WikiRating Proposition
Version 0.3

Davide Valsecchi <valsecchi.davide94@gmail.com>
Alessandro Tundo <alessandrotundo94@gmail.com>

March 20, 2017

Contents

1 Introduction

The main purpose of this project is the creation of an automated system to evalute the reliability and quality of articles and pages on a wiki-style learning platform.

The Problem In wiki-style platform, new contents have to be evaluated by expert users to become reliable and to be shared with the public. That work is fondamental to guarantee the correctness of the platform's content but could slow down the development of highly partecipated works with a lot of changes. Moreover finding a lot of experienced editors in a big community could be really difficult. How can we ensure the quality of contents and meanwhile make the system automatic?

The Solution Our base idea is measuring the trustworthiness of the users to evaluate their writing reliability and their ability to review other's works. Every document is voted by users and, using our estimate of their reliability, we can get an overall rating of it.
The complete history of the document is followed to have a precise determination of its quality: every version has a weight related to its importance in the total document.
Every user's action implies an adjustment of his rating, trying to measure his real reliability. In a system like this, the activity of the users is also important: more active users are a resource for the platform and have to be prized for their work.

2 The User

Every user of ***WikiRating*** is identified by a unique ID and by two parameters that measures his performance: the trust (or reliability) coefficient (α) and the activity coefficient (θ).

In the system also an *Anonymous* user is present wit a fixed and very low reliability. Since the unregistered users cannot write any page on the platform, this user is used only as a reference for anonymous votes.

2.1 Trust Coefficient

This parameter is a number between 0 to 1, we call it α. The higher it is, the more we can trust the user's judgements and we can consider his writings correct. We want α to depend on two parameter a and b that can vary between 0 and 1.

a is linked to the quality of user writing evaluated by other users' votes (as we'll describe in section 4.1, pag. 10).

b is linked to the measured ability of the user to vote other users' documents. It's is an important parameter because we have to measure the reliability also the users that don't write anything on the platform but act only as readers.

In the section 4 at page 10, we will describe how a and b are calculated and how they varies during user activity.

α can be calculated, in the simplest way possible, as a weighted mean of the two parameters:

$$\alpha = \frac{Aa + Bb}{A + B} \tag{1}$$

2.2 Activity Coefficient

This coefficient measures the activity level of a user in the platform. We call this number γ. If the value is between 0 and 1 the user is considered little active or inactive. If the value is between 1 to 2 the user is considered regular and if the value is greater than 2 he is regarded as really active. This coefficient is important in the calculations to modify ths user's trust coefficient, as we will see later. The activity coefficient is proportional to the ratio of the number of actions done by the user over time. Furthermore this coefficient decreases quickly over the time if the user is absent from the platform (See section 4.2, pag. 11).

2.3 Initial values for the user

The initial values for the coefficients must be set properly according to our estimation of the user's initial trust level. We can link it to domains of users' email. For example accounts that use a recognized institutional domain (like *@unimib.it*) will start with a trust coefficient higher that account that are registered with a student domain (like *@campus.unimib.it*). That methods can be used to distinguish teachers from students and from common users. The initial value of α could also vary over time, studying periodically the overall reliability of the users associated with every domain.

3 The Chain of Changes

We want now to describe how the ***WikiRating*** system acts on the Chain of Changes (CC) of a wiki page or document.

The history of a document can be described as a descendant chain. First of all a user creates a new document, then another user makes changes on the top of the first version and saves it: a new ring of the chain is added. Every time an user works on that document new subsequent rings are added, multiple branches are not allowed.

Let's start introducing how the rating system works.

3.1 Marks

Every registered user that visits a page can assign a vote between 1 to 10 that it is then scaled between 0 to 1 for the calculations. This mark refers to the latest version of the document that is always shown to visitors.

Every vote is associated with user's trust coefficient that determines the weight of the mark. The global evaluation of *the single version* is calculated as a weighted average of single votes over the trust coefficients:

$$\bar{V} = \frac{\displaystyle\sum_{i=1}^{n} \alpha_i \cdot V_i}{\displaystyle\sum_{i=1}^{n} \alpha_i} \tag{2}$$

where V_i are the votes and α_i are trust coefficients of the users.

So, let's make an example: Alice creates the page and receives some votes, then Bob makes some changes and a new ring of the chain is created. Obviously new votes on Bob's version not only evaluate Bob's work but also Alice's old work, depending on the number of Bob's changes. Instead, votes of Alice's work are not related at all with Bob's subsequent editing but cannot be thrown away because Bob has not created an entirely new document. We need a new method to calculate global evaluation for the page and above all a criterion to establish when a version is considered stable. Let's first explorate the latter.

3.2 Trustworthiness of Marks

Users are not all equally reliable: for example marks from an ungraduate student must be less influential than marks from a teacher. A calculation of global vote for a page does not ensure us that a document has been checked and evaluated *sufficently.* If a page is evaluated as nine but by five users only, we are not satisfied by the level of judgment. Furthermore our opinion could change depending on the five users' trust coefficient. We need a measure of the *trustworthiness of the marks* from now called $\boldsymbol{\sigma}$.

In our model, the reliability of the evaluation of a page version is not related only to the votes that it has received , but it is related to the quality and quantity of the votes, i.e. to their distribution and to the trust coefficient of the users. A well-evaluated version has not necessarily an excellent average of votes, but it has been carefully evaluated by a sufficient number of users that agree to a certain extent.

$\boldsymbol{\sigma}$ is an evaluation of the reliability of the average of marks of a document. That's the goal of σ parameter.

The σ of a user's vote, called σ_i, is calculated as (i is refered to the i-th users):

$$\sigma_i = \alpha_i^B \cdot (1 - \Delta V)^C \qquad \Delta V = |\bar{V} - V_i| \tag{3}$$

In a *single version* the overall σ value is calculated as:

$$\sigma = \sum_{i=1}^{n} \sigma_i = \sum_{i=1}^{n} \alpha_i^B \cdot (1 - \Delta V)^C \tag{4}$$

where B and C are constant that have to be chosen. Thus, σ depends on α, the trust coefficient of the user, and on the deviation of his vote from the weithed average of the votes as calculated in the eq. 2. If votes are distributed in a large range, their mean is less reliable as a global judgment, so σ is smaller. Wherease if votes are all close to each other, their mean has more significance and the *trustworthiness of the marks* σ increases. Obviously votes from more reliable users are more important than votes from users with a low α. Parameters B and C could be used to make higher α and $(1 - \Delta V)$ more important than smaller values.

Now, returning to the problem of the Chain of Changes, what happens if Bob edits the document before σ threshold is reached? As we have already said, new votes are more important that old ones, but Alice's work judgements cannot be thrown away.

3.3 The Decay Process

For every version of a page, we have a method to calculate the average vote (eq. 2, pag. 6) and the cumulative trustworthiness of marks, σ (eq. 4, pag. 7). Every version added to the document changes it but users votes the page in different point of its history.

Let's make an example: Alice creates a new document of 1500 characters; Bob finds two spelling errors and creates a new revision, made of 10 characters. Some time later, Alice's work has been voted by two people only because Bob has worked quickly. Bob's version instead has been voted by a lot of people.
In this case considering only Bob's revision's marks (the latest) as valid it's not a problem: the votes are related mainly to Alice's work because she has made the real big part of the writing. Votes given to Alice could almost be added directly.

If we reverse the situation things change a lot. Imagine that Alice, with her 1500 characters, receives a load of votes. Then, after some time, Bob makes a small correction and receives only a few votes. Can we consider only the lates marks as valid? In this case it would be wrong because the real judgements has been made by marks of the first version. After Bob's intervention, new votes are still related to previous version.

After the first two revisions, Charlie arrives on the page, decides to complete a poor paragraph of the document and saves a new version with 300 characters. He even receives a lot of valutations. In this case the writing has changed considerably and consequentially old votes are less significative: they lose importance compared to new votes relatively to the number of modified (added or removed) characters.

If votes lose importance with the addiction of new rings to the Chain of Changes, also trustworthiness of marks σ has to be scaled along the revisions to make the system coherent.

Now we'll descrive how this sort of ***Decay*** takes part in ***WikiRating*** system.

3.3.1 V Decay

The Decay of Votes or *V Decay* must give us a way to calculate the average mark of a stack of versions of a document given the votes and the number of changes of every ring. We called this mechanism "decay" just because at every new step of the chain, all previous calculations have to be scaled by a particular factor to take part in the next computing. Proceeding this way, step by step, is the best practice to make the computations the less complex possible with the accumulating of revisions.

Initial data Consider now a stack of revisions that begins with a new document [1]. Every version has received a number N_i of votes by N_i users, everyone with his α_i coefficient. Every user can vote only one time every version. Taking individually every version, we calculate it's weighted *partial* average of marks according to eq. 2, pag. 6, and we call those partial means $\bar{V_n}$ (n is related to the n-th version).

Decay factor (DF) Every version is characterized by an amount of changes: as parameters we have the previous and current length of a page, the number of characters added and removed. We have to create a **Decay Factor (DF) $\boldsymbol{\Phi}$**, function of these parameter to represent the information of how much a page has changed between two versions.

Total average of vtes Now, we start calculating a progressive *total average* $\tilde{V_i}$ of marks, starting from the first revision. We can use this simple recursive formula :

$$\tilde{V_1} = \bar{V_1} \quad \longrightarrow \quad \tilde{V_n} = \frac{\bar{V_n} + \Phi_n \cdot \tilde{V}_{n-1}}{1 + \Phi_n} \tag{5}$$

For every step, the *total* average depends on the *partial* average of votes for the current revision, but the previos *total* average is not lost. Actually, the new *total*

[1] what to do after a given stable version will be explained later in section **??**, pag. **??**

mark is a weighted mean between the current *partial* mark and the previous *total* mark over the **Decay Factor (DF) $\mathbf{\Phi}$**, that it's different for every n.

Now we have a clear view of how votes are scaled over the Chain of Changes, we must look at how σ parameter scales.

3.3.2 σ Decay

As we said in section 3.2, σ parameter measures the trustworthiness of marks, so if marks loses importance along the chain, the total sigma σ calculation should consider all the marks but has to evaluate more newer votes.

Initial data We know how to calculate σ for every *single* version from equations 4 and 3 on page 7. Consider a stack of revisions that begins with a new document. For every version, excepted the current, we can calculate che *partial* σ and we call that σ_n.

$$\sigma_n = \sum_{i=1}^{N_i} \sigma_i = \sum_{i=1}^{N_i} \alpha_i^B \cdot (1 - \Delta V)^C \qquad \Delta V = |\bar{V_n} - V_i| \tag{6}$$

where N_i is the number of votes to the n-th version, σ_i and α_i are related to the single vote and user. First of all, note that for *every version* the overall σ_n is calculated using only votes given to that version and that the partial average of marks $\bar{V_n}$ is used, not the total average $\tilde{V_n}$.

The calculation for past versions can be done only one time because data used doesn't change any more. For the current active version things change a little: in fact every new vote changes the partial average of marks. So, for the latest version σ_n must be calculated every time a vote is added.

Total σ Now, as in the case of Votes Decay, we have to calculate the *total* σ , called $\tilde{\sigma_n}$, starting from the first version.

$$\tilde{\sigma}_1 = \sigma_1 \quad \longrightarrow \quad \tilde{\sigma}_n = \sigma_n + \Phi_n \cdot \tilde{\sigma}_{n-1} \tag{7}$$

As Decay Factor (DF) Φ, obviously we use the same DF of the V Decay.

The effect of the DF is the same of that described in section 3.3.1. The only difference is that $\tilde{\sigma_n}$ is a linear combination of contibutes of every version and not a weighted mean.

4 Update User's Parameters

As described in section 2, α trust coefficient depends on two parameters: the quality of writing a and the qualilty of reviewing b. Both these parameters are changed every time an update process of the ***WikiRating*** database is completed: if the user is the author of one of the revision his a parameter changes, if the user has voted one on more of the revisions his b parameter changes.

4.1 Δa calculation

Influencial parameters

- In our model, since the α coeffient is related to the expected rank of the user if evaluated by other users, Δa must be proportional to the difference between the user's current α and the average mark assigned to his work, $\Delta V = (\bar{V} - \alpha)$.
- Then Δa should depend on the importance of the user's work in the page, thus on the ration λ_n / μ_{tot}, where λ_n is the number of changes by the user and the μ_{tot} is the total number of characters of the document at the moment of his revision.
- Also σ of votes given to the user could influence the width of Δa. Note that we are considering the *partial* σ or σ_n of the user's version, as calculated in formula 6. If a lot of trusted users vote the document, Δa should be more important. To eliminate the influence of the number of voters to this parameter we should use the *reduced* σ , $\mathring{\sigma}_n$:

  $$\mathring{\sigma}_n = \frac{\sigma_n}{N_v} \tag{8}$$

  where N_v is the number of votes.

When a user writes a revision for a document, every subsequent version is related in some measure with his work. Thus, to evaluate the writing qualities of a user we must consider not only the votes assigned to his version but also all the votes given to subsequent versions till the stable one. Obviously the importance of following versions' votes must be related to the number of changes *from* the first user's version. We need a kind of *Forward Decay* for Δa calculation.

User's version First of all let's consider the versions directly written by the current user in exam, Alice. For every Alice's version n, $\Delta\tilde{a}_n$ is calculated using the following formula:

$$\Delta\tilde{a}_n = \Delta V_n \cdot \mathring{\sigma}_n \cdot \frac{\lambda_n}{\mu_{tot}} \tag{9}$$

using the notation convention of the previous section 3.3 we can expand the formula as:

$$\Delta\tilde{a}_n = \left(\bar{V}_n - \alpha\right) \cdot \frac{\sigma_n}{N_v} \cdot \frac{\lambda_n}{\mu_{tot}} \tag{10}$$

Forward Decay Now let's consider versions after Alice's work. Starting with the first version after Alice's version we can calculate every *partial* $\Delta\bar{a}_j$ as:

$$\Delta\bar{a}_j = \Psi_j \cdot \Delta V_j \cdot \mathring{\sigma}_j \tag{11}$$

$$\Delta\bar{a}_j = \Psi_j \cdot \left(\bar{V}_j - \alpha_{Alice}\right) \cdot \frac{\sigma_j}{N_v} \tag{12}$$

where $\bar{V}_j$ and σ_j are related the j-th version, and α_{Alice} is always the α coeffient of Alice.

$\boldsymbol{\Psi_j}$ is the Decay Factor (DF) that weighs as usually the importance of following versions (instead of Φ_n that is used with the previous versions). One proposed model for this factor is the inverse of the previous decays factor:

$$\Psi_j = \frac{1}{\Phi_j} \tag{13}$$

We can observe that if the versions after Alice's one introduces a lot of changes, their votes are not influential for Alice's Δa. Otherwise the newer votes are also related to Alice's work and consequentally are important for her rank.

Total $\boldsymbol{\Delta a}$ We can now calculate the total Δa for Alice:

$$\Delta a_{tot} = F \cdot \gamma \cdot \left(\sum_{n=1}^{A} \Delta\tilde{a}_n + \sum_{j=1}^{B} \Delta\bar{a}_j \right) \tag{14}$$

where F is a parameter chosen a posteriori, A is the number of the versions written by Alice and B is the number of versions between Alice's one and the stable one.

If Alice has written more than once in a single chain of changes:

- $\Delta\tilde{a}_n$ is calculated for every Alice's version with formula 10.
- after every Alice's version $\Delta\bar{a}_j$ is calculated with the Forward Decay process with formula 12 for every version till the Decay Factor is below a certain thresold (to improve performances) or another Alice's version is met.

As you can see finally Δa_{tot} depends on the user's activity coefficient γ. If the user is inactive ($0 < \gamma < 1$) then Δa_{tot} is decreased. If the user is active ($\gamma > 1$), then it is increased. An active user could quickly increase his α if he receives a lot of good votes.

4.2 Δb calculation

The b parameter measures the user's ability to judge other users' works. This parameter is important because a lot of users on the platform are only viewers and not writers: their α is fundamental because they are who votes the documents, hence it has to reflect the quality of their actions.

Influencial parameters

- First of all, an important parameter is the deviation of the user's vote from the mean of votes. If the vote is too distant from the mean it could imply that the user is not good at evaluate properly the document. We could define a parameter φ calculated as:

  $$\varphi = \left(\frac{\delta - \Delta V}{\delta}\right)^G \quad \longrightarrow \quad \Delta b \propto \delta \tag{15}$$

  where G is a parameter determined a posteriori and δ is the *standard deviation* of votes assigned to a *single* version:

  $$\delta = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(V_i - \bar{V}\right)^2} \tag{16}$$

  where N is the number of votes and $\bar{V}$ is the *partial* average of votes of the current version. ΔV is the absoulute value of the difference between the vote V_i in exam and $\bar{V}$:

  $$\Delta V = |V_i - \bar{V}| \tag{17}$$

  There are different scenarios:

  - if $\Delta V < \delta$, φ is always less then 1 and positive. Thus $\Delta b \propto \varphi$ is positive and user's α increases.
  - if $\delta < \Delta V < 2\delta$, φ is negative and less then 1. Thus b decreases by a small amount and α decreases.
  - if $\Delta V > 2\delta$, φ is negative and greater then 1. So b decreases much and so does α .
  - finally is $\Delta V \simeq \delta$ then $\varphi \simeq 0$. b and α doesn't change because the vote is distributed normally.

- Also user's α is important to measure Δb. If a trusted user votes far from the average is little probable that he has judged in the wrong way: we trust that user! Variations of b linked with the distribution of votes must be weighted by user's α in this way:

  $$\Delta b \propto (1 - \alpha)^H \tag{18}$$

  The exponent H could be setted a posteriori. If a user is already trusted his Δb changes slowly.

Total Δb Given a Chain of Changes between two stable versions and a specific user that has partecipated in the poll, we calculate Δb_n for each of his vote using the *partial* average of marks $\bar{V_n}$ of every different version:

$$\Delta b_n = (1 - \alpha)^H \cdot \varphi \tag{19}$$

$$\Delta b_n = (1-\alpha)^H \cdot \left(\frac{\delta - \Delta V}{\delta}\right)^G \tag{20}$$

Remeber that every user can vote only one time a revision but can vote more than one in Chain of Changes. Then Δb_n are summed up to make *total* Δb of a single user:

$$\Delta b_{tot} = L \cdot \gamma \cdot \sum_{i=1}^{N} \Delta b_i \tag{21}$$

where N is the number of the user's vote and L is a parameter to determine a posteriori as usual. Also the γ activity coefficient of the user is present as in the Δa_{tot} process. Active users' coefficients could change more quickly than inactive users' ones.

4.3 γ calculation

User's activity coefficient γ is important because it influences the size of the variation of a and b, so it directly influences changes of α . The logic behind the calculations is that if a user is active we can change a lot his rating because it is frequently updated. If a user is inactive changes must occur more slowly because we don't know when coefficients will be updated again and it could be not in the short period. Activity coefficient could be calcaluted as:

$$\gamma = P \cdot \frac{actions}{days} \cdot Q^{-inactive\ days} \tag{22}$$

γ is proportional to the number of actions by day and to a constant P. Then, for every day [2] of inactivity, γ is scaled by an exponential factor with base $Q > 1$.

[2] another fixed period of time could be chosen

5 Pros and Cons, Further development

5.1 Pros

Expandable For how it is build, ***WikiRating*** system is only a valid *base* for a more complex and complete rating system. A lot of parameters can be further modeled and a lot of mechanism have to be developed more.

Flexible The system has a lot of parameters that influence the calculation to be determined a posteriori with tests and simulations. This increases a lot the flexibility of the system.

Scalable Calculations involved in the system have been thought keeping in mind computational complexity. We try to minimize the number of necessary operations.

Fair The system tries to evaluate fairly the quality of every user and of every document. The more a user is active, the more he can improve his reputation. Users' engagement is prized and measured. Document's reliability is evaluated thoughtful and we try to discourage bad polling.

5.2 Cons

Number of users To work correctly, ***WikiRating*** system needs a lot of users and a lot of votes to every document. Only with a lot of data, marks and their trustworthiness, measured by σ, make sense.

Review ability Users' ability to review other works measurament (See section 4.2, pag. 11) could be improve a lot. We need to test the current mechanism because it could evaluate normal votes as bad ones if not set correctly. The system is more good at rating users how writes documents than users that vote them. This could be a problem because the majority of users are reviewers. The problem could be solved measuring more precisely the b coefficient.