A method for discovering the knowledge of item rank from consumer reviews

Song Luo

1 Graduate School of Systems and Information Engineering, University of Tsukuba, 1-1-1 Tennodai, Tsukuba, Ibaraki 305-8577, Japan
E-mail s1230158@u.tsukuba.ac.jp
Received March 30, 2018, Accepted September 25, 2018

Abstract

When observing a collection of ranked items, we may be interested in the questions of why and how one item is ranked over another. This paper presents a method for discovering the knowledge about the rank of the items from consumer reviews. We formulate the questions of interest as a single biconvex minimization problem which has a relationship with SVM(Support Vector Machines). To facilitate the process of knowledge discovery, we propose a two-stage learning algorithm for discovering knowledge from small data. Finally, we evaluate the method by showing our simulation and experiment results.

Keywords knowledge discovery, statistical learning, biconvex optimization

1. Introduction

Ranking items is a fundamental task in many industrial and commercial systems. Traditionally, a ranking model’s purpose is to produce a permutation of items in a way which can mimic a ranker’s behavior in some sense. However, we may also be interested in the questions of why and how one item is ranked over another. Motivated by these questions, this paper discusses (1) how to discover the knowledge of a ranker’s behavior from observable data and (2) how this knowledge may help to explain and predict the ranker’s behavior. More concretely, this paper studies the behavior of a book ranker by using textual data of online consumer reviews.

2. Problem description

Observing k ranked items (books) \( I_1 \prec \cdots \prec I_k \) as their consumer reviews, we think of that the item \( t \)'s consumer reviews and description are described as either positive or negative, denoting respectively by \( f^+ \) and \( f^- \). Let \( T = (t_1^+, \ldots, t_m^+, t_1^-, \ldots, t_m^-) \in \{0, 1\}^{2m} \) be a vector of a consumer review, for \( s = 1, \ldots, m \),

\[
t_s^+ = \begin{cases} 1, & \text{if a positive description of } f_s \text{ is observed,} \\ 0, & \text{otherwise.} \end{cases}
\]

We define \( t_s^- \) in a similar way. In addition, \( t_s^+ + t_s^- \leq 1 \).

For each vector, let \( T_i = \{T_{i,1}, \ldots, T_{i,N(t_i)}\} \) be the set of vectors of item \( i \)'s consumer reviews and define \( p_i = \sum_{j=1}^{N(t_i)} T_{i,j} / \sum_{j=1}^{N(t_i)} |T_{i,j}| \) as the ranker’s impression vector towards item \( i \), where \( |T| \) is the number of ones in \( T \). It can be seen \( p_i \in \mathcal{P} \), where \( \mathcal{P} = \{(p_1^+, \ldots, p_m^+, p_1^-, \ldots, p_m^-) \in [0, 1]^{2m} : \sum_{j=1}^{m}(p_j^+ + p_j^-) = 1\} \).

Moreover, for some integer \( r \) \((0 < r < k)\), associate item \( I_i \) with label \( y_i \), such that \( y_i = 1 \) if and only if \( i \leq r \) otherwise \( y_i = -1 \). Given the data \( (p_1, y_1), \ldots, (p_k, y_k) \), our goal is to simultaneously explain and predict the behavior of the ranker.

3. Problem formulation

This section first introduces a generative ranking model in order to explain the behavior of the ranker. Then, with this model we formulate the questions of interest as a learning problem for prediction purpose.

3.1 A generative model of the ranker’s behavior

To establish a dependency relationship between \( p \) and \( y \), we introduce a latent vector \( q \) called anchor vector. The purpose of introducing \( q \) is to create a standard to be compared with \( p \). Intuitively, if \( p \) is placed over \( q \), then \( p \) is expected to be associated with a high rank.

With this idea, we construct a scoring function \( S : \mathcal{P} \times \mathcal{P} \rightarrow \mathbb{R} \) with following appealing properties:

- \( B_1 \) \( q \in \mathcal{P} \) (existence)
- \( B_2 \) \( S(p, q) \geq S(p', q) \) if \( p \geq p' \) (monotonicity)
- \( B_3 \) \( S(p, q) + S(q, p) = 0 \) (asymmetry)
- \( B_4 \) \( S(\theta p + (1-\theta)p', q) = \theta S(p, q) + (1-\theta)S(p', q) \), \( \forall \theta \in [0, 1] \) (linearity).

In \( B_2 \), \( p \geq p' \) if and only if \( p_j^+ \geq p_j'^+ \) and \( p_j^- \leq p_j'^- \), for \( j = 1, \ldots, m \). It can be verified that \( S(p, q) = p^T A q \), for some asymmetric matrix \( A \) called interaction matrix,

\[
A = \begin{bmatrix} A_{m \times m}^+ & A_{m \times m}^- \\ A_{m \times m}^+ & A_{m \times m}^- \end{bmatrix}.
\]
where $A_{m \times m}^{+} \leq A_{m \times m}^{-}$ and $A_{m \times m}^{-} \leq A_{m \times m}^{+}$. There may be other prior information about $A$. Let $\mathcal{A}$ be a convex set of all possible interaction matrices constructed by available prior information. In this way, a space of hypotheses concerning the structure of the ranker’s behavior can be constructed and parameterized by $A$ and $q$. Given $(p_1, y_1), \ldots, (p_k, y_k)$, we would like to estimate the values of $A$ and $q$. However, the immediate challenges are that (i) under what inductive principle to infer $A$ and $q$, (ii) how to obtain robust estimation even when $k$ is small, and (iii) how to compute $A$ and $q$.

### 3.2 Problem reformulation with learning theory

We provide a possible answer to challenges (i) and (ii) with learning theory.

Assume that $p_1, \ldots, p_k$ are independent and identically distributed (i.i.d.) observations drawn according to a fixed but unknown distribution $F(p)$ and that $y_1, \ldots, y_k$ are i.i.d. observations drawn according to a fixed but unknown conditional distribution $F(y|p)$. Therefore, it follows that $(p_1, y_1), \ldots, (p_k, y_k)$ can be seen as a random sample drawn according to a fixed but unknown joint distribution $F(p, y)$, where $F(p, y) = F(p)F(y|p)$.

Let $\gamma(p, \alpha), \alpha \in \Lambda$ be a set of functionals, where $\Lambda$ is determined by available prior information. Given $(p_1, y_1), \ldots, (p_k, y_k)$, the goal is to choose from $\gamma(p, \alpha), \alpha \in \Lambda$, the one which best predicts $F(y|p)$.

More concretely, consider the loss function

$$L(y, \gamma(p, \alpha)) = \begin{cases} 0 & \text{if } y = \gamma(p, \alpha), \\ 1 & \text{if } y \neq \gamma(p, \alpha) \end{cases}$$

and the expected value of the loss, given by the risk function

$$R(\alpha) = \int L(y, \gamma(p, \alpha))dF(p, y).$$

Then, the goal is to find the function $\gamma(p, \alpha^*)$ that minimizes the risk function $R(\alpha), \alpha \in \Lambda$. To minimize risk function $R(\alpha)$ for unknown $F(p, y)$, we consider the Structural Risk Minimization (SRM) Principle [1] instead of the Empirical Risk Minimization (ERM) Principle. The empirical risk functional is defined as $R_{emp}(\alpha) \triangleq (1/k)\sum_{i=1}^{k} L(y_i, \gamma(p_i, \alpha))$. According to ERM principle, one chooses $\gamma(p, \alpha^*)$ by minimizing $R_{emp}(\alpha), \alpha \in \Lambda$, which may be unfavorable when $k$ is small. Instead, given the size of empirical data is fixed, SRM principle finds the function that achieves the minimum of the guaranteed risk by optimizing the relationship between the quality of approximation of empirical data and the complexity of the set of functionals. To estimate $A$ and $q$, we adopt the support vector estimation strategy of SRM principle which keeps the value of the empirical risk fixed and then minimizes the complexity of the set of the functionals. In this paper, we first consider the case where $p_1, \ldots, p_k$ can be separated by a hyperplane $p^Tw - b = 0$, for some $w \in \mathbb{R}^{2m}$ and $b \in \mathbb{R}$, and then relax the assumption of linear separability for the non-separable case. To this end, we use a special type of hyperplane, called maximal margin hyperplane [1]. A set of vectors is separated by a maximal margin hyperplane if it is separated without error and the distance between the closest vector to the hyperplane is maximal. The maximal margin hyperplane is given by

$$\min \frac{1}{2}\|w\|^2$$

$$\text{s.t. } y_i(p_i^Tw - b) \geq 1, \quad i = 1, \ldots, k.$$ 

In this case $\gamma(p, \alpha) = p^Tw - b$, where $\alpha = (w, b)$. Noting that $S(p, q) = p^TAq$ is linear in $p$, we replace $Aq$ with $w$ and let $b = 0$. This leads to the support vector estimation of $A$ and $q$, and hence formulates the questions of interest as a single biconvex optimization problem (for a survey of biconvex problem, see [2])

$$\min_{\Lambda, q} \frac{1}{2}\|Aq\|^2$$

$$\text{s.t. } y_i(p_i^TAq) \geq 1, \quad i = 1, \ldots, k,$$

$$q \in P, \quad A \in \mathcal{A}.$$ 

To relax the assumption of linear separability, we consider a generalized concept of maximal margin hyperplane determined by the following problem $Q$,

$$Q : \min \frac{1}{2}\|Aq\|^2 + C\sum_{i=1}^{k}\eta_i$$

$$\text{s.t. } y_i(p_i^TAq) \geq 1 - \eta_i, \quad i = 1, \ldots, k,$$

$$q \in P, \quad A \in \mathcal{A},$$

$$\eta_i \geq 0, \quad i = 1, \ldots, k.$$ 

Here $C$ is a fixed positive value. This relaxation also plays a role in discriminating between plausible explanations and unlikely ones, as will be seen in the next section.

### 4. Two-stage learning algorithm

This section deals with the challenge of how to obtain robust estimation of $A$ and $q$. The difficulty lies in that (1) in general, it is hard to find a global optimal solution of a biconvex minimization problem; (2) to obtain several competing explanations, we need to find several distinguished optimal global solutions.

To cope with the difficulty, a two-stage learning algorithm is proposed. In the information extraction stage, we extract as many high-quality suboptimal solutions as possible by making use of the biconvex structure of $Q$. Then, in the knowledge formation stage, the suboptimal solutions are clustered and further processed into knowledge.

The obtained knowledge may be seen as a stable pattern underlying the high-quality suboptimal solutions, suggesting a balance between explanation plausibility and prediction accuracy.

#### 4.1 Information extraction stage

**Stage 1.1** Given a fixed integer $N$, draw $q^{(1)}, \ldots, q^{(N)}$ independently and uniformly from $\mathcal{P}$. Intuitively
speaking, in order to avoid investigating the complicated structure of the solution space of \( Q \), an uninformative prior and hence uniform sampling on \( \mathcal{P} \) will be used in a Monte Carlo style for searching high-quality suboptimal solutions.

**Stage 1.2** For each \( q^{(i)} \), solve \( Q(q^{(i)}) \)

\[
Q(q^{(i)}): \min \frac{1}{2} \| Aq^{(i)} \|^2 + C \sum_{i=1}^{k} \eta_i \\
\text{s.t.} \quad y_i(p_i^T Aq^{(i)}) \geq 1 - \eta_i, \; i = 1, \ldots, k, \\
A \in \mathcal{A}, \\
\eta_i \geq 0, \; i = 1, \ldots, k.
\]

Since \( Q \) is biconvex, \( Q(q^{(i)}) \) is a convex optimization problem for each \( q^{(i)} \), and hence we can efficiently obtain its optimal value \( v^{(j)} \) and its unique solution \( A^{(j)} \). If \( v^{(j)} \) is relatively small, \( (A^{(j)}, q^{(j)}) \) is probably an unlikely explanation.

**Stage 1.3** Sort the \( v^{(j)} \)’s, say \( v^{(j_1)} \leq \cdots \leq v^{(j_N)} \). Then, for some proper integer \( N < N \), find the elite samples \( q^{(j_1)}, \ldots, q^{(j_N)} \) and their corresponding \( A^{(j_1)}, \ldots, A^{(j_N)} \). We define \( A^{(j)}_t, q^{(j_t)} \), \( t = 1, \ldots, N \), as \( Q \) suboptimal solutions of \( Q \). A good choice of the value of \( N \) should successfully discriminate as many suboptimal solutions of \( Q \) as possible from the \( N \) samples, and hence it depends on the values of \( N \) and \( v^{(j)} \)’s in general.

4.2 Knowledge formation stage

**Stage 2.1** Group the elite samples of \( q \)’s and \( A \)’s obtained in stage 1.3 into \( K_q \) and \( K_A \) clusters \( C^q = \{ C^q_1, \ldots, C^q_{K_q} \} \) and \( C^A = \{ C^A_1, \ldots, C^A_{K_A} \} \), respectively, by solving

\[
\min_{C^q = \{ C^q_1, \ldots, C^q_{K_q} \}} \sum_{h=1}^{K_q} \sum_{q \in C_h^q} \| q - q_h \|^2, \\
\min_{C^A = \{ C^A_1, \ldots, C^A_{K_A} \}} \sum_{l=1}^{K_A} \sum_{A \in C_l^A} \| A \| - A_l \|^2,
\]

where \( K_q \) and \( K_A \) are fixed and relatively small integers. This is exact the standard formulation of K-Means clustering. Denote by \( q_h^* \) and \( A_l^* \) the obtained clustering centers. These centers can be seen as a summary of the explanation plausibility of the high-quality suboptimal solutions.

**Stage 2.2** To arrive at a stable pattern that is able to balance explanation plausibility with prediction accuracy, the centers are further recombined as follows. Let \( w_F \) be the optimal solution of the following problem

\[
\min \frac{1}{2} \| w \|^2 + C \sum_{i=1}^{k} \eta_i \\
\text{s.t.} \quad y_i(p_i^T w) \geq 1 - \eta_i, \; i = 1, \ldots, k, \\
\eta_i \geq 0, \; i = 1, \ldots, k.
\]

For each center \( q_h^* \), let

\[
A_q^* = \arg \min_{A \in \{ A_1, \ldots, A_r \}} \| Aq_h^* - w_F \|.
\]

Finally, \( (A_q^*, q_h^*) \), \( h = 1, \ldots, K_q \), are the desired knowledge about the ranker’s behavior.

5. Simulation

This section inspects the effectiveness of the two-stage learning algorithm. Particularly, we investigate how the prior information helps to predict the ranker’s behavior and whether the knowledge discovered by the two-stage learning algorithm is useful to explain its behavior.

5.1 Simulation procedure

1. **Step 1** Generate the underlying true interaction matrix \( A_T \) and anchor vector \( q_T \).

2. **Step 2** For pre-fixed \( k \) and \( \Delta \), generate \( k \) random samples \( p \) independently and uniformly drawn from \( \mathcal{P} \) with a balanced number of labels given by

\[
y_i = \begin{cases} 
1, & \text{if } p_i^T A_T q_T \geq \Delta, \\
-1, & \text{if } p_i^T A_T q_T < -\Delta.
\end{cases}
\]

These \( p \)’s and \( y \)’s are used as the simulation data.

3. **Step 3** Determine \( A \). We give each non-diagonal entry of the interaction matrix \( A \) a box constraint \( (A_T)_{ij} - B \leq A_{ij} \leq (A_T)_{ij} + B \), for \( i \neq j \). Here, \( B \) is a fixed positive number serving as a measure of the amount of prior information.

4. **Step 4** Apply the two-stage learning algorithm to the simulation data under various parameter settings.

5. **Step 5** Compare (1) \( q_h^* \) with \( q_T \), (2) \( A_q^* \) with \( A_T \), and (3) \( A_T q_h^* \) with \( A_T q_T \) and \( w_F \).

5.2 Parameter setting

The two-stage learning algorithm was simulated under the following parameter settings: \( m = 4, k \in \{10, 100, 950\} \), \( \Delta \in \{1, 1.5\} \), \( B \in \{0.1, 1, 5, 10\} \), \( N \in \{100, 1000, 3000, 10000\} \), \( N = 5/5 \) and \( K_q = K_A = 4 \).

5.3 Simulation result

5.3.1 Summary of simulation result

**Result 1** Even when \( \Delta = 1, k = 10 \) and \( B = 10 \), the learning algorithm still tends to give good estimation of \( q \) and moderate estimation of \( A \). This simulation result is appealing since it suggests a validation of the effectiveness of the algorithm. When \( \Delta = 1.5 \) (the maximal margin gets larger) while keeping the other parameters unchanged, the estimation of \( q \) and \( A \) seems robust. When \( B = 0.1 \), the quality of the estimation of \( q \) has a noticeable but limited improvement. When \( k = 950 \), the quality of the estimation of \( q \) has a slight improvement. \( N = 3000 \) seems a good trade-off between estimation quality and computation time.

**Result 2** Under various parameter settings, the learning algorithm tends to find \( q_h^* \) and \( A_q^* \) such that

\[
\| A_T q_T - A_q^* q_h^* \| < \| A_T q_T - w_F \|.
\]
This numerical result seems to support that the prior knowledge of the ranker’s behavior helps to improve the accuracy of the prediction.

5.3.2 An instance of simulation result

In Fig. 1 and 2, we show an instance of simulation result when \( m = 4 \), \( \Delta = 1 \), \( k = 10 \), \( B = 10 \), \( C = 1000 \), \( N = 3000 \) and \( K_q = K_A = 4 \). The values of \( q_T \) and \( A_T \) (\( A_T \) has 28 free variables) are connected by black lines, and the dots represent the support vector estimations of the values of \( q_T \) and \( A_T \). This result shows the learning ability of the two-stage learning algorithm in the linear separable case when only small data are available.

6. Experiment

We used the data collected from the website [3] consisting of a list of ranked python programming books and their consumer reviews. The consumers reviewed the books mainly from four aspects: readability, beginner-friendliness, content highlight and content coverage. After removing some irrelevant books, the data set contains ten books as shown in Table 1. The unnormalized impression vectors of the books are shown in the second column. The books 4 and 8 are removed from the data set since they had few consumer reviews. In addition, the book 10 is removed since it is an outlier. Notice that the data is not linearly separable, and correspondingly, we set \( C = 10 \) in the experiment for relaxing the assumption of linear separability.

Moreover, there are zeros in some unnormalized impression vectors. To deal with the zeros, we consider the smoothed impression vectors given by \( p_i = S1 + \sum_{j=1}^{N} T_{i,j}/(2mS + \sum_{j=1}^{N} |T_{i,j}|) \), where \( S \) is a positive number and \( 1 \) is the all-ones vector. In this experiment, we set \( S = 15 \), \( m = 4 \), \( N = 3000 \), \( B = 10 \), \( K_q = K_A = 4 \) and \( A_T = \begin{bmatrix} 0 & 5J \\ -5J & 0 \end{bmatrix} \). Here, \( J \) is the all-ones matrix. Applying the two-stage learning algorithm, the experiment result of

\[
q^* = (0.35, 0.1, 0.1, 0.1, 0.13, 0.07, 0.06, 0.09)
\]

showed that the ranker seems to have much higher expectation for positive readability and also have a slightly higher tolerance for negative readability. This implies that the positive evaluation of readability may be an important factor to produce a high rank but the negative evaluation of readability may not lead to a low rank. Moreover, the first row of interaction matrix \((0, 1, 1, 2, 0, 2, 0, 1, 1)\) implies that positive readability is relatively more important. However, since the data are not linearly separable and the value of maximal margin is relatively small, we caution that the ranking model may suffer from being falsified.

7. Conclusion

This paper proposed a method for discovering the knowledge about item rank by learning the ranker’s latent behavior structure. This method showed a possibility that the questions concerning explanation and prediction may be simultaneously answered under some conditions. The simulation result gave a positive support to this possibility. Although, as the data experiment showed, this method needs a further extension to deal with more general cases, it still seems promising.

Acknowledgments

I thank Professor Maiko Shigeno for her helpful suggestions and kind encouragement. I also thank the anonymous reviewers for their valuable comments.

References


Table 1. Experiment Data.

<table>
<thead>
<tr>
<th>Item</th>
<th>Attribute description</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(27, 17, 35, 24, 6, 6, 12, 25)</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>(18, 8, 16, 9, 2, 3, 3, 4)</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>(3, 3, 7, 1, 4, 0, 2, 3)</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>(1, 0, 3, 1, 0, 2, 1, 1)</td>
<td>removed</td>
</tr>
<tr>
<td>5</td>
<td>(17, 20, 36, 33, 33, 14, 6, 2)</td>
<td>−1</td>
</tr>
<tr>
<td>6</td>
<td>(8, 6, 14, 6, 1, 7, 1, 0)</td>
<td>−1</td>
</tr>
<tr>
<td>7</td>
<td>(3, 5, 11, 2, 0, 0, 1, 3)</td>
<td>−1</td>
</tr>
<tr>
<td>8</td>
<td>(1, 1, 3, 0, 1, 0, 1, 1)</td>
<td>removed</td>
</tr>
<tr>
<td>9</td>
<td>(14, 20, 17, 6, 1, 1, 0, 5)</td>
<td>−1</td>
</tr>
<tr>
<td>10</td>
<td>(6, 6, 4, 2, 0, 0, 0, 0)</td>
<td>removed</td>
</tr>
</tbody>
</table>