

# WikiRating Proposition

## Version 0.1

Davide Valsecchi <valsecchi.davide94@gmail.com>

Mario Marini <mamarini94@gmail.com>

August 31, 2015

## Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>The User</b>	<b>4</b>
2.1	Trust Coefficient . . . . .	4
2.2	Activity Coefficient . . . . .	5
<b>3</b>	<b>The Chain of Changes</b>	<b>6</b>
3.1	Marks . . . . .	6
3.2	Trustworthiness of Marks . . . . .	7
3.3	The Decay Process . . . . .	8
3.3.1	V Decay . . . . .	8
3.3.2	$\sigma$ Decay . . . . .	10
3.3.3	After a stable version . . . . .	11
3.4	Reaching the threshold . . . . .	11
3.5	Finding $\sigma_s$ . . . . .	12
<b>4</b>	<b>Update User's Parameters</b>	<b>13</b>
4.1	$\Delta a$ calculation . . . . .	13
4.2	$\Delta b$ calculation . . . . .	15
4.3	$\gamma$ calculation . . . . .	16
<b>5</b>	<b>Pros and Cons, Further development</b>	<b>17</b>
5.1	Pros . . . . .	17
5.2	Cons . . . . .	17

## 1 Introduction

The main purpose of this project is the creation of an automated system to evaluate 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 fundamental to guarantee the correctness of the platform's content but could slow down the development of highly participated 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. Then, we found a way to identify stable versions using the distribution of the voters and votes: a kind of limit that it has to be reached to guarantee that a sufficient number of users has rated the content.

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 ( $\alpha$ ) and the activity coefficient ( $\theta$ ).

### 2.1 Trust Coefficient

This parameter is a number between 0 to 1, we call it  $\alpha$ . The higher it is, the more we can trust the user's judgements and we can consider his writings correct. This coefficient comes from the mean of Beta probability density function.

$$B[x; a, b] = \frac{x^{a-1} (1-x)^{b-1}}{\int_0^1 x^{a-1} (1-x)^{b-1} dx} \quad (1)$$

$$\alpha = \bar{x} = \frac{a}{a+b} \quad (2)$$

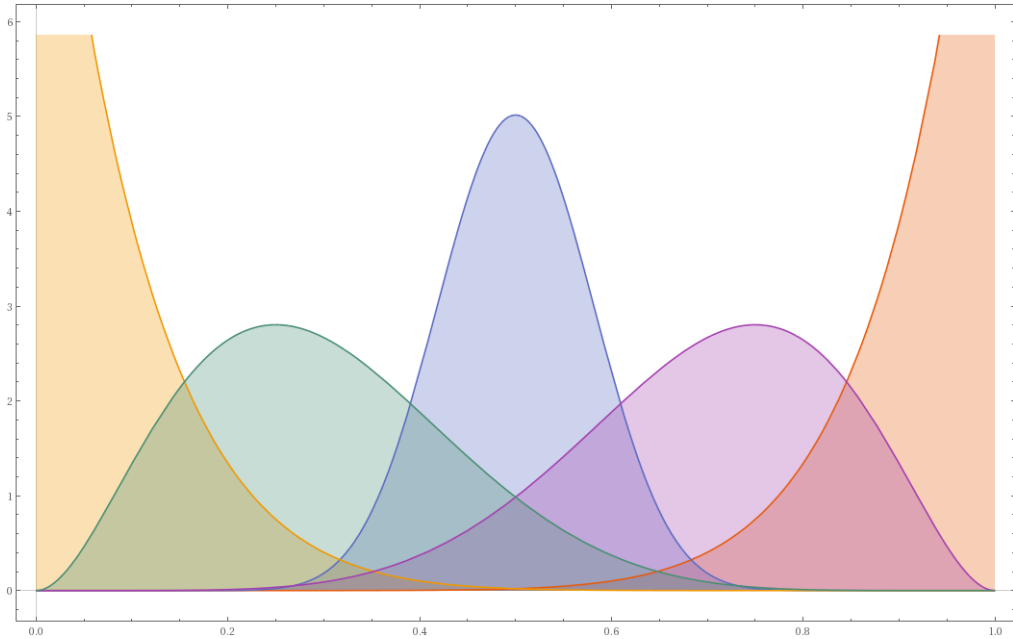


Figure 1: Trust distribution

We can see that  $\alpha$  depends on two parameter  $a$  and  $b$  that can vary between 0 and arbitrary positive values. If  $a$  is high,  $\alpha$  tends to 1. If  $b$  is high,  $\alpha$  tends to 0. In the Figure 1 we can see Beta distribution with different  $a, b$  combination:  $a = b$  (blue),  $a > b$  (pink),  $a < b$  (green),  $a \gg b$  (dark orange),  $a \ll b$  (orange).

In our model we decide to link  $a$  to the quality of user writing (evaluated by other users as we'll describe in section 4.1, pag. 13). The max value of  $a$  is called  $A_1$  and it has to be determined a posteriori. Then we substitute  $b$  for

$(A_2 - b)$  (where  $A_2$  is a constant equal to the max value of  $b$  ) and we link  $b$  to the reliability of user judgements.

$$\alpha = \bar{x} = \frac{a}{a + (A - b)} \quad (3)$$

Thus, with high  $a$  and  $b \rightarrow A$  the trust coefficient tends to 1. Therefore, the trust coefficient depends on user writing and user ability to evaluate other works on the platform. In the section 4 at page 13, we will describe how  $a$  and  $b$  are calculated and how they varies during user activity.

## 2.2 Activity Coefficient

This coefficient measures the activity level of a user in the platform. We call this number  $\gamma$ . 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. 15).

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  $\alpha$  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 (*ChCh*) 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.

Versions are not all equally important: in the life of a page there are periods of heavy work and frequent changes and periods when the content is quite stable and definite. Usually, in wiki platform, stable versions must be marked by expert editors who check the document and decide that it is ready. In *WikiRating* system we thought about a threshold mechanism able to decide automatically when a version must be marked as stable.

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. Even if the last version is not marked as stable, it must be shown to users (reporting that as a warning) because it has to be evaluated. Votes are not shown on the page because other judgements must not be influenced (See 3.4, pag. 11).

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:

$$\bar{V} = \frac{\sum_{i=1}^n \alpha_i \cdot V_i}{\sum_{i=1}^n \alpha_i} \quad (4)$$

where  $V_i$  are the votes and  $\alpha_i$  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 evaluated 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 explore 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 *sufficiently*. 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  $\sigma$ .

**A document's version is marked as stable when it has been sufficiently evaluated by users, thus when the overall  $\sigma$  has passed an established threshold.** In our model, stability of a document's 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 stable version has not necessarily an excellent average of votes, but if it's stable we can be sure that it's vote has been carefully evaluated by sufficient number of users and they agree to a certain extent.  **$\sigma$  is an evaluation of the reliability of the average of marks of a document.** That's the function of  $\sigma$  parameter.

The  $\sigma$  of a user's vote, called  $\sigma_i$ , is calculated as ( $i$  is referred to the  $i$ -th users):

$$\sigma_i = \alpha_i^B \cdot (1 - \Delta V)^C \quad \Delta V = |\bar{V} - V_i| \quad (5)$$

In a *single version* the overall  $\sigma$  value is calculated as:

$$\sigma = \sum_{i=1}^n \sigma_i = \sum_{i=1}^n \alpha_i^B \cdot (1 - \Delta V)^C \quad (6)$$

where  $B$  and  $C$  are constant that have to be chosen. Thus,  $\sigma$  depends on  $\alpha$ , 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. 4. If votes are distributed in a large range, their mean is less reliable as a global judgment, so  $\sigma$  is smaller. Wherease if votes are all close to each other, their mean has more significance and the *trustworthiness of the marks*  $\sigma$  increases. Obviously votes from more reliable users are more important than votes from users with a low  $\alpha$ . Parameters  $B$  and  $C$  could be used to make higher  $\alpha$  and  $(1 - \Delta V)$  more important than smaller values.

All the power of this system lives in the determination of the value of the threshold of  $\sigma$  that established it has been reach a new stable versione. Ideas about how to determine dinamically this fundamental value will be explored later (See section 3.5, pag. 12).

Now, returning to the problem of the Chain of Changes, what happens if Bob edits the document before  $\sigma$  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

When new not stable versions stack on the latest milestone of a document, every ring of the famous Chain of Changes is ranked by a variable number of users. For every version, we have a method to calculate the average vote (eq. 4, pag. 6) and the cumulative trustworthiness of marks,  $\sigma$  (eq. 6, pag. 7), but it's not sufficient to be able to determinate when a new milestone must be created and it's final mark. Furthermore, version are not all the same: there could be little corrections or big addictions and they are really different.

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 latest 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  $\sigma$  has to be scaled along the revisions to make the system coherent.

Now we'll describe 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 <sup>1</sup>. Every version has received a number  $N_i$  of votes by  $N_i$  users, everyone

---

<sup>1</sup>what to do after a given stable version will be explained later in section 3.3.3, pag. 11



with his  $\alpha_i$  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. 4, pag. 6, and we call those partial means  $\bar{V}_n$  ( $n$  is related to the  $n$ -th version).

Every version is characterized by an amount of changes: considering numerically both addition and deletions as equal changes, we call that number  $\lambda_n$  for every version.

**Decay factor (DF)** Now, we start calculating a progressive *total average*  $\tilde{V}_i$  of marks, starting from the first revision. We use this simple recursive formula, :

$$\tilde{V}_1 = \bar{V}_1 \quad \longrightarrow \quad \tilde{V}_n = \frac{\bar{V}_n + \Phi_n \cdot \tilde{V}_{n-1}}{1 + \Phi_n} \quad (7)$$

For every step, the *total* average depends on the *partial* average of votes for the current revision, but the previous *total* average is not lost. Actually, the new *total* mark is a weighted mean between the current *partial* mark and the previous *total* mark over the **Decay Factor (DF)**  $\Phi$ , that it's different for every  $n$ .

As we explained with examples at the beginning of 3.3 section, *DF*  $\Phi$  must depend on the relative number of changes between subsequent revisions.

Doing so  $\Phi_n$  can be expressed as:

$$\Phi_n = \left(1 - \frac{\lambda_n}{\lambda_{n_{tot}}}\right)^D \quad (8)$$

where  $\lambda_{n_{tot}}$  is given by the partial sum of all the changes from the beginning to the  $n$ -th version *included*:

$$\lambda_{n_{tot}} = \sum_{i=1}^n \lambda_i \quad (9)$$

The exponent  $D$  can be determined a posteriori. Let's analyze the behavior of the *DF*  $\Phi$ .

- If the  $n$ -th revision of the document introduces small changes compared to all the previous ones of other users, then  $\Phi_n \sim 1$ . Thus, old votes will be mediated with new votes with almost the same weight in formula 7. That's coherent with the fact that old votes are still important for the document because only little changes has been made.
- If the  $n$ -th revision of the document introduces a number of changes of the order of the partial sum of the previous ones,  $\lambda_{(n-1)_{tot}}$ , then  $\Phi_n \sim \left(\frac{1}{2}\right)^A$ . In fact:

$$\Phi_n = \left(1 - \frac{\lambda_n}{\lambda_{n_{tot}}}\right)^D = \left(1 - \frac{\lambda_n}{\lambda_{(n-1)_{tot}} + \lambda_n}\right)^D \quad (10)$$

and we are considering the case  $\lambda_n \sim \lambda_{(n-1)_{tot}}$ . In this case older votes are becoming less and less important than newer and that's consistent with the fact that the number of new changes is of the same order of the previous changes: there's a big alteration in the document and must be reevaluated.

- Finally, if the n-th revision introduces a number of changes really bigger than the partial sum of all the previous ones,  $\Phi_n \rightarrow 0$ . Because  $\lambda_n$  is included in  $\lambda_{n_{tot}}$ , to make  $\Phi \rightarrow 0$  the n-th changes must be a really bend in the document. In this case old votes are practically deleted from the weighted average as eq. 7 shows.

A general observation is that older votes are never more important than newer ones. In the limit case of a new version with very little number of changes  $\lambda$ , old and new votes have the same importance. This is consistent with the fact the updated judgements are always more reliable.

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

### 3.3.2 $\sigma$ Decay

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

**Initial data** We know how to calculate  $\sigma$  for every *single* version from equations 6 and 5 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*  $\sigma$  and we call that  $\sigma_n$ .

$$\sigma_n = \sum_{i=1}^{N_i} \sigma_i = \sum_{i=1}^{N_i} \alpha_i^B \cdot (1 - \Delta V)^C \quad \Delta V = |\bar{V}_n - V_i| \quad (11)$$

where  $N_i$  is the number of votes to the n-th version,  $\sigma_i$  and  $\alpha_i$  are related to the single vote and user. First of all, note that for *every version* the overall  $\sigma_n$  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  $\sigma_n$  must be calculated every time a vote is added.

**Decay Factor (DF)** Now, as in the case of V Decay, we have to calculate the *total*  $\sigma$ , called  $\tilde{\sigma}_n$ , starting from the first version.

$$\tilde{\sigma}_1 = \sigma_1 \quad \longrightarrow \quad \tilde{\sigma}_n = \sigma_n + \Phi_n \cdot \tilde{\sigma}_{n-1} \quad (12)$$

As Decay Factor (DF)  $\Phi$ , obviously we use the same DF of the V Decay:

$$\Phi_n = \left( 1 - \frac{\lambda_n}{\sum_{i=1}^n \lambda_i} \right)^D \quad (13)$$

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.

### 3.3.3 After a stable version

So far we have considered decays starting from the creation of the document, so that  $\tilde{V}_1 = \bar{V}_1$  and  $\tilde{\sigma}_1 = \sigma_1$ . If a document is modified after a version marked as stable, its vote must influence the later calculations on new revisions: the work on the document does not restart from the beginning. All the formulas and decays described previously are valid with a change:

$$\tilde{V}_1 = \bar{V}_1 + \Theta \cdot V_{stable} \quad \tilde{\sigma}_1 = \sigma_1 + \Theta \cdot \sigma_{stable} \quad (14)$$

The total average mark and  $\sigma$  of the stable base version are inserted in the decay with a DF  $\Theta$ . This factor  $\Theta$  could be modeled in various way:

- $\Theta$  could depend on the number of characters of the stable version. The longer the latest stable version is, the more its votes are important.
- $\Theta$  could be a fixed parameter, chosen a posteriori
- $\Theta$  could depend on the number of users have voted the stable version. The more it has been voted, the less must lose importance.

## 3.4 Reaching the threshold

Let's now focus on the mechanism that makes a modified document reaching the threshold of  $\sigma$  ( $\sigma_s$ ), i.e. when the latest revision becomes a stable milestone.

Every time the latest version receives a new vote:

1. the *partial* average of marks  $\bar{V}_n$  is updated
2. the *partial*  $\sigma_n$  is updated (as described in section 3.3.2), because  $\bar{V}_n$  has changed.
3. the *total*  $\tilde{\sigma}_n$  is updated by formula 12
4. new value of  $\tilde{\sigma}_n$  is compared to  $\sigma_s$
5. if  $\tilde{\sigma}_n \geq \sigma_s$  the latest version must be labeled as stable because it has been sufficiently evaluated:
  - (a) the *total* average of marks  $\tilde{V}_n$  is calculated by formula 7 and assigned to the new stable milestone as ultimate mark.
  - (b) the final mark is published on the page and it's no more secret.
  - (c) all calculations to update users' coefficient are triggered as we will see in section 4 on pag. 13.
  - (d) polling can be stopped or no stopped. There are two scenery:
    - i. polling is stopped: reaching the threshold  $\sigma_s$  the document has been sufficiently evaluated. The final vote remains immutable until new revisions come.

- ii. polling is not stopped: new votes for the stable version are computed.  $\bar{V}_n$  and  $\tilde{V}_n$  are updated, but  $\sigma_n$  and  $\tilde{\sigma}_n$  are not. For every new vote,  $\sigma_i$  is calculated by formula 5 using  $\tilde{V}_n$  as average of marks. Then every  $\sigma_i$  is added directly to  $\tilde{\sigma}_n$  without any DF (document is not changed). Doing so  $\sigma$  could only grow<sup>2</sup>.
6. if  $\tilde{\sigma}_n \leq \sigma_s$  the threshold has not be reached yet.
  7. when a new vote is received and we return to point 1.

If the document is modified before its latest version has reached  $\sigma_s$ , then a new ring of the Chain of Changes is created and the following operations take place for the previous version:

1. the definitive  $\bar{V}_{n-1}$  is calculated.
2. the definitive  $\tilde{V}_{n-1}$  is calculated with the correct DF  $\Phi_{n-1}$ .
3. the definitive  $\sigma_{n-1}$  is calculated given the definitive  $\bar{V}_{n-1}$ .
4. the definitive  $\tilde{\sigma}_{n-1}$  is calculated given the correct DF  $\Phi_{n-1}$ .

With all the parameters set for the previous version, we can now do the calculations described in the previous mechanism for the current version.

### 3.5 Finding $\sigma_s$

Choosing the right  $\sigma_s$  threshold of every document is the key of all *WikiRating* system. This parameter is not fixed and it could be determined by some dynamical conditions: here some proposals.

- $\sigma_s$  could depend on the number of visitors on the page. The more users have viewed a page, the more it's necessary a depth evaluation.
- $\sigma_s$  could depend on the category of the subject of the document
- $\sigma_s$  could depend on the number on internal and external links to the page.
- $\sigma_s$  must be linked also to the history of a document. When a page is created from scratch  $\sigma_s$  should be high, because it should be more difficult to reach the first stable version than to modify later the document. Then  $\sigma_s$  for following revisions could depend on the number of users involved in the writing of the document or on the lenght of it.

In general the more a page is important for the platform, the more  $\sigma_s$  must be high to guarantee an evaluation in deep.

---

<sup>2</sup>This point is not so clear: polling could continue, but could  $\sigma$  change after it has reached the threshold?

## 4 Update User's Parameters

When a stack of revisions become stable, users' work has been carefully evaluated: it's time to change users' trust coefficient to reflect their action within this document. As we described in section 2,  $\alpha$  trust coefficient depends on two parameters: the quality of writing  $a$  and the quality of reviewing  $b$ . Both these parameters are changed when a user has taken part of a chain of revision become stable: 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 $\Delta a$ calculation

#### Influential parameters

- In our model, since the  $\alpha$  coefficient is related to the expected rank of the user if evaluated by other users,  $\Delta a$  must be proportional to the difference between the user's current  $\alpha$  and the average mark assigned to his work,  $\Delta V = (\bar{V} - \alpha)$ .
- Then  $\Delta a$  should depend on the importance of the user's work in the overall document, thus on the ration  $\lambda_n/\mu_{tot}$ , where  $\lambda_n$  is the number of changes by the user and the  $\mu_{tot}$  is the total number of characters of the document.
- Also  $\sigma$  of votes given to the user could influence the width of  $\Delta a$ . Note that we are considering the *partial*  $\sigma$  or  $\sigma_n$  of the user's version, as calculated in formula 11. If a lot of trusted users vote the document,  $\Delta a$  should be more important. To eliminate the influence of the number of voters to this parameter we should use the *reduced*  $\sigma$ ,  $\hat{\sigma}_n$ :

$$\hat{\sigma}_n = \frac{\sigma_n}{N_v} \quad (15)$$

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  $\Delta a$  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:

$$\Delta \tilde{a}_n = \Delta V_n \cdot \hat{\sigma}_n \cdot \frac{\lambda_n}{\mu_{tot}} \quad (16)$$

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

$$\Delta \tilde{a}_n = (\bar{V}_n - \alpha) \cdot \frac{\sigma_n}{N_v} \cdot \frac{\lambda_n}{\mu_{tot}} \quad (17)$$

**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:

$$\Delta\bar{a}_j = \Psi_j \cdot \Delta V_j \cdot \sigma_j \quad (18)$$

$$\Delta\bar{a}_j = \Psi_j \cdot (\bar{V}_j - \alpha_{Alice}) \cdot \frac{\sigma_j}{N_v} \quad (19)$$

where  $\bar{V}_j$  and  $\sigma_j$  are related the  $j$ -th version, and  $\alpha_{Alice}$  is always the  $\alpha$  coefficient of Alice.

$\Psi_j$  is the Decay Factor (DF) that weighs as usually the importance of following versions (instead of  $\Phi_n$  that is used with the previous versions). This DF depend on the number of changes of the  $j$ -th version related to the total number of changes from Alice's version  $\lambda_n$ .

$$\Psi_j = \left(1 - \frac{\lambda_j}{\lambda_{j_{tot}}}\right)^E \quad (20)$$

$$\lambda_{j_{tot}} = \sum_{i=n}^j \lambda_i \quad (21)$$

where  $E$  is a parameter defined a posteriori.

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

**Total  $\Delta a$**  We can now calculate the total  $\Delta a$  for Alice:

$$\Delta a_{tot} = F \cdot \gamma \cdot \left( \sum_{n=1}^A \Delta\tilde{a}_n + \sum_{j=1}^B \Delta\bar{a}_j \right) \quad (22)$$

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 17.
- after every Alice's version  $\Delta\bar{a}_j$  is calculated with the Forward Decay process with formula 19 for every version till it is met the stable version or another Alice's version.

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

## 4.2 $\Delta b$ 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  $\alpha$  is fundamental because they are who votes the documents, hence it has to reflect the quality of their actions.

### Influential 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  $\varphi$  calculated as:

$$\varphi = \left( \frac{\delta - \Delta V}{\delta} \right)^G \quad \longrightarrow \quad \Delta b \propto \delta \quad (23)$$

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

$$\delta = \sqrt{\frac{1}{N} \sum_{i=1}^N (V_i - \bar{V})^2} \quad (24)$$

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

$$\Delta V = |V_i - \bar{V}| \quad (25)$$

There are different scenarios:

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

$$\Delta b \propto (1 - \alpha)^H \quad (26)$$

The exponent  $H$  could be setted a posteriori. If a user is already trusted his  $\Delta b$  changes slowly.

**Total  $\Delta b$**  Given a Chain of Changes between two stable versions and a specific user that has participated in the poll, we calculate  $\Delta b_n$  for each of his vote using the *partial* average of marks  $\bar{V}_n$  of every different version:

$$\Delta b_n = (1 - \alpha)^H \cdot \varphi \quad (27)$$

$$\Delta b_n = (1 - \alpha)^H \cdot \left( \frac{\delta - \Delta V}{\delta} \right)^G \quad (28)$$

Remember that every user can vote only one time a revision but can vote more than one in Chain of Changes. Then  $\Delta b_n$  are summed up to make *total*  $\Delta b$  of a single user:

$$\Delta b_{tot} = L \cdot \gamma \cdot \sum_{i=1}^N \Delta b_i \quad (29)$$

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

### 4.3 $\gamma$ calculation

User's activity coefficient  $\gamma$  is important because it influences the size of the variation of  $a$  and  $b$ , so it directly influences changes of  $\alpha$ . 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 calculated as:

$$\gamma = P \cdot \frac{\text{actions}}{\text{days}} \cdot Q^{-\text{inactive days}} \quad (30)$$

$\gamma$  is proportional to the number of actions by day and to a constant  $P$ . Then, for every day<sup>3</sup> of inactivity,  $\gamma$  is scaled by an exponential factor with base  $Q > 1$ .

---

<sup>3</sup>another fixed period of time could be chosen



## 5 Pros and Cons, Further development

### 5.1 Pros

**Automatic** The  $\sigma$  *Threshold* mechanism tries to automate the determination of the stability of the documents. We decide to link stability with the quality of the overall judgement in the sense described in section 3.2, pag. 7.

**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 13 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  $\sigma$ , make sense.

**Review ability** Users' ability to review other works measurement (See section 4.2, pag. 15) 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.