TESI DI LAUREA

A crowdsourcing platform to collect and rank pros and cons on controversial issues

candidato
Fela Winkelmolen

relatore
Marina Ribaudo

UNIVERSITÀ DEGLI STUDI DI GENOVA

Luglio 2014

Contents

Chapter 1

Introduction

This is a very early draft. The sections marked as *[Work in progress]* are, well very much work in progress so should not be read yet :) They might contain incomplete thoughts and sentences, spelling and logical errors and similar. Even what is not marked as such is in a very early draft state.

1.1 Background and motivation

Technologies such as wikis, forums, blogs and question and answer platforms are widely used to help internet users to collaboratively collect information and make the result available on the internet. Different users might have different opinions about what the most relevant information is, so there are many ways in which to determine the end result.

Forums and blogs chronologically display everything that gets written, the main differences between the two is that blogs do not have threads and are written by a single person or a small group of people and are therefore generally less inclusive. Wikis either require the users to reach a consensus about what to display and how to display it or there might be some hierarchy of administrators that determines who has the latest word.

In question and answer applications votes determine what answers get displayed first. Voting works best when it is relatively easy to agree on what a good answer is. On controvertial issues where opinions are polarized users will often vote based on political opinion rather then relevance and quality. This is the reason why for example the StackExchange questions and answers platform, to maintain the quality of its content, bans questions that are prone to strong opinions[1]. This leaves out many questions, such as those in the form 'What are the pros and cons of ...'. In this thesis I will

[1] for example http://stackoverflow.com/questions/6548826/angular-js-vs-backbone-js

describe and implement a web applications that tries to deal exactly with this kind of questions.

The main tecnique I use to counter the effect of polarized votes is to rank pros and cons indipendently. This way there will be no competition between the two factions. The voting merely determines the internal rankings of the pros, and the internal ranking of the cons.

1.2 Building an online community

The content and voting behaviour will depend on the community that forms itself arount the website. As Aaron Swarz notes[2] when discussing wikipedia and bringing ideas of the open enciclopedia to other contexts, a community is a combination of technology, rules and people. Each component influences the other: different people will want different rules and guidelines, while the specific rules and conventions will attract certain kind of people. Technology can keep people away or attract them, and can help enforce some of the rules.

While this work concentrates on the technological aspects, it is important to bear in mind that the technology will have strong bidirectional ties to the community that uses it and whith the way this community behaves.

1.3 Similar web applications

To my knowledge there is no application with the exact same goal and with the same approach, but there are many that share it at least partially and have interesting properties worth looking at.

1.3.1 Procon.org

This is a website with the goal of collecting pros and cons on controversial issues. This is, on the surface, very similar to what this work aims at. A deeper look, however, will show some very fundamental differences. The most important one is the role of the community and the way content is selected. The main content of the procon.org website is collected by a selected group of paid editors. Nobody ouside of this group has any effective influence on the content and on how it is displayed. The role of the internet users is relegated to the comments section, which are a secondary feature of the website. While the comments are subdivided in pros and cons and provide a simple voting

[2] http://www.aaronsw.com/weblog/morewikipedias

mechanism, there are no rules nor guidelines to what kind of comments are allowed and based on what they should be up or downvoted.

The resulting comments are instructive of how the technology (the voting mechanism), and the community rules influence the content. On most comment pages the votes are heavily biased toward one side of the issue, for example the commenters are videly favorable to medical use of marijuana, so much that the lowest rated pro comment has a higher rating than the highest rated con comment[3]. Other topics, while less drastic, show similar trends. The ratings mostly indicates if the view of the commenter is viewed favourably, rather than indicating if it raises any interesting and important point useful to evaluate the issue. If the comments where mixed together, as opposed to being subdivided in pros and cons one would have to look very much down the page to find any reasons for the minority opinion.

Additionaly, as the commenting behaviour is not subject to any guideline there are a lot of duplicate comments, very subjective opinions, rethorical answers without much substance, and personal experiences. In the specific case it is probably not meant to be otherwise, but if users are to create the main content, rules about the kind of content allowed are needed.

1.3.2 StackExchange

There exist a vast number of question and answer platforms, I will concentrate on the StackExchange network as it is one of the most successful ones, has many interesting properties, and strongly influences my work. A general property of question and answer websites is that they allow everybody to answer and to vote to decide the best questions, which will be displayed first. While many websites allow a great variety of quesitons to be asked, Stack Exchange is composed of multiple websites, each focused on a single topic, with fairly restricted rules about the questions and answers that are allowed. This, coupled with the reputation system, has the effect of creating very strong and healty communities centered around the various topics.

1.3.3 Wikipedia

On wikipedia many people successfully cooperate to write about a wide range of topics, including controversial ones. It is an example of a community build by the combination of people, rules and technology. On wikipedia community guidelines and consensus are of great importance.

[3] as of June 2013 http://medicalmarijuana.procon.org/view.answers.reader-comments.php?questionID=1325

As wikipedia it will be possible to edit content, athough the type of edits are more restricted, and new users will not be allowed to edit other peoples content.

Chapter 2

Authentication with multiple email addresses

[Work in progress]

2.1 Mozilla Persona

User registration and authentication is done through Mozilla Persona[1]. Mozilla Persona allows the user to log in using any existing email address, without the need of sharing a password with the web application [3]. Tipically, the user will be required to choose a password that is shared only with `persona.org` and can be used for an unlimited number of websites that support the protocol, without incurring in the security risks normally associated with password reuse. If the email provider supports the identity provider (IDP) portion of the protocol the password is not needed and the user authenticates directly with its provider, similarly to what happens with OpenID and OAuth. In the worst case, if the user has never used Persona before, the user interaction will be analogous to standard email registration, with the only drawback that it happens in a popup owned by a third party domain (`persona.org`).

2.2 Linking together multiple email addresses

Many internet users own and manage more than one email address [2], and it can happen that a user might does not remember which one was used during registration. To overcome this inconvenience a user that logs in with a new email address will be prompted with two choices. He can either create a

[1] `https://developer.mozilla.org/en-US/docs/Persona`

new account or link the new email address to an existing account. If more addresses are linked to an account any of them can be used to sign in.

Offering the ability to link different email addresses should be done with care as it can introduce a number of security vulnerabilities:

1. it increases the **attack surface**: with multiple email addresses linked to an account the attacker needs to compromise just one to gain unauthorized access to the user account; if the ability to login with multiple email addresses is to be retained this cannot be totally countered, but three kind of measures can be taken to help minimize the damage: first, making sure not all kind of actions are available through all email addresses, thus limiting the damage; second, improving detection; and third, in case of detection, making recovery easy

2. if an email address is temporarily compromized, the attacker might be able to **add a new email address** to be able to access the account even after control of the original email address is lost, this relies on the author not noticing the added address or somohow being unable to remove it

3. if in the above scenario the attacker has the ability to also **remove the original email** address he will gain full control of the account and the original owner will have lost it completely

The feasability of these attacks depends on the exact circumstances in which email addresses can be removed and added. A combination of some of the following measures could be used to minimize the security risk, while keeping the application usable.

2.2.1 Removal of addresses

If it is discovered that an attacker has added an email address (as in scenario 2) it is important to be able to remove that address. Removing old adresses can also be used to avoid an unecessary number of addresses to be linked to an account. However, giving users the possibility to remove email addresses makes it possible for attackers to remove legitimate ones, as in scenario 3. It is thus clear that email removal alone is not enough to protect from the outlined security risks.

2.2.2 Delayed actions

Security sensitive actions such as removing and/or adding email addresses can be delayed, for example by a fixed number of days. In the meantime, after every login, the user will be shown a notice informing him he can cancel the action.

Delayed *removal*, while helping to avoid the removal of legitimate addresses, would slow down actions to counter scenario 2 (illegitimate email addresses added by an attacker). This latter scenario can be countered by also delaying the addition of a new email address, which however would have usability drawbacks. So again delayed actions alone might not provide a satisfying solution.

2.2.3 Email confirmation

Email confirmation from every email address for certain actions makes it harder for the attacker to take those actions. Unfortunately it also can block the user from accomplishing legitimate tasks in the case of compromised email addresses. Email confirmation from some of the email address is required to change the password of mozilla persona.

2.2.4 Primary email

To prevent the kind of escalation described, sensitive actions can be restricted to a single primary email address. This way, for the given actions the security provided will be equal to the case of a single email address. Other email addresses can still be used to login, until the user wants to perform one of the restricted actions, which will require him to use its primary email address.

One drawback is that if the control of the primary email is lost, control of the whole account is also lost. This is the same as what happens with single email sign in, and it is the users burden to choose an email provider with good recovery options.

There can be various choises regarding which actions will require a primary address. A liberal approach would be to only restrict the actions of adding and removing other email addresses. It would still be possible for an attacker to partly compromize an account by compromizing one of its secondary email addresses, but as soon as the breach is discovered, the affected email address can be removed.

The most conservative approach, on the other hand, would be to not allow any actions other than displaying to primary email address which the user should use to be able to actively use the account. While more secure, it requires an additional action on the users part when he logs in with a secondary email address. This means that other email addresses serve the sole purpose of reminding the user which was his primary address in the case he forgets it.

To add dependency of primary email, some way to recover by secondary email addresses could be available...

2.2.5 Proposed solution

The proposed solution is to require primary email addresses. Furthermore, delayed addresses are provided.

Email confirmation is used my mozilla persona, so implicitely if the user wants to change the password.

[Note on possible usability corner case: when linking email addresses on new address login, care should be taken it is really the users intention to link the addresses]

The two most important use cases are that of a user logging in for the first time, and one logging in with the correct email address.

From the primary email address it is possible to change the address

Chapter 3

Ranking by vote

3.1 Bayesian average

The objective when ranking elements is to show the most relevant and interesting information at the top. This is done by using the votes as an indication of how interesting the information was for past users, and assuming that what holds for past users is likely to hold for future users[1].

The simplest and most widespread ranking schemes suffer from well known weaknesses ([7]). Zhang et al. [7] propose two axioms that should reasonably be respected by a good ranking algorithms: first, up and downvotes should respectively (strictly) increase and (strictly) decrease the score, and second, if a vote of the same type gets added its weight should be strictly smaller than the previous vote. They find three ranking algorithm consistent with these axioms, their most general result uses what they call Dirichlet prior smoothing. Let $n_\uparrow$and $n_\downarrow$ be the number of up and down-votes, they will be ranked by the following score

$$\frac{n_\uparrow + \mu p_\uparrow}{n_\uparrow + n_\downarrow + \mu}$$

where $p_\uparrow \epsilon (0,1)$ and μ are fixed parameters that will determine how the ranking behaves in the case of few available votes. Intuitively, as the total number of votes increases, the score assigned by the algorithm will tend towards the ratio $positive\, votes/total\, votes$, if there are few votes it will tend towards the fixed "default" score $p_\uparrow$; the higher μ the higher the weight of

[1] When different users have very different opinions this might not hold, an example is on controversial topic, where polarization could make opinions diverge widely. Although this is exactly the setting of this thesis, the main component of this polarization is removed by raking the claims of the two main positions separately.

$p_\uparrow$. To better understand this, the score could be rewritten as the weighted average between $p = positive\,votes/total\,votes$ and the user defined $p_\uparrow$

$$\frac{w_1 p + w_2 p_\uparrow}{w_1 + w_2}$$

where the weight w_1 is equal to the number of total votes $n_\uparrow + n_\downarrow$, and w_2 is equal to μ, the weight given to the prior information.

Here is an example of the result on random imaginary votes, the first column shows the resulting score, followed by number of upvotes and the number of downvotes:

97	+119	−3
88	+12	−1
85	+85	−14
82	+7	−1
80	+2	−0
68	+11	−5
67	+0	−0
57	+50	−39
48	+13	−15
38	+1	−4
23	+7	−29
10	+0	−18

There is a generalization of this algorithm for the case where the votes can assume more than two values, this version is currently in use by a number of websites. IMDB, for example, uses it for its top 250 list and calls it true bayesian estimate.

The formula also has a more elegant probabilistic giustification. It has also been variously referred to online as *bayesian average* and *bayesian ranking*, but without to my knoledge ever giving a complete explanation. [5], although without giving the above formula, sketches the basic reasoning from which the result can be derived.

The model rests on the assumption that there is a certain population of potential voting users, and that we would like to estimate the percentage q of them that, if given the chance, would vote positively. A detailed derivation of the final formula is given in appendix B.

I will use bayesian average to denote the score calculated this way, and bayesian ranking to denote the ranking process that uses that score.

3.2 Limitations and assumptions

"All models are wrong, but some are useful" – George E. P. Box

Every model is just an approximation of reality, but it is hoped to capture the main interactions that are of interest. Very often approximations or assumptions are made without giving them much thought. I believe that a honest analyses should give them some space, to be able to better assess how well and under which condition a model might be warranted. This is specially important in the case where where there are no obvious benchmarks to assess quality. I will also note that assuming that the result of a benchmark will correlate to usefulness in real life is in itself a model.

3.2.1 Justification of the model

The probabilistic setting assumes there is a population of voters from which a number of individuals selected at random cast their vote. To start to understand the possible weaknesses of the model it is worth asking how much this is an accurate description of what happens in practice.

There are at least two ways in which the independence assumption might fail. First, the users that visit the webpage might not really be randomly chosen from the population of interest, for a number of reasons, for example if somebody likes a comment very much he might refer other people to it, which might have similar opinions. Second, users might be influenced by the current score[2].

Another implicit assumption which subtly breaks, is that the total number of votes has been assumed as being given, where in fact it is unlikely to be totally independent of q. The score will influence the position that an element will have on the page, which in turn influences the number of votes.

Notwithstanding all these assumptions, most of which are unlikely to have a big effect overall, if the only information available is the number of upvotes and downvotes, estimating q is the best that can be done, and the bayesian approach yield optimal results *if* the prior and loss function are chosen correctly.

[See also http://www.bloomberg.com/news/2013-07-19/j-k-rowling-and-the-chamber-of-literary-fame.html which shows the votes are not really independent and influence each other]

[2] The effect of peer pressure on judgment, also known as Bandwagon effect, has been extensively documented, see for example [1]

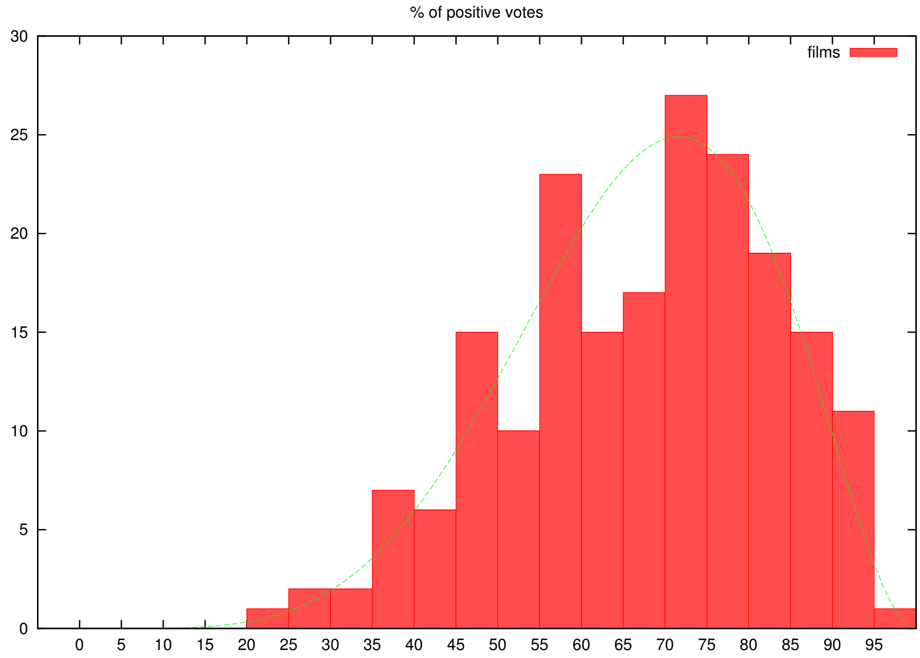

Figure 3.1: The graph shows, in red, a number of random films grouped by their Rotten Tomatoes score, which is equal to the percentage of positive user ratings. The green line shows an ideal distribution, assuming a beta distributed prior.

3.2.2 Prior

When applying bayesian inference a few additional assumptions have been made that merit further consideration.

When calculating the posterior probability the prior has been assumed to be known and be equal to the beta distribution, without giving much justification other than algebraical convenience. The beta distribution does however have a number of desirable properties. It is worth noting that the assumption does not force a single prior, but rather allows for a family of distribution functions, by varying $p \uparrow$ and μ. $p_\uparrow$ is equal to the mean of the distribution, and will be the "default" value in the case when no information is available, the obvious way to chose it is to set it equal to the mean of all historical votes in the application. μ is inversely proportional to the variance, and indicates how much the score tent to vary in historical data. The higher μ the more votes will be required before the score will differ significantly from $p_\uparrow$.

Too see how realistic such a prior is, I analyzed the distribution in a real example, to do this I chose the Rotten Tomatoes user scores. Rotten

Tomatoes is a website containing information and reviews about movies; the users can score the movies and the final score will be equal to the percentage of positive votes. Given the large number of votes, from many hundreds to many thousands in the case of the movies I have analyzed, this percentage can be assumed a good approximation of q. The films analyzed have been collected from available lists such as movies currently playing in the theaters and recent DVD releases. Lists such as best and worst movies have not been used as obviously biased for our use. The API does not allow to directly access the number of user votes, so the number of critics reviews has been used to exclude obscure movies which might not have enough votes to make the data reliable. This might add a slight bias, however the fact that the distribution is still approximately beta distributed is an indication of the good properties of this distribution. The result can be seen in Figure 3.1, the overlayed green line represents a beta distribution with parameters set to match the data.

3.2.3 Loss function

As the prior, the loss was chosen mainly for its convenient algebraic properties. The loss function is the cost of estimating that an object has a percentage of approval equal to q where, if more information had been available, we would discover that the real percentage is $\tilde{q}$. In our case this cost is assumed to be $(q - \tilde{q})^2$, which penalizes outliers; this is to say that having an error of 2Δ is assumed as bad as having 4 errors of magnitude Δ. Another option would have been the absolute error. Given that there is no strong reason to prefer either, the more tractable square error has been chosen.

A case could also be made for asymmetric loss functions, for example [5] proposes a loss function that tends to give greater cost to over-ranking as opposed to under-ranking, with the result that in the case of uncertainty the score will be lower. A similar effect, without the additional computational cost, could be obtained by setting $p_{\uparrow}$ to a lower value in the prior distribution. There is however no conclusive justification for being conservative in sorting comments, and having uncertain scores come quickly to the top might help counter the "fastest gun in the west" problem (see below 3.2.5).

3.2.4 Using additional information

The above model takes into account only the votes. There might be other information that could be useful for ranking. For example which authors voted and the number of people who have viewed an element without voting on it. How to best use views is analyzed in the next section, while reputation systems which look into who voted are described in the next chapter.

3.2.5 Fastest gun in the west problem

A know problem often arising when ranking comments by votes is that the oldest comments tend to have a much higher chance making it to the top, purely as a consequence of being there first. This is particularly true in the case of ranking by number of votes, or by the difference of upvotes and downvotes. Newer comments will start at the bottom and thus, other things being equal, will be much less likely to collect votes, making their difference in standing relative to the top only bigger as time passes.

This shows ranking in a different perspective, because it is not just important to show the most relevant information first, but also show the most relevant information to future users. There could be cases where it is not particularly likely that an object is relevant, but it is still advantageous to make it visible to attract votes and be able to show more accurate information in the future.

Any ranking that estimates the proportion of positive votes has the following property: given a large enough number of viewers a comment will eventually reach its real score. However the number of viewers, specially for new comment, is finite, furthermore at any given time there might be a big number of of recent comment that do not yet have much votes. This has been a reason for not using score algorithms that tend to underevaluate the score in case of high uncertainty.

3.3 Number of views

An interesting piece of information that has not been used is the number of users who have seen an object but did not find it interesting enough to vote on it. So the current algorithm would give the same score to an item viewed by 100 people all of which voted it up, and one viewed by 1000 people only 100 of which voted it up. The next section deals with estimating the number of views, but assuming they are known, how can they be incorporated into the algorithm?

Starting from the obvious consideration that a view is better than a downvote and worse than an upvote, the easiest way to generalize the bayesian average is to suppose a view equals to $v \epsilon [0, 1]$ upvotes and $1 - v$ downvotes. This could be formally justified by assuming that if forced to vote the fraction of non voters that would cast a positive vote is equal to v.

3.3.1 Estimation of number of views

Obviously the number of page views alone is not a good measure, when a page is loaded only a fraction of the content of a page is really viewed [6]. Furthermore not everybody is as likely to vote on content they view.

Therefore, on a given page, only users that voted on at least one item get counted as having viewed its content, and will get counted only once. This solves the statistical noise that would otherwise be caused by different people behaving differently. Elements further down are less likely to be viewed, so their view count should not be increased by as much. To be able to do this the number of views can assume non discreet values, for example 0.5 views is a valid number and equals to 50% chance of a view.

It is thus needed to have a a function that maps page position to probability of being viewed. This function can, aside from a multiplication factor, be approximated by the probability of a attracting a vote given the page position. This probability can be estimated by using the historical data in the whole application. Until this data is available a rough estimate will have to do. Lets suppose the probability of obtaining a vote at position i has been calculated and is equal to

$$f(i) = \frac{votes}{pageviews}$$

This can be normalized by setting the maximum value equal to 1

$$\tilde{f}(i) = \frac{f(i)}{\underset{i}{max}\, f(i)}$$

At this point if elements in position $i_1, i_2, ..., i_n$ get voted by a user, the views count of element at position i will be incremented by the value assumed by the above function, scaled such that the less visible item that has been voted on equals 1:

$$min(\frac{\widetilde{f}(i)}{\underset{j=i_1,i_2,...,i_n}{min} \widetilde{f}(j)}, 1)$$

A more complex algorithm to estimate views could also incorporate cursor movement, which has been shown to strongly correlate with eye movement[4].

3.3.2 Interpretation of number of views

[Work in progress]

There are different way too look at incorporating the number of views in the ranking.

One way is to consider them neutral votes, with a value between 0 and 1. Using a low value means ignored scores will rank low compared to controversial ones, and viceversa.

A different approach would be to calculate the bayesian average separatedly from the bayesian average of the percentage of people that voted on a given argument. So you would obtain two scores: one that estimates the number of positive votes/the number of negative votes, another that estimates the number of votes/number of views. Those two can then be combine with an arbitrary function. With the appropriate function the result would be equivalent to the previous case.

There might be some variants, for example: number of positive votes/number of views and number of negative votes/number of views. Or even number of positive votes/number of votes and number of positive votes/number of votes. There are three variables: number of positive votes, number of negative votes, number of views, that can be paired in any way to obtain a bayesian average.

3.4 Normalization

The bayesian average is normalized to the interval $[0, 10]$ before being displayed to the user with one digit after the decimal point. Having one digit after the decimal point makes sure the user understands that the score is not a simple sum of the up and downvotes.

Chapter 4

Reputation

[Work in progress]

There are many kind of reputation systems and different reasons why to use one. A frequent reason to create a reputation system is to better recognize good content and trustworthy scores, with the ultimate goal of ranking the content, an example of this is PageRank. Another reason is to encourage users to contribute and to vote (karma). This is one of the main reasons why I have used a reputation system, together with requiring a minimum reputation for certain actions. This is a simple but effective way to conter unsofisticated sybil attacks.

4.1 Sybil attack

4.2 Anonimous users

To encourage participation of anonimous users and to be able to provide "some" ranking even if no other users provided any vote, two tools are used: first anonimous users and 0-reputation users have a vote that is an order of magnitude less strong that that of registered users, second there is a maximum number of votes allowed by anonimous users per item, this way if there are even very few votes by normal users at most anonimous users can be deciding in tie breaking. If there are no votes, the risk of sybil attacks is warranted by the possibility of providing some ranking and allowing anonimous users to participate.

Bibliography

[1] SE Asch. *Effects of group pressure upon the modification and distortion of judgments.* Carnegie Press, 1951.

[2] Benjamin M. Gross and Elizabeth F. Churchill. Addressing constraints: multiple usernames task spillage and notions of identity. In Mary Beth Rosson and David J. Gilmore, editors, *CHI Extended Abstracts*, pages 2393–2398. ACM, 2007.

[3] M. Hanson, D. Mills, and B. Adida. Federated Browser-Based Identity using Email Addresses. In *W3C Workshop on Identity in the Browser*, 2011.

[4] Jeff Huang, Ryen W. White, and Susan T. Dumais. No clicks, no problem: using cursor movements to understand and improve search. In Desney S. Tan, Saleema Amershi, Bo Begole, Wendy A. Kellogg, and Manas Tungare, editors, *CHI*, pages 1225–1234. ACM, 2011.

[5] Evan Miller. Bayesian Average Ratings. Online article, http://www.evanmiller.org/bayesian-average-ratings.html, 2012.

[6] J. Nielsen. How Users Read on the Web, 1997. Online article, http://www.useit.com/alertbox/9710a.html.

[7] Dell Zhang, Robert Mao, Haitao Li, and Joanne Mao. How to Count Thumb-Ups and Thumb-Downs: User-Rating Based Ranking of Items from an Axiomatic Perspective. In Giambattista Amati and Fabio Crestani, editors, *ICTIR*, volume 6931 of *Lecture Notes in Computer Science*, pages 238–249. Springer, 2011.

Appendix A

FAQ

See website :)

Appendix B

Here follows the proof that the Dirichlet Prior proposed by Zhang et al. ([7]) can be obtained by bayesian inference, under the assumption of a prior following the beta distribution and a minimum square error loss function. A formalization of the voting problem that uses bayesian reasoning has been sketched in a 2012 online article by Miller ([5]). Similar results have been obtained in the literature under the name additive smoothing.

Let q denote the proportion of the population that would give a positive vote[1], i.e. the value we want to estimate. Let the prior probability of q be known and be equal to the beta-distribution in the parameters α and β

$$p(q) = Beta(\alpha, \beta) = \frac{q^{\alpha-1}(1-q)^{\beta-1}}{B(\alpha, \beta)}$$

where the normalization factor B denotes the beta function.

Every vote is assumed to come from a randomly selected user, therefore every vote will have q probability of being positive and $1-q$ probability of being negative. The probability of having $n_\uparrow$ positive votes and $n_\downarrow$ negative votes after $n = n_\uparrow + n_\downarrow$independent trials is given by the probability of a binomial experiment

$$p(n_\uparrow, n_\downarrow | q) = \binom{n_\uparrow + n_\downarrow}{n_\uparrow} q^{n_\uparrow}(1-q)^{n_\downarrow}$$

The posterior probability of q can be calculated by Bayes' theorem

$$p(q | n_\uparrow, n_\downarrow) = \frac{p(n_\uparrow, n_\downarrow | q)p(q)}{\int p(n_\uparrow, n_\downarrow | q)p(q)dq}$$

The specific prior distribution has been chosen because it is the conjugate prior of a binomial experiment, meaning that the posterior probability of

[1] Here it has been assumed the population is equal to the users that would vote, which means we leave out the users that would vote neither up nor down.

Bayes' theorem can be written in term of elementary functions, condition which does not hold in general. More specifically, the posterior probability in the case of beta-distributed prior is known to be another beta-distribution, in the specific case

$$p(q|n_{\uparrow}, n_{\downarrow}) = Beta(\alpha + n_{\uparrow}, \beta + n_{\downarrow})$$

At this point the usual approach is to choose q by minimizing a loss function, in our case we choose to minimize the mean square error (MSE), which can be shown to be equal to finding the expected value:

$$\underset{y}{min}\, \mathrm{MSE}[x] = \underset{y}{min} \int_0^1 p(x)(x-y)^2 dx$$

$$= \underset{y}{min} \int_0^1 p(x)x^2 dx + y^2 \int_0^1 p(x) dx - 2y \int_0^1 p(x) x dx$$

$$= \underset{y}{min}(y - 2y\mathrm{E}[x]) = \mathrm{E}[x]$$

in the case of our distribution $Beta(\alpha + n_{\uparrow}, \beta + n_{\downarrow})$ the expected value is equal to

$$\frac{n_{\uparrow} + \alpha}{n_{\uparrow} + n_{\downarrow} + \alpha + \beta}$$

from which it will suffice to replace $\mu = \alpha + \beta$ and $p_{\uparrow} = \alpha/(\alpha + \beta)$ to obtain the formula used by Zhang et al.

$$\frac{n_{\uparrow} + p_{\uparrow}\mu}{n_{\uparrow} + n_{\downarrow} + \mu}$$