INTERNATIONAL
HELLENIC
UNIVERSITY

Human Rating System

Ioannis Papikas

SID: 3301140002

SCHOOL OF SCIENCE & TECHNOLOGY

A thesis submitted for the degree of

Master of Science (MSc) in Information and Communication Systems

DECEMBER 2016

THESSALONIKI – GREECE

INTERNATIONAL
HELLENIC
UNIVERSITY

Human Rating System

Ioannis Papikas

SID: 3301140002

Supervisor: Assist. Prof. Christos Tjortjis

Supervising Committee Members: Dr. Christos Berberidis
Assoc. Prof. Nikolaos Bassiliades

SCHOOL OF SCIENCE & TECHNOLOGY

A thesis submitted for the degree of

Master of Science (MSc) in Information and Communication Systems

DECEMBER 2016

THESSALONIKI – GREECE

Abstract

Feedback systems exist in the market for a long time. Many retail stores are using similar systems in order to bring more sales, promote their business and drive customers into becoming more active and interested in their products. At the same time, feedback systems have been also used to provide a personal opinion or a recommendation about a person and their skills.

This document is going to analyze the existing rating systems and more specifically LinkedIn endorsements feature, how it works and how it affects people and drives job applications and recruitments. In the analysis various flaws will be presented, that exist with the current rating systems and how the community behaves and reacts based on the existing flaws and the concept of a flawed system.

Finally, the result of the analyses is a new model for rating and ranking users for their skills in order to fix the existing flaws. Although the community might say that the idea is broken beyond repair, this is a game changer introducing the concept of PageRank in combination with the ratings. The new model ranks users for their skills based on the graph itself allowing propagation of knowledge and experience as well as setting limits and bounds to the amount of experience sharing.

Ioannis Papikas

December 23rd, 2016

Contents

1 Introduction

A long time ago, sales was a process during which a person, after having the need to buy a new product, she was going to a retail store asking to buy a product of their desire. This was a simple process including simple steps like the following:

- Investigate the buyer's needs
- Understand the budget available
- Propose 3 - 4 products
- Close the sale

and everything was done in a proper way. People were going to a store that they liked or trusted and they would buy what they wished for. The game started to change when the first malls were built, a strong example was the Selfridges[1], which opened in London for the first time. They promoted a different kind of process in sales where people had different opportunities and behavior on behalf of the store.

Long story short, sales processes evolve. Nowadays people are using the internet as the main medium to buy the products they need. This evolution started to happen using products that people can buy blindly, meaning that they match to anyone. Usually these kind of products are technology products. These days the evolution of online purchasing has changed a lot and people can buy almost anything.

The most critical point here, which is going to be analyzed further in this document, is trust. How do you trust a store or people about products when they describe a product to you? Do you trust them that they will be credible and tell the truth? When you are there, you can see for yourself, but is it the same as online? Can you trust the seller/store? Usually people trust big and known brands but there are also cases that big and well known does not align with credible. On the other hand, it is not always about the product or the seller's side. Going on the other side, can the seller trust the buyer? Will they pay in time? So, there are two sides of the same coin, where coin is the trust itself.

Ebay and Amazon were some of the pioneers to introduce a massive feedback system for products[2], sellers and buyers, people in general. This feedback allowed people to have an insight of the seller (and/or buyer) so that they can be assured for many things like the product's description accuracy, the seller's speed in dispatching the product and many more. This was one of the first feedback systems that became so popular and people are using it every day since, maybe without realizing how important it is for the company's community itself.

Further down this document is going to analyze this trust, break it down and analyze how people can be rated based on different characteristics and share this knowledge through a trusted "authority" and allow better and safer transactions of any kind.

1.1 Research disclaimer

This research is new in this specific business section, along with the methodology and the product that is going to be proposed in the end. The focus of this research is how community behaves, reacts and works using products and services, mainly working with trusting other people. Most of the references that are going to be presented here are going to be community articles, blogs and opinions. At the time of writing this document, there are no official researches for most of the features that are going to be analyzed. For those that can be supported with scientific articles, references are going to be presented for all the sources properly.

2 Literature Review

This chapter contains an extended literature review on the feedback and rating process and how it works using different scenarios and cases. Feedback and rating systems for products but also for people and their skills will be taken into consideration as well.

2.1 Background Research

Feedback systems exist in the market for a long time. Many retail stores are using similar systems in order to improve their processes, products and services, also bring more sales, promote their business and drive customers into becoming more active and interested in their products. Feedback systems in general are being used as a valuable metric for the companies themselves but also for other customers, to facilitate their choice in buying the products they need.

This sub-chapter is going to analyze how rating systems work and how other companies have used this approach/feature to build a new feedback system about people's skills. Analysis will contain descriptions on how ratings bring value and drive people into actions, either buying products, visiting a point of interest or even getting a job.

2.1.1 eBay feedback system

Before proceeding with the analysis of a feedback process and explaining why this is important, it is required to mention in a few words how the eBay feedback system works.

According to eBay, feedback is:

> *Each time you buy or sell something, you have an opportunity to leave Feedback about your experience. That Feedback is an essential part of what makes eBay a successful community.*[2]

The above sentence describes a community that eBay has successfully created that consists of sellers, buyers and a very good feedback system. In eBay's case, feedback can be positive, negative or just neutral and it can be combined with a short comment describing the experience of the purchase on the seller's or the buyer's side. It is a well known fact that in eBay's community, feedback is a very important system that allows sellers to build a reputation and buyers to be trustworthy when they bid on items they need (eBay is above all an auction community).

This information, the total amount of feedback, is accumulated and displayed on each seller's and buyer's profile. It is visible on each product whenever a person views a product.

2.1.2 Feedback and rating systems

The first step is to start examining the entire process of making a purchase in a retail store. In this case study eBay is going to be used as the retail store/platform. The steps that were followed and analyzed are the following:

1. looking for a product,
2. placing an order,
3. paying for that order,
4. receiving the order,
5. giving feedback and
6. going back to 1

Disclaimer about the process that is going to be analyzed: This part of the document is not interested in telling the story right from eBay processes and functionalities perspective, but telling a story about user experience, feedback and looping this information into valuable data for future uses.

Looking for a product

Every process starts with an interest, and in this case the user is looking for a product. The interest generates the demand for a purchase. As the end of the story will present, this demand will be the main reason for creating the rating and review systems.

Since this is the first round, there is no rating yet on the product so the buyer decides to buy a product from *AmazingITSeller2016*, a seller on eBay. As mentioned in a previous paragraph, the buyer can see the seller's rating so far from previous purchases and it looks that this seller does not have enough feedback yet, however she decides to buy the product because it appears ok and she really needs it.

Steps 2 - 4

These steps, placing the order, paying for the product and receiving the product, are quite trivial for almost everyone that has already bought a product online and every similar company, so the buyer will choose not to go into more details about the process of an order. All 3 steps are important because these are the true source of users' thinking when they leave feedback. Although the steps are trivial, they define the user experience and affect seller and buyer feedback. Especially in the case of eBay, after receiving an order you have the option to leave a feedback as a buyer (and the seller can do the same as well) regarding these 3 steps.

So, in this example it can be said hypothetically that the buyer receives the order with a delay, which is enough to change the experience from good to bad. In many cases a delay might not do us any harm because usually there is a large date span given by eBay (that probably comes from the seller) about a possible delivery date (for example between 3rd of November and 15th of November). In the end, this small delay was the reason why she missed her nephew's birthday and no present was delivered, at least not the one she ordered from eBay.

Giving feedback

This is the step where the buyer receives the order and it is time she provided some feedback regarding the experience towards the seller. In general, it is a very simple process where users can rate their experience on different subjects that eBay points out on every purchase.

Since the buyer received the product with a big delay and she missed an important date, this is going to affect in a negative way the feedback. In the case of missing an important date, it is considered a big problem and she would rate it as 1, in scale from 1 to 5. Besides the delayed delivery, it is safe to assume (for the purposes of this example) that the order was quite good and the rest of the required feedback ratings are at 4.

In this case, the buyer ended up providing a bad feedback based on a real experience for a real product on a real case scenario. So, this seller would have 1 (out of 5) for fast delivery.

Looking for a product - round 2

Next, there is another client looking for a product (back to step 1) which is also available from the same seller seller, *AmazingITSeller2016*. This time the second buyer can see that there is some feedback for this seller and she can see that although the product is good, there is a possibility that the order will arrive with a big delay.

At this point it is important to mention that usually communities like the one that is being referred to discourage negative feedback and suggest that any buyer should contact the seller before giving any negative feedback in order to solve the problem in a different way, because this kind of feedback is permanent and affects the seller's final feedback score. Of course there are cases where buyers do not have options or there was miscommunication during the purchase.

Coming back to the order, having described the first buyer that bought an item from the same seller, it is understandable the fact that the seller's feedback score is not good. This has as a result to skip this seller or choose another seller that has bigger feedback score.

Buyer feedback

The example that has just been described was a normal case that can happen any day. In the first case the buyer was disappointed from the seller and there was a low rating feedback (we make it simpler and present a low rating feedback instead of a negative feedback).

Now, it is time to see the case from another perspective, from the side of the seller. At this point it is important to mention that in eBay, when you select to buy a product, you commit to buy it and then you have to pay for it. This process is there because it also covers cases where there are bids and you win an auction. As also mentioned in the beginning of this example process, eBay is also an auction platform and the community is strong enough to provide feedback from both sides.

In the end of this case, and for the purposes of the example, the buyer committed to buy the product but changed their mind and did not pay after all. Although the seller (according to eBay) cannot leave a negative feedback, they can leave the lowest feedback in the payments section, with the description that the buyer never paid for the product.

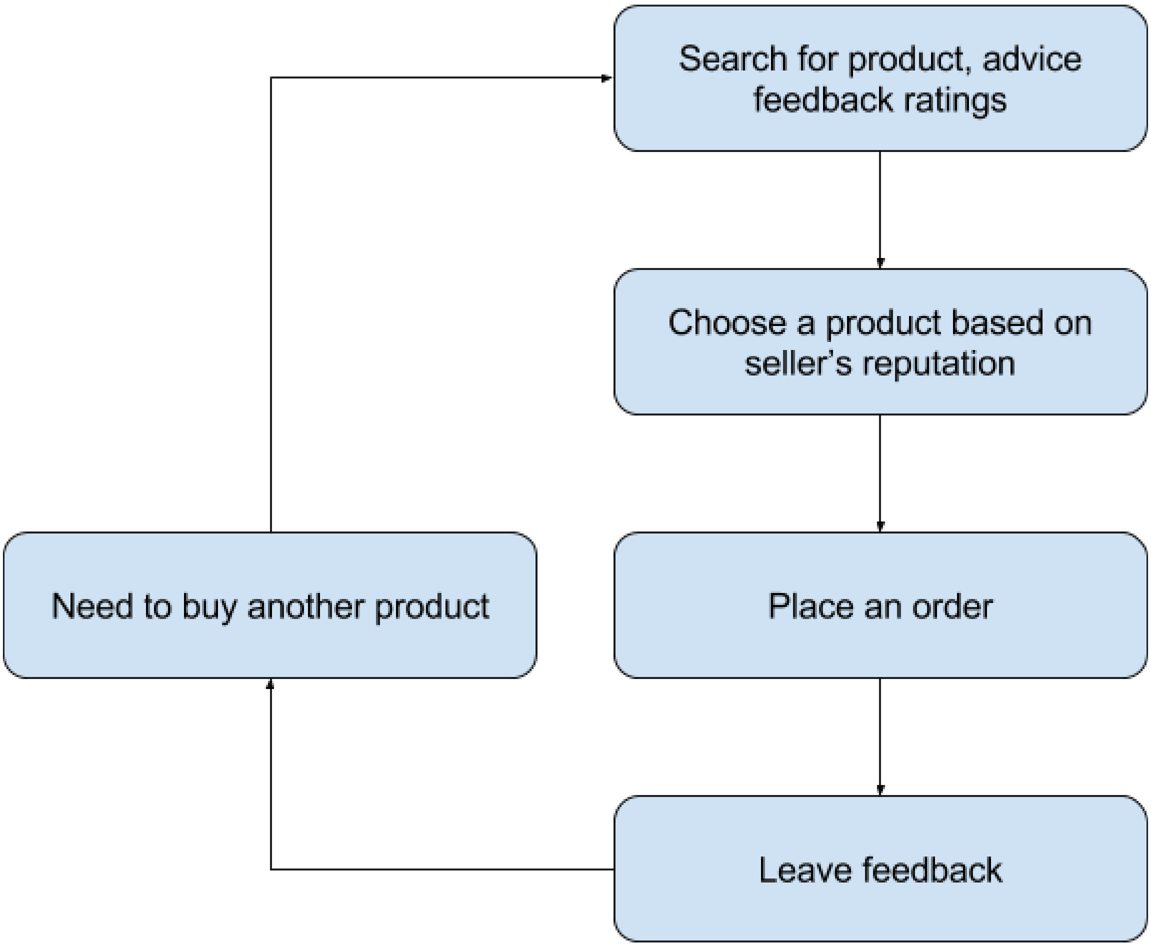
The above example presents in a very graphic way the relation between buyers and sellers and how it affects the community.

Conclusions

This story can be used so that one can see the power of feedback and rating systems. Although feedback can be anonymous or from a complete stranger (we can see that you do not have to know someone else in order to give a feedback, it is important to clarify this now as more details will be presented later on), it can affect people. It is a powerful tool that can affect the way people see companies and brands, generate either good or bad reputation. It is also a good factor to see that companies are using these kind of systems, because they believe that they have good services and satisfied customers.

On the other hand, people get feedback also. In the case of the seller's feedback, the buyer was bad during payment and actually there was no payment at all. Next time that the same person will try to buy a product from eBay, sellers will probably not take those offers seriously and they can deny selling the product or they prioritize their orders in a different way.

Feedback system is a powerful tool, but it does not aim at punishing people but making people more careful and follow normal and simple instructions. Even eBay, as mentioned before, tries to teach the community not to leave negative feedback and try to solve any problems using communication which can result in a positive feedback after all. Feedback can also build a *brand* and a *reputation* for both people and companies.


Search for product, advice feedback ratings
Choose a product based on seller's reputation
Place an order
Leave feedback
Need to buy another product

Feedback Process

2.1.3 Entire companies based on ratings

Since feedback systems became so powerful for people and companies, driving sales, visits and generic conversion to high levels, there are companies that were created to provide this kind of feedback for different services and gain from the value that they create. These companies have seen the value that end users can get from the power of the feedback from the crowed and they decided to take advantage of it and provide a model of providing free suggestions based on user feedback for different services or products.

Companies have been creating a business model based on ratings for places and products. A very simple and popular example is TripAdvisor[3]. TripAdvisor has created a product where people can review and rate hotels and restaurants. These ratings have created an entire service where people now can search for the best hotels and restaurants based on the reviews of other people. So, basically, they have created a business out of people's reviews for points of interest like hotels, restaurants, coffee shops etc.

In addition to TripAdvisor, there are different companies like Yelp and Foursquare which are in a similar business, providing points of interest based on user ratings.

Yelp started as a food restaurant metasearch engine that offers the best way to find great local businesses. Now they support all kinds of businesses including doctors and others. The categories so far are the following: *Food* | *Nightlife* | *Restaurants* | *Shopping* | *Active Life* | *Arts & Entertainment* | *Automotive* | *Beauty and Spas* | *Education* | *Event Planning & services* | *Health & Medical* | *Home Services* | *Local Services* | *Financial Services* | *Hotels & Travel* | *Local Flavor* | *Mass Media* | *Pets* | *Professional Services* | *Public Services* | *Real Estate* etc.

Foursquare can be described as a website that offers the best places to eat, drink, shop or visit and they are supported by local experts. Another company which offers information based on ratings and reviews from other users and local experts.

Companies that work with User Feedback

Again, one can understand how powerful is the combination of feedback/rating/review systems and communities. They provide the value of public opinion through a model that allows people to have insights and take decisions.

2.2 Rating people for their skills

In addition to the above, there is a big research regarding feedback and people's ratings on a different business sector, the recruitment services. There are HR departments in most of the companies that handle the inner communication and they include recruiting as part of their processes. Also, for companies that are not big enough to support such department, or they just do not want to have a similar department, there are companies dedicated to offer recruiting services with small fees.

Their model can be described in very simple words, however it is the process itself that is complicated because it includes people and factors that are not clear and there are many reasons that can affect a decision. For a long time, recruiting companies have been creating profiles for people who are looking for a job so that they can match them with available positions in companies. Recruiters are following this process of creating profiles for people they are interested in, or when they apply for a position. In a generic description, recruiters add skills to candidates based on their experience and they evaluate them. It is a detailed profile which can be used to perfectly identify a candidate, from the recruiter's perspective.

The goal of this document and the model suggested is to facilitate both freelancers and employers. It facilitates freelancers so that they can find the best collaborators they can from their network or area and it facilitates employers so that they can find the best employees they can hire.

2.2.1 LinkedIn, Recommendations and Endorsements

We mentioned in the previous chapter that a recruiting process is very complicated but it has sub-processes that are simple enough. LinkedIn, as a business network, has clearly taken their position regarding this process. Being a business network, they promote the creation of a more professional profile that users can have in order to either apply for a job or look for collaborators and partners.

Initially, LinkedIn added a feature where users can create projects that they are working on with other LinkedIn users. In addition to projects and/or companies, users can provide recommendations. A recommendation is a short subjective text that a user can write on another user's profile page about their collaboration and provide some valuable feedback about the user. The recommendation information itself is more valuable than any scalar metric system. It is pure information to the point. The disadvantage is that recruiters have to gather all this information, read it and create the profiles themselves.

In 2012, LinkedIn introduced endorsements to the world[4]. Although one cannot say with confidence, LinkedIn created this feature so that they can use metrics and enable better search of the users. According to LinkedIn, endorsements are a quick step of adding a rating to a user without writing a descriptive text explaining what and how. The recommendations still exist but an endorsement is a quick *tag* to users representing a skill that they have. It's LinkedIn's way to tag a user with an one-click action. In addition to recommendations, endorsements are a way for recruiters (or just users with a premium account) to easily find candidates for their companies. Although most of the users are using endorsements and invite their connections to endorse them for their skills, however community opinions are negative about endorsement because it does not offer a true value under many circumstances, which is going to be analyzed further in the next chapters.

2.2.2 Flaws with endorsements

Endorsements exist since 2012 and it's a very successful feature for all of LinkedIn users. Recruiters are also using this feature to filter other users and approach candidates using the premium account.

Although endorsements are very successful, they are not accurate enough. This assumption is supported by the fact that there is no control at all on who endorses who. A deeper analysis, based on articles and opinions from the community, can generate some facts that make endorsements not so accurate and friendly, but mostly a feature to sell. Different opinions from different users have been analysed, in blogs and special websites, that can teach someone how to get more endorsements, which means that LinkedIn endorsements have become like a game where the one that has more wins, although counting stops at 99 and then it just displays 99+.

After the analysis, there are some flaws that were visible in the current implementation of LinkedIn Endorsements.

Endorsements are open

Endorsements are open. This sentence describes the ability/freedom of LinkedIn users to provide endorsements to everyone that they are connected to. When user A endorses user B for skill X, user B can select to add the skill to their profile or not, if the skill does not exist yet. This can be a nice feature to help people build their skills but it also generates a lot of noise and false data.

The generic rule for LinkedIn is that a user can endorse any user in their network for any skill. Even when the receiver does not have a skill, this can be a nice feature for suggesting skills from collaborators and colleagues.

Endorsements are not controlled

No control means that there is no condition to prevent a user from giving an endorsement, regardless of the personal skills. For example, the current LinkedIn structure allows user A to endorse user B for skill X although user A does not have the skill X. This no-control of the endorsements means that they lose credibility regarding where they come from and what it is their true value.

We can explain this flaw using a simple example. Given a LinkedIn user, she has a skill called “PHP”. In the user’s connections there other users that are from their business circle but not everyone knows what PHP is. So, it is possible to get an endorsement from a colleague that works in HR or Marketing. In this example, the real question is how this user knows whether one is good with PHP or not.

Endorsements are not weighted

One of the most important flaws of the current architecture, which will be examined thoroughly in this document, is that endorsements are not weighted. Reading LinkedIn’s announcement for the new feature, it is visible the fact that it is not about accuracy but about tags and keywords for the users so that they can match candidates with positions faster and easier. Endorsements' weights (or lack of it) mean that every endorsement has the same value from any user. Although one can see the list of endorsers, there is no real value because it fallbacks to the same position it was before endorsements, where one had to read all the reviews, but now one has to go through the list of endorsers. Finally there are some discussions and thoughts from users about not having weights[5] and the final conclusion is that LinkedIn Recommendations are the ones that really matter.

We can analyse the weight problem using the example from the previous flaw. Assuming that one has a colleague from the Marketing department who endorsed her for the "PHP" skill. In this case one can also add another endorsement from a senior developer of the company who has the PHP skill and he is good at it (she personally knows it). Without weights, these two endorsements have exactly the same value for LinkedIn, for both endorsements the skill value will increase by 1. This equality does not really make sense as one could have the same result just by having the same endorsements from people that have no idea about PHP.

Conclusions

Using the three previous facts, a good conclusion is that there is doubt about the accuracy of LinkedIn endorsements. Although it is being used widely to facilitate tagging and skill listing, it is not an accurate way of skill measurement. Recruiters are using endorsements for guidance. Recruiters cannot use these systems to create a fully accurate system to measure a person's skills. It will inevitably include false positive data created by *noise* in the system, users that do not really have the skills listed.

In the current approach, using weighted graphs, noise can be eliminated using a smart skill calculation algorithm that takes into consideration more factors and transfers the value of the endorser to the endorsee.

2.2.3 Endorsement psychology

Extended analysis and research in articles and special blogs has shown that people are more concerned about endorsements and they can be manipulated or follow instructions that allow them to either get more endorsements or misuse them.

Research has shown that people can be categorized into specific models according to their behavior. These models describe how people react and behave before and after an endorsement and one can see that the community can be get more flaws and misconceptions.

Adding and getting endorsements can become a *game* or a *habit*. People can be categorized based on their behavior in order to get more endorsements[6][7]. People have been analysing how the community behaves regarding LinkedIn endorsements and they reached into the conclusion that people can be categorized in some *strange* categories[7]:

- **The Beggar**: “Please endorse me. Pretty please? If not, I’ll just send you another message next week with a pretty, PRETTY please.”
- **The Stranger**: “You don’t really know me that well and we haven’t actually worked together, but please endorse all of my wonderful skills. It’s only a little white lie, really…”
- **The Guilt-Tripper**: “I just endorsed you, so please endorse me back.” (Note: May turn into The Beggar or The Threatener, see below.)
- **The Threatener**: “If you don’t endorse me, I’m gonna remove my Endorsements of you.”

The above models can easily present how people have been using LinkedIn to get endorsements either for having a better profile and increasing the chances for an interview or just to have increased skill values.

Since endorsements are a quick and easy way to list your skills and find a desired position at a company, users are focusing on getting more and more endorsements. Searching the internet you can also find a lot of articles[8][9][10] that describe how you can increase your endorsements, to help you get more position proposals and get more profile views. This is another proof that endorsements can be built based on not valid data.

3 Methodology Analysis

This chapter is an extended analysis on the methodology in order to build a smarter skill rating system for people based on the previous assumptions and the flaws of the existing system(s).

3.1 Rating people's skills

In the previous chapters there was an extended analysis on how people can rate other people for their skills, or in other words, endorse[4] a person for a skill. Also, some generic observations on people's behavior around these endorsements have been described in detail.

Building a rating system for people require a different structure from other systems. Since people are connected to each other, the suggested model is going to look like a social graph with users connected to each other on a skill level. A graph has vertices and edges, in this case it is a directed graph which means that edges are directed. To apply a graph to the example, like social graphs, users will be vertices (when a user has a skill, it becomes a vertex on the skill graph) and a skill rating as a directed edge (arrow) between two vertices. When a person is rating another person for a skill, an edge is added on the skill graph.

So far the existing rating system(s) presents flaws that can vary from excessive usage of ratings, to no control and meaningless tags on people. This chapter is going to introduce a new approach where endorsements do not have all the same value and impact and where each endorsement can change the values of the skills either in a positive or sometimes in a negative way.

3.2 Social Impact

Recruiters from all over the world are using LinkedIn as a tool to find candidates for job positions. Their research include many aspects of a person's profile, including experience, endorsements and recommendations. As LinkedIn mentions in their announcement about endorsements[4], they have created a tool where users can add keywords or tags to other users to enable faster search.

Endorsements work in both directions. Recruiters are filtering candidates based on their endorsements and users are trying to get more to increase their profile strength.

Community portals and blogs mention that nowadays recruiters do not pay much attention to endorsements as they are *meaningless*, *fake* and *driven by popularity* and not the actual value of the skill. Recruiters are paying more attention to recommendations and less in endorsements in order to create an initial profile about a candidate.

The rating system that is being introduced here can help recruiters save time and use the ratings that this system is going to generate in a more accurate way. Some of the potential features are the following:

1. Access to the rating graph to see where the ratings are coming from
2. Faster filtering using the skills and the location of the profile
3. Use different input sources for adding skills to the graph, not only social networks

Creating a service that can implement the graph described, recruiters are going to spend less time finding people with the skills needed and candidates will build their skills in a more accurate way based on other people's opinion, which is going to matter a lot (or not, it depends on the connection's skill value).

As a conclusion, the goal of this research is double. It aims at creating a better and more accurate model for rating people's skills but also aims at providing recruiters and candidates better tools that will enable a faster and more accurate matching between candidates and job descriptions.

3.3 Building the graph

LinkedIn endorsements are one way of saying that a registered user has a given skill. LinkedIn is using its social graph to connect users and propose connections, however it is not known if it uses the social graph to export data and insights from endorsements, since this is an internal process and there is no visible proof of taking advantage of the graph. One can see that, using this feature, anyone can endorse any connection and the only result will be the increase of the connection skill by 1.

We are introducing a new way of building a skill-rating graph between users. In this graph, the nodes will be the users and the arrows will be the ratings between users. This chapter is going to analyze the big structure of a single graph and the combination of different skill graphs, how it is going to be weighted, how initial weights are going to be set and the effects of every new connection added to the graph (both vertices and edges).

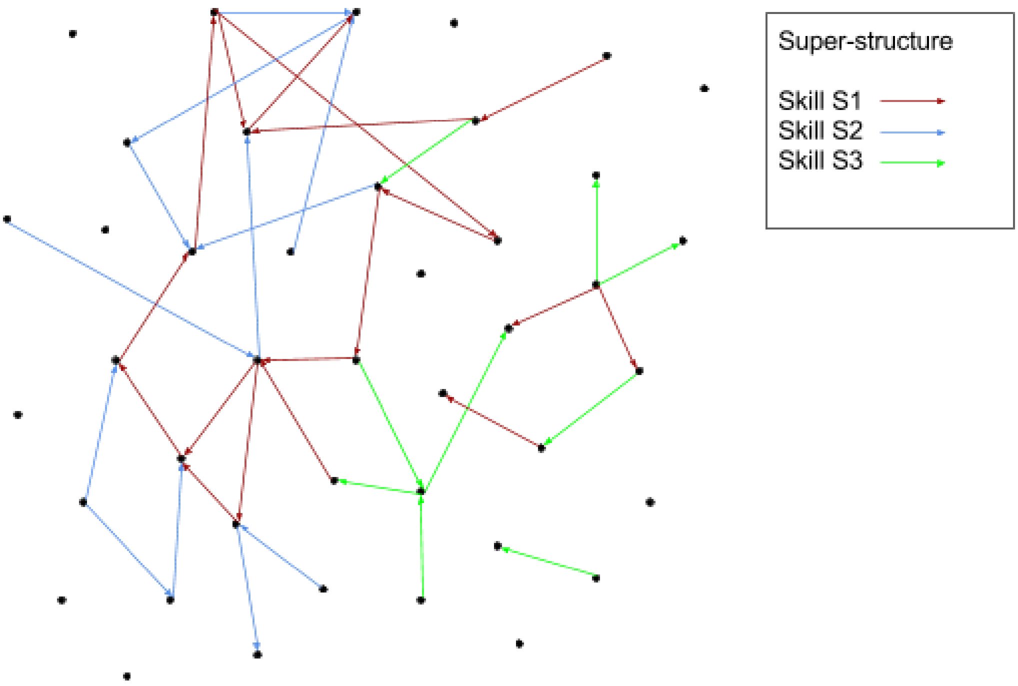
3.3.1 Graph super-structure

To understand better how the approach is going to work using graphs, it would be good to explain first how it is going to face the entire problem, allowing users to have skills and rate other users. The full size of the graph will be mentioned as super-structure not because it is a very big and complicated structure but because it consists of multiple levels, from a single *social* graph one can generate multiple skill graphs using the same nodes/users.

In the world of connected and weighted endorsements, users will be able to create a profile and add personal skills. It is normal and acceptable to assume that users will have common skills. Next, users will be able to create their *neighborhood*, including other users that they know, either from the business world or from the social world. Social networks are going to be used to allow users to register to the application and use the network, LinkedIn, Facebook or Google+, in order to create the neighborhood. This first approach will be to allow users to rate only users in their neighborhood. A quick assumption now is that it will not matter if users are connected with each other, the graph algorithm will work properly and it will not damage the node values.

The mathematical analysis is simple and there are already some definitions. Let’s assume that user A has N skills and M connections, and S1 is one of N skills of user A. User A will be able to endorse this skill to the subset of its M connections that have S1, which can be any amount [0, M]. This process will start creating a graph about skill S1. For each skill S out of the N skills of user A (and out of any skill submitted in the application) a different graph is going to be created, a different skill-graph. Nodes of the graph are going to be the users and the connections between these nodes are going to be the endorsements.

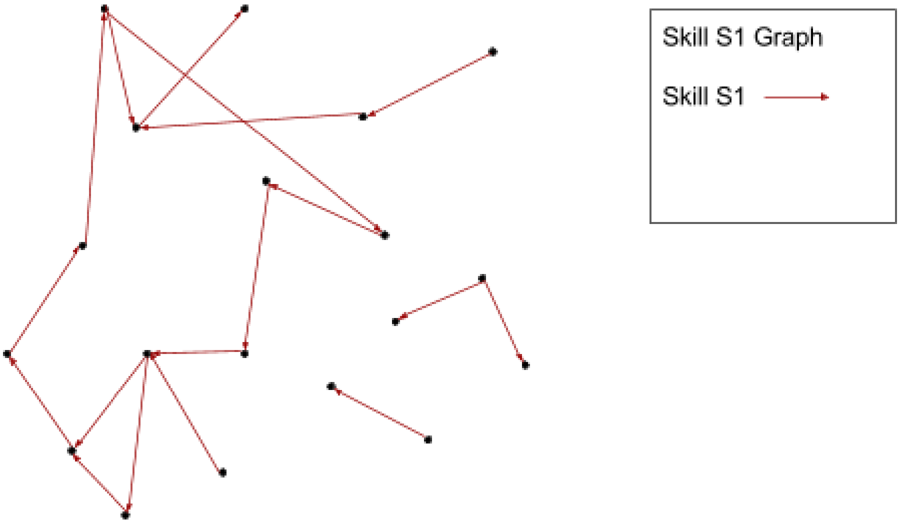
The result will be, as mentioned above, a multi-level, multi-dimensional graph that illustrates how users are connected with each other in each skill graph. The result could look like this:



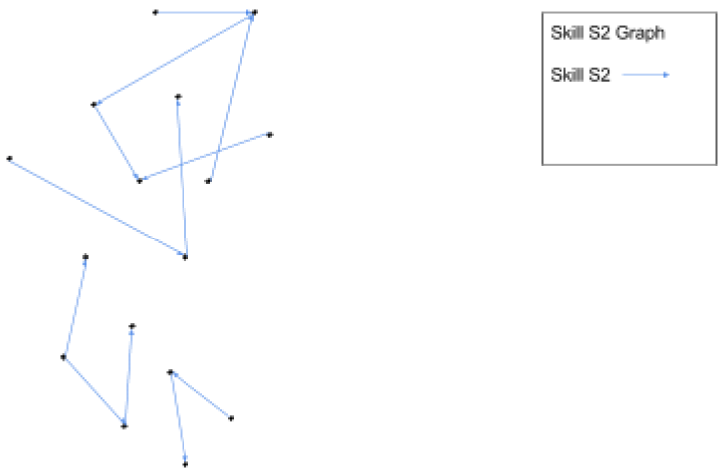
Full skill graph

The above graph is a super-structure and includes all the registered users and their connections in different skills. Observing the example graph, one can see that there are users that participate in no graphs, there are users that participate only in one color graph and finally users that participate in more than 1 graphs.

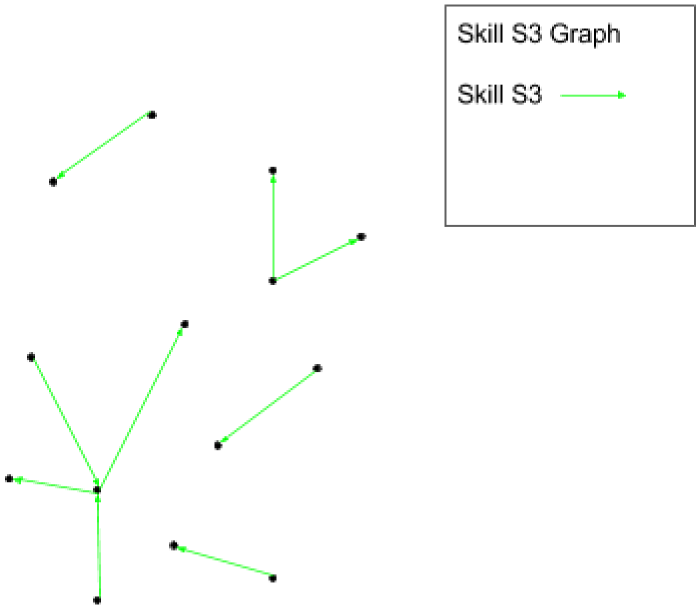
The above big graph can break in 3 different skill graphs:



Skill S1 Graph



Skill S2 Graph



Skill S3 Graph

In the initial graph one can discover 4 types of nodes:

1. Nodes that don't connect with other nodes
2. Nodes that have only outgoing connections, the indegree is 0
3. Nodes that have only incoming connections, the outdegree is 0
4. Nodes that have both outgoing and incoming connections

As also mentioned above, different nodes can participate in different graphs according to skills that they have.

3.3.2 Graph weights and values

The goal of this research is to provide a better way of rating people for their skills, to be more accurate and avoid overuse. The goal is to allow users to define skills which in the end are going to have some values that will show how good the user is at the given skill. The level of knowledge of a skill of a user is going to be called *node value*.

Graph weights depend on two factors:

1. node value
2. outdegree of the sender

In order to build a smart algorithm that allows the distribution of the node values to their outgoing connections but also punish node with many outgoing connections, the algorithm is going to be a variation of the PageRank[11]. PageRank is an algorithm used by Google search engine to rank websites in their results. PageRank was named after Larry Page, one of the founders of Google. PageRank is a way of measuring the importance of website pages. According to Google:

PageRank works by counting the number and quality of links to a page to determine a rough estimate of how important the website is. The underlying assumption is that more important websites are likely to receive more links from other websites[12].

Like PageRank, the current implementation estimates the value of a person's skill based on the incoming endorsements and their values. As mentioned above, the weight of each connection depends on the node value and the number of the outgoing connections. For user X, the weights of all the outgoing connections is going to be the same and equal to:

$$GW(x,\ s)\ =\ \frac{1}{O(x,\, s)}$$

Where:

1. GW(x, s) is the weight of each outgoing connection for user x for skill s, based on the graph connections

2. O(x, s) is the number of outoging connections of user x on the graph of skill s

The above equation shows how the weights decrease as the number of outgoing connections increase. The logic behind this model is to distribute the user's value to their connections equally. This allows a user to distribute their knowledge and prevent misuse and excessive use of ratings. A simple example is when a very experienced user on a specific skill rates other users. Whenever a new outgoing rating is added to the graph, the skill's value is divided again and the weights are being updated (are getting smaller in this case). This is a simple example that indicates how a skill value can decrease, as a user starts giving more ratings on a specific skill.

In addition to the weight that is coming from the graph structure, there is a rating weight variable to allow users to rate using a scale instead of simple boolean rate with 1 and 0. Based on the weights of the graph, one can define that at any moment, the value of node x for skill s will be:

$$V(x,\ s)\ = D(x,\ s) + \sum_{i \in I(x,s)} (V(i,\ s) * RW * GW(i,\ s) * d)$$

Symbols explanation:

Symbol	Description
V(x, s)	the value of node x for skill s
D(x, s)	the initial non-zero value of node x for skill s
I(x, s)	the number of incoming connections of node s for skill s
RW	the user's rating weight <= 1
GW(x, s)	the graph's rating weight of node x for skill s
d	a constant, where d < 1, which guarantees the termination of the algorithm due to circles between two nodes also allows the linear system of equations to be solvable and have a solution

The above equation is for any node x. Given a graph with N nodes, this approach is going to create a linear system of N equations with N unknown variables.

We are using constant d as a very important factor of the equation as it allows the termination of the algorithm. Having $d < 1$ allows the propagation of the values to end after a finite number of iterations.

3.3.3 Challenges

Although it might seem easy for small systems, the size if the linear system increases as the graph size increases. In the current situation, since the main use case is to store social graphs and user connections, the graph can possible have (in scale) millions or billions of nodes with even more connections.

Taking into consideration the fact that not all users will have the same skill, skill graphs are going to be smaller than the full social graph, a subset of the full graph. Also, graphs can have nodes that are not all connected to each other. A quick reference to the example graphs that were presented in chapter 3.3.1 can present that in a skill graph there might be sub-graphs. The skill value equation can only be applied to nodes that have incoming and outgoing connections. The linear system can be applied separately to each sub-graph as nodes that are not connected in any way do not affect each other in any way.

The above approach can increase performance and reduce calculation times, considering that skill graphs are a subset of the full social graph and there are sub-graphs in each skill graph. This is very important at scale, when there are many skill graphs, many users and there are sub-graphs where there is need to calculate and propagate the ratings.

3.3.4 Initial weights

Building the skill graph is part two of the whole process. Part one is about initialization of the skill graph. The skill value equation includes a factor D(x, s) which is the initial skill value of the node.

D(x, s) is a non-zero initial value. It needs to be a non-zero value because outgoing nodes depend on a fracture of the current node value. This value will affect the growth scale of the skill values in the long run, it will act as a skill value size factor. It will also affect precision and distances between users and their skills. It’s a good option to keep it in low levels, usually <= 1.

Choosing a suitable initial value is not a critical decision. This factor is part of the skill-value equation. There is a normalization process scheduled to run to re-calculate all the skill values using the new initial weight. For the context of this application, the initial value is going to be set to 1.

3.3.5 Final node values

Node values depend on the initial value and the number of connections. The initial value acts as a scale factor and it will determine the final node values. The algorithm can normalize the results by simply changing the initial value D(x,s), either increase or decrease the values. This normalization process can take part in the future if there is an exponential rate of increasing node values.

Presenting the node values as they are, plain numbers, will be difficult to understand the meaning since there will be no measurement. If for example the application presents for a user a skill value of 1253, or a skill value of 0.632. Having the node value without any context information will be difficult to understand how high or low a user is in the ranking. In the current example the numbers do not mean anything in particular because the ranking is not known in its full length. For example the value of 1253 can mean that you know nothing about that skill if the higher ranking is 432002, or the value of 0.632 could mean that you are an expert if the highest value could be 0.7.

Due to the above challenges, a first approach will be to present the node/skill value in relation to the biggest node value in the graph (the entire skill graph, not only the connected nodes). This approach will help us present normalized values with an upper limit of 100, where 100 is the most expert user of this skill.

Another challenge is the full range of the node values. If only the upper value is taken into consideration, then it is possible that all the values can be above a limit, based on the lowest value. Having this in mind, there is a second percentage which will represent the ratio between the lowest and the highest value. For example, one can say that user A knows skill X as follows (highest value: 10000, lowest value: 4000, user value: 7000):

- V_ABS(A, X) = 7000/10000 = 70%

- V_REL(A, X) = (7000-4000)/(10000-4000) = 50%

V_ABS is the absolute value of the skill. It represents the ranking of the user in relation to the highest value.

V_REL is the relative value of the skill in relation to the highest and lowest value. It represents the ranking of the user in relation to the full range of the community that has this skill.

Outliers

As part of the calculation process, there are chances that skill values will have spikes from users that receive ratings more than normal. In the current implementation, which is a proof of concept, these outliers are not going to be removed from the population. This can be considered as *protection* for the future if irregular ratings are found and users have rankings a lot bigger than the average population value.

Selecting population

An extra step that can be added in the future implementations is to limit the population while getting the final node values. In order to generate more attractive results, there is an approach in which node values are retrieved based on the geographical location of a user. A user would be able to select to see their skill values based on their hometown or based on the current location.

This is more like a feature for recruiters who can focus their attention into limiting the area of selection. This feature can ask questions like:

- Which user has the biggest skill value for skill A in Europe?
- Which user has the biggest skill value for skill B in Greece?

Ratings visibility

We choose to hide all the incoming ratings (the user, not the number of ratings) for all the users to prevent users from requesting or revoking a rating because the rating was not returned or for similar reasons. Hiding the incoming ratings will disable any negative effect that those ratings will have on the users.

Being able to show the number of ratings can simple help users analyze the graph and take decisions. For example, two users can have the same skill value but one user has 50 incoming connections and the other has only 30. This means that the second user has more “expert” connections but the first user has a bigger community. This is a decision left for the recruiter to decide which is more important whereas the model's goal is to facilitate the entire process.

3.3.6 Converge Algorithm and Tolerance

Another approach of calculating the node values is through a convergent algorithm. Although the linear system seems simpler, implementing it in a large scale can be very resource consuming.

The converge algorithm will be applied every time a new connection is added to the graph. A simple pseudo code is the following:

```
GC = clone G(skill_s);
insert_connection(user_a, user_b, skill_s);
nodes_to_check_stack = [];
nodes_to_check_stack.push(O(user_a));
while (nodes_to_check_stack.length > 0) {
     foreach (n in nodes_to_check_stack) {
          Vn = calculate_node_value(n);
          GC(n) = Vn;
          if (abs(Vn - GC(n)) > tolerance) {
               nodes_to_check_stack.push(O(n));
          }
     }
}
G(skill_s) = GC;
```

Symbols explained:

Symbol	Description
G(s)	the graph for skill s
insert_connection(a, b, s)	inserts a connection between nodes a and b in the graph for skill s
O(x)	the ougoing neighbourhood of node x
calculate_node_value(a)	calculates the node value for node based on the skill value equation and the entire graph state
tolerance	a percentage under which the algorithm doesn't propagate the changes made

We select tolerance to be the amount of change in percentage that should not propagate to neighbor nodes. For the initial implementation of this algorithm, the initial tolerance will be set to 15%. This means that a change in a node value under 15% will not propagate to its neighbors.

Choosing the right time to trigger the algorithm can be an important factor for performance and the application itself. Some conditions can identified so that a graph has to meet to trigger the algorithm:

1. More than X connections have been added to the graph
2. The number of outgoing connections for a node has increased more than X%. X can be somewhere around 10 or 15.
3. Using a fixed time interval, based on the size of the graph and the calculation time

According to the skill calculation equation, it is logical to conclude that the biggest the graph, the less impact a new connection will have. This means that the graph will be stable after some iterations. Although new nodes and connections will be added in the future, the impact is going to be less as the node values increase.

For this proof of concept application, the third option is going to be used for the algorithm trigger. The implementation will run only if there are new ratings in the last fixed timebox. In this case a timebox of 5 minutes is going to be used, meaning that every user will have an average of 2.5 minutes of waiting after sending and receiving ratings for them to see the rankings.

The complexity of the above algorithm is a hard element to investigate on. The number of iterations is strictly dependent on the number of ratings, the ratings' values and the propagation threshold. Provided the value $d < 1$, the algorithm will stop after a finite number of iterations because on each propagation the value will decrease by a d%. On the other hand, if the equation is applied as a linear system, the complexity depends solely on the implementation of the tool that will solve the equation system..

3.3.7 Node values paradox

In 3.3.2 the node value calculation equation was introduced. This equation takes into consideration the outgoing neighborhood of each node and adjusts the weights based on the number of connections. For each new connection, this weight reduces. To visualize this weight, here is a table of weights according to the number of outgoing connections (we will take into consideration also d as d=0.85):

# of connetions	weight
1	0.85
2	0.425
10	0.085
50	0.017
100	0.0085

In the above table it is visible that the weight decreases as more and more outgoing ratings are being added. This weight is not visible to any user and it is more of an internal value that the model calculates on the fly based on the outdegree.

The paradox

A paradox that has been appeared here is that node values can decrease after adding a new connection to the graph. In more detail, inserting a connection from user A to user B, there are the following effects:

1. we expect user B value to *increase* because a new incoming connection is added
2. we expect all the node values of the outgoing connections of user A to *decrease* because the weight just decreased

Node values depend on the number of connections they have, both incoming and outgoing. Thus, one cannot predict with accuracy any node value after a new connection is added to the graph without any context information as it depends on the graph structure and the connections.

3.3.8 Solving the existing flaws

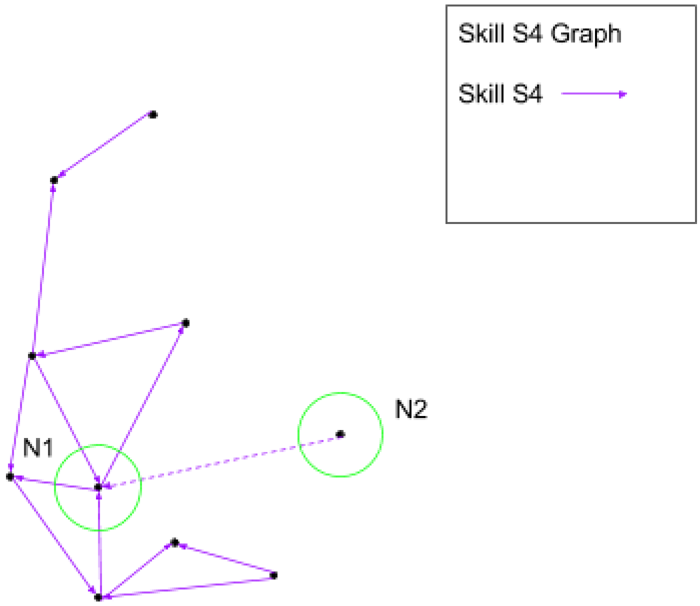
The approach that has been described in the previous chapters solves the flaws that were listed in chapter 2.2.2. In more details, this chapter analyzes how the new solution solves each one of the flaws.

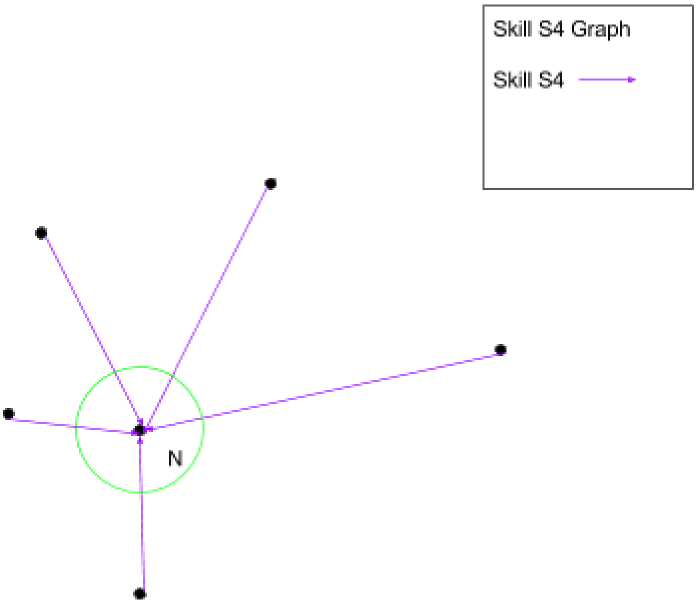
Endorsements are open

In this new solution the application can select the endorsements to be limited only to the users that actually have the skill they want to be rated for. However, the fact that a node's weights are decreasing each time that a new outgoing connection is added can allow us to make this feature more open than it was before. The application allows users to rate everyone, with no limitation at all. Rating any user in the system in this case will not allow values to increase uncontrollably because of the weight calculation equation.

Endorsements are not controlled

In this new approach, in order to create an outgoing connection to the graph, you have to be member of the graph, meaning that you have to have the skill required. Although this is a prerequisite to add a rating to another user, anyone can add the skill required to their profile and then make the rating. Example:





star topology

As one can see in the above example, new single connections can have a big impact in the graph. In an example where any node like N1 has many nodes like N2 as incoming connections, the impact will be significant and N1 value will increase and get the value of the indegree. This is an extreme situation where there is not an actual graph but it is a special structure. It is expected to have this kind of *anomalies* in any graph, that can be caused by special formations. This kind of formations and their impact to the graph are examined further in chapter [4.3].

Endorsements are not weighted

In chapter 3.3.2 a graph weight equation for all the nodes that participate in any graph was introduced. This weight solves the balance flaw of the graph and the impact that any connection will have on a node.

Having a weight structure allows us to differentiate and separate the rating that is coming from the Marketing colleague and the one that is coming from the experienced developer. Using weights it will be easier to say that since the Marketing colleague does not know anything about that skill, the impact will be small whilst the experienced developer will transfer more value.

According to the node value paradox, it is worth mentioning that it can be a case where the Marketing colleague will transfer more value than the experienced user. Here is a possible scenario: the Marketing colleague has rated only my user for that skill whilst the developer has rated their entire network. As it is already mentioned in chapter 3.3.7, the weight of the rating can become really small as the outdegree increases.

4 Implementation and Field Study

This chapter is are going to briefly analyze the structure of the web application that is built as a proof of concept for the skill rating algorithm.

4.1 Web application

For a proof of concept application, to prove the accuracy of the model and the social impact, a full web application will be implemented. Web applications nowadays are easy to build, deploy, maintain and it is very easy to share it with the entire world.

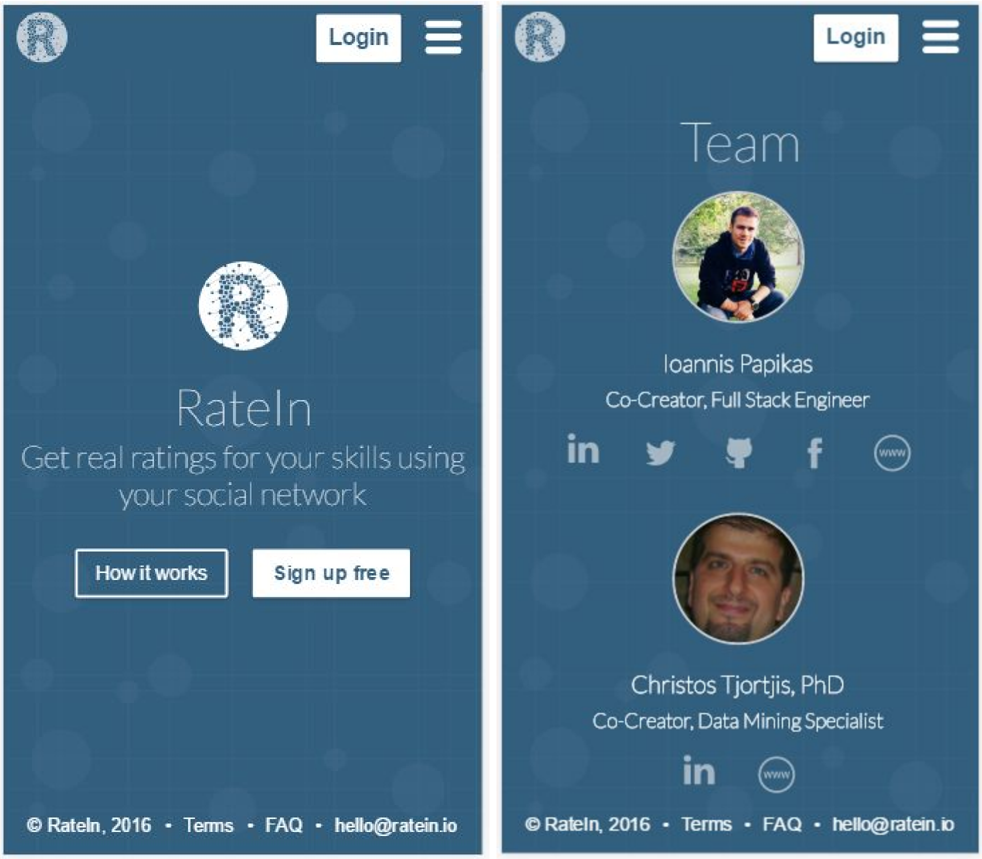
The application's schema has two basic and unique ingredients:

1. Users
2. Skills

The application is written in *php7*[13] to power the servers and the logic of the application. For the social graph the application is going to use a simple RDBMS database, MySQL. As the application grows, there are scaling plans to move into a social graph database, a solution like *neo4j*[14], which will facilitate the entire graph communication and propagation, offering a better interface for updating the graph and finding all the connections.

The main part of the application is the server part which also includes the api, it has the entire logic of the application. The server is responsible for storing all the skills, the users, the users' skills and all the ratings. It exposes different apis to allow the communication between the frontend application and future applications and external sources for gathering external ratings. For user authentication, an external service is going to be used as a scheme, loginBox[14] (https://loginbox.io), which offers an easy way to build all the authentication forms and mechanisms.

The main application, for this proof of concept implementation, will be a mobile-friendly website application that will provide all the necessary mechanisms for users to login, create a profile, add their skills and start rating other users. The application will include an easy to use interface that will allow users to see their skills, their ratings and their rankings based on the population.



Get real ratings for your skills using your social network

© RateIn, 2016 • Terms • FAQ • hello@ratein.io

© RateIn, 2016 • Terms • FAQ • hello@ratein.io

Next sub-chapter is are going to analyse some simple processes that will be part of the application, like the on-boarding process, the search engine and the rating mechanism.

4.1.1 On-boarding process

On-boarding process is the process during which users are registered to an application/service and they follow some steps to create their profile or other settings that the application might need.

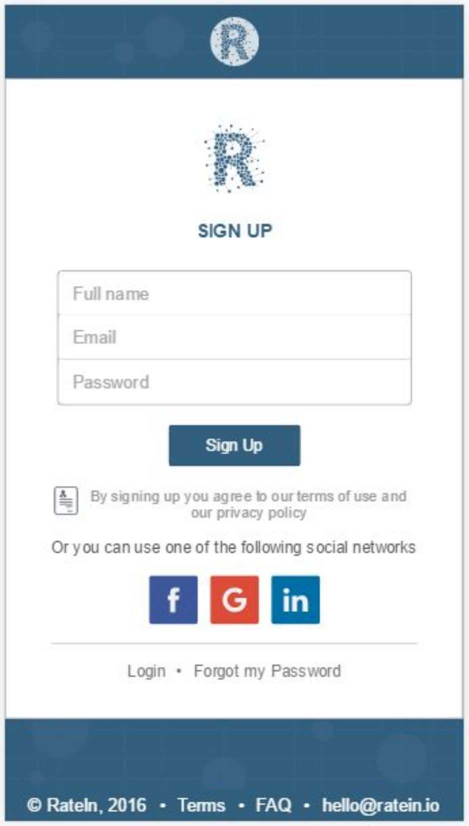
The on-boarding process is very important and essential for the user to create a unique experience that will drive them into using the application the best way possible. For this reason, is is needed to build an on-boarding process for the application that will offer the following to the users:

- Unique experience
- Easy to use interface
- Easy setup of the profile, including profile information

- Easy skill management
- Easy connection with other users

Using loginBox as the authentication scheme, any social login is very easy to be added. The application is going to use 3 basic social networks to all of the clients: Facebook, LinkedIn and Google Plus. The on-boarding process can be separated in the following detailed steps:

1. **Register** to the application using email or one of the available social networks
    a. The application will use the email, the name and the job title (in case of LinkedIn) to setup the user's profile



SIGN UP

Full name

Email

Password

Sign Up

By signing up you agree to our terms of use and our privacy policy

Or you can use one of the following social networks

Login • Forgot my Password

© RateIn, 2016 • Terms • FAQ • hello@ratein.io

2. **Add skills** to your profile. The application shows a quick dialog with some of the most popular skills for the user to add.
    a. If the user cannot find their skills in the list, they can simply add skills manually using the main add-skill form
    b. LinkedIn skill addition is not possible because LinkedIn has made this API private

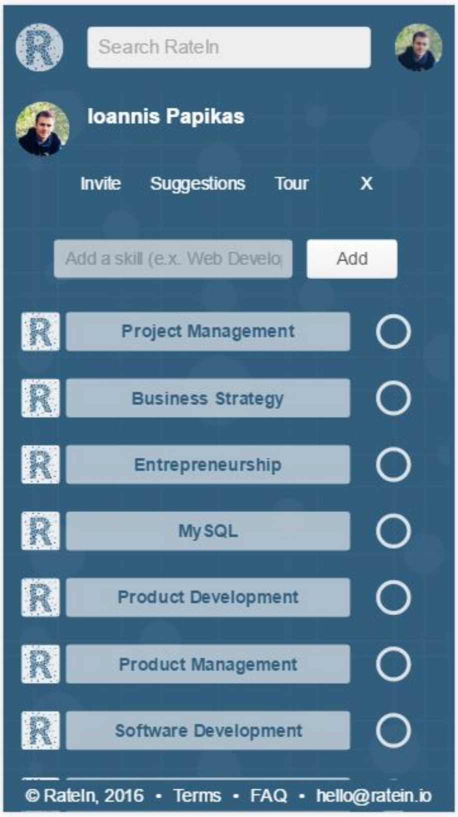
3. After adding skills, the application will guide you through a **list of suggestions** including users from the user's social network (if one is used) or users with common skills to rate
4. Final step, the application will prompt the user to **invite other people** to join the application using emails or one of the available social networks

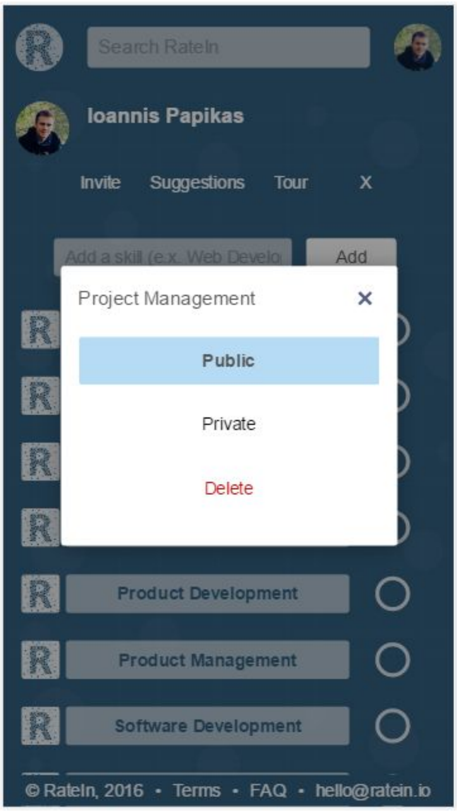
As part of the application, there is an on-boarding tour that presents to the user all the necessary steps and functionalities of the application in order to understand the basic concepts and get on-board faster.

Finally, when the on-boarding process is finished, the user has access to their profile and they will be able to use the explore mode to search for other users based on their names and skills.

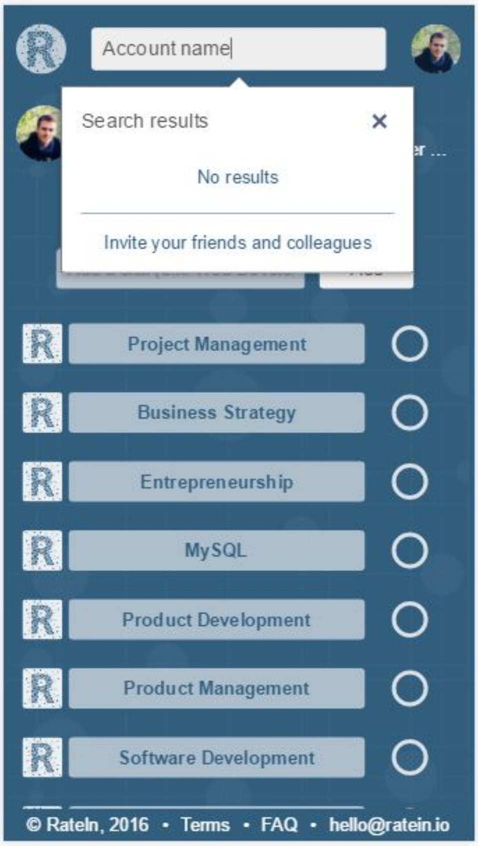
4.1.2 Explore mode

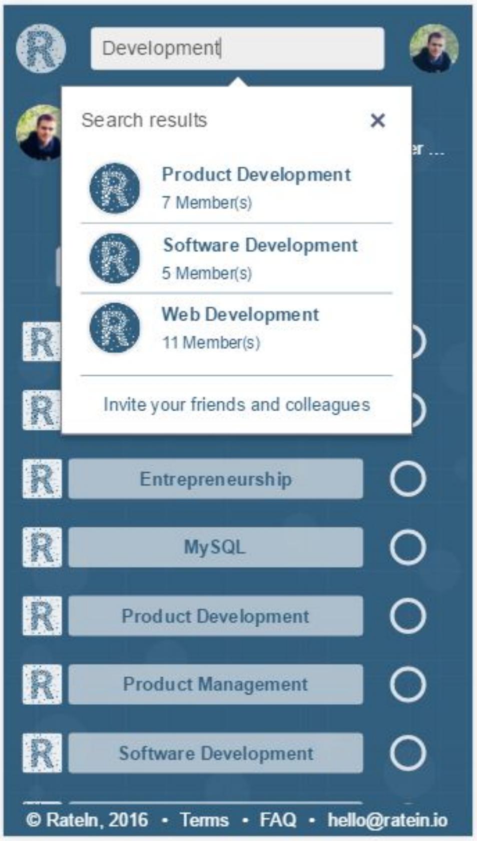
After registering and going through the on-boarding process, all users will be redirected to their profile page. On the profile page, the main functionality is to allow the user to build their profile by adding new skills and editing existing skills. A user can select a skill to be private, visible only by the owner and not ratable, and public, visible by everyone and anyone can rate.





This page is the basic page for the explore mode. The user can always search for a user by their name/title or search for skills and see their members. This allows a full exploration of the RateIn network in this proof of concept application.





The above screenshots display the search dialog for a user or a specific skill. Through the search results the user can navigate on another user or see all the users (the skill network) of a given skill.

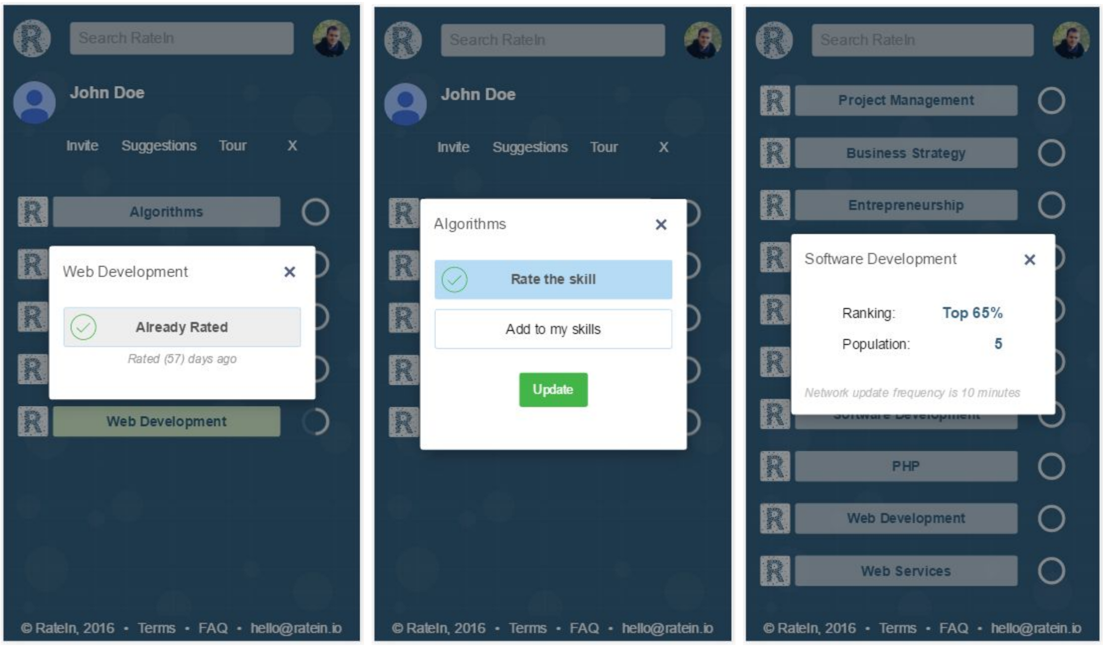
Next step is to be able to rate another user’s skill. This can easily be achieved by going to the profile of the user that one want to rate and, based on the skills that the user has, to rate the ones that the user knows (best). The process is simple, any user can do it for any other registered user without any limitations. The rating dialog offers two options:

1. Rate the skill (if not rated yet)
2. Add the skill to one’s profile (if one does not have it already)

At this point it is important to describe better the open and not controlled flow that endorsements have and what it is implemented instead. In the current implementation, any user can also rate other users for any skill, but it is not like endorsements. This means that the user that rates and does not have the skill gets the skill at the first rate, but the skill is hidden.

The user needs to have the skill in order for the algorithm to calculate values and apply the equation in general. Since the skill is hidden, the skill value is the default initial value and it will not increase (it requires ratings) unless the user adds the skill to their profile.

In the screenshots below one can see the rating dialog and the ranking dialog. The rating dialog appears when the user wants to rate a user's skill and the ranking dialog appears when the user clickß on the ranking circle on the right of the skill to see how the user ranks based on the population and the highest value.



Rated (57) days ago

Ranking: Top 65%

Population: 5

Network update frequency is 10 minutes

For this proof of concept application, rating visibility will be limited. Users will be able to see only the outgoing ratings (first screenshot, green skill) and not their incomings. They will be able to see how many ratings they have on which skills but not the users that rate them. This approach is used to limit distractions and biased ratings.

4.1.3 Skill calculation

In the first version of the application, the converge algorithm is designed to process new ratings every 5 minutes. This means that there is an average of 2.5 minutes that a user must wait for their ratings to be applied. As seen above, the ranking is a percentage of their score over the lowest and highest rate in the skill graph. When a ranking is described as Top 20%, it means that the user ranks in the first 20% of the population from the top. In other words, given a population with 100 users, ordered descending based on the skill value, top 20% means that the user is in the first 20 users.

Part of the paradox in the skill calculation, as described before, is the fact that not all values increase. Adding a new rating to the system will not cause other ratings to increase. Depending on the graph structure, it will cause at least one value to increase (the recipient of the rating) and from 0 to more values to decrease. If the user that gives the rating has also other outgoing ratings for the same skill, then all the 1st degree neighbors will probably get their skill value decreased, however this is just a possibility and it depends fully on the graph structure.

The application has an activity notification mechanism. According to user preferences, the application is going to send daily or weekly updates (first version has email notifications) about the user's activity in receiving and also sending ratings from and to other users accordingly.

4.1.4 Connect with 3rd party services

The application server exposes a RESTful API to the world. Through this API the existing web application is able to connect and perform all the following actions:

- Authentication mechanism, register and login a user
- Profile editing, update user information and settings
- Create, update and delete skills
- Create ratings
- Read skill rankings
- Search users
- Search skills
- Read users for a given skill

The API is going to be public and external services will be able to access the API after receiving an API key. This will allow external engines to register users and provide skills and ratings for their network. They will be also flexible to define rating weights and read user info and rankings. These 3rd party services will help expand the skill graphs or create new ones for different markets. Other services can include LinkedIn, eBay, stackoverflow and many more that can create skills and skill ratings equivalents for the users.

4.2 Recruiters and future implementation

The previous chapters have described a simple and proof of concept application in order to see how weighted and directed graphs can generate better and more valuable results regarding user skills. This chapter is going to describe a business model that can be adopted and a future implementation to enable growth towards the global market.

4.2.1 Business model

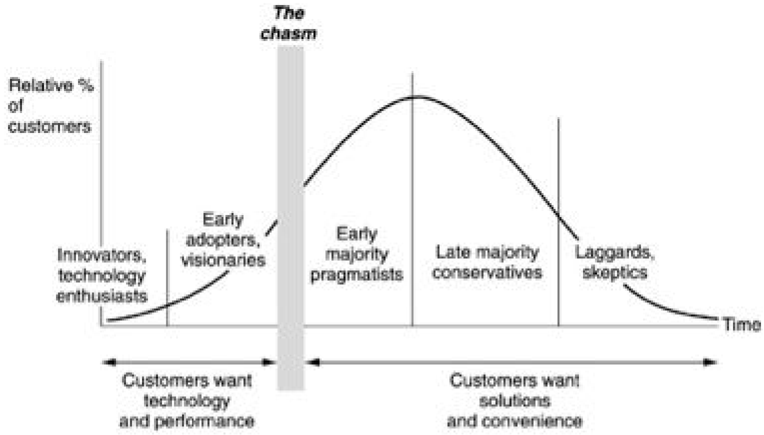
Recruiters use services like LinkedIn or facebook to find candidates for positions that they have available. Their job includes a process of creating a profile for different people based on their skills. Using this application, recruiters will be able to find users with the top skills in their area or sector and target them more efficiently.

Users will be able to create a profile for free and add skills, even load skills from external sources like LinkedIn or stackoverflow. Users will be able to start rating other users, friends or connections to grow their skill graph and their skills.

Recruiters will be able to create a premium account to unlock extra features of the application. Users with premium account will be able to use the application's advanced search to search for users in specific regions or users with a given skill combination. Also, recruiters will be able to see rankings in much more detail using graph connections and discover full paths between distant nodes. Finally, the most useful feature for recruiters is to be able to see advanced analytics for a specific combination of skills or within a given geographical area.

The revenue model of the application is a Freemium model where the application is free for all users but premium users will have to pay a monthly fee to access the premium features. Recruitment companies will have to pay per user per month.

The initial target audience is going to be users with technical skills like developers who have a better understanding of the underlying infrastructure and goal, they are going to be the early adopters according to the law of diffusion of innovation. To be able to cross the gap between early adopters and the mass, the application has to reach 15% of the population and be adopted by users via existing social networks.


The chasm
Relative % of customers
Innovators, technology enthusiasts
Early adopters, visionaries
Early majority pragmatists
Late majority conservatives
Laggards, skeptics
Time
Customers want technology and performance
Customers want solutions and convenience

Law of diffusion of innovation

4.2.2 Future implementation

The current proof of concept application can be characterized as an early MVP. This means that it currently has the minimum amount of the described features so that it can prove that the underlying previously described model works, both in accuracy and in performance.

Part of the future implementation, the application's growth strategy is going to include the following features:

1. Create a skill helper engine which will allow users to choose the skills that match their needs. This engine is going to be part of a smart algorithm that will group skills together in categories and will merge similar skills. It will also propose new skills to users based on preferences, history and other user's skills.
2. Connect 3rd party services and allow them to add users and ratings. As described in a previous chapter, regarding social impact, 3rd party services and their connection with the application api is an important part of the application growth and part of the expansion strategy to different graphs and networks.
3. Implement the business model described in the previous chapter that will allow recruiters to have access to premium features of the application. The implementation is going to include all the features of the premium user like the following:
    a. Advanced user and skill search
    b. Advanced ratings and better graph display
    c. Advanced analytics and rankings using regional data

4.3 Results and Decision Making

The application was tested in two ways. Different and multiple simulations were put into the microscope using fake user accounts, fake skills and fake ratings to test the accuracy of the model but also the performance. The algorithm is designed to work in a increased performance schema and handle one skill graph at a time. This means that it is ready for scaling, in case there is enough payload, to separate the skill graphs in different threads or even servers.

This chapter is going to present the social impact of the application but also detailed simulation results regarding the performance of the algorithm.

4.3.1 Social impact and reactions

After publishing the application, with its purpose to prove the model, the application managed to get 30 real users to sign up and they added a total of 140 skills, 60 unique skills in total. All 30 users have created 70 ratings, for 24 skills in total.

We published the idea and the concept of the application to known social media (facebook, linkedin and twitter). Some of the positive feedback received can be listed here:

- *"Sounds like pagerank for endorsements"*
- *"...algorithm that would visualise a rating system. I would like to know more..."*

In general, there was a positive feedback regarding the concept of the model. However, not all audiences have the same positive feedback. After uploading the idea on the hacker news website (https://news.ycombinator.com/item?id=12966375, show section), there was no attention at all and a single comment from a user along the following lines:

- *"Improving something that has the underlying idea broken beyond repair"*

We could really drive into a final conclusion that the feature cannot stand on its own and it has to be connected with a social network and allow the users of the network to create skills through this model.

As already mentioned in the **Future Implementation** chapter, the plan is to integrate the current application with an existing social network and really test its impact to people's skills and ratings.

4.3.2 Performance and accuracy

As already mentioned in the previous chapter, the real user reach was very limited to a simple testing audience of users that were curious enough to test the application.

Seeing that the application cannot get big numbers of users really fast, a graph and algorithm performance simulation is needed. In order to test the performance of the algorithm in the smallest VPS server available on the market, the strategy included the creation of fake accounts, skills and ratings.

Before moving forward and analyze the simulation, it is important to mention here that the algorithm works in 5 minutes intervals. Because it’s a converge algorithm, it can be a bottleneck to trigger it as soon as a new rating is added because it might not finish until a second rating is added again. For this reason, it is designed to run every 5 minutes and get all the ratings that were created in the last 5-6 minutes (to be sure that there are no ratings in between) and update the values based on the new graph edges.

Note: Re-running the algorithm over and over again will not change significantly the values of the skill graphs. Because of the convege algorithm, only the direct neighbors of the affected nodes might slightly change their values but the values will not propagate because of the threshold.

To resume to the simulation scenario, during the first step a total of 500 fake accounts were created, 10.000 skills for all the accounts and 10.000 ratings.

The next sections will analyse the way new ratings were added and what was the impact of the algorithm and the model.

Random Simulations

The process includes a randomization of the ratings. The first attempt ended with 500 ratings creating no circles, which had as a result each account skill to be updated slightly.

Simulation No	#1
Number of users	500
Number of skills	10000
Number of ratings	500
Nodes affected	484

Iterations	484
Elapsed time	16.150847196579s

Since the previous results are not clear and decisive, another 2000 ratings are added in the system to add the possibilities of creating circles and actual graphs.

Simulation No	**#2**
Number of users	2000
Number of skills	10000
Number of ratings	2500
Nodes affected	2050
Iterations	2050
Elapsed time	98.727710008621s

The simulation experiments use random numbers but the above results show that the number of users is very high and cannot create a connected graph of more than two nodes. The process is being reset and start again with 500 accounts in order to make the account pool smaller and increase the possibilities of selecting the same account and create graphs with more than 2 nodes. The number of skills will be also limited to 30.

Simulation No	**#3**
Number of users	500
Number of skills	30
Number of ratings	10000
Nodes affected	500
Iterations	2310
Elapsed time	115.05077910423s

Although the result show that there were 2300 iterations, which means that there were propagations of the ratings, the rating values were between 1 and 16.27, with rankings between 0.0185 and 0.9999 (0.9999 can be considered as 1 but it is part of a graph). There was a maximum of 5 outgoing rankings for each user for a single skill.

The above result is still not good because the ratings are still sparse, among the 500 users and 30 skills. A full graph can have 500x500x30=7500000 edges and only 10000 were inserted. The process is repeated using only 50 accounts to limit the number of total edges of a full graph in 50x50x20=50000. Let's try to increase the number of outgoing rankings per account per skill.

Simulation No	#4
Number of users	50
Number of skills	20
Number of ratings	10000
Nodes affected	50
Iterations	461
Elapsed time	4.5805480480194s

The algorithm this time did 9 times more iterations than the users, which means that there were many cases where it had to propagate the changes to the rest of the graph. This approach generated 10.000 ratings with a maximum of 18 ratings both received and sent per user per skill.

Here are some accumulated statistics that were gathered from all the previous random simulations:

Total users	~500
Total skills	~1000
Total account skills	~10500, from 50 users and 20 skills
Total ratings	~10000
Maximum outgoing ratings per user per skill	18

Maximum ingoing ratings per user per skill	18
Maximum generated value	9.3286303947521
Minimum generated value	2.5743783466849 (1, the base, is excluded)
Average ranking generated value	5.914111211256835 (excluding the entries with 1)

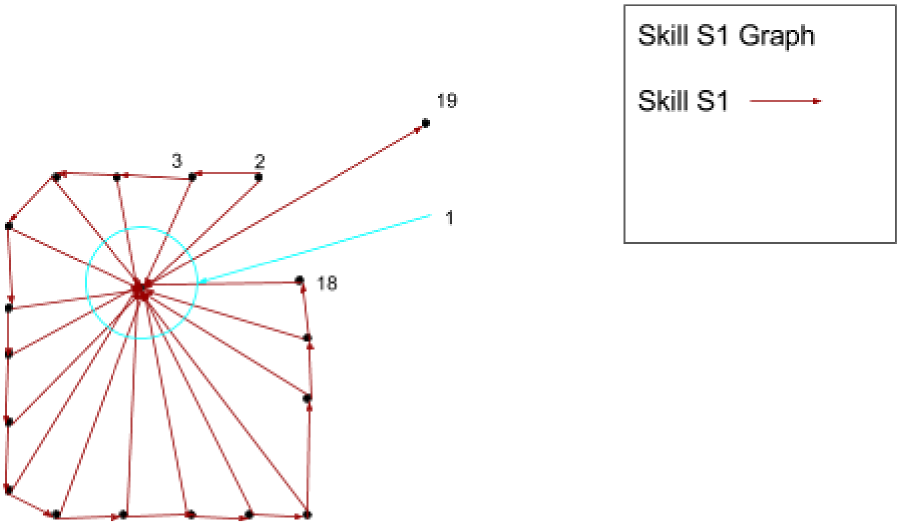
Structure-Specific Simulations

As seen in the previous simulations, randomization can be useful to generate a big amount of data and analyze the, but at the same it makes the ratings random and most of the time equally spread among the users. Also, in general there was a standard average number of ratings (18 ratings on average) meaning that most of the users had 15-22 ratings, without having any outliers, either a lot less ratings or too many.

In order to study in more depth specific cases and examine different graph structures, specific graph structures were created to allow the study of these graphs regarding performance, rating and ranking values and paradoxes.

Simulation #5

This is a specific simulation on a specific graph. The graph that the simulation is going to create is the following:



Simulation No	#5
Number of users	19
Number of skills	1
Number of ratings	35
Nodes affected	18
Iterations	21
Elapsed time	0.25875997543335s

The above simulation creates a specific graph structure where all users rate a single user for a skill and the user in the middle rates an external user. Also, users create a rate chain, as seen in the graph. In this simulation there are two kinds of results:

1. It is visible how, having this chain, the skill rating increases on every step
2. It is also visible how the external user, although it has one rating, the ranking is big

Data:

User Id	Skill Id	Outgoing ratings	Rating Value	Ranking
1	5	1	13.01890385	1
19	5	0	12.80519836	0.9822192188
18	5	2	1.739129597	0.06149725518
17	5	2	1.739128464	0.06149716092
16	5	2	1.739125799	0.06149693911
15	5	2	1.739119526	0.06149641722
14	5	2	1.739104767	0.06149518924
13	5	2	1.73907004	0.06149229988
12	5	2	1.738988329	0.06148550137
11	5	2	1.738796069	0.06146950489
10	5	2	1.738343692	0.06143186611
9	5	2	1.737279276	0.06134330428
8	5	2	1.734774768	0.06113492351
7	5	2	1.728881807	0.0606446158
6	5	2	1.715016016	0.05949095061
5	5	2	1.682390625	0.05677644428
4	5	2	1.605625	0.05038937055

3	5	2	1.425	0.03536096179
2	5	2	1	1

The user with user id 2 has value 1 because it has not received a single rating.

Simulation #6

Another case of a specific simulation is to present the power of the mass. This case will include a similar solution like above, but instead of having 1 center it is going to have 5 center nodes and 100 users around rating 3 out of 5 center nodes.

After running the simulation, a total number of 231 ratings were created in the following amounts:

User Id	Number of Incoming Ratings
1	47
2	46
3	35
4	61
5	41

Simulation performance:

Simulation No	#6
Number of users	5
Number of skills	1
Number of ratings	231
Nodes affected	5
Iterations	5
Elapsed time	0.49890995025635s

Ratings and rankings values:

User Id	Rating Value	Ranking
1	16.86666667	0.7179487179
2	17.43333333	0.7435897436
3	13.04166667	0.5448717949

4	23.1	1
5	14.45833333	0.608974359

As expected, the user with the most incoming ratings has the bigger ranking. But it is also notable the fact that although user with id 1 has 47 incoming ratings and user with id 2 has 46, user with id 2 has a bigger rating value and a bigger ranking. A short analysis in the data shows that the users that rated user id 1 had more outgoing ratings and according to the weight equation, they transferred less value.

Conclusions

The above simulations had two important outcomes of the new model:

1. The model accuracy works very good, as already mentioned
2. The model is capable of giving value to the users depending on the community and the value of incoming ratings, regardless the count.

5 Research references

The research was based on existing testimonies and community opinions regarding the impact of LinkedIn endorsement feature. So far there is no scientific analysis about reviewing and rating people's skills and characteristics.

Extended research was done in combination with the PageRank algorithm in order to be applied in the rating model and the converge algorithm.

The work presented in this paper is a first sample of the new rating model and how it can be applied to known systems and services used everyday.

5.1 Future studies, implementation and global impact

Future studies of the algorithm include an extensive research of the social impact that this model can have to users. Early results have shown that the idea is interesting and can be implemented with a meaningful impact on users and rating their skills but there was also criticism about the problem itself, having flaws as a concept and it cannot be fixed.

The application that was built as a proof of concept for this dissertation is a stand alone model-application just for the purposes of the research and it has to be connected with a social network in order to function properly, as an underlying system behind user skills. Future implementation is going to include attempts to be embedded in some kind of social networks to see the reactions of the users and the recruiters.

5.2 References

[1] "Our Heritage." *Designer Fashion, Accessories & More - Shop Online at Selfridges*. Web. 30 Nov. 2016. http://www.selfridges.com/GB/en/content/our-heritage

[2] "How Feedback Works." *How Feedback Works*. Web. 30 Nov. 2016. http://pages.ebay.com/help/feedback/howitworks.html

[3] "TripAdvisor: Read Reviews, Compare Prices & Book." *TripAdvisor: Read Reviews, Compare Prices & Book*. Web. 30 Nov. 2016. https://www.tripadvisor.com/

[4] "Introducing Endorsements: Give Kudos with Just One Click." *Official LinkedIn Blog*. Web. 30 Nov. 2016. https://blog.linkedin.com/2012/09/24/introducing-endorsements-give-kudos-with-just-one-click

[5] Http://www.cityhyd.info/author/admin/. "ARE LINKEDIN ENDORSEMENTS MEANINGLESS?" *CITY HYDERABAD*. 30 Jan. 2013. Web. 30 Nov. 2016. http://www.cityhyd.info/2013/01/31/are-linkedin-endorsements-meaningless/

[6] @andyxfoote. "The Truth About LinkedIn Endorsements." *LINKEDINSIGHTS.COM*. 06 Jan. 2015. Web. 30 Nov. 2016. https://www.linkedinsights.com/the-truth-about-linkedin-endorsements/

[7] @undercoverrec. "LinkedIn Endorsements: The Good, The Bad and The Ugly." *Undercover Recruiter*. 17 Mar. 2016. Web. 30 Nov. 2016. http://theundercoverrecruiter.com/linkedin-endorsements/

[8] @sjtousley. "The Quickest Way to Increase Your Linkedin Endorsements." *Portfolium Academy*. 08 June 2015. Web. 30 Nov. 2016. http://blog.portfolium.com/increase-linkedin-skills/

[9] "The Best Way to Increase Your LinkedIn Endorsements." *Business 2 Community*. Web.

30 Nov. 2016.

http://www.business2community.com/linkedin/the-best-way-to-increase-your-linkedin-endorsements-01243970

[10] @smexaminer. "How to Maximize Your LinkedIn Endorsements : Social Media Examiner." *Social Media Examiner*. Web. 30 Nov. 2016.

http://www.socialmediaexaminer.com/manage-linkedin-endorsements/

[11] "PageRank Algorithm, 1998; Brin, Page." *SpringerReference*. Print.

[12] "Algorithms – Inside Search – Google." *Algorithms – Inside Search – Google*. Web. 30 Nov. 2016. https://www.google.com/insidesearch/howsearchworks/algorithms.html

[13] "PHP: Hypertext Preprocessor." *PHP: Hypertext Preprocessor*. Web. 30 Nov. 2016.

https://secure.php.net/

[14] "Authentication, Payments and Memberships from a Single Box." *Login Box*. Web. 11 Dec.