How to Count Thumb-Ups and Thumb-Downs: User-Rating based Ranking of Items from an Axiomatic Perspective

Dell Zhang[1], Robert Mao[2], Haitao Li[3], and Joanne Mao[4]

[1] Birkbeck, University of London
Malet Street, London WC1E 7HX, UK
dell.z@ieee.org

[2] Microsoft Research
1 Microsoft Way, Redmond, WA 98052, USA
robmao@microsoft.com

[3] Microsoft Corporation
1 Microsoft Way, Redmond, WA 98052, USA
lht1999@gmail.com

[4] MNX Consulting
9833 Wilden Lane, Potomac, MD 20854, USA
joanne.mao@mnxconsulting.com

Abstract. It is a common practice among Web 2.0 services to allow users to rate items on their sites. In this paper, we first point out the flaws of the popular methods for user-rating based ranking of items, and then argue that two well-known Information Retrieval (IR) techniques, namely the Probability Ranking Principle and Statistical Language Modelling, provide simple but effective solutions to this problem. Furthermore, we examine the existing and proposed methods in an axiomatic framework, and prove that only the score functions given by the Dirichlet Prior smoothing method as well as its special cases can satisfy both of the two axioms borrowed from economics.

1 Introduction

Suppose that you are building a Web 2.0 service which allows users to rate items (such as commercial-products, photos, videos, songs, news-reports, and answers-to-questions) on your site, you probably want to sort items according to their user-ratings so that stuff "liked" by users would be ranked higher than those "disliked". How should you do that? What is the best way to count such thumb-ups and thumb-downs? Although this problem — user-rating based ranking of items — looks easy and occurs in numerous applications, the right solution to it is actually not very obvious.

In this paper, we first point out the flaws of the popular methods for user-rating based ranking of items (see Section 3), and then argue that two well-known Information Retrieval (IR) techniques, namely the Probability Ranking Principle [1] and Statistical Language Modelling [2, 3], provide simple but effective solutions to this problem (see Section 4). Furthermore, we examine the existing and proposed methods in an axiomatic framework, and prove that only the score functions given by the Dirichlet Prior smoothing [3] method as well as its special cases can satisfy both of the two axioms borrowed from economics, namely the Law of Increasing Total Utility and the Law of Diminishing Marginal Utility [4] (see Section 5).

2 Problem

Let's focus on binary rating systems first and then generalise to graded rating systems later. Given an item i, let $n_{\uparrow}(i)$ denote the number of thumb-ups and $n_{\downarrow}(i)$ denote the number of thumb-downs. In the rest of this paper, we shall omit the index i to simplify the notation when it is clear from the context that we are talking about an item i in general. To sort the relevant items based on user-ratings, a score function $s(n_{\uparrow}, n_{\downarrow}) \in \mathbb{R}$ would need to be calculated for each of them.

3 Popular Methods

There are currently three popular methods widely used in practice for this problem, each of which has some flaws.

3.1 Difference

The first method is to use the *difference* between the number of thumb-ups and the number of thumb-downs as the score function, i.e.,

$$s(n_{\uparrow}, n_{\downarrow}) = n_{\uparrow} - n_{\downarrow} \ . \tag{1}$$

For example, Urban Dictionary, a web-based dictionary of slang words and phrases, is said to be using this method, as shown in Figure 1.

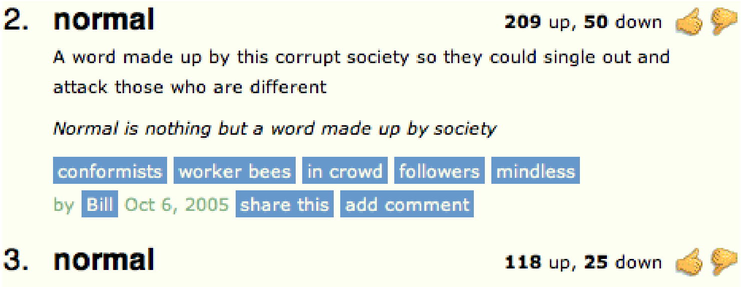
Assume that item i has 200 thumb-ups and 100 thumb-downs, while item j has 1,200 thumb-ups and 1,000 thumb-downs, this method would rank item i (whose score is 100) lower than item j (whose score is 200). However, this does not sound plausible, because item i has twice thumb-ups than thumb-downs, while item j has only slightly more thumb-ups than thumb-downs.

3.2 Proportion

The second method is to use the *proportion* of thumb-ups in all user-ratings as the score function, i.e.,

$$s(n_{\uparrow}, n_{\downarrow}) = \frac{n_{\uparrow}}{n_{\uparrow} + n_{\downarrow}} \ . \tag{2}$$

For example, Amazon, the largest online retailer company in the United States, is said to be using this method, as shown in Figure 2.



2. **normal** **209** up, **50** down

A word made up by this corrupt society so they could single out and attack those who are different

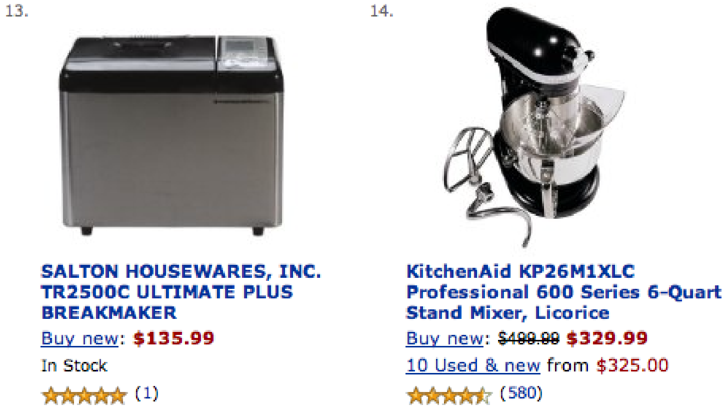
Normal is nothing but a word made up by society

conformists worker bees in crowd followers mindless

by Bill Oct 6, 2005 share this add comment

3. **normal** **118** up, **25** down

Fig. 1. An example of Urban Dictionary's ranking methods for user rated items, adapted from Evan Miller's online article[5].



13.

SALTON HOUSEWARES, INC. TR2500C ULTIMATE PLUS BREAKMAKER

Buy new: **$135.99**

In Stock

(1)

14.

KitchenAid KP26M1XLC Professional 600 Series 6-Quart Stand Mixer, Licorice

Buy new: ~~$499.99~~ **$329.99**

10 Used & new from $325.00

(580)

Fig. 2. An example of Amazon's ranking methods for user rated items, adapted from Evan Miller's online article[5]

Assume that item i has 200 thumb-ups and 1 thumb-down, while item j has 2 thumb-ups and 0 thumb-down, this method would rank item i (whose score is 0.995) lower than item j (whose score is 1.000). However, this does not sound plausible, because although both item i and item j have almost none thumb-down, item i has hundreds of thumb-ups, while item j has only a couple of thumb-ups.

3.3 Wilson Interval

The third method was advocated by Evan Miller's online article[5] on this topic to avoid the flaws of the above two simple methods. The idea is to treat the existing set of user-ratings as a statistical sampling of a hypothetical set of user-ratings from all users, and then use the *lower bound* of *Wilson score confidence*

[5] http://www.evanmiller.org/how-not-to-sort-by-average-rating.html

interval [5] for the proportion of thumb-ups as the score function, i.e.,

$$s(n_\uparrow, n_\downarrow) = \frac{\hat{p} + \frac{z^2_{1-\alpha/2}}{2n} - \sqrt{\frac{z^2_{1-\alpha/2}}{n}\left[\hat{p}(1-\hat{p}) + \frac{z^2_{1-\alpha/2}}{4n}\right]}}{1 + \frac{z^2_{1-\alpha/2}}{n}} \ , \qquad (3)$$

where $n = n_\uparrow + n_\downarrow$ is the total number of user-ratings, $\hat{p} = n_\uparrow/n$ is the observed proportion of thumb-ups, and $z_{1-\alpha/2}$ is the $(1-\alpha/2)$ quantile of the standard normal distribution. With the default parameter value $\alpha = 0.10$, the above score function estimates what the "real" proportion of thumb-ups at least is at 95% chance, therefore it balances the proportion of thumb-ups with the uncertainty due to a small number of observations. This method is considered as the current state of the art and thus adopted by many sites. For example, Reddit, a famous social news site, has mentioned in its official blog post[6] that this method is used for their ranking of comments, as shown in Figure 3.

Nevertheless, this method is not well justified either.

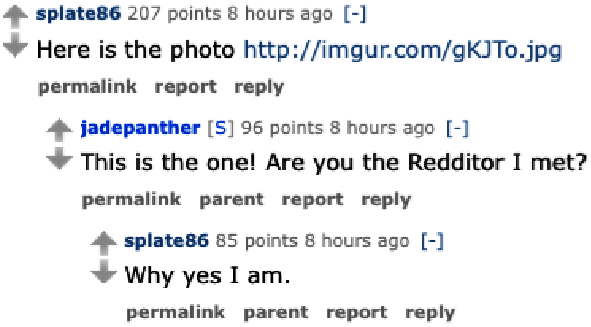
- First, the above formula cannot be applied to calculate scores for the items that have not received any user-rating yet: the prevailing implementation assigns score 0 to such items, which is wrong since this implies that "no user-rating yet" is roughly same as "zero thumb-up vs. one billion thumb-downs".
- Second, as the lower bound is biased towards one side only, it always under-estimates the "real" proportion of thumb-ups.
- Third, it is not clear how tight the lower bound is, i.e., how far it deviates away from the "real" proportion of thumb-ups.
- Fourth, the difference between the lower bound and the "real" proportion of thumb-ups are inconsistent for items with different number of user-ratings.

Assume that item i has 1 thumb-up and 2 thumb-downs, while item j has 100 thumb-ups and 200 thumb-downs, this method would rank item i (whose score is 0.386) lower than item j (whose score is 0.575). However, this does not sound plausible, because while we are not really sure whether item i is good or bad, we have a lot of evidence that item j is bad, so we should rank item i higher than item j. For another example, using this method, we have $s(500, 501) > s(5, 1)$, i.e., an item with 500 thumb-ups and 501 thumb-downs would be ranked higher than an item with 5 thumb-ups and one thumb-down, which does not make much sense.

4 Proposed Approach

In this paper, we propose to address the problem of user-rating based ranking of items by formulating it as an extremely simple Information Retrieval (IR) system: each user-rating — thumb-up or thumb-down — is considered as a *term*;

[6] http://blog.reddit.com/2009/10/reddits-new-comment-sorting-system.html



splate86 207 points 8 hours ago [-]

Here is the photo http://imgur.com/gKJTo.jpg

jadepanther [S] 96 points 8 hours ago [-]

This is the one! Are you the Redditor I met?

splate86 85 points 8 hours ago [-]

Why yes I am.

Fig. 3. An example of Reddit's ranking methods for user rated items, extracted from Reddit's blog post[6].

each item is considered as a *document* that consists of a number of those two terms. Since users would like to find good items from the collection, the ranking of the items could be regarded as searching the collection with a virtual *query* of one term — thumb-up ($q = \uparrow$). The better ratings an item has received from users, the more *relevant* it is to the query thumb-up.

According to the Probability Ranking Principle [1], we should rank documents by their probabilities of being relevant to the query, in our case, $\Pr[R = 1|i, \uparrow]$. This has been strictly proved to be the optimal retrieval strategy, in the sense that it minimises the expected loss (a.k.a. the Bayes risk) under 1/0 loss (i.e., you lose a point for either returning a non-relevant document or failing to return a relevant document) [6].

Making use of the Statistical Language Modelling [2,3] technique for retrieval, we treat each item i as a bag of user-ratings and construct a *unigram* model $M(i)$ for it, then the probability of an item being good (i.e., relevant to the query thumb-up) $\Pr[R = 1|i, \uparrow]$ can be calculated as the probability of the query being generated from its corresponding unigram model: $\Pr[\uparrow |M(i)]$.

So the problem becomes how we can accurately estimate the probability $\Pr[\uparrow |M(i)]$ for each item i. Given only a small number of observed user-ratings, the maximum likelihood estimator using the proportion of thumb-ups (i.e., the second method mentioned in Section 3) does not work due to the limitation of its frequentist view of probabilities, which is a well-known fact in the Information Retrieval community. For example, if item i has got 1 thumb-up and 0 thumb-down, the maximum likelihood estimator gives $\Pr[\uparrow |M(i)] = 1/(1 + 0) = 1$ and $\Pr[\downarrow |M(i)] = 0/(1 + 0) = 0$, which is apparently unreasonable — no thumb-downs so far does not mean that it is not possible to receive thumb-downs in the future, especially when we have seen one user-rating only. The solution is to *smooth* the maximum likelihood estimator so that we do not assign zero probability to unseen terms (user-ratings) and improve the accuracy of the estimated language model in general [7,8,3].

4.1 Additive Smoothing

Laplace Smoothing One of the simplest way to assign nonzero probabilities to unseen terms is Laplace smoothing (a.k.a. Laplace's rule of succession), which assumes that every item "by default" has 1 thumb-up and 1 thumb-down (known as pseudo-counts):

$$s(n_{\uparrow}, n_{\downarrow}) = \Pr[\uparrow |M] = \frac{n_{\uparrow} + 1}{(n_{\uparrow} + 1) + (n_{\downarrow} + 1)} \ . \tag{4}$$

If item i has received 2 thumb-ups and 0 thumb-down from users, it would have 1+2=3 thumb-ups and 1+0=1 thumb-downs in total, so $\Pr[\uparrow |M(i)] = 3/(3+1) = 0.75$. If item j has got 100 thumb-ups and 1 thumb-down, it would have 100+1=101 thumb-ups and 1+1=2 thumb-downs in total, so $\Pr[\uparrow |M(j)] = 101/(101+2) = 0.98$. Thus we see that item j would be ranked higher than item i, which is indeed desirable.

Lidstone Smoothing Although Laplace smoothing avoids most flaws of those popular methods (such as getting zero probability for unseen user-ratings), it probably puts too much weight on the pseudo-counts. A better choice is its more generalised form, Lidstone smoothing, which assumes that every item "by default" has ϵ thumb-ups and ϵ thumb-downs:

$$s(n_{\uparrow}, n_{\downarrow}) = \Pr[\uparrow |M] = \frac{n_{\uparrow} + \epsilon}{(n_{\uparrow} + \epsilon) + (n_{\downarrow} + \epsilon)} \ , \tag{5}$$

where $\epsilon > 0$ is a parameter. Previous research studies have shown that the performance of Lidstone Smoothing with $0 < \epsilon < 1$ is usually superior to $\epsilon = 1$ (i.e., Laplace Smoothing) [9].

4.2 Interpolation Smoothing

The above additive smoothing methods give all unseen user-ratings the same probability, which is not desirable if the user-ratings are generally imbalanced. A more reasonable smoothing strategy is to give different unseen user-ratings potentially different probabilities. This can be achieved by interpolating the maximum likelihood estimator of the item language model with a *background* language model M_b. Such a background language model can be specified a priori based on the domain knowledge. For example, in on-line shopping, users tend to be risk-averse so thumb-up should probably be given a lower probability than thumb-down in the background language model. More often, we may want to estimate the background language model based on the entire item catalogue. Suppose that there are totally N items in the catalogue. Let $p_{\uparrow}$ and $p_{\downarrow}$ denote the thumb-up probability and the thumb-down probability respectively in the background language model. Obviously $p_{\downarrow} = 1 - p_{\uparrow}$, so the background language model is determined as long as $p_\uparrow$ is known. There are two possible ways to estimate $p_\uparrow$ based on all the items $1, 2 \ldots, N$:

$$p_\uparrow = \Pr[\uparrow | M_b] = \frac{\sum_{i=1}^{N} n_\uparrow(i)}{\sum_{i=1}^{N}(n_\uparrow(i) + n_\downarrow(i))} \ , \tag{6}$$

$$p_\uparrow = \Pr[\uparrow | M_b] = \frac{1}{N}\sum_{i=1}^{N} \frac{n_\uparrow(i)}{n_\uparrow(i) + n_\downarrow(i)} \ . \tag{7}$$

Their difference is that in the former equation each user-rating contributes equally while in the latter equation each item contributes equally to the background language model. Which way is a better choice depends on which of these two assumptions is more sensible for the application domain.

Absolute Discounting Smoothing The idea of this smoothing method is to lower the probability of seen user-ratings by subtracting a constant from their counts, and then interpolate it with the background language model:

$$s(n_\uparrow, n_\downarrow) = \Pr[\uparrow | M] = \frac{\max(n_\uparrow - \delta, 0)}{n_\uparrow + n_\downarrow} + \sigma p_\uparrow \ , \tag{8}$$

where $\delta \in [0, 1]$ is the discount constant parameter, and $\sigma = 1 - (\max(n_\uparrow - \delta, 0) + \max(n_\downarrow - \delta, 0))/n$ so that all probabilities sum up to one.

Jelinek-Mercer Smoothing The idea of this smoothing method is to interpolate the maximum likelihood estimator of each document language model with the background language model using a fixed coefficient to control the amount of smoothing:

$$s(n_\uparrow, n_\downarrow) = \Pr[\uparrow | M] = (1 - \lambda)\frac{n_\uparrow}{n_\uparrow + n_\downarrow} + \lambda p_\uparrow \ , \tag{9}$$

where $\lambda \in [0, 1]$ is the fixed coefficient parameter.

Dirichlet Prior Smoothing The idea of this smoothing method is to move from frequentist inference to Bayesian inference where probabilities are measures of uncertainty about an event. Before we see any user-rating for item i, we should have a prior belief about the probability for it to get thumb-ups which is given by $p_\uparrow$ from the background languagde model. After we see a user-rating for item i, we should revise or update our belief accordingly, i.e., increase $\Pr[\uparrow |M]$ when we see a thumb-up and decrease it otherwise. How much adjustment is appropriate depends on the probability distributions. Since there are only two random events (thumb-up or thumb-down), the natural choice is to model their occurrences as a binomial distribution for which the conjugate prior is a beta distribution. The beta distribution is the special case of the Dirichlet distribution with only two parameters. In order to keep the terminology consistent with the Information Retrieval literature, we call this Bayesian smoothing method Dirichlet Prior smoothing [3]. Such a prior essentially assumes that every item "by default" has $\mu p_{\uparrow}$ thumb-ups and $\mu p_{\downarrow} = \mu(1 - p_{\uparrow})$ thumb-downs:

$$s(n_{\uparrow}, n_{\downarrow}) = \Pr[\uparrow | M] = \frac{n_{\uparrow} + \mu p_{\uparrow}}{n_{\uparrow} + n_{\downarrow} + \mu} \ , \tag{10}$$

where $\mu > 0$ is a parameter that determines the influence of our prior. Consequently, when we pool these pseudo-counts with the actual counts of user-ratings observed in the data, we would effectively interpolate the maximum-likelihood estimator of each item language model $M(i)$ with the background language model M_b using a dynamic coefficient that changes according to the number of user-ratings received so far: with more and more user-ratings available, the probabilities estimated using Drichlet Prior smoothing would be closer and closer to the maximum-likelihood estimator based on the observed data only.

4.3 Other Smoothing Techniques

There are many other smoothing techniques in Statistical Language Modelling, such as Good-Turing smoothing [7], but they do not seem to be suitable for our task because we only have two distinct "terms": thumb-ups and thumb-downs.

4.4 Generalisations

The proposed approach to ranking of items based on binary ratings (thumb-ups and thumb-downs) can be generalised to graded rating systems straightforwardly by taking each graded rating as multiple thumb-ups and thump-downs. Thus the "query" is still just one thumb-up, and each "document" (item) is still just a bag of thumb-ups and thumb-downs. For example, a 3-star rating in the 5-star scale system can simply be regarded as 3 thumb-ups and 5-3=2 thumb-downs. However, the semantic distance between 2-stars and 3-stars may be different from that between 3-stars and 4-stars. It is possible to take this into account by learning a real number of semantic thumb-ups for each graded rating from the user clickthrough data etc.

Furthermore, our approach can also be easily extended to take the ageing of user-ratings into account without affecting the computational efficiency through Time-Sensitive Language Modelling [10] techniques.

5 Axiomatic Examination

Which of the above mentioned ranking method, existing or proposed, is the best? To answer this question, we propose to examine their score functions in an axiomatic framework. The axioms that we use here are two fundamental principles in economics developed by Carl Menger [4] which nowadays are accepted as "irrefutably true" and widely used to interpret numerous economic phenomena.

Definition 1. *Given a score function s for user-rating based ranking of items, the* ***marginal utility*** *u of an additional thumb-up or thumb-down is the amount of difference that it can make to the score:*

$$\Delta_{\uparrow}^{(s)}(n_{\uparrow}, n_{\downarrow}) = s(n_{\uparrow}+1, n_{\downarrow}) - s(n_{\uparrow}, n_{\downarrow}) \ ,$$

$$\Delta_{\downarrow}^{(s)}(n_{\uparrow}, n_{\downarrow}) = s(n_{\uparrow}, n_{\downarrow}) - s(n_{\uparrow}, n_{\downarrow}+1) \ ,$$

where $n_{\uparrow}$ and $n_{\downarrow}$ are the current numbers of thumb-ups and thumb-downs respectively.

Axiom 1. ***The Law of Increasing Total Utility***
For any pair of non-negative integer numbers of thumb-ups and thumb-downs $n_{\uparrow}, n_{\downarrow} \in \mathbb{Z}^*$, a reasonable score function s must satisfy the following rules:

$$\Delta_{\uparrow}^{(s)}(n_{\uparrow}, n_{\downarrow}) > 0 \ ,$$

$$\Delta_{\downarrow}^{(s)}(n_{\uparrow}, n_{\downarrow}) > 0 \ ,$$

which imply that each additional thumb-up or thumb-down should always make the score higher or lower correspondingly.

Axiom 2. ***The Law of Diminishing Marginal Utility***
For any pair of non-negative integer numbers of thumb-ups and thumb-downs $n_{\uparrow}, n_{\downarrow} \in \mathbb{Z}^*$, a reasonable score function s must satisfy the following rules:

$$\Delta_{\uparrow}^{(s)}(n_{\uparrow}, n_{\downarrow}) > \Delta_{\uparrow}^{(s)}(n_{\uparrow}+1, n_{\downarrow}) \ ,$$

$$\Delta_{\downarrow}^{(s)}(n_{\uparrow}, n_{\downarrow}) > \Delta_{\downarrow}^{(s)}(n_{\uparrow}, n_{\downarrow}+1) \ ,$$

which imply that the difference made by each additional thumb-up or thumb-down to the score should decrease as the number of thumb-ups or thumb-downs increases.

The above two axioms reflect our intuition about what a reasonable score function should be like.

Proposition 1. *The Difference method satisfies Axiom 1 but violates Axiom 2.*

Proposition 2. *The Proportion method violates both Axiom 1 and Axiom 2.*

Proposition 3. *The Absolute Discounting smoothing method violates both Axiom 1 and Axiom 2.*

Proposition 4. *The Jelinek-Mercer smoothing method violates both Axiom 1 and Axiom 2.*

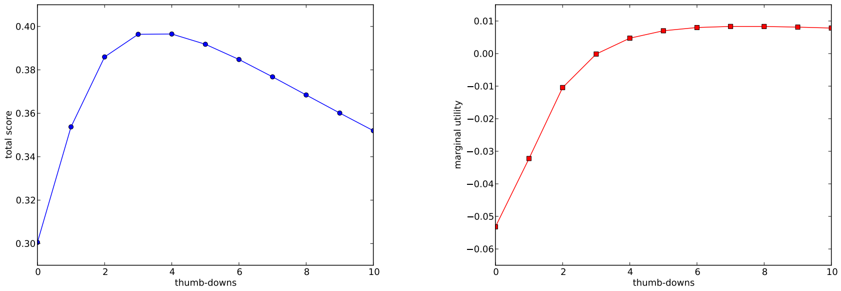
It is relatively easy to show that the above propositions are true, by checking the score functions at the boundary condition $n_{\downarrow} = 0$, so their proofs are omitted.

Proposition 5. *The Wilson Interval method violates both Axiom 1 and Axiom 2.*

Proof. This can be shown by checking the score function (3) with $n_\uparrow = 1$.

It violates the Law of Increasing Total Utility, because along with the increase of $n_\downarrow$ the total score is not monotonically decreasing, as shown in Figure 4(a).

It violates the Law of Diminishing Marginal Utility, because along with the increase of $n_\downarrow$ the marginal utility is not decreasing but increasing, as shown in Figure 4(b). $\square$



(a) total score

(b) marginal utility

Fig. 4. The Wilson interval $s(n_\uparrow, n_\downarrow)$ with $n_\uparrow = 1$.

Theorem 1. *The Dirichlet Prior smoothing method satisfies both Axiom 1 and Axiom 2.*

Proof. The score function (10) obeys the Law of Increasing Total Utility because

$$\begin{aligned}
& \mathit{\Delta}_\uparrow^{(s)}(n_\uparrow, n_\downarrow) \\
= & \; s(n_\uparrow + 1, n_\downarrow) - s(n_\uparrow, n_\downarrow) \\
= & \; \frac{n_\uparrow + 1 + \mu p_\uparrow}{n_\uparrow + 1 + n_\downarrow + \mu} - \frac{n_\uparrow + \mu p_\uparrow}{n_\uparrow + n_\downarrow + \mu} \\
= & \; \frac{n_\downarrow + \mu(1 - p_\uparrow)}{(n_\uparrow + n_\downarrow + \mu)(n_\uparrow + n_\downarrow + \mu + 1)} \\
> & \; 0 \; ;
\end{aligned}$$

$$\begin{aligned}
& \mathit{\Delta}_\downarrow^{(s)}(n_\uparrow, n_\downarrow) \\
= & \; s(n_\uparrow, n_\downarrow) - s(n_\uparrow, n_\downarrow + 1) \\
= & \; \frac{n_\uparrow + \mu p_\uparrow}{n_\uparrow + n_\downarrow + \mu} - \frac{n_\uparrow + \mu p_\uparrow}{n_\uparrow + n_\downarrow + 1 + \mu} \\
= & \; \frac{n_\uparrow + \mu p_\uparrow}{(n_\uparrow + n_\downarrow + \mu)(n_\uparrow + n_\downarrow + \mu + 1)} \\
> & \; 0 \; .
\end{aligned}$$

The score function (10) obeys the Law of Diminishing Marginal Utility, because

$$\begin{aligned}
&\Delta_{\uparrow}^{(s)}(n_{\uparrow}, n_{\downarrow}) - \Delta_{\uparrow}^{(s)}(n_{\uparrow}+1, n_{\downarrow}) \\
&= \frac{n_{\downarrow} + \mu(1-p_{\uparrow})}{(n_{\uparrow}+n_{\downarrow}+\mu)(n_{\uparrow}+n_{\downarrow}+\mu+1)} - \frac{n_{\downarrow} + \mu(1-p_{\uparrow})}{(n_{\uparrow}+1+n_{\downarrow}+\mu)(n_{\uparrow}+1+n_{\downarrow}+\mu+1)} \\
&= \frac{n_{\downarrow} + \mu(1-p_{\uparrow})}{n_{\uparrow}+n_{\downarrow}+\mu+1} \left(\frac{1}{n_{\uparrow}+n_{\downarrow}+\mu} - \frac{1}{n_{\uparrow}+n_{\downarrow}+\mu+2} \right) \\
&> 0 \ ;
\end{aligned}$$

$$\begin{aligned}
&\Delta_{\downarrow}^{(s)}(n_{\uparrow}, n_{\downarrow}) - \Delta_{\downarrow}^{(s)}(n_{\uparrow}, n_{\downarrow}+1) \\
&= \frac{n_{\uparrow} + \mu p_{\uparrow}}{(n_{\uparrow}+n_{\downarrow}+\mu)(n_{\uparrow}+n_{\downarrow}+\mu+1)} - \frac{n_{\uparrow} + \mu p_{\uparrow}}{(n_{\uparrow}+n_{\downarrow}+1+\mu)(n_{\uparrow}+n_{\downarrow}+1+\mu+1)} \\
&= \frac{n_{\uparrow} + \mu p_{\uparrow}}{n_{\uparrow}+n_{\downarrow}+\mu+1} \left(\frac{1}{n_{\uparrow}+n_{\downarrow}+\mu} - \frac{1}{n_{\uparrow}+n_{\downarrow}+\mu+2} \right) \\
&> 0 \ .
\end{aligned}$$

□

Corollary 1. *The Laplace smoothing method satisfies both Axiom 1 and Axiom 2.*

Proof. It is because the Laplace smoothing method (4) is a special case of the Dirichlet Prior Smoothing method (10) with $\mu = 2$ and $p_{\uparrow} = 1/2$. □

Corollary 2. *The Lidstone smoothing method satisfies both Axiom 1 and Axiom 2.*

Proof. It is because the Lidstone smoothing method (5) is a special case of the Dirichlet Prior Smoothing method (10) with $\mu = 2\epsilon$ and $p_{\uparrow} = 1/2$. □

The axiomatic examination results about the existing and proposed ranking methods are summarised in Table 1. It is clear that only the score functions given by the Dirichlet Prior smoothing method as well as its special cases (Laplace smoothing and Lidstone smoothing) can satisfy both axioms borrowed from economics. Therefore the Dirichlet Prior smoothing method is our recommended solution for user-rating based ranking of items.

6 Conclusions

The main contribution of this paper is to show how the Information Retrieval techniques — Probability Ranking Principle and Statistical Language Modelling (with Dirichlet Prior smoothing) — can provide a *well justified* solution to the problem of user-rating based ranking of items in Web 2.0 applications.

The axiomatic approach to Information Retrieval has been studied by Bruza and Huibers [11], Fang and Zhai [12], and a few other researchers. To our knowledge, this paper is the first work that formulates user-rating based ranking of items as an Information Retrieval problem and examines the ranking methods for this problem from an axiomatic perspective.

	Increasing Total Utility	**Diminishing Marginal Utility**
Difference	Y	N
Proportion	N	N
Wilson Interval	N	N
Laplace smoothing	Y	Y
Lidstone smoothing	Y	Y
Absolute Discounting smoothing	N	N
Jelinek-Mercer smoothing	N	N
Dirichlet Prior smoothing	Y	Y

Table 1. The axiomatic examination results.

References

1. Robertson, S.E.: The Probability Ranking Principle in IR. In: Readings in Information Retrieval. Morgan Kaufmann (1997) 281–286
2. Ponte, J.M., Croft, W.B.: A language modeling approach to information retrieval. In: Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR), Melbourne, Australia (1998) 275–281
3. Zhai, C.: Statistical Language Models for Information Retrieval. Morgan and Claypool (2008)
4. Menger, C.: Principles of Economics. New York University Press (1981)
5. Wilson, E.B.: Probable inference, the law of succession, and statistical inference. Journal of the American Statistical Association **22** (1927) 209–212
6. Ripley, B.D.: Pattern Recognition and Neural Networks. Cambridge University Press (1996)
7. Chen, S.F., Goodman, J.: An empirical study of smoothing techniques for language modeling. Technical Report TR-10-98, Harvard University (1998)
8. Zhai, C., Lafferty, J.D.: A study of smoothing methods for language models applied to ad hoc information retrieval. In: Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR), New Orleans, LA, USA (2001) 334–342
9. Agrawal, R., Bayardo, R., Srikant, R.: Athena: Mining-based interactive management of text databases. In: Proceedings of the 7th International Conference on Extending Database Technology (EDBT), Konstanz, Germany (2000) 365–379
10. Zhang, D., Lu, J., Mao, R., Nie, J.Y.: Time-sensitive language modelling for online term recurrence prediction. In: In Proceedings of the 2nd International Conference on the Theory of Information Retrieval (ICTIR), Cambridge, UK (2009) 128–138
11. Bruza, P., Huibers, T.W.C.: Investigating aboutness axioms using information fields. In: Proceedings of the 17th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval (SIGIR), Dublin, Ireland (1994) 112–121
12. Fang, H., Zhai, C.: An exploration of axiomatic approaches to information retrieval. In: Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR), Salvador, Brazil (2005) 480–487