Breadth First Search for Music Recommendation in Last.fm and the Music Genome Project

Adhika Sigit Ramanto - 13512060
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13512060@std.stei.itb.ac.id

***Abstract*— In behalf of the increasing number of services that allow users to download or stream music, each one must have its own distinguishable imprints with which the services differentiate themselves amongst the others. One of these imprints is how the service can generate a list that recommends new music to the user by filtering a vast amount of data on the users' past preferences and taste. An extension to this recommendation system is how the music service determines what artists to recommend and makes sure the same recommendations doesn't recur. In this paper I will outline the design of a Breadth-First-Search approach in recommending new music using the data from both the user through two recommender system approaches, given a sizable and comprehensive information on both ends.**

***Index Terms*— Breadth First Search, Recommendation System, last.fm, Music Genome Project, Collaborative Filtering, Content Based Recommender**

I. INTRODUCTION

We are truly living in the age of information technology. Since the mid-90s, the world has seen a rapid advancement in IT and information sharing since the founding of the internet. Civilization has finally found a way to almost completely ignore spatial and geographical boundaries with ways previously unthinkable. Currently we almost take for granted what the people of the past would call life-changing simply because it has encompassed every aspect in our life.

This tremendous accomplishment owes its thanks to the efforts and inventions achieved in the field of networking throughout the 80s, and content digitalization beginning in the 90s, still undergoing vast improvements even to this day. The interconnectedness of the world has never been greater, with people in opposite parts of the earth sharing information in seamless rapid transfer. Even we can now witness events ongoing in the International Space Station with new connectivity technology using lasers.

The ubiquity and the increasing reliability of the internet, a globally connected and accessible network of computers utilizing a standard TCP/IP protocol, gives us a new and often preferred way of handling everyday activities. People open their laptops while still sleeping on their beds to go on a news website and check their e-mail instead of walking up to their porch to pick up this morning's newspaper and open the mailbox. They stream YouTube and download shows they want to watch instead of waiting for the right moment for the television to show the wanted program. Listeners download the latest music from iTunes with the ability to buy a specific song track instead of going to the record store with only the constraint to buy the whole album.

Better yet, with the escalating surge in internet connection speeds, now people doesn't even have to download the songs to keep in their personal hard drives. Instead they stream music from a cloud storage, another innovation made possible with the internet, and preserve memory space.

These ways of enjoying music are now seen by the industry as the most lucrative and therefore sought after both by the service provider and the client themselves. It has never been easier for people to listen to music. However, this rise in ease does have a side effect, in which people will want to discover more music than what is already in their playlist, but still clings to their tastes.

There are already a plethora of services concerning themselves with providing music to the users, whether they are streaming services, "personal radio" services, or download services. One thing they all have in common is that they each have their way in recommending new music to the listener.

The recommendation system is a staple of services that typically sells products client, and generate a list of the next potential sales conformed to the client's tastes to increase the chance of the customer actually making the next purchase. The system is in use not only in the entertainment industry, but also in online retail shops, and countless other services. There are even services specifically dedicated in providing recommendations.

II. THEORY

A recommender system is a filtering system which recommends products that the user may be interested in based on their previous choices. These systems usually work by comparing a user's profile to a reference trait, recommending items the user previously had not considered or had knowledge of. A recommender system generally belongs in one of two categories, based

on how the system determines the items to recommend. These two categories are Collaborative Filtering, and Content-Based Recommendation.

A. Collaborative Filtering

Collaborative Filtering (from now on referred to as CF) is a technique used by recommender systems where a user gets a recommendation based on other users' reference trait with a similar taste. CF has two senses, a narrow one and a more general one. In general, CF is the process of filtering for patterns using techniques involving multiple viewpoints and data sources, usually in the form of vast database of users and the information of their preferences. There are two methods to CF. The first type comes from the item rating system, where users rate items they have purchased, rented, or consumed on a scale which comes in varying forms such as the 5 star system or the 10 point system. This method proceeds as follows :

1. The system saves the items and ratings previously consumed by a user for whom the recommendation is made as their main reference and commonly referred to as the user's "library"
2. The system searches randomly for users with the same rating patterns as the main reference
3. The system saves a number of highly rated items not in the main reference's library into a list.
4. Repeat until a specified quota is met
5. The list is then shown to the user acting as the main reference.

This method is called the user-based collaborative filtering, which is one of the utilizations of the Nearest Neighbor Algorithm. The user-based approach proves ineffective when the items are highly subject to user taste, such as music, movies, and books. However, it is very useful when dealing with hard products that should be doing the same function for everyone, which leads to a more objective rating system.

A fundamentally different approach is used when dealing with items which have highly subjective ratings, first the system builds an item to item matrix determining the relationship between pairs of items, then method of machine learning is used to infer user taste using this matrix and data regarding user taste.

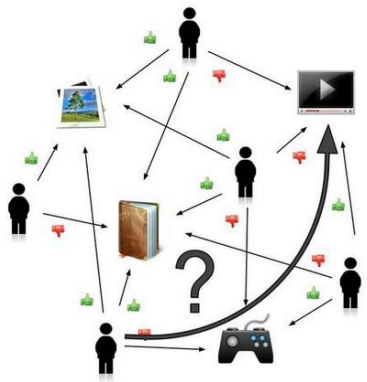

Figure 1. Collaborative Filtering (source : Wikipedia)

B. Content Based

The other commonly used method utilized in recommender systems is the Content Based Recommender

Through a different approach, the content based approach also tries to recommend items similar to those a given user has consumed, enjoyed, or purchased in the past. The basic process performed by a content-based recommender consists in matching up attributes of a user profile in which preferences and interests are stored, with the attributes of an item.

Systems implementing a content-based recommendation approach analyze a set of documents and/or descriptions of items previously rated by a user, and build a model or profile of user interests based on the features of the object rated by that user. The profile is a structured representation of user interests, adopted to recommend new interesting items. The recommendation process basically consists in matching up the attributes of the user profile against the attributes of a content object. The result is a relevance judgment that represents the user's level of interest in that object. If a profile accurately reflects user preferences, it is a huge advantage for the effectiveness of an information access process. For instance, it could be used to filter search results by deciding whether a user is interested in a specific Web page or not and, in the negative case, preventing it from being displayed.

Both the Content-based recommender systems need proper techniques for representing the items and producing the user profile, and some strategies for comparing the user profile with the item representation. The high level architecture of a content-based recommender system is as follows :

- Content Analyzer – When information has no structure (e.g. text), some kind of pre-processing step is needed to extract structured relevant information, the main responsibility of the component is to represent the content of items coming from information sources in a form suitable for the next processing steps. Data items are analyzed by feature extraction techniques in order to shift item representation from the original information space to the target one(e.g. Music represented as musical attributes with a standard scale of values in the Music Genome Project). This representation is the input to the Profile Learner
- Profile Learner – This module collects data representative of the of the user preferences and tries to generalize this data, in order to construct the user profile.
- Filtering Component – This module uses the user profile to suggest relevant items by matching the profile representation against that of items to be recommended. The result is a binary or continuous relevance judgment (computed with various modifiable algorithms).

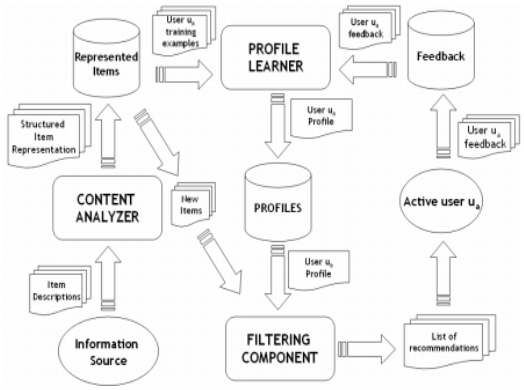

Figure 2. Content Based Recommender System

Typically it is possible to distinguish between two kinds of relevance feedback: positive information (inferring features liked by the user) an d negative information (inferring features the user is not interested) Users can also explicitly define their areas of interests as an initial profile without providing any feedback. There are three main forms of giving a feedback. The first one is a simple like/dislike, with translations that generally mean relevant or irrelevant with no intermediate values. Next, there are ratings, which can be a discrete numeric scale or symbolic ratings mapped to a numeric scale. Text comments are the most diverse kind of relevance feedback, and might need more advanced processing techniques for it to be used efficiently.

C. Music Genome Project

Most music sites allow some form of new music discovery. With some, the system is based on a form of rating or recommendation from other users. This could show up like 'Other users also bought XXX,' implying that if you liked this song, there is a good chance that you might like the other too. However, there are a few music services living on the Web that take a different approach. One of these is the Music Genome Project.

Conceived by Tim Westergren, the Music Genome Project attempts to take a more analytical approach. This project was launched in January of 2000 with an attempt to analyze the structure of a song so that they could identify similar songs that a searcher might like. Originally a project of his company Savage Beast, it took five years and 30 experts in music theory to build their database to the point of usefulness. Five years because each of the songs had to be individually listened to and up to 400 musical attributes manually assessed, which required 20 to 30 minutes for each four minutes of song. As of May 2006, this library contains in excess of 400,000 analyzed songs from 20,000+ contemporary artists.

Last year saw the release of its online interface, called Pandora Radio, and a company name change from Savage Beast to Pandora Media. Pandora, from Greek mythology, received many gifts from the gods, including both the gifts of music and curiosity which, in this case, are celebrated and rewarded. This binary Pandora generally requires no installation, as it works its magic through the use of Adobe's Flash plug-in (version 7 or 8, 8 being preferred) and some judicious Java Script. It will work on machines running Windows 2000 or Windows XP with either MS Internet Explorer 7 or 8, or with Firefox. Unlike many services, it will also run on the MacOS X 10.3+ with either Safari or Firefox. The processor should be running at 1 GHz or more, with a minimum of 256 MB of RAM. A broadband Internet connection is also required, as dial-up connections are not supported.

Pandora is best for broadening your musical exposure. While this may be true to some extent, at the very least, it is a good way to discover additional artists working in a style you like. The songs it tends to pick definitely have a different sound to them than say Yahoo!'s Launchcast radio. As most of the Music Genome selectors have not been revealed, an attempt to create an Open Source version of the Music Genome is being made.

D. Breadth-First-Search

The breadth first search algorithm (henceforth referred to as BFS) is a graph-traversal algorithm to systematically traverse the nodes of a graph. The BFS algorithm derives its name from the fact that it traverses graph sideways'; it first visits the root node, after which it visits a node v which is its child and is situated leftmost in a tree diagram. After this it visits its siblings –nodes situated to its right in a tree diagram—after which it visits its child nodes, so on and so forth until it has visited all the nodes in the tree diagram.

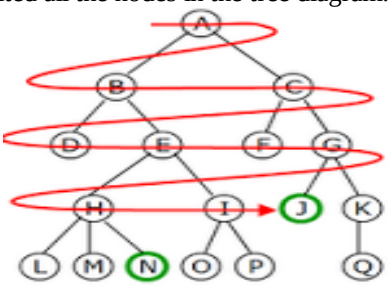

Figure 3. Visualization of the BFS state space tree (source: Google)

This diagram shows the order in which nodes are traversed in a BFS algorithm, 1 being the root node and beginning of the algorithm. For a data structure that builds nodes dynamically, as we will be using, the algorithm can be represented as a function that executes a function and puts the results into a queue, executing the function with the result at the head of the queue as a parameter while end conditions are

III. IMPLEMENTATION

We will see how Breadth-First-Search can be useful for both collaborative filtering and content-based recommender system, with an example for each method of recommendations.

1. Collaborative Filtering

One example of recommender system that uses the collaborative filtering system is from the music discovery and listening service Last.fm. The users installs an application in their computer and chooses a media with which said user will listen to music. After listening to at least half a song, the users' listening library is then updated in a process called "scrobbling". Last.fm has a vast library of recognized and profiled tracks, while all artists well-known enough get their own page. A section called "Similar Artists" is situated in each of these artists' pages. The similar artists list is generated by obtaining information from listeners of said artist and recommending the artists most listened to next after the webpage's artist.

Last.fm users also have a personalized recommendation in their profile page that makes use of the similar artists list. The personalized recommender takes the user's listening history list in descending order, then takes into account the artists. The artists will then each have their own Similar Artists list, the system checks for repeat mentions of artists in the multitude of Similar Artists list, and puts the artist with the most mentions not in the user's history on top of the recommendation list, and the list goes on with the second most mention, and so on.

Here is an example on how the Breadth First Search works in Last.fm music recommendation. Currently there are 6 random artists that the user adhikasigit (acts as the root node) recently listens to quite often. Namely Minks, Wild Nothing, Mac DeMarco, Craft Spells, Real Estate, and Yuck.

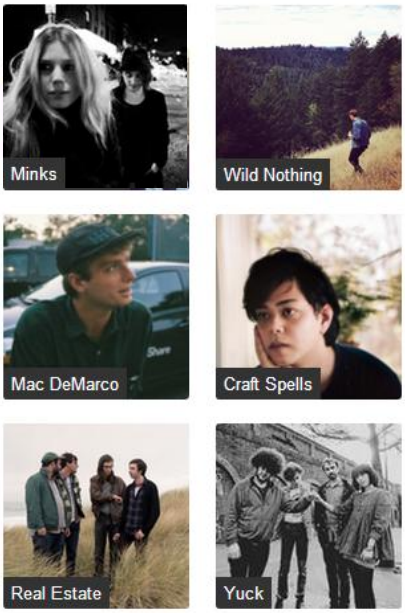

Figure 4. adhikasigit's " recently listened to" artists (source : Last.fm)

Here are the "Similar Artists" section on Yuck's and Wild Nothing's Artist/Musician webpage.

Figure 5. Yuck's and Wild Nothing's Similar Artists section (source : Last.fm)

The name DIIV appeared in 5 out of 6 Similar Artists' pages from the artists on Figure 4. Here is adhikasigit's personalized recommendations list

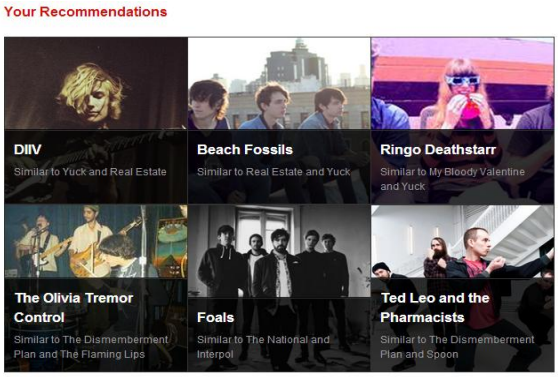

Figure 6. adhikasigit's personalized recommendations (source : Last.fm)

DIIV comes out on top because it is mentioned in Similar Artists the most amongst others, as you can see Beach Fossils got mentioned in both Yuck and Wild Nothing's Similar Artists List, which is why it is the second artist to be recommended after DIIV. Here is the BFS representation of how the personalized recommendation came to be.

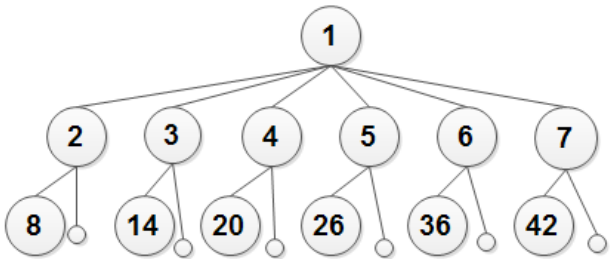

Figure 7. BFS tree for personalized recommendation

- 1 : User adhikasigit
- 2: Minks
- 3: Wild Nothing

- 4 : Mac DeMarco
- 5: Craft Spells
- 6 : Real Estate
- 7 :Yuck
- 14,20,26,36,42 : DIIV
- The smaller nodes represent that there are 6 children per musician (5 more than the numbered one) because there are 6 Similar Artists to that musician to a total of 47 nodes raised in the BFS tree

2. Content-Based Recommender

The following section will outline the design for a content-based recommender system that uses a BFS approach in its Filtering Component to actually determine what comes next in the playlist recommendation. It is also worth noting that our recommendation is one where it is created while the user begins listening, and may evolve differently according to some action by the user. The experience is hoped to be akin to a personalized music radio.

Until the last decade, it is almost impossible to build a content-based recommender system for music recommendation mainly because of the lack of a standard assessment with which songs are given attributes save for a biased genre tag that some music player tries to implement, and it often differs from one user to another. With the Music Genome Project, we finally have a comprehensive standardization of musical attribute assessment with a database sufficient enough to guarantee a playlist possibly larger than anyone's music library.

As stated before, there are three main components in the architecture of a content-based recommender: Content Analyzer, Profile Learner, and the Filtering Component. We will be going through the first and second briefly, then focus on the Filtering Component which will utilize a Breadth-First-Search approach out of a number of other applicable algorithms.

A. Content Analyzer

The content analyzing component of our music recommendation system the Music Genome Project, an ongoing project that tries to give songs and artists a standardized attribute assignment with values of a certain scale. Each song has its own "genes" analogous to the human DNA imprint that makes each song unique with the combination of about 400 genes on average. The genes are attributes that identify a song e.g. "Used Instruments", "Lead Vocal Style", "Rhythm", "Beats per minute", etc. We now have a large enough amount of songs and artists already analyzed.

B. Profile Learner

Users of the personalized music station will have a simplistic profile that saves the listening history of the user, along with the number of times the song has been played, from the listening history, the system might infer the user's preferences, although in most cases the user would get to choose the first song that "jump starts" the personalized music station. As the playlist is dynamically made, the user has the ability to disapprove or dislike a song that he/she felt not suitable in the music station, and help improve the next recommendations. The profile learner can be said to have the responsibility of supplying root nodes to the BFS search tree in the filtering component.

C. Filtering Component

After obtaining a large database of music attributes from the Content Analyzer, also a profile of the user's taste and preference to help with making the music station, there needs to be a component to actually determine what songs to be added next to the playlist. To have a good playlist, the songs that the music station explore needs to be diverse enough, while still clinging to the original style derived from the user's taste; Furthermore, it needs to branch out progressively as the user continues on listening to the music station. To achieve this stability of music recommendation, we use a Breadth-First-Search approach in exploring the songs.

There are two cases in starting the personalized music station, either the user chooses a first song, or the Profile Learner feeds the filtering component with a song that conforms to a "Mood" settings in the player. In either case, the song becomes the root node in our BFS tree.

The Breadth-First-Search component is easier to explain with an example. Let's say we have a user play the first song, which is the song "The Daily Mail" by Radiohead. Now we have a root node 1= "Radiohead – The Daily Mail", this root node will have its genes,, which we obtain from the Music Genome Project. Next, the Profile Learner will feed the Filtering Component with a diversity setting which acts as a constraint on how many children a node can have. For this example, the setting is set to four children.

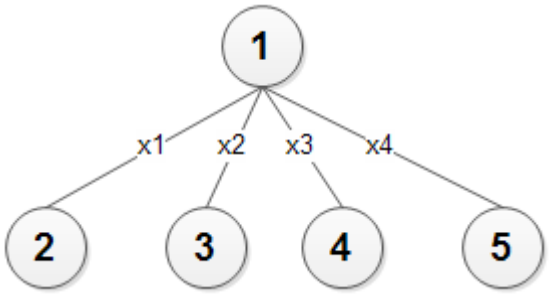

Figure 8. Showing a level of children, can also represent a single recursion

- 1 : "Radiohead – The Daily Mail"
- 2 : "The National – England"
- 3 : "Five for Fighting – 100 Years"
- 4 : "Radiohead – You and Whose Army?"
- 5 : "Swans – Song for a Warrior"
- X1 : "Used Instruments"
- X2: "Lead Vocals Style"
- X3: "Soundwave Formation"

- X4 : “Melodic Progression”

Now, the root node 1 will have four children, 2, 3, 4, 5, but on what decision should the making of the child node be? The decision is either a random gene from the parent node or a user-demanded attribute that is determined beforehand in the Profile Learner. For 2, decision x1 = “Used Instruments”, so 2 will be a selected song that has the same “Used Instruments” gene values as “Radiohead – The Daily Mail”. This returns “The National – England” the filtering component adds this to the playlist. For 3, decision x2 = “Lead Vocals Style”, and returns “Five for Fighting – 100 Years” and is added to the playlist. For 4, decision x3 = “Soundwave Formation”, and returns another Radiohead song, “Radiohead – You and Whose Army?”, suppose that when the song plays, the user clicks the dislike button. For 5, decision x4 = “Melodic Progression” and returns “Swans – Song for a Warrior”, then added to the playlist. Now, we have reached the maximum number of children that 1 can have.

We do the recursion for node 2 and 3 with the same decisions; this should return different songs for each of 2’s and 3’s children. Now, because the user disliked node 4 “Radiohead – You and Whose Army?” the BFS algorithm won’t generate the children for node 4 and instead skips to 5 and generates 5’s children. This can be done ad infinitum to make a never ending playlist, hence, a personalized music station.

IV. Analysis

Music taste is a highly subjective trait, with an immense variety from one person to another. Since the beginning of the music streaming and music download service the collaborative filtering method takes the dominant market share of use. Though the writer finds no fault in using collaborative filtering, there is one unavoidable weakness in using it, which is the tremendous subjectivity of each and everyone’s music taste; no one has the same exact musical preference. Through collaborative filtering, the recommender system is bound with the opinion of users, without knowing anything at all about the actual content of the product, or in this case, music, but it still is capable of giving the users a viable recommendation list. But until a decade ago, there is no workaround to this weakness mainly because no one has made large database that concerns a relatively large number of musical experts trying to unanimously assign attributes to music. With the Music Genome Project, another fresh window of opportunity is opened. There is now a way to judge musical preferences by the content itself and the results are different. In Last.fm, Radiohead’s page has the similar artists Atoms for Peace, Thom Yorke, Jonny Greenwood, Muse, Coldplay, Portishead, and Arcade Fire, Sigur Ros. Three of those similar artists are members of Radiohead, by logic we can assume that fans of a band tends to follow the Spotify’s Discover Radio also has the same artists in the first ten song skips. Pandora, which uses the Music Genome Project, on the other hand only had Muse from Last.fm’s similar artists in the first twenty song skips.

Through the Breadth First Search, it is almost guaranteed that the playlists stay in the same veins of style while getting more diverse and progressive as the user continues listening. In the examples above, the next songs will be musically close to the previous ones. If other searching algorithms are used, say Depth First Search, the next few songs after the first one will tend to musically stray too far, and will come back to the style of the first song in the next songs after the DFS tree backtracks, losing the progressiveness of the playlist, and would exhaust the first gene that is used as the decision. Using a Brute Force approach would also mean exhausting all the songs in one gene before moving on to the other gene of the song, this will make for a redundant radio. It is not wrong since music has high subjectivity and people might actually prefer the methods of the DFS and Brute Force search, but for the purpose of broadening musical discovery, the BFS would fit best.

V. Conclusion

Recently there has been a boom of brimming opportunity in the music industry for streaming and downloading music through the internet, because of the same products and price range, the services try to best each other by the way they allow customers to discover new music, the recommender system in the music industry is dominated by the collaborative filtering, such as the one used by Last.fm’s recommendation system, but the decade-old Music Genome Project allows a content-based recommender approach for music. The user-personalized recommendations in Last.fm uses a Breadth-First-Search on their Similar Artists recommendations and finding the most mentioned artists. With the Music Genome Project, using a Breadth-First- Search algorithm for the Filtering Component in the Content-Based Recommender System to determine the playlist provides a better experience for broadening musical discovery.

VII. Acknowledgment

I would like to thank my mentor, Ir Rinaldi Munir, M.T., as it was due to his guidance and his textbook that this paper was written and completed, Tim Berners-Lee for having invented the internet. Sergey Brin for founding Google. Felix Miller for founding Last.fm, and Tim Westergren for initiating the Music Genome Project.

References

[1] M. J. Pazzani, “A framework for collaborative, content-based and demographic filtering,” *Artificial Intelligence Review*, vol. 13, no. 5–6, pp. 393–408, 1999.

[2]

"About The Music Genome Project." [Online]. Available: http://www.pandora.com/about/mgp. [Accessed: 16-May-2014].

[3] "Last.fm: 50 billion scrobbles, the return of the mix tape - TNW Media." [Online]. Available: http://thenextweb.com/media/2011/04/16/last-fm-50-billion-scrobbles-and-the-return-of-the-mix-tape/3/. [Accessed: 18-May-2014].

[4] Joyce, John, "Pandora and the Music Genome Project." [Online]. Available: http://www.scientificcomputing.com/blogs/2006/08/pandora-and-music-genome-project?terms=music%2520genome. [Accessed: 16-May-2014].

[5] "Patent US7003515 - Consumer item matching method and system - Google Patents." [Online]. Available: http://www.google.com/patents/US7003515?dq=7,003,515. [Accessed: 16-May-2014].

[6] P. Melville and V. Sindhwani, "Recommender systems," in *Encyclopedia of machine learning*, Springer, 2010, pp. 829–838.

[7] F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, Eds., *Recommender Systems Handbook*. Boston, MA: Springer US, 2011.

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 29 April 2010

ttd

Nama dan NIM