | Online post | Deleted post | Expired post | Flagged post | Total post |
|---|---|---|---|---|---|
| Number | 17,103 | 26,658 | 14,119 | 3063 | 60,943 |
| Percentage | 28.06 | 43.74 | 23.17 | 5.03 | 100 |

## 4.3 Feature selection and extraction

In order to train our proposed ranking model to rank used products introduced in Sect. 3.3.2, we need to design features to represent them. The features can be categorized into two groups: regular product-specific features and used product-specific features. The former ones are directly related to products themselves, while the latter ones are related to the condition and owners of used products. The regular product-specific features might be different in different domains (i.e., item categories). For example, the features of used cameras may include lens, autofocus, resolution, etc. The features of used laptops may include CPU, memory, storage, etc.

Different from purchasing products from conventional suppliers, the trustworthiness of used products as well as people who provide those products might not be guaranteed in a used product market. Thus, the used product-specific features are designed to measure the used product conditions and trustworthiness of the owner of a used product. Examples of used product-specific features may include the length of content describing a used product, the sentiment expressed in the content, the number of images, the credibility of the owner, etc. Lewis (2011) argued that the more information, such as photos, text, and graphics, are provided to buyers, the less information gap exists between owners and buyers. Further, the information asymmetry caused by lemon market (Akerlof 1970), where owners of bad used items try to sell them to ill-informed buyers, might be mitigated.

In our study, we inspect a number of used vehicle posts from Craigslist displaying used vehicles and identify the following features.

### *4.3.1 Regular product-specific features*

Such type of features can be usually obtained from structured fields of a post. Those structured attributes directly reflect the basic information about a vehicle.

- *Make* This categorical feature describes the manufacturer of a used car. Examples of makes include: Ford, BMW, and Toyota.
- *Model* This categorical feature describes a specific type of a car. Examples of models include: Focus, 325i, and Camry.
- *Color* This categorical feature describes the exterior color of a car.
- *Transmission* This categorical feature describes the transmission type of a vehicle, which is the most important mechanic that controls a vehicle. The major two types of transmissions in our dataset is automatic and manual.
- *Car type* This categorical feature describes the design of a road vehicle. Examples of car types include: sedan, coupe, SUV, hatchback, etc.
- *Fuel* This categorical feature describes the motor fuel that is used to provide power to vehicles. Examples of motor fuel include gas, diesel, electric, etc.

- *Number of cylinders* This categorical features describe the power that a vehicle's engine can make. The larger number of cylinders, the more power an engine can make.
- *Wheel drive* This categorical feature describes the number of driven wheels of a vehicle.

### 4.3.2 Used product-specific features

In addition to features that are relevant to the basic information about a vehicle, we also design various features that are potentially useful to characterize used vehicles.

*Used Vehicle Condition*

- *Year* This numeric feature indicates how old a used car is. The less the age of a car, the lower its expected odometer reading.
- *Odometer* How many mileage a used car runs. This feature is usually negatively correlated with the feature *year*. The more mileage a car runs, the more year it likely has been used, and the less expensive price it is.
- *Price* This feature indicates the price asked from a used car's owner. It is highly possible that the price could be negotiable when a buyer and an owner make a deal. We discuss whether the asked price is important to make a used car post rank high in Sect. 7.1.
- *Car title status* This categorical feature indicates if there is any major physical issues with a used car. Such feature takes three values: clean, salvage, and rebuilt. This feature is associated with the quality of a car. A used car with salvage title usually comes with a very low price. A buyer might not want to buy a car with salvage title unless he has very little money.
- *Car condition* As mentioned, the quality of a used car must be different from a brand new car. The car condition is usually associated with the previously mentioned features. This categorical feature is provided based on owners' evaluations of their cars, so it is subjective. This feature takes four values: excellent, good, fair, and poor condition.

*Information disclosure by owners* The reputation of sellers is one of the important aspects for successfully trading on E-commerce sites. Therefore, it is beneficial to include features that are relevant to a seller's reputation, such as buyers' ratings or reviews. In this study, as Craigslist is an advertisements website, it does not keep track of transaction information between sellers and buyers. Therefore, we are unable to obtain such information. To mitigate this problem, we design several features that are used to measure whether a seller is willing to disclose more information about him/herself or the used product he/she owns. Lewis found that such features can be used to reflect the trustworthiness of a seller for used product trading (Lewis 2011).

- *Location* This categorical feature describes the city where a used vehicle is located. Ideally, using the distance of a seller from a buyer is better than using the location of the seller. However, as mentioned in Sect. 4.2, we cannot access to the server, so we are unable to obtain buyer's information.
- *Content length* This numerical feature indicates the number of words that the content of a post contains. Sellers may provide additional descriptions about their used cars instead of structured attributes mentioned above. Intuitively, longer content is apt to contain more information that a short one is. We rely on NLTK (Bird 2006) to

preprocess the content, such as sentence segmentation, tokenization, and lemmatization.

- *Ratio of positive/negative sentiment words* We adopt a sentiment lexicon Multi-perspective Question Answering (MPQA) (Wiebe et al. 2005) to identify sentiment words from post content. This feature is used to measure the degree of emotion expressed by a seller. Intuitively, the more sentiment words occur, the more likely that a seller is emotionally invested in his/her car.
- *Ratio of sentences containing sentiment words* Similar to the previous feature, this numerical feature also indicates the emotion from a seller. An example of such sentences is: *This car is very reliable.*
- *Ratio of capitalized words*: Similar to sentiment words, some capitalized words also indicate sentiments, such as *GREAT CAR*.
- *Ratio of modifiers* This feature aims to indicate the richness of post contents. More adjectives or adverbs make content more descriptive.
- *Overall content sentiment* This categorical feature presents the sentiment polarity of each post, namely positive, negative, or neural. We rely on the Sentiment Tool[8] to derive this feature.
- *Confidence score of overall content sentiment* This numerical feature describes the confidence score of the prediction of sentiment polarity obtained by Sentiment Tool.
- *Existence of VIN number* This boolean feature indicates whether the owner of a used car posts the Vehicle Identification Number (VIN), which is a unique 17 letters and numbers assigned to a vehicle when it is built. If such a number of provided, its maintenance records could be retrieved from online services, such as CARFAX.[9]
- *Number of images* This numeric feature identifies how many photos an owner posts about his/her used car. The photos usually show a car’s exterior, interior, and mechanical engines, as well as whether any parts of the car are broken or scratched. Applying image processing techniques to identify those sections is out of the scope of our study.
- *Pixels of images* This numeric feature identifies the clarity of the images posted. Clearer images might attract more attention than blur ones.
- *Missing attributes* Due to the characteristics of user generated content, the datasets contain a lot of noises and missing attributes. This feature aims to measure the completeness of a post. Specifically, we design a boolean feature for each categorical feature (e.g., the odometer or the age of a used vehicle) indicating whether its value is missing or not.

# 5 Experiments

## 5.1 Constructing queries

In order to train a ranking model, we need a set of queries associated with posts and their labels. The posts and label data can be obtained using the data collection method introduced in Sect. 4. However, due to unavailable access to commercial log servers, it is

[8] http://sentiment.vivekn.com/.

[9] http://www.carfax.com/.

challenging to obtain real user queries. Therefore, we follow the strategy introduced in Telang et al. (2012) to construct user queries and thus obtain datasets for experiments.

We first select five attributes that Craigslist provides for searching a used vehicle: *year*, *price*, *odometer*, *make*, and *model*. For each attribute, we construct a set of selection queries to retrieve posts obtained in Sect. 3. Examples of queries include "\$10,000 $\leq$*price*$<$ \$11,000" or "*model* = *focus*". While designing the queries, we make sure two different queries do not result in overlapping results. However, given a simple query, there are too many retrieved posts, e.g., the averaged number of posts per query in regards to different price ranges is 1, 240. So we partition each subsets of posts with respect to each query based on their posted dates under the assumption that the post distribution on each day is the same. Queries whose number of associated posts is less than 5 are removed. Hence, ranking results will be generated and evaluated for each daily query. Table 4 summarizes the query information. In addition to the five single classes of query, we also combine two of those queries to make composite queries, such as "*car_price* $>$ \$10,000 *and car_price* $<=$ \$11,000 *and model* = *Camry*". Note that it is possible that users construct more complex queries, e.g., three query conditions. Since the average number of posts per query for the two query condition dataset is only 6.4, the data that satisfies complex queries would be quite sparse. Hence, we only include data retrieved by two query conditions in our dataset and leave complex queries for future work.

Each of the query types is associated with a dataset, namely *price*, *year*, *odometer*, *make*, *model*, and *composite* query datasets. For each dataset, we follow the data parition of benchmark datasets [e.g., LETOR (Liu et al. 2007)] for learning to rank models to create our experimental datasets. Specifically, we randomly split those datasets into three partitions based on the number of daily queries (the second line in Table 4), for training (60%), development (20%), and testing (20%), respectively. The statistics of data splits are summarized in Table 5.

## 5.2 Experimental setups

Our algorithm has five parameters required tuning as mentioned in Algorithm 1. In order to test the effectiveness of our Combined Poisson Regression and Listwise ranking model (CPL), we explore different settings of $\alpha$ in intervals of 0.1 between 0 and 1. The changes of the regularization term $\lambda$ do not have significant impacts on the performance of validation sets, so we set $\lambda = 0.005$ for all experiments. The learning rate $\eta$ is set to $5e-4$. The algorithm dynamically shrinks it by 50% during the SGD optimization when the total loss is greater than that obtained in previous iteration. To end up the algorithm, the number

**Table 4** Statistics of retrieved results using different types of queries

| Query type | Price | Year | Odometer | Make | Model | Composite |
|---|---|---|---|---|---|---|
| #Queries | 18 | 28 | 22 | 7 | 24 | 12 |
| #Daily queries | 1192 | 1660 | 887 | 487 | 510 | 1584 |
| #Avg. posts | 18.73 | 12.73 | 13.53 | 48.47 | 30.74 | 6.4 |
| #Std. posts | 13.38 | 6.69 | 6.2 | 34.87 | 19.67 | 1.8 |
| #Min. posts | 5 | 5 | 5 | 5 | 5 | 5 |
| #Max. posts | 77 | 40 | 39 | 135 | 84 | 18 |

**Table 5** Statistics of data splits for different types of queries

| Query type | Price | Year | Odometer | Make | Model | Composite |
|---|---|---|---|---|---|---|
| #Daily queries (train) | 716 | 996 | 533 | 293 | 306 | 952 |
| #Daily queries (dev.) | 238 | 332 | 177 | 97 | 102 | 316 |
| #Daily queries (test) | 238 | 332 | 177 | 97 | 102 | 316 |
| #Posts (train) | 12,929 | 12,571 | 7217 | 13,277 | 9626 | 6021 |
| #Posts (dev.) | 4688 | 4375 | 2396 | 5315 | 2964 | 2095 |
| #Posts (test) | 4711 | 4199 | 2392 | 5015 | 3087 | 2027 |

of iterations $T$ is set to 30 and convergence threshold $\varepsilon$ is set to $1e-4$. Our proposed models are obtained using the entire training data, and their performance is evaluated on test data. The parameters are determined on validation data.

### 5.3 Baselines

We compare our combined Poisson Regression and ListMLE (CPL) ranking approach against the following nine baselines. The first three non-machine learning (non-ML) ranking approaches rank posts of used products using specific factors, and consequently, they do not require any training steps. The other six approaches are the state-of-the-art pointwise, pairwise, and listwise learning to rank methods (Liu 2009). As a type of supervised machine learning techniques, learning to rank methods require a set of training examples consisting of a query set along with their associated posts represented by feature vectors and labels. Similar to training our proposed ranking model introduced in Sect. 3, we use the same data (see Sect. 5.1) and feature set (Sect. 4.3) to train the six learning to rank baselines. As introduced in Sect. 4.2, we use the number of days taken for a post to be removed to *approximate* the "sellability" value of each used product, and thus to represent the label of each query-product pair. It is noted that all the ranking models will rank used product posts with the same ranking scores arbitrarily.

*Non-Machine learning baselines*

- *Posted time* Ranked based on the posted time of a post. This is a normal ranking approach adopted by Craigslist.
- *Price from low to high (PriceLow)* Ranked based on the price value in ascending order. This is an alternative ranking approach provided by Craigslist.
- *Price from high to low (PriceHigh)* Ranked based on the price value in descending order. This is also an alternative ranking approach provided by Craigslist.

*Learning to rank baselines*

- *Linear regression* Ranked based on prediction scores by a linear regression model. Similar to our proposed ranking model, the linear regression model is trained with a number of posts using used products' approximated "sellability" values as labels. This approach has been applied to product conversion probability prediction (Wu and Bolivar 2009).

- *Logistic regression* Ranked based on prediction scores by a multi-class logistic regression model. Similar to our proposed ranking model, the logistic regression model is trained with a number of posts using used products' approximated "sellability" values as labels. The training algorithm uses the one-vs-rest scheme. This approach has been applied to product conversion probability prediction (Wu and Bolivar 2009) and click prediction for product search (Wang et al. 2010).
- *Poisson regression* Ranked based on prediction scores by a Poisson regression model optimized with pointwise errors between the predicted value of "sellability" and ground truth value of a number of posts. We train Poisson regression models using our proposed approach with the same parameter settings introduced in Sect. 5.2.
- *RankNet* Ranked based on the prediction scores by a neural network learned with pairwise examples (Burges et al. 2005). We rely on RankLib[10] to train the model with default parameter settings.
- *ListMLE* Ranked based on prediction scores by a linear regression model optimized with listwise errors between a predicted ranked list and a perfect ranked list of posts. We train ListMLE models using our proposed approach without adding the pointwise Poisson regression loss.
- *Combined regression and ranking (CRR)* Ranked based on the prediction scores by a combined pointwise and pairwise ranking model (Sculley 2010). This combined model differs from ours in that it first used a tradeoff coefficient to select pointwise or pairwise examples, and then was trained with a pointwise loss function (e.g. squared loss), while ours use a ranked list as an example and is trained with a combination of pointwise and listwise loss functions.

### 5.4 Evaluation metrics

In order to determine the best ranking models, we use the following metrics for evaluation.

*Mean squared error* (*MSE*)

$$\mathrm{MSE} = \frac{1}{\mathrm{m}}\sum_{\mathrm{i}=1}^{\mathrm{m}}(\mathrm{f}(\boldsymbol{x}^{(\mathrm{i})}) - \mathrm{y}^{(\mathrm{i})})^2, \tag{6}$$

where $f(\boldsymbol{x}^{(i)})$ denotes the predicted "sellability" value of a post, $y^{(i)}$ denotes the ground truth value, and $m$ is the total number of posts in the test set. MSE is designed to measure the performance of regression approaches, e.g., linear regression or Poisson regression. It does not consider query information.

*Normalized discounted cumulative gain* (*NDCG*)

$$\mathrm{NDCG@k} = \frac{1}{|\mathrm{Q}|}\sum_{\mathrm{q}\in\mathrm{Q}}\frac{1}{\mathrm{Z_q}}\sum_{\mathrm{i}=1}^{\mathrm{k}}\frac{2^{\mathrm{r_i^q}} - 1}{\log(1+\mathrm{i})}, \tag{7}$$

where $Q$ is the set of test queries, $k$ indicates the top $k$ results in a ranked list, $Z_q$ is a normalization factor, and $r_i^q$ indicates the "sellability" value of the result in the $i$th position in the ranked list in descending order of "sellability" values for query $q$. This metric measures the ranking quality with respect queries at position $k$ (Järvelin and Kekäläinen 2000). We set $k$ to 1, 3, 5, and 10, respectively.

[10] http://sourceforge.net/p/lemur/wiki/RankLib/.

# 6 Results

## 6.1 Performance evaluation

The averaged results of five fold cross validation on the six datasets introduced in Sect. 5.1 obtained by different methods are shown from Tables 6, 7, 8, 9, 10 and 11. The three non-machine learning (non-ML) baselines cannot predict the "sellability" values of used products, so the MSEs are not provided. Boldface stands for best performance per column. We conduct paired *t* tests without applying corrections for all the comparisons of results achieved by two different methods. Interestingly, for all datasets but the price and composite query datasets, the NDCG values of ranking results sorted by *price* in ascending order are significantly better than that by posted time and price in descending order ($p < 0.001$). The price and composite query datasets, priceLow outputforms postedTime significantly at $p < 0.001$. Moreover, for the three pointwise learning to rank baselines, in terms of NDCG values, it is surprising that linear regression and logistic regression are inferior to PriceLow on all but the price dataset, except NDCG@1 achieved by logistic regression on the make dataset. PriceLow even outperforms RankNet on the odometer, make, and composite query dataset. These figures indicate that lower price is an important factor that determines a "sellable" used vehicle (see Sects. 6.3, 7.1 for more analysis).

For learning to rank baselines, Poisson regression outperforms the other two pointwise approaches in terms of MSE and NDCG values on all the datasets. This is probably because of the fact that Poisson regression fits the distribution of our dataset well. Poisson regression achieves the best MSEs on all datasets, due to the fact that RankNet and listMLE only consider the pairwise or listwise ranked order for each query but ignore the difference between predicted values and ground truth values of "sellability". Poisson regression and listMLE outperform RankNet on all the datasets for NDCG at different levels. The combined model CRR outperforms linear regression and logistic regression on all datasets due to the benefit of training with pairwise examples. It also yields better results than RankNet does on all datasets.

**Table 6** Price query ranking results

| Method | MSE | N@1 | N@3 | N@5 | N@10 |
|---|---|---|---|---|---|
| postedTime | – | 0.209 | 0.275 | 0.347 | 0.457 |
| priceLow | – | 0.286 | 0.359 | 0.430 | 0.528 |
| priceHigh | – | 0.281 | 0.356 | 0.421 | 0.524 |
| Linear | 72.759 | 0.348 | 0.408 | 0.473 | 0.564 |
| Logistic | 77.128 | 0.326 | 0.385 | 0.451 | 0.550 |
| Poisson | 44.018 | 0.496 | 0.559 | 0.632 | 0.694 |
| RankNet | 571.814 | 0.393 | 0.446 | 0.509 | 0.595 |
| listMLE | 539.908 | 0.472 | 0.534 | 0.616 | 0.681 |
| CRR | 49.197 | 0.422 | 0.489 | 0.576 | 0.647 |
| CPL | **43.923**$^{\dagger,\ddagger}$ | **0.502**$^{\dagger,\ddagger}$ | **0.563**$^{\dagger,\ddagger}$ | **0.635**$^{\dagger,\ddagger}$ | **0.697**$^{\dagger,\ddagger}$ |

**Table 7** Year query ranking results

| Method | MSE | N@1 | N@3 | N@5 | N@10 |
|---|---|---|---|---|---|
| postedTime | – | 0.236 | 0.320 | 0.408 | 0.532 |
| priceLow | – | 0.377 | 0.462 | 0.540 | 0.643 |
| priceHigh | – | 0.207 | 0.279 | 0.378 | 0.519 |
| Linear | 62.728 | 0.354 | 0.421 | 0.498 | 0.607 |
| Logistic | 74.752 | 0.361 | 0.429 | 0.507 | 0.618 |
| Poisson | 43.494 | 0.528 | 0.597 | 0.691 | 0.746 |
| RankNet | 578.681 | 0.381 | 0.457 | 0.535 | 0.636 |
| listMLE | 541.457 | 0.502 | 0.579 | 0.674 | 0.731 |
| CRR | 48.838 | 0.463 | 0.541 | 0.642 | 0.707 |
| CPL | **43.390**$^{\dagger,\ddagger}$ | **0.531**$^{\ddagger}$ | **0.599**$^{\ddagger}$ | **0.691**$^{\ddagger}$ | **0.746**$^{\ddagger}$ |

**Table 8** Odometer query ranking results

| Method | MSE | N@1 | N@3 | N@5 | N@10 |
|---|---|---|---|---|---|
| postedTime | – | 0.259 | 0.320 | 0.390 | 0.525 |
| priceLow | – | 0.373 | 0.437 | 0.510 | 0.630 |
| priceHigh | – | 0.219 | 0.301 | 0.374 | 0.516 |
| Linear | 68.654 | 0.358 | 0.422 | 0.492 | 0.614 |
| Logistic | 77.678 | 0.347 | 0.405 | 0.471 | 0.598 |
| Poisson | 45.040 | 0.499 | 0.559 | 0.675 | 0.737 |
| RankNet | 573.825 | 0.340 | 0.396 | 0.468 | 0.593 |
| listMLE | 543.648 | 0.484 | 0.545 | 0.657 | 0.724 |
| CRR | 50.039 | 0.447 | 0.510 | 0.629 | 0.704 |
| CPL | **44.741**$^{\dagger,\ddagger}$ | **0.509**$^{\ddagger}$ | **0.572**$^{\dagger,\ddagger}$ | **0.681**$^{\dagger,\ddagger}$ | **0.742**$^{\dagger,\ddagger}$ |

**Table 9** Make query ranking results

| Method | MSE | N@1 | N@3 | N@5 | N@10 |
|---|---|---|---|---|---|
| postedTime | – | 0.179 | 0.210 | 0.249 | 0.308 |
| priceLow | – | 0.326 | 0.341 | 0.379 | 0.437 |
| priceHigh | – | 0.180 | 0.212 | 0.248 | 0.319 |
| Linear | 63.244 | 0.310 | 0.315 | 0.344 | 0.403 |
| Logistic | 75.897 | 0.291 | 0.304 | 0.332 | 0.403 |
| Poisson | 43.478 | 0.402 | 0.427 | 0.484 | 0.556 |
| RankNet | 573.754 | 0.322 | 0.348 | 0.385 | 0.442 |
| listMLE | 546.002 | 0.360 | 0.389 | 0.448 | 0.525 |
| CRR | 48.491 | 0.333 | 0.367 | 0.427 | 0.507 |
| CPL | **43.403**$^{\dagger,\ddagger}$ | **0.407**$^{\ddagger}$ | **0.431**$^{\ddagger}$ | **0.488**$^{\ddagger}$ | **0.559**$^{\ddagger}$ |

Our proposed method CPL achieves the lowest residual errors and highest ranking performance. $^{\dagger}$ and $^{\ddagger}$ indicate whether the evaluation metrics achieved by CPL are significantly ($p < 0.001$) better that those achieved by Poisson regression and listMLE respectively. Specifically, CPL can reduce MSEs achieved by listMLE significantly on all datasets. Even though the improvement on the MSEs is small (between 2% and 7%), CPL

**Table 10** Model query ranking results

| Method | MSE | N@1 | N@3 | N@5 | N@10 |
|---|---|---|---|---|---|
| postedTime | – | 0.239 | 0.289 | 0.333 | 0.400 |
| priceLow | – | 0.397 | 0.426 | 0.463 | 0.520 |
| priceHigh | – | 0.236 | 0.287 | 0.351 | 0.425 |
| Linear | 48.890 | 0.341 | 0.384 | 0.429 | 0.484 |
| Logistic | 58.260 | 0.348 | 0.384 | 0.429 | 0.492 |
| Poisson | 37.321 | 0.469 | 0.511 | 0.561 | 0.627 |
| RankNet | 615.821 | 0.379 | 0.433 | 0.475 | 0.532 |
| listMLE | 588.612 | 0.437 | 0.480 | 0.535 | 0.602 |
| CRR | 42.021 | 0.411 | 0.455 | 0.510 | 0.582 |
| CPL | **37.133**$^{\dagger,\ddagger}$ | **0.484**$^{\ddagger}$ | **0.526**$^{\dagger,\ddagger}$ | **0.574**$^{\dagger,\ddagger}$ | **0.637**$^{\dagger,\ddagger}$ |

**Table 11** Two query condition ranking results

| Method | MSE | N@1 | N@3 | N@5 | N@10 |
|---|---|---|---|---|---|
| postedTime | – | 0.337 | 0.459 | 0.609 | 0.697 |
| priceLow | – | 0.424 | 0.532 | 0.666 | 0.738 |
| priceHigh | – | 0.394 | 0.504 | 0.650 | 0.722 |
| Linear | 65.851 | 0.411 | 0.530 | 0.667 | 0.735 |
| Logistic | 69.309 | 0.416 | 0.529 | 0.665 | 0.736 |
| Poisson | 41.934 | 0.597 | 0.719 | 0.773 | 0.776 |
| RankNet | 592.781 | 0.408 | 0.520 | 0.660 | 0.730 |
| listMLE | 563.599 | 0.593 | 0.714 | 0.772 | 0.774 |
| CRR | 47.332 | 0.562 | 0.687 | 0.752 | 0.754 |
| CPL | **41.692**$^{\dagger,\ddagger}$ | **0.605**$^{\ddagger}$ | **0.725**$^{\dagger,\ddagger}$ | **0.778**$^{\dagger,\ddagger}$ | **0.780**$^{\dagger,\ddagger}$ |

yields significantly better MSEs even at 0.001 level on all datasets compared with Poisson regression. For ranking performance, CPL significantly improves the NDCG values at all levels achieved by learning to rank baselines on the odometer, model, and composite query datasets except NDCG@1. Although CPL does not achieve significant NDCG values compared with Poisson regression on the other three datasets, it outperforms listMLE significantly at all levels of NDCG. It also outperforms CRR on all datasets in terms of all evaluation metrics.

In summary, PriceLow surpasses the other two non-ML baselines and even linear regression, logistic regression, and RankNet in terms of NDCG values. The results achieved by learning to rank approaches for different datasets are different due to the nature of their varied partitions. Our combined method CPL integrates the advantages from Poisson regression for prediction and ListMLE for ranking, and shows robustness to different partitions of datasets. It achieves the best performance of NDCG values and MSEs on all datasets with significant improvements.

## 6.2 Parameter analysis

In order to examine the impact of the regularization parameter $\alpha$ on our proposed method CPL, we examine performances achieved by different settings of $\alpha$ on one fold of test set. Figures 3 and 4 show the MSE and NDCG@10 values achieved by different $\alpha$ between 0

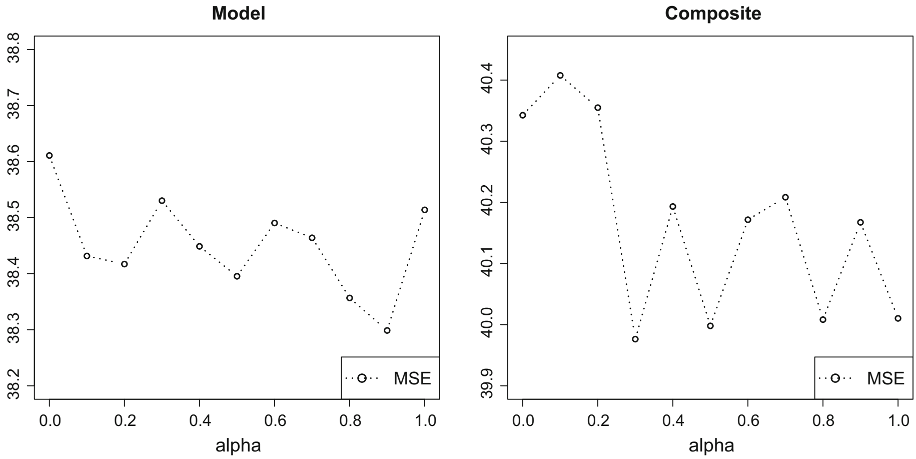


**Fig. 3** Regression performance

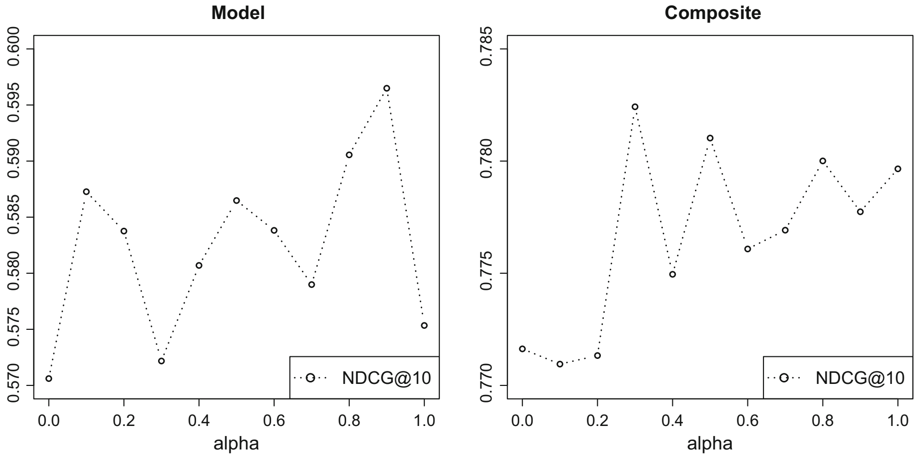


**Fig. 4** Ranking performance

and 1 with an interval of 0.1 on single (model) and composite query datasets. Other datasets also demonstrate similar patterns so we leave them out. When $\alpha = 0$, the MSE is obtained by Poisson regression. While $\alpha$ is increasing, the MSEs achieved on the model dataset are decreasing except when $\alpha = 0.3$, $\alpha = 0.6$ and $\alpha = 1.0$. On the composite query dataset, the MSE are increasing first and then decreasing sharply when $\alpha = 0.3$, when the MSEs reach the lower points. As $\alpha$ is getting bigger, the MSEs start to fluctuate, but they are still lower than that achieved by Poisson regression.

In terms of NDCG@10, the values are higher than that achieved by Poisson regression while $\alpha$ is growing, except when $\alpha = 0.1$ and 0.2 on the composite query dataset. The trend of NDCG curves is opposite to that of MSE curves. When MSE decreases, the corresponding NDCG value increases, and vice verse. In our experiment, we select the best $\alpha$ values based on the lowest MSEs on the validation set. On the model dataset, $\alpha = 0.9$ is the best choice, but since the MSEs achieved on other $\alpha$ values are also lower than that by

**Table 12** Summary of feature importance on different datasets

| Query type | Feature impact | Top_1 | Top_2 | Top_3 |
|---|---|---|---|---|
| Price | Positive | Location (San Jose) | Location (Saratoga) | Make (Toyota) |
| | Negative | Price | Location (Stockton) | Location (Campbell) |
| Year | Positive | Location (San Jose) | Location (Saratoga) | Make (Honda) |
| | Negative | Price | Location (Stockton) | Location (Concord) |
| Odometer | Positive | Location (Vallejo) | Condition (like new) | Make (Toyota) |
| | Negative | Price | Location (Concord) | Location (Hayward) |
| Make | Positive | 4 cylinders | Make (Toyota) | Title (clean) |
| | Negative | Price | Location (Concord) | Location (Stockton) |
| Model | Positive | Title (clean) | Transmission (automatic) | 4 cylinders |
| | Negative | Is_price | Price | Is_odometer |
| Composite | Positive | Condition (like new) | Location (Vallejo) | Make (Toyota) |
| | Negative | Location (Stockton) | Model (Protege) | Location (Concord) |

Poisson regression, they could also be selected. On the composite query dataset, $\alpha$ between 0.3 and 1.0 would be good choices.

In summary, our model takes advantage of Poisson regression, which is capable of predicting "sellability" values, and ListMLE, which enjoys the property of preserving order of a ranked list. It yields lower regression errors and higher ranking performance.

### 6.3 Feature analysis

In order to analyze features that are particularly important for used vehicle ranking, we examine the feature weights of the best CPL model on each dataset and list the top 3 important features that have positive and negative impacts on predicting "sellability" in Table 12. Overall, the feature weighting across different datasets are slightly different.

As shown in Sect. 6.1, lower price is a powerful indicator to achieve higher performance of a ranked list. Not surprisingly, the price feature plays the most important role on all single query datasets. The negative feature weight indicates lower prices will have smaller negative impact on the predicted value, and thus the corresponding vehicles are likely to be "sellable" used vehicles.

Location is another important factor that buyers would consider when finding a used car. Used vehicles are more likely to be sold quickly in San Jose and Saratoga when price range or year range is similar, and in Vallejo when mileage range is similar. Stockton, Campbell and Concord are the locations where used vehicles are not sold very quickly. Such differences are probably related with the population difference. The population of San Jose is 945,942, while that of Stockton is 291,707 as of the 2010 U.S. Census.[11] It is possible that more people consume more vehicles and thus drive more used-vehicles sales. Different from price, year, and odometer datasets, where top important features are more relevant to used product features (e.g., location, car condition), on the make and model-generated datasets, the top 3 positive features are related to regular product features (Sect. 4.3.1). On the composite query dataset, the features are mixed.

[11] http://factfinder.census.gov/faces/tableservices/jsf/pages/productview.xhtml?pid=DEC_10_SF1_GCTPH1.ST10.

Among those features, missing attribute, text, and sentiment features are those that neither play the most or least important roles in deriving a good prediction/ranking model. On the model dataset, two features indicating providing price or odometer by a seller might not contribute to high “sellable” used vehicles. This seems to be inconsistent with previous finding that information disclosure will mitigate the problem of “information asymmetry” between a buyer and a seller and thus makes used vehicle trading easily (Lewis 2011). One possible reason is that we leverage only deleted posts to train those models (Sect. 4.2). The used vehicles of expired posts are possibly those that take longer time to be sold, and could be representative for “unsellable” used vehicles. Thus, in next section, we examine some factors that contribute to differentiate “sellable” used vehicles and “unsellable” used vehicles.

## 7 Empirical analysis

### 7.1 The impact of price

As shown in performance comparison of different non-ML baselines in Sect. 6.1, lower price is a powerful indicator to achieve higher accuracy for a ranked list. Considering price value is relevant to some factors, such as the age of a used vehicle, how many mileage it runs, the manufacture and car models, etc., we analyze the price differences in terms of four attributes, year, odometer, make, and model, which are also the queries we apply to build ranking models.

According to Table 3, we select top 10,000 sellable used vehicles from deleted posts and top 10,000 unsellable used vehicles from expired posts based on their “sellability” values, and examine the price differences between the two groups with respect to year, odometer, make and model. Figure 5a shows the plots of averaged price values in terms of how old a used vehicle is. Basically, the older a used vehicle is, the less expensive the price is. The averaged prices with respect to different ages of sellable used vehicles are consistently lower than that of unsellable used vehicles. Similarly, the averaged price per odometer range of sellable used vehicles is also lower than that of unsellable used vehicles

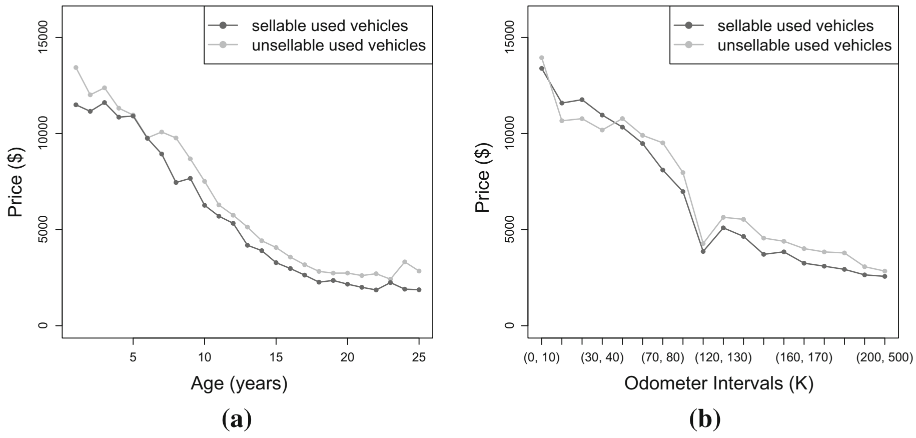


**Fig. 5** Price distributions. **a** Age, **b** odometer

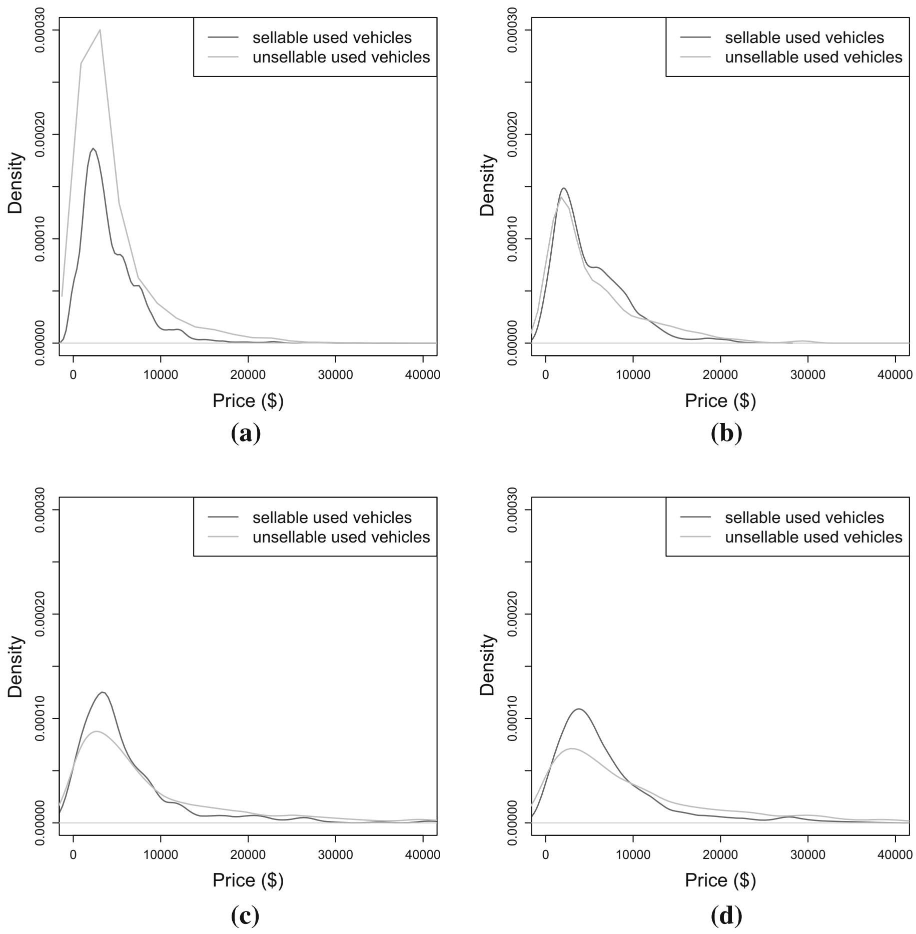


**Fig. 6** Price distribution w.r.t. car makes. **a** Honda, **b** Toyota, **c** Chevrolet, and **d** BMW

except odometers between 10 and 50 k (shown in Fig. 5b). These two findings are consistent with the previous findings that PriceLow is superior to the other two non-ML baselines to generate a ranked list on year and odometer-based datasets.

Figure 6 shows the density of price values in terms of four vehicle manufacturers, Honda, Toyota, Chevrolet, and BMW. The price range is fixed to the same range (0 to $40,000) for easy comparison. We also include those whose prices are not provided by sellers (indicated by the value −1). Unsellable used vehicles contain more unpriced vehicles compared to sellable vehicles. The price distributions of the four car makes are different. For Japanese cars, there are more cheap but unsellable Honda vehicles; while the price range for Toyota vehicles is similar for both sellable and unsellable used vehicles. On the contrary, Chevrolet and BMW vehicles have more cheap unsellable used vehicles.

Similarly, the price density of different car models are also different (shown in Fig. 7). There are more cheap but unsellable Civic cars, which correspond to the large number of unsellable Honda vehicles. There are more cheap sellable Prius used vehicles but more expensive unsellable Prius used vehicles. For the two models of vehicles made in BMW,

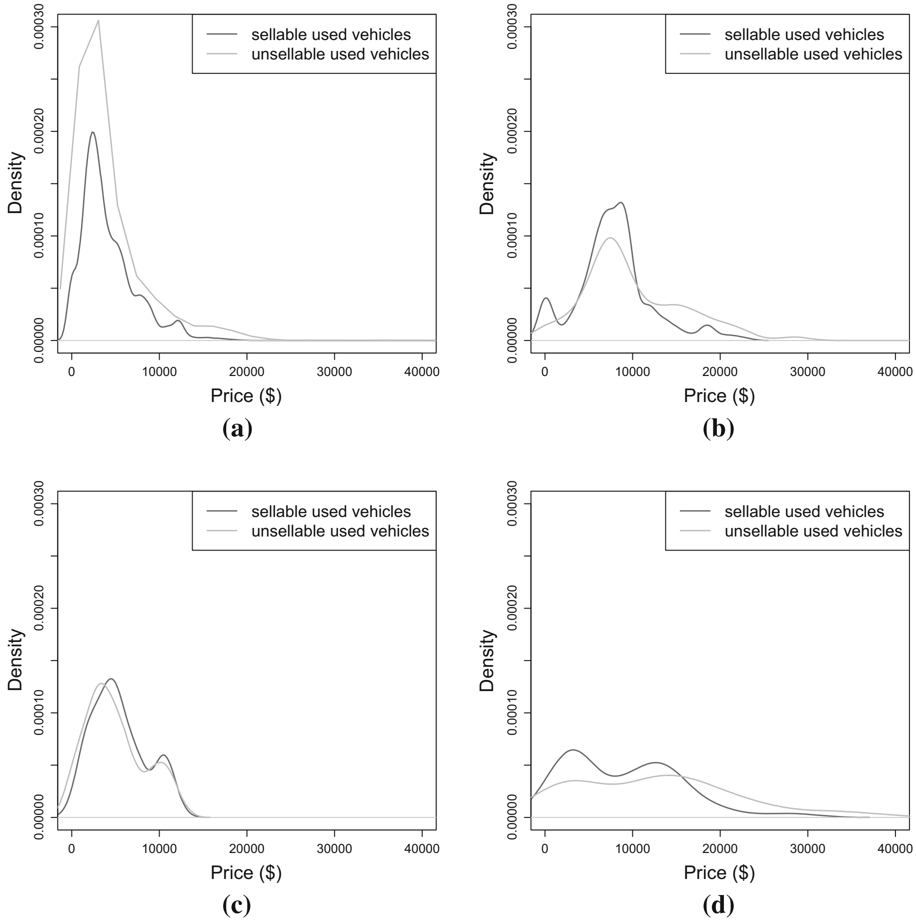


**Fig. 7** Price distribution w.r.t. car models. **a** Civic, **b** Prius, **c** 325i, and **d** 328i

328i have more cheap sellable used vehicles, but the range distributions between sellable and unsellable used vehicles are quiet similar to each other. Even though the relationships between price and make/model are not as consistent as that between price and year/odometer, the non-ML baseline PriceLow is still useful because some manufactures or models are more incline to cheaper deals.

## 7.2 The impact of used product-specific features

In addition to price, we aim to explore whether used product-specific features affect the "sellability" of used vehicles. We use the features introduced in Sect. 4.3 to represent the two groups of data introduced in the previous section, and compute the mutual information of each feature. Table 13 list the top important features that contribute to separating the two categories of used vehicles. Note that this table is different from Table 12, which shows the important features obtained by CPL trained with only deleted posts on different datasets. It shows that top six features are all related to the content of a post, in particular the sentiments expressed in the posts ($p < 0.001$). The results of two-sample $t$ tests indicate that

**Table 13** Top 10 important features in differentiating sellable and unsellable used vehicles

| Feature | Mutual information | Sig. level of* mean difference |
|---|---|---|
| Ratio of modifiers | 1.249 | $p = 0.816(+)$ |
| Ratio of negative sentiment words | 1.192 | $p < 0.001(+)$ |
| Content length | 1.151 | $p = 0.6(+)$ |
| Ratio of positive sentiment words | 1.144 | $p < 0.001(+)$ |
| Ratio of capitalized words | 1 | $p < 0.001$ |
| Ratio of sentiment sentences | 0.251 | $p < 0.001(+)$ |
| Fuel = electric | 0.173 | – |
| Car type = offroad | 0.170 | – |
| Model = explorer | 0.170 | – |
| Model = ranger | 0.170 | – |

* (+) the mean value of sellable used vehicles is higher

the post contents of sellable used vehicles are longer, have more modifiers and uppercases letters, and express more emotional signals than that of unsellable used vehicles do.

Below are some examples of post contents of sellable used vehicles (sentiment words and modifiers are bolded)

> Up for sale is my 1999 Honda Accord EX Coupe. -Manual transmission -2 Door -107K miles **Clean** Title -Dark Green -Sunroof -Lowered -VIN: XXXX If interested, email, call, or text if you have questions.

> 1997 Toyota Corolla Sedan CE (**classic** edition) Selling by First Owner We owned the car since **new**, Runs **Excellent**, has been a **very reliable** car, **Very Clean** inside and outside, gets **great** gas mileage, has 4 Micheline tires, with AC, Radio, and more, i have all the service records kept worth of over $7,000, all services are up to date, This would be **great commuter/starter** car, and could be used for **long** commute.

Compare to the post contents of sellable used vehicles, which are characterized by descriptive long sentences or phrases, the post contents of unsellable vehicles are shorter, concise, and contain less sentiment words:

> Up for sale is my 1999 Honda Accord EX Coupe. -Manual transmission -2 Door -107K miles **Clean** Title -Dark Green -Sunroof -Lowered -VIN: XXXX If interested, email, call, or text if you have questions.

This finding is consistent with Lewis' argument that "information asymmetry" could be reduced by disclosures of text written by sellers (Lewis 2011). Thus, it is our future work to include expired posts to improve the performance of the ranking model.

## 8 Conclusions and future work

In this paper, we address the task of used product ranking by introducing a novel time-aware metric to measure the "goodness" of used items. A combined Poisson regression and listwise ranking model is proposed to rank used products based on "sellability". The experiments were conducted in the domain of used vehicles with data collected from

Craigslist. The results demonstrate the effectiveness of the proposed model. By analyzing the importance of different features, we conclude that a "sellable" used product is highly related to its basic information, location, price, and how its owner describes it. In the future work, we plan to evaluate the proposed approach on other types of used products and explore more useful features. Additional works could also include further exploration of combinations of other ranking loss functions and evaluation on real user queries if available.

**Acknowledgments** Special thanks to Chenzi Qian for her help in maintaining the crawler.

## References

Akerlof, G. A. (1970). The market for lemons: Quality uncertainty and the market mechanism. *The Quarterly Journal of Economics*, *84*(3), 488–500.

Bird, S. (2006). NLTK: The natural language toolkit. In *COLING/ACL* (pp. 69–72). ACL.

Borghol, Y., Ardon, S., Carlsson, N., Eager, D., & Mahanti, A. (2012). The untold story of the clones: Content-agnostic factors that impact YouTube video popularity. In *KDD* (pp. 1186–1194).

Burges, C., Shaked, T., Renshaw, E., Lazier, A., Deeds, M., Hamilton, N., & Hullender, G. (2005). Learning to rank using gradient descent. In *ICML* (pp. 89–96).

Cao, Z., Qin, T., Liu, T.-Y., Tsai, M.-F., & Li, H. (2007). Learning to rank: From pairwise approach to listwise approach. In *ICML* (pp. 129–136).

Chang, K. C.-C., He, B., Li, C., Patel, M., & Zhang, Z. (2004). Structured databases on the web: Observations and implications. *ACM SIGMOD Record*, *33*(3), 61–70.

Chaudhuri, S., Das, G., Hristidis, V., & Weikum, G. (2004). Probabilistic ranking of database query results. In *VLDB* (pp. 888–899). VLDB Endowment.

Chaudhuri, S., Das, G., Hristidis, V., & Weikum, G. (2006). Probabilistic information retrieval approach for ranking of database query results. *ACM Transactions on Database Systems*, *31*(3), 1134–1168.

Chung, S. H., Goswami, A., Lee, H., & Hu, J. (2012). The impact of images on user clicks in product search. In *MDMKDD* (pp. 25–33).

Duan, H., Zhai, C., Cheng, J., & Gattani, A. (2013). Supporting keyword search in product database: A probabilistic approach. *VLDB Endowment*, *6*(14), 1786–1797.

Duan, Y., Jiang, L., Qin, T., Zhou, M., & Shum, H.-Y. (2010). An empirical study on learning to rank of tweets. In *Coling* (pp. 295–303). ACL.

Guo, Q., & Agichtein, E. (2010). Ready to buy or just browsing? Detecting web searcher goals from interaction data. In *SIGIR* (pp. 130–137).

Hong, L., Dan, O., & Davison, B. D. (2011). Predicting popular messages in twitter. In *WWW* (pp 57–58).

Hristidis, V., Gravano, L., & Papakonstantinou, Y. (2003). Efficient IR-style keyword search over relational databases. In *VLDB* (pp. 850–861). VLDB Endowment.

Huang, M., Yang, Y., & Zhu, X. (2011). Quality-biased ranking of short texts in microblogging services. In *IJCNLP* (pp. 373–382). ACL.

Järvelin, K., & Kekäläinen, J. (2000). IR evaluation methods for retrieving highly relevant documents. In *SIGIR* (pp. 41–48).

Lan, Y., Niu, S., Guo, J., & Cheng, X. (2013). Is top-k sufficient for ranking? In *CIKM* (pp. 1261–1270).

Lewis, G. (2011). Asymmetric information, adverse selection and online disclosure: The case of ebay motors. *The American Economic Review, 101*(4), 1535–1546.

Li, B., Ghose, A., & Ipeirotis, P. G. (2011). Towards a theory model for product search. In *WWW* (pp. 327–336).

Liu, F., Yu, C., Meng, W., & Chowdhury, A. (2006). Effective keyword search in relational databases. In *SIGMOD* (pp. 563–574).

Liu, T.-Y. (2009). Learning to rank for information retrieval. *Foundations and Trends in Information Retrieval*, *3*(3), 225–331.

Liu, T.-Y., Xu, J., Qin, T., Xiong, W., & Li, H. (2007). Letor: Benchmark dataset for research on learning to rank for information retrieval. In *Proceedings of SIGIR 2007 workshop on learning to rank for information retrieval* (pp. 3–10).

Long, B., Bian, J., Dong, A., & Chang, Y. (2012). Enhancing product search by best-selling prediction in e-commerce. In *CIKM* (pp. 2479–2482).

Luo, Y., Lin, X., Wang, W., & Zhou, X. (2007). Spark: Top-k keyword query in relational databases. In *SIGMOD* (pp. 115–126).

Lynn, M. (1989). Scarcity effects on desirability: Mediated by assumed expensiveness? *Journal of Economic Psychology*, *10*(2), 257–274.

Ma, Z., Sun, A., & Cong, G. (2012). Will this # hashtag be popular tomorrow? In *SIGIR* (pp. 1173–1174).

Park, D. H., Liu, M., Zhai, C., & Wang, H. (2015). Leveraging user reviews to improve accuracy for mobile app retrieval. In *SIGIR* (pp. 533–542).

Park, Y., Jones, J., McCoy, D., Shi, E., & Jakobsson, M. (2014). Scambaiter: Understanding targeted nigerian scams on craigslist. *NDSS*, *1*, 2.

Pu, P., Chen, L., & Kumar, P. (2008). Evaluating product search and recommender systems for e-commerce environments. *Electronic Commerce Research*, *8*(1–2), 1–27.

Sculley, D. (2010). Combined regression and ranking. In *SIGKDD* (pp. 979–988).

Su, W., Wang, J., Huang, Q., & Lochovsky, F. (2006). Query result ranking over e-commerce web databases. In *CIKM* (pp. 575–584).

Telang, A., Li, C., & Chakravarthy, S. (2012). One size does not fit all: Toward user-and query-dependent ranking for web databases. *TKDE*, *24*(9), 1671–1685.

Vandic, D., Frasincar, F., & Kaymak, U. (2013). Facet selection algorithms for web product search. In *CIKM* (pp. 2327–2332).

Wadleigh, J., Drew, J., & Moore, T. (2015). The e-commerce market for lemons: Identification and analysis of websites selling counterfeit goods. In *WWW* (pp. 1188–1197).

Wang, X., Liu, C., Xue, G., & Yu, Y. (2010). Click prediction for product search on C2C web sites. In L. Cao, J. Zhong, & Y. Feng (Eds.), *Advanced data mining and applications: 6th international conference, ADMA 2010, Chongqing, China, November 19–21, 2010, Proceedings, Part II* (pp. 387–398). Berlin, Heidelberg: Springer Berlin Heidelberg.

Wiebe, J., Wilson, T., & Cardie, C. (2005). Annotating expressions of opinions and emotions in language. *Language Resources and Evaluation*, *39*(2–3), 165–210.

Wu, X., & Bolivar, A. (2009). Predicting the conversion probability for items on C2C ecommerce sites. In *CIKM* (pp. 1377–1386).

Xia, F., Liu, T.-Y., Wang, J., Zhang, W., & Li, H. (2008). Listwise approach to learning to rank: theory and algorithm. In *ICML* (pp. 1192–1199).

Zhang, Y., Sondhi, P., Goswami, A., & Zhai, C. (2014). A Bayesian framework for modeling price preference in product search. *NIPS Workshop*, *51*, 61801.