RECOMMENDER SYSTEMS ALGORITHMS

Project report submitted in fulfilment of the requirement for the degree of

Bachelor of Technology
In
Computer Science and Engineering
By

Ritwik Tiwari (151325)

Vaibhav Dutta(151332)

Under the supervision of

Dr. Pradeep Kumar Gupta

To

Department of Computer Science & Engineering and Information Technology

Jaypee University of Information Technology Waknaghat, Solan-173234, Himachal Pradesh

Certificate

Candidate's Declaration

I hereby declare that the work presented in this report entitled RECOMMENDER SYSTEMS ALGORITHMS in fulfilment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering/Information Technology submitted in the department of Computer Science & Engineering and Information Technology, Jaypee University of Information Technology, Waknaghat is an authentic record of my own work carried out over a period from August 2018 to May 2019 under the supervision of Dr. Pradeep Kumar Gupta, Associate Professor, Computer Science and Engineering/Information Technology.

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

Ritwik Tiwari, 151325
Vaibhav Dutta, 151332

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

Dr. Pradeep Kumar Gupta
Associate Professor

Computer Science and Engineering / Information Technology

Dated: 5/9/2019

ACKNOWLEDGEMENT

I have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organizations. I would like to extend our sincere thanks to all of them.

I am highly indebted to Dr. Pradeep Kumar Gupta for their guidance and constant supervision as well as for providing necessary information regarding the project and also for their support in completing the project.

I would like to express our gratitude towards our parents and Jaypee University of Information Technology for their kind cooperation and encouragement which helped us in completion of this project.

Mine thanks and appreciations also go to our colleague in developing the project and people who have willingly helped us out with their abilities.

Chapter 1

INTRODUCTION

1.1 Introduction:

Machine learning is a subset of Artificial Intelligence. It is field where algorithms are designed so that machines can learn autonomously from data. Machine Learning is a sub-set of computerized reasoning where PC calculations are utilized to self-sufficiently gain from information and data. In machine learning PCs don't need to be unequivocally modified however can change and enhance their calculations independent from anyone else. Today, machine learning calculations empower PCs to speak with people, self-governing drive vehicles, compose and distribute sport coordinate reports, and discover psychological militant suspects. I immovably trust machine learning will seriously affect most ventures and the employments inside them, which is the reason each director ought to have probably some grip of what machine realizing is and how it is developing. In this post I offer a speedy excursion through time to look at the roots of machine learning and in addition the latest achievements.

During the 1950s, we see the principal PC amusement program professing to have the capacity to beat the checkers best on the planet. This program helped checkers players a considerable measure in enhancing their aptitudes! Around a similar time, Frank Rosenblatt concocted the Perceptron which was an, extremely straightforward classifier however when it was joined in vast numbers, in a system, it turned into a ground-breaking beast. Indeed, beast is with respect to the time and in that time, it was a genuine leap forward. At that point we see quite a long while of stagnation of the neural system field because of its challenges in tackling certain issues.

On account of measurements, machine learning turned out to be exceptionally celebrated in 1990s. The convergence of software engineering and insights brought forth probabilistic methodologies in AI. This moved the field further toward information driven methodologies. Having extensive scale information accessible, researchers began to construct canny frameworks that could dissect and gain from a lot of information

Recommender frameworks utilize calculations to give clients item or administration suggestions. As of late, these frameworks have been utilizing machine taking in calculations from the field of man-made consciousness. In any case, picking a reasonable machine learning calculation for a recommender framework is troublesome on account of the quantity of calculations depicted in the writing. Specialists and experts creating recommender frameworks are left with little data about the current methodologies in calculation utilization. In addition, the improvement of recommender frameworks utilizing machine learning calculations regularly faces issues and brings up issues that must be settled.

Recommender System

It all starts from what is the recommendation? If we google it then the first thing that will pop out is “a suggestion or proposal as to the best course of action, especially one put forward by an authoritative body”. Generally, humans have a tendency to take some sort of recommendation to initiate some sort of activity. For example, it might be taking some friend's recommendation before making a car purchase or it might be buying some sort of clothing. Recommendations are ubiquitous and humans worked assiduously to make this possible. With the rise of artificial intelligence and lots of data processing recommendations are there on our screens. If someone wants to purchase something, Amazon provides recommendations. Bored want to watch something? YouTube is there and will recommend you some videos so that you can bust your time off. Another fine example is Facebook, it recommends who you should be friends with. There are many algorithms to achieve this some of them are efficient and some of them are not. Here we will discuss some of the most widely used algorithms and compare them.

1.2 Problem Statement:

Building a probabilistic model for recommendations. Model is based on Game Theory. Game Theoretic approach involves linear algebra and probability.

1.3 Objective:

Proving probabilistic model with neural networks outperforms the traditional approach with and without neural networks for generating recommendations.

1.4 Methodology:

Data of various types will be gathered. An improved scientific classification isolates recommender systems into substance based mostly versuscollaborative-separating based frameworks. Content based methodology: the attributes begin form the info itemcollaborative winnow approach: the qualities begin frame the consumer condition (social, consumer inclinations, designs, etc.)One of the elemental problems for with the 2 methodologies is that the chilly begin issue. New shoppers got to interface with the framework before to own aprofile developed and also the framework finishes up practiced for his or her wants. Hybrid methodology is often thought of, by connexion highlights fromcollaborative and content-sifting techniques, to counteract such impediments.

1.4.1 Content-based methodology

The substance primarily based methodology includes in breaking down the substance of the items being prescribed. Each consumer is prohibited severally. There isno presumption of gathering or network. The framework works preponderantly by dissecting things and also the distance of the selected things to othersselected by the consumer. At that time this stuff square measure chosen to be steered as they will intrigue the user.This approach is smartly based on that things square measure being thought of and on the consumer condition. There was no all-inclusive material foundon general exchange of the topic.

1.4.2 Cooperative Data Filtering

Synergistic separating emulates verbal proposals. Herlocker et al. states that "a standout amongst the simplest technologies for recommender frameworks known as synergistic separating". Shared separating frameworks originate from the previous knowledge filteringsystems. Those frameworks were created with the tip goal to convey simply vital knowledge to the consumer by observance past practices and therefore,building a consumer profile. This framework depends on the build-up of style knowledge from varied shoppers. A sense of network is underlying. It settles for that a gathering of shoppers can have a comparative gratitude to things at that time suggests that to "anticipate the on the QT inclinations of a functioning

Client enthusiastic about a straight weighted mixture of alternative individuals' inclination" [1]. Dynamic winnow is isolated from uninvolved separating on the grounds that utilizing dynamic separating need the consumer to commit a while with the tip goal to rate the information things once, utilizing latent winnow, shoppers consequently offer data by simply attending to the issue. Another methodology is that the item based approach.

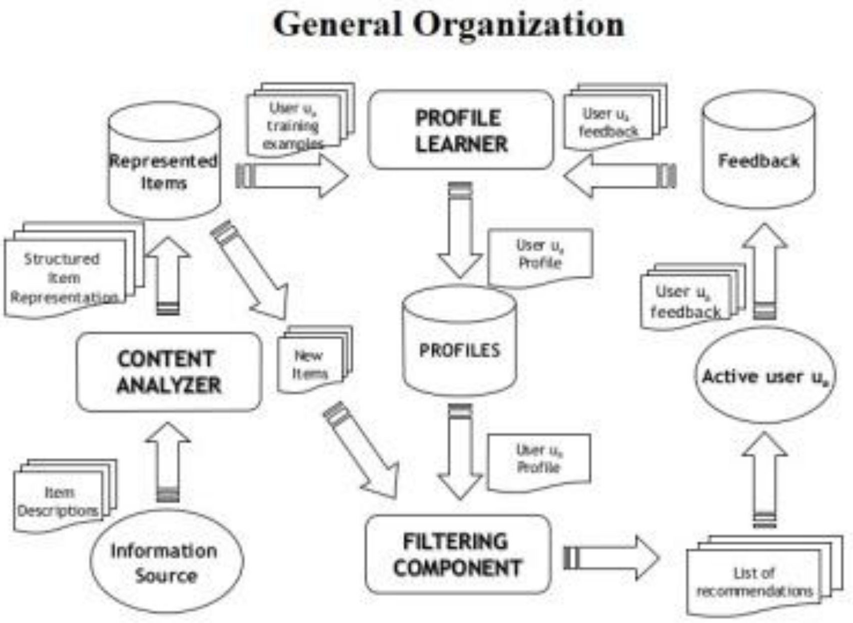
1.4.3 Dynamic separating (or unequivocal data accumulation)

Dynamic separating could be a technique for communitarian winnow in sight of its shared coordinative methodology. Totally different profiles from associatesare matched to get rid of comparable interests. This system depends on the certainties that peers trade knowledge such square measure evaluations and appreciation of specific things. It mirrors the regular methodology of companions prescribing outlets to at least one another. This type of separating is particularly viable in cases wherever people don't seem to be educated concerning the mass of knowledge accessible to them. One of the principle focal points of dynamic winnow is that the info rating is given by a real individual United Nations agency has seen the issue with interest. Another most popular point of view in intensely social-arranged frameworks is that it offers the open door for willing people to be detected and provide highly vital data. The principle inconvenience is that this framework needs some activity by the consumer and on these lines makes the data a lot of expensive to urge and rarer. Another prevalence of getting AN activity needed is that the inputs given could also be one-sided, as an example towards a negative or positive experience, contingent upon the target consumer. Another issue of these substance separating frameworks originates from the averaging impact occurring in some specific circumstances. Over a scope of comparative things, the framework will not understand the separating attributes between things. This ultimately oft makes the foremost acknowledge things be prescribed all the lot of often as they're going to have a lot of appraisals. Theproblem occurs for brand new things with no past rating and also the Cold begin issue happens for brand new shoppers with no past inclinations.

1.4.4 Inactive winnow (or verifiable data gathering)

Inactive winnow is consumer to collect knowledge verifiably. a number of precedents are: purchasing anitem using, stinting printing, adjusting, remarking over and over on an item Referring or connecting to a web site (in another setting than rating, as an example social media) Number of times an issue is queried Time estimations to determine whether or not the consumer is examining, perusal or operating with a document.The principle vantage of inactive separating is that it expands the amount of inhabitants in consumer giving criticisms. Basically, simply a number of populaces of shoppers come to the framework to rate issues although all interface with the framework to urge to the thing. Their conduct amid that stage is most probably able to offer knowledge concerning their advantage.

1.5 Organization:



Chapter 2
LITERATURE SURVEY

1. Matrix Factorization Model in Collaborative Filtering Algorithms: A Survey

Authors: Dheeraj Bokde, Sheetal Girase, Debajyoti Mukhopadhyay

Publisher: Procedia Computer Science, Elsevier

Year: 2015

This paper attempts to present a comprehensive survey of Factorization model like Singular Value Decomposition to address the challenges of Collaborative Filtering algorithms, which can be served as a roadmap for research and practice in this area.

2. Latent Factor Models for Web Recommender Systems

Authors: Bee-Chung Chen, Deepak Agarwal, Pradheep Elango, Raghu Ramakrishnan

Publisher: Yahoo! Research & Yahoo! Labs

Year: 2014

This paper discusses about the model that:

- uses feature-based regression to predict the initial point for online learning, and

- reduces the dimensionality of online learning

Rapidly update online models once new data is received:

- Fast learning: Low dimensional and easily parallelizable

- Online selection for the best dimensionality

[3] Matrix Factorization Techniques for Recommender Systems

3. Authors: Yehuda Koren, Yahoo Research; Robert Bell and Chris Volinsky, AT&T Labs—Research

Publisher: IEEE Computer Society Press Los Alamitos, CA, USA

Year: 2009

Used in Netflix Competitions

This paper discusses about matrix factorization techniques have become a dominant methodology within collaborative filtering recommenders.

Experience with datasets such as the Netflix Prize data has shown that they deliver accuracy superior to classical nearest-neighbour techniques. At the same time, they offer a compact memory-efficient model that systems can learn relatively easily. What makes these techniques even more convenient is that models can integrate naturally many crucial aspects of the data, such as multiple forms of feedback, temporal dynamics, and confidence levels.

Chapter 3
SYSTEM DESIGN

To see how recommender system functions, consider an example of five mobile phones with two noteworthy highlights "Battery and Display". These five mobile phones have the following properties:

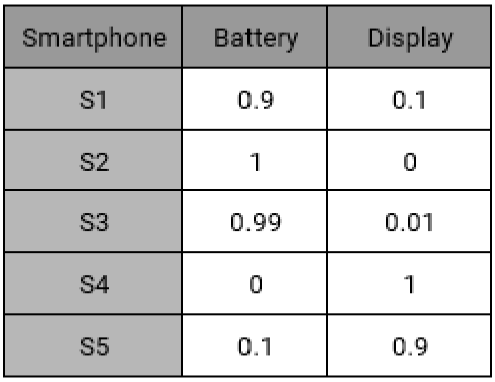
S1: Great battery life & poor display

S2: Class leading battery life & poor display

S3: Great battery life & very poor display

S4 and S5 have nice display nevertheless poor battery execution.

Utilizing these attributes, we are able to create an Item – Feature Matrix.



Smartphone	Battery	Display
S1	0.9	0.1
S2	1	0
S3	0.99	0.01
S4	0	1
S5	0.1	0.9

Our example set likewise comprise of four consumers and their choices or preference of the mobile phones.

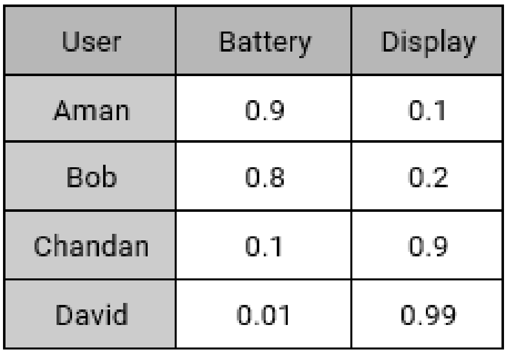
Aman: Prefers battery over display.

Bob: Prefers battery over display.

Chandan: Prefers display over battery.

David: Prefers display strongly over battery.

Utilizing their interests, we are able to create a User – Feature Matrix:



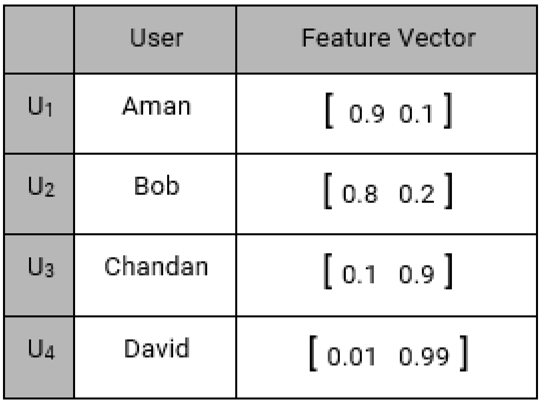
User	Battery	Display
Aman	0.9	0.1
Bob	0.8	0.2
Chandan	0.1	0.9
David	0.01	0.99

We have two matrices namely Item-Feature Matrix and User-Feature Matrix. Using these matrices we will generate the recommendations by utilizing the following algorithms. These algorithms are basic stepping stone for recommendation systems. Advanced recommendation systems are made by exploiting these algorithms and implementing them in several layers.

Content-based Recommendations:

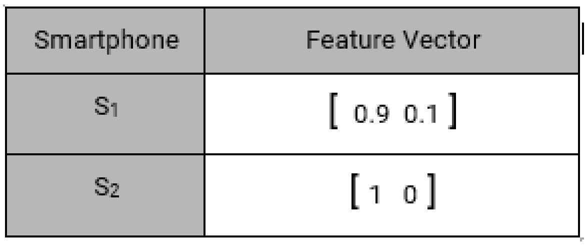
In this algorithm it generates the recommendations based on similarity index. It calculates the similarity based on the relationship between item specification and the user preference. The feature matrices are converted into vectors as shown and the recommendations are calculated as shown below:

Vectorizing the User-Feature matrix:



	User	Feature Vector
U_1	Aman	[0.9 0.1]
U_2	Bob	[0.8 0.2]
U_3	Chandan	[0.1 0.9]
U_4	David	[0.01 0.99]

Vectorizing the Item-Feature matrix:



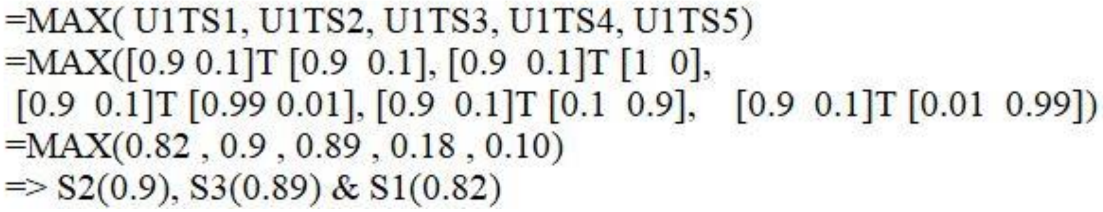
Smartphone	Feature Vector
S1	[0.9 0.1]
S2	[1 0]

User and item recommendations are mapped by the following relation:

$\mathrm{MAX}\ (U_{(j)}{}^{T}.\ I_{(i)})$
i, j -> n, m

This mapping procedure is called content based recommendation.

Recommendation calculation for user U_1 is shown below:



=MAX(U1TS1, U1TS2, U1TS3, U1TS4, U1TS5)
=MAX([0.9 0.1]T [0.9 0.1], [0.9 0.1]T [1 0],
[0.9 0.1]T [0.99 0.01], [0.9 0.1]T [0.1 0.9], [0.9 0.1]T [0.01 0.99])
=MAX(0.82 , 0.9 , 0.89 , 0.18 , 0.10)
=> S2(0.9), S3(0.89) & S1(0.82)

From the above result we see that mobile phone S2 has the highest score of 0.9 followed by mobile phone S3 (0.89) and then mobile phone S1(0.82). Mobile phone S2 will be recommended to U_1 (Aman).

Collaborative filtering-based Recommendations:

Content-based desires in characteristic repose conditions or complicated practices. For example, a consumer might like a mobile phone with very particular features. Like he/she might like a mobile phone only if it has infinity display with a qHD resolution and wouldn't otherwise.

This algorithm computes the recommendation by considering the recommendations of the other people which can be termed as "Client Behaviour". It is based on the idea that if a person has preferred an item in the past are likely to prefer same item in the future.They abuse the conduct of various shoppers and things relating to exchange history, appraisals, determination and get knowledge. Totally different shoppers conduct and inclinations over the items square measure utilized to impose things on the new shoppers. Collaborative filtering approach has two parts:

A. Memory based approach

Uses cosine based similarity or pearson correlation to find the similarity between the users

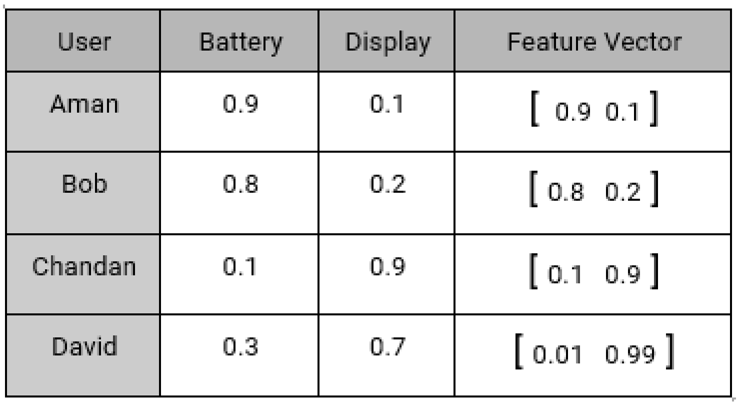
B. Model based approach

Uses machine learning algorithms to find the user ratings. Algorithms like matrix factorization, singular value decomposition, neural networks etc. are used.

Following example is based on the memory based approach. Since this approach is most basic to understand so we will explore this first.

As in content based recommendation system approach, we have used a User-Feature matrix, similarly we will start with it here:

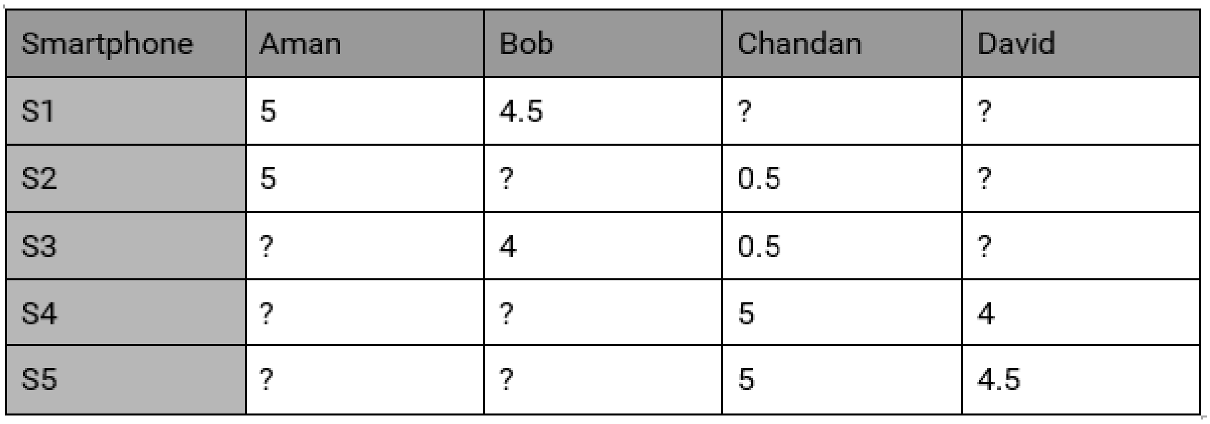
User – Feature Matrix (along with feature vector):



User	Battery	Display	Feature Vector
Aman	0.9	0.1	[0.9 0.1]
Bob	0.8	0.2	[0.8 0.2]
Chandan	0.1	0.9	[0.1 0.9]
David	0.3	0.7	[0.01 0.99]

Now we will record the interactions of the users. How the users interacts with a product, whether he/she likes it or not. How much rating will a user give to a product? These type of interactions are recorded and put into a matrix called as User-Behaviour Matrix. Consider the following User-Behaviour matrix for this example:

User- Behaviour Matrix:



Smartphone	Aman	Bob	Chandan	David
S1	5	4.5	?	?
S2	5	?	0.5	?
S3	?	4	0.5	?
S4	?	?	5	4
S5	?	?	5	4.5

Where, the values of the behaviour matrix can be described as:

Bi,j = {p , ifUj has given “p” rating to a S_i

?, if no rating is given}

This matrix is used to compute the ratings of the unknown products. We can find the similarity between the users and with User-Feature matrix and User-Behaviour matrix we can calculate the feature ratings of unknown products.

To calculate the features of S_1 using the User-Behaviour matrix is shown below:

U_1 has rated S_1 5.

U_2 has rated S_1 4.5.

U_3 and U_4 has not rated S_1

Since we have two features Battery and Display for S_1 it can be vectorised as:

S_1: [x_1 x_2]

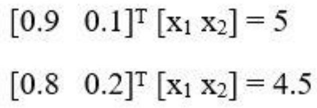
Where, x_1 is battery feature and x_2 is display feature.

This vector can be computed by the following equation:

$U_1^TS_1 = 5$

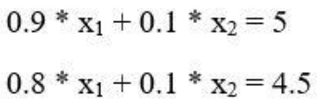
$U_2^TS_1 = 4.5$

Substituting the values of U and S we get,



$[0.9 \quad 0.1]^T \, [x_1 \; x_2] = 5$

$[0.8 \quad 0.2]^T \, [x_1 \; x_2] = 4.5$



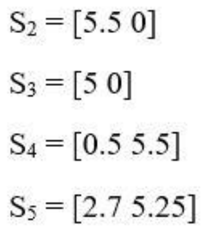
$0.9 * x_1 + 0.1 * x_2 = 5$

$0.8 * x_1 + 0.1 * x_2 = 4.5$

Now we have two equations and two unknowns, we can solve them by substitution or elimination. On solving these two equations, we have $x_1 = 5.5$ and $x_2 = 0.5$. This implies that for mobile phone S_1, its battery feature is rated as 5.5 and display feature is rated as 0.5.

$S_1 = [5.5 \; 0.5]$

Similarly, we can compute the feature ratings for all the mobile phones. The result as per the calculation is given:



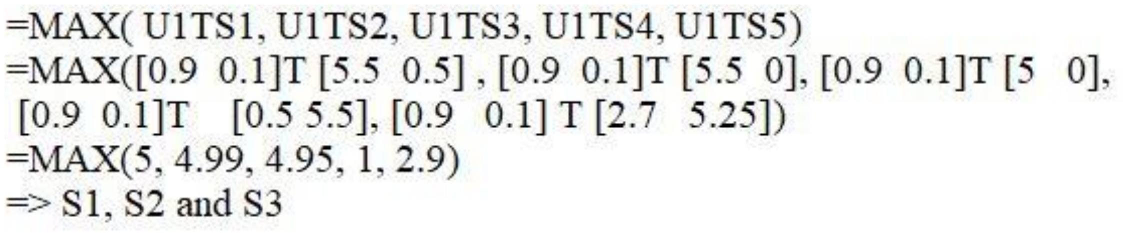
$S_2 = [5.5 \; 0]$

$S_3 = [5 \; 0]$

$S_4 = [0.5 \; 5.5]$

$S_5 = [2.7 \; 5.25]$

After calculating all the feature vectors of all the mobile phones we will calculate the recommendations as we have calculated in content based recommendation algorithm. Now we will map the user feature and item feature as we did in content based filtering method. Calculation for U_1 (Aman), the product recommendation will be:



=MAX(U1TS1, U1TS2, U1TS3, U1TS4, U1TS5)
=MAX([0.9 0.1]T [5.5 0.5] , [0.9 0.1]T [5.5 0], [0.9 0.1]T [5 0], [0.9 0.1]T [0.5 5.5], [0.9 0.1] T [2.7 5.25])
=MAX(5, 4.99, 4.95, 1, 2.9)
=> S1, S2 and S3

Result is S1, S2 and S3 again since S1 and S2 are already rated by Aman, mobile phone S3 is new for Aman so, we will recommend him.

Here for simplicity we have used only two features of the mobile phone i.e. display feature and battery feature. But in real life features may be very much diverse. For N number of features our feature vector for mobile phone S1 will look like:

S_1: [x_1 x_2 x_3 x_4 x_5 … x_N]

CHAPTER 4
ALGORITHMS

1. Content-based Methods:

- Uses attributes of items/users
- Recommend items similar to those liked by user in past

2. Collaborative Filtering Methods:

- Recommend items liked by similar users
- Enable exploration of diverse content

3. K Nearest Neighbours:

In the domain of pattern recognition nearest neighbour rule is one of the pioneer algorithm. It is quite pragmatic and quite obvious and no requirement is there to actually define it as an algorithm. For every programme that would be coded in any language would bear same result. It is therefore quite apt as a standard benchmark in studies involving comparison.

The nearest neighbour rule

This method of classification is exceptional. Vanilla version contains all sorts of examples. Practising it requires no use of training and therefore no density information is used. Its principle is entirely dependent on the examples provided and a distance measure which is define by user. The classification is derived from constant comparisons with whatever that is stored in storage spaces.

Since no probabilistic data is utilized in the 1-nn rule, as opposed to every other classifier, including the K-NN rule, it has the enormous favourable position that the preparation set ought not to be fundamentally founded on a specific testing. the educator, i.e. the application master who is in charge of structuring the classifier, is permitted to utilize his insight for finding a decent arrangement of models that speaks to the area of intrigue and not the likelihood thickness work. Learning of the area is typically preferable accessible over probabilistic information and is hard to be utilized something else.

There some significant steps that are to be followed. First of all, we are supposed to have a proper distance measure. What would actually happen during the classification process is determined by the instances that have been chosen and the distance measure technique is used.

Generally, we use nearest neighbouring technique as a base or initial procedure in compassion of datasets and there benchmarking. Being an untrained technique it used as a benchmark. If any untrained set if underperforms this benchmark is generally being considered as useless. All the hard work put in training such a training set is vain. Being very instinctive it is at times is much underrated or being overlooked. an example of this can be cited when neural networks got prominence than there were many ways that were worked on but they produced some very complex than nearest neighbour but the results generated by these ways were less accurate despite using complex training algorithms and computation power.

Being very simple is one of the reason that is most used algorithm. The algorithm basically doesn't consider any parameters and quite lazy in respect of complexity. Talking about the working principle of this algorithm it works upon databases which classified into various classes (data points) and predicts the classification for new entry or sample point.

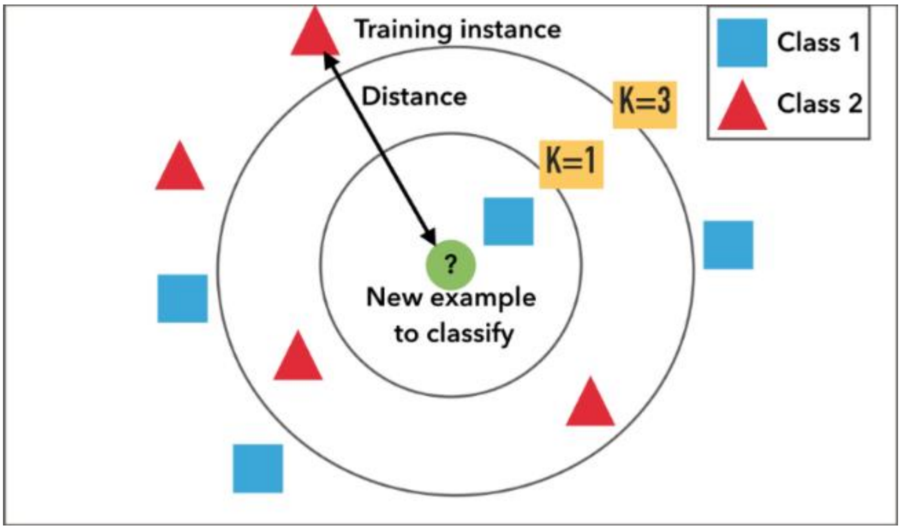
What does exactly the term non parametric means?

Non parametric means while classifying data it doesn't take into account any assumptions. What it means is what is the structure of the model is determined by characteristics of data. Now it seems pretty reasonable as real world doesn't obey theoretical concepts or assumptions. Therefore, it is considered as one of the first choices for classification of data when one has no prior knowledge about how to classify data.

Why lazy?

K nearest neighbour is also considered as a lazy algorithm but why? Does lazy imply doing nothing. It means not using training data points to do any kind of generalization. The using of training phasors is very minimal which leads to the training phase being quite fast. Not generalising means KNN keeps all the training data into account. To make it easy, during testing phase all the training data is needed.

Feature similarity is the principle on which KNN algorithm is based. Out of sample features resemblance with training set determines how to classify a given data point.



KNN’s one of the application is that it can be used as a classifier. That is prediction a discreet value for a class. Majority of the votes of the neighbour is used as a parameter classification, with object given the class which is most common among its nearest neighbours.
Regression is one of its application which tells us the value for object this value could be an average.

What are the applications of KNN?

- Loan predictor

Is bank going to approve my loan or not? Bank takes in to account various parameters while approving loan to an entity. Now looking on previous data that has all the parameters involved in providing loan to an entity. Using the previous data, we predict the class for the new object whether loan approved or not.

- Credit Ratings

Matching financial characteristics and people with same financial features to a database assigned. Objects having same financial points are given similar credit ratings. Now use this table to predict credit ratings of new objects.

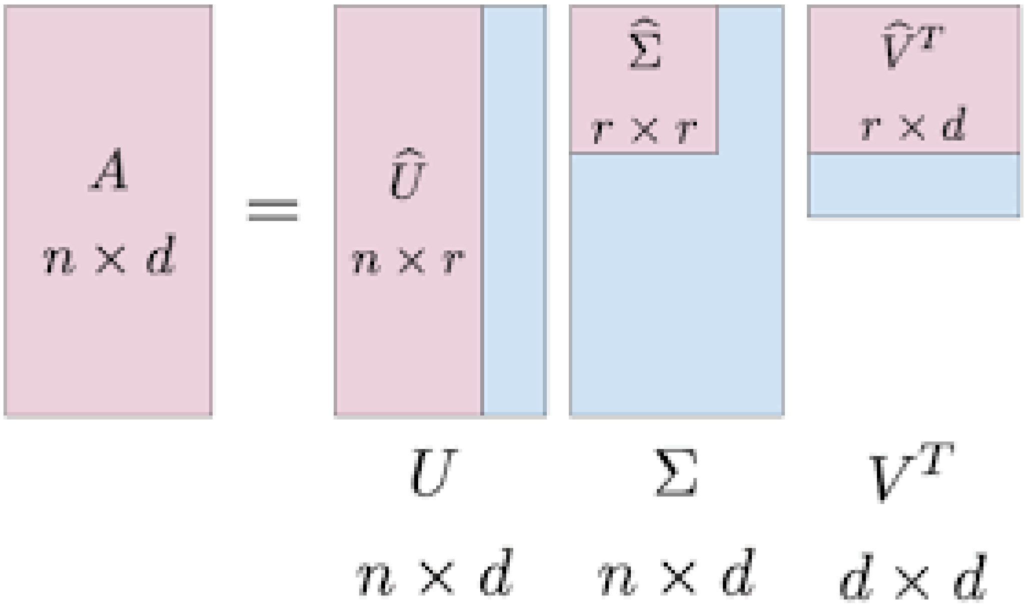
- Predicting Votes

Check if the potential voter will vote or not based upon the parameters asked to the voter.
Also the database can be used to predict the class of the vote that has been cited by the voter.

- Recommender Systems

Recording the responses of user and then dividing into class and based upon his classification you can predict what would be the class that may be desired by the user for next time.

4. Singular Value Decomposition:



Singular Value Decomposition is a mathematical tool in which matrix is decomposed into three parts. This technique is used widely in statistical modelling, data science, machine learning applications and many other mathematical fields. Recent developments in the field of computer science and machine learning have employed SVD and this mathematical technique is now used more extensively than ever.

Matrix Factorization: In simple terms SVD is representing a matrix as the product of three matrices.

Mathematics behind SVD
Consider a matrix M of the dimension m x n, then there existsa singular value decomposition of (SVD) of matrix M such that:

$$M=U\Sigma V^*$$

Where,
U represents the singular matrix (left singular vector of dimension m x m)

Σ represents the square matrix with non-negative real numbers (of dimension m x n).

V represents the singular matrix (right singular vector of dimension n x n)

V* represents the conjugate transpose of V (of dimension n x n).

The singular values of M are the diagonal values of Σ.

Conjugate Transpose: Interchanging row and column index.

Identity matrix: A square matrix in which all the diagonal values are 1 and other values are 0.

Diagonal Matrix: A matrix whose non diagonal elements are 0.

Unitary matrix: If UU*=I then matrix U is said to be unitary matrix.

Singular Values: In simpler terms roots of the Eigen values are called singular values.
Geometrical Representation:

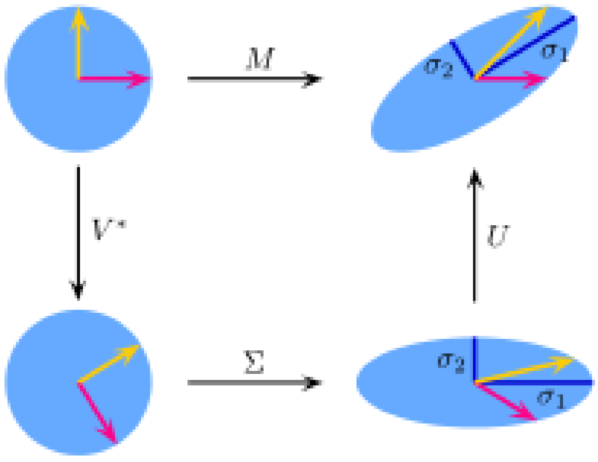
Consider any two vectors lying in a plane each start at the birthplace. Pivot the vectors by any total point and altogether such the vector with the bigger x worth consolidates a lower supreme y worth than the inverse. The two points can lie on the limit conic segment focused on the starting point.

Differ the edge of the turn and rehash the scaling till the scaled unit vectors are symmetrical.

At last turn the unit vectors by point β till they relate to organize hub and arrange pivot.

The over technique depicts is that the opposite of SVD.

The articulation UΣV* will be taken as an arrangement of 3 geometrical changes: a revolution, a scaling, and another turn. SVD is generally utilized for information pressure in fluctuated fields. Besides learning pressure the resultant networks has a few remarkable properties.

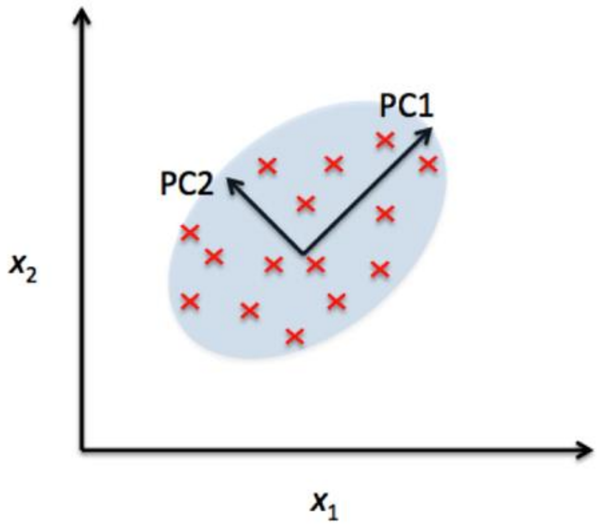


$$M = U \cdot \Sigma \cdot V^*$$

On the off chance that M is isosceles positive distinct its eigenvectors are symmetrical and that we will compose M= QΣQ*. This can be an extraordinary instance of a SVD, with U = V = Q

Essential Component Analysis (PCA):

Essential parts examination might be a methodology for trademark a little scope of disconnected factors, known as "key segments", from an outsized arrangement of information. The objective of chief parts investigation is to clear up the most extreme amount of fluctuation with the least scope of key parts.



Connection among PCA and SVD:

Give M a chance to be m × n network wherever m is that the scope of tests and n is that the scope of alternatives

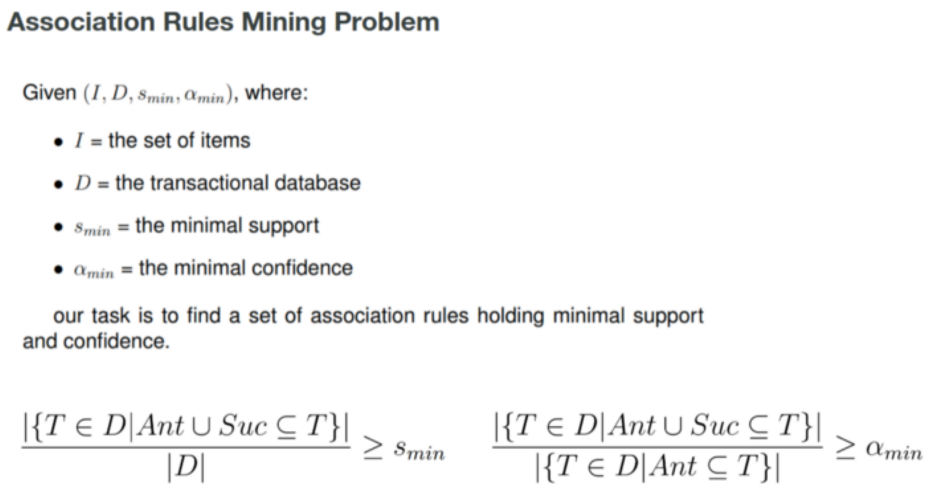
The difference network of M is C

Note: The over is right given that M is centred (scaled). Exclusively then is difference lattice fit M*M/(m−1). That is the reason we'd get a kick out of the chance to focus the underlying dataset M for PCA.

The correct particular vectors V are foremost bearings and Principal parts are given by $MV=U\Sigma V^*V=U\Sigma$. To downsize the spatiality of the information from m to k wherever k where $k<m$, select the principal k segments of $U\sigma$.

5 Association Rule:

Association rules may be used for recommendation. Things that area unit oftentimes consumed along area unit connected with a foothold within the graph. We will be able to see clusters of best sellers (densely connected things that just about everyone interacted with) and little separated clusters of niche content.



Association Rules Mining Problem

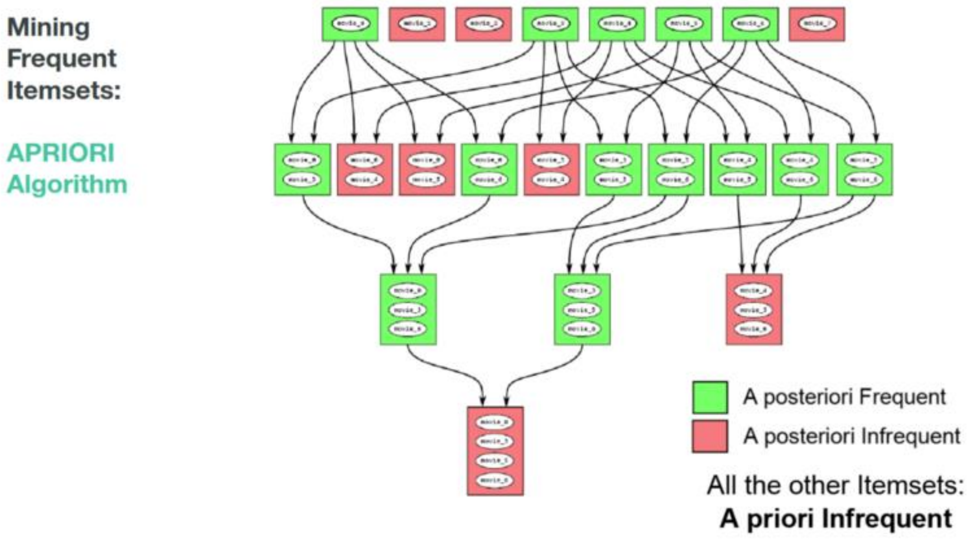
Given $(I, D, s_{min}, \alpha_{min})$, where:

- I = the set of items
- D = the transactional database
- s_{min} = the minimal support
- α_{min} = the minimal confidence

our task is to find a set of association rules holding minimal support and confidence.

$$\frac{|\{T \in D | Ant \cup Suc \subseteq T\}|}{|D|} \geq s_{min} \qquad \frac{|\{T \in D | Ant \cup Suc \subseteq T\}|}{|\{T \in D | Ant \subseteq T\}|} \geq \alpha_{min}$$

Rules well-mined from the interaction matrix ought to have a minimum of some lowest support and confidence. Support is expounded to the frequency of occurrence — implications of bestsellers have high support. High confidence implies that rules don't seem to be typically profaned.



Mining Frequent Itemsets:

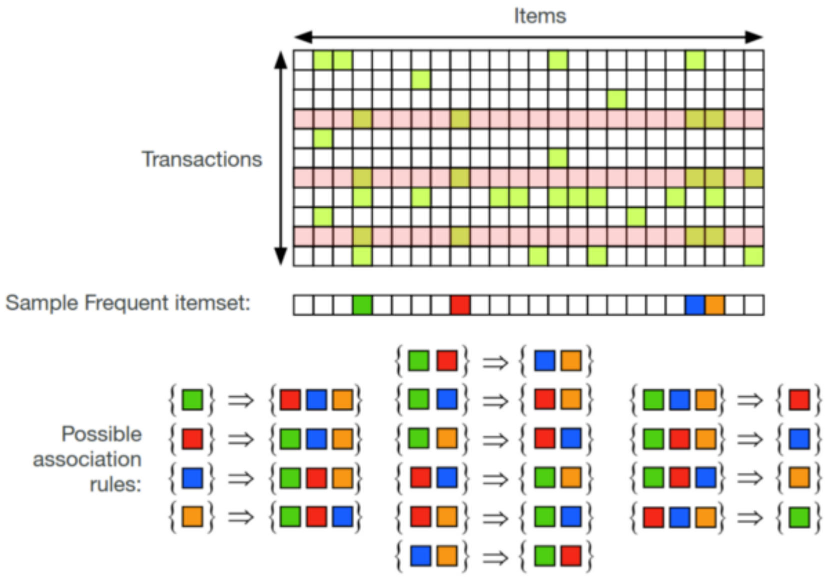
APRIORI Algorithm

🟩 A posteriori Frequent

🟥 A posteriori Infrequent

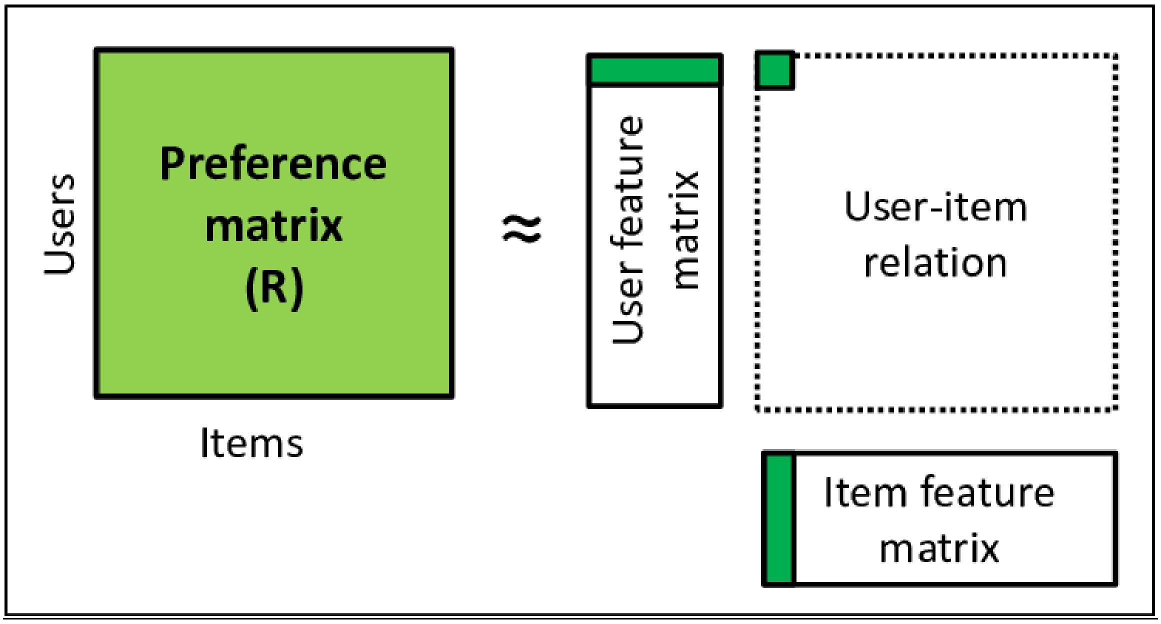
All the other Itemsets:
A priori Infrequent

Mining rules don't seem to be terribly ascendable. The APRIORI rule explores the state area of potential frequent itemsets and eliminates branches of the search area, that don't seem to be frequent.



Frequent itemsets area unit won’t to generate rules and these rules generate recommendations.

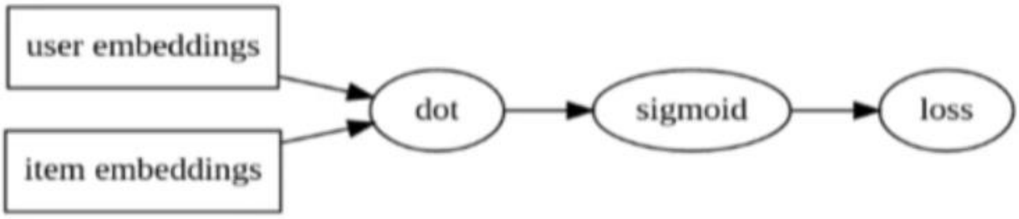
6 Factorization Model:



A resolving model takes in associate implicit or express interaction. We will use the implicit one for straightforward illustration.

$$I = \begin{pmatrix} 1.0 & 0.0 & \cdots & 1.0 \\ 0.0 & 1.0 & \cdots & 0.0 \\ \vdots & \vdots & \ddots & \vdots \\ 1.0 & 1.0 & \cdots & 1.0 \end{pmatrix}$$

Its plan is incredibly like SVD wherever users and things are mapped into a latent house so they're directly comparable. Primarily, we tend to use two embedding layers to represent users and things, severally. The target is that the interaction (utility matrix) that we tend to passed in. To reason the score for a user-item try, we tend to take the inner product of the latent illustration for that user and item, and spending it through a sigmoid activation perform. By computing the loss (more on it later) for all the user-item pairs with relation to verity interaction, we will back-propagate and optimize the embedding layers. The network structure is shown because the figure below.



We solely would like a couple of lines of code to coach such model with Spotlight, that appearance terribly like scikit-learn toolkit:

```
fromspotlight.factorization.implicit import ImplicitFactorizationModel
implicit_model = ImplicitFactorizationModel()
implicit_model.fit(implicit_interactions)
implicit_model.predict(user_ids, item_ids=None)
```

Gradient Descent

Gradient descent calculation is a standout amongst the most well known improvement calculations for finding ideal parameters for the model. Objective is to discover the parameter which limit the cost capacity.

Process

Neighbourhood gradient of the mistake work as for the parameter is estimated and it goes toward diminishing gradient. Cost capacity should diminish after each cycle. At the point when the diminishing in cost work is under 10^ (- 3), it shows that base point is drawing closer or exceptionally close.

Learning Rate

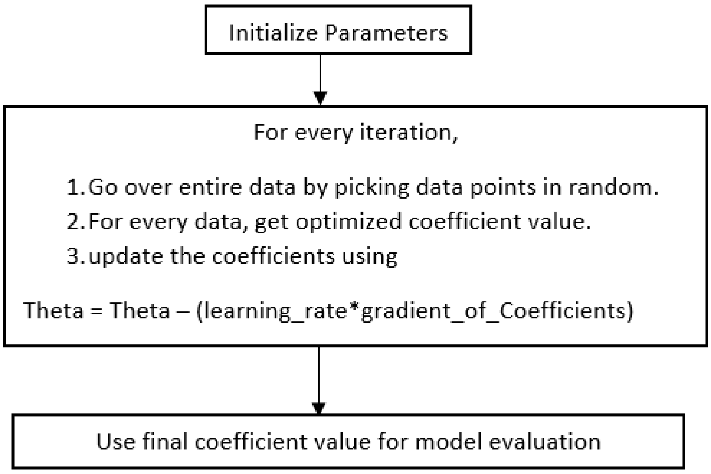
Learning rate is the one of the significant hyper-parameters that controls how enormous the means are amid gradient descent. Littler the learning rate, calculation makes littler strides during the time spent achieving worldwide least. In the event that the learning rate is expansive, it might overshoot least and neglect to meet. Now and again, it might even veer.

Feature Scaling

On the off chance that there are more than one element, at that point highlight scaling will help gradient descent to combine rapidly. Two component scaling strategies can be connected.

1. Normalization—This is utilized to change the distinctions among traits as far as recurrence of event, mean, range and fluctuation. Supplant x(i) with x(i)— Mean(x).
2. Standardization—This is finished by subtracting the mean and separating by standard deviation.

Stochastic Gradient Descent

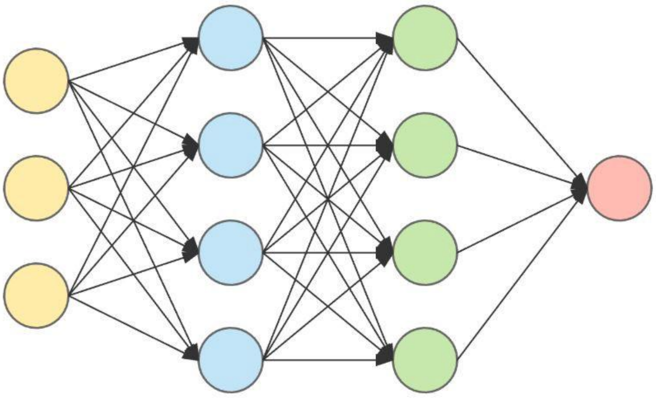

Initialize Parameters
For every iteration,
1. Go over entire data by picking data points in random.
2. For every data, get optimized coefficient value.
3. update the coefficients using
Theta = Theta – (learning_rate*gradient_of_Coefficients)
Use final coefficient value for model evaluation

Cost work go here and there. There is just steady reduction in the cost capacity on a normal.

This can help in leaving neighborhood least and odds of finding worldwide least isn't as flawless as group gradient strategy. Yet, by diminishing learning rate over the time can help achieving the ideal least point.

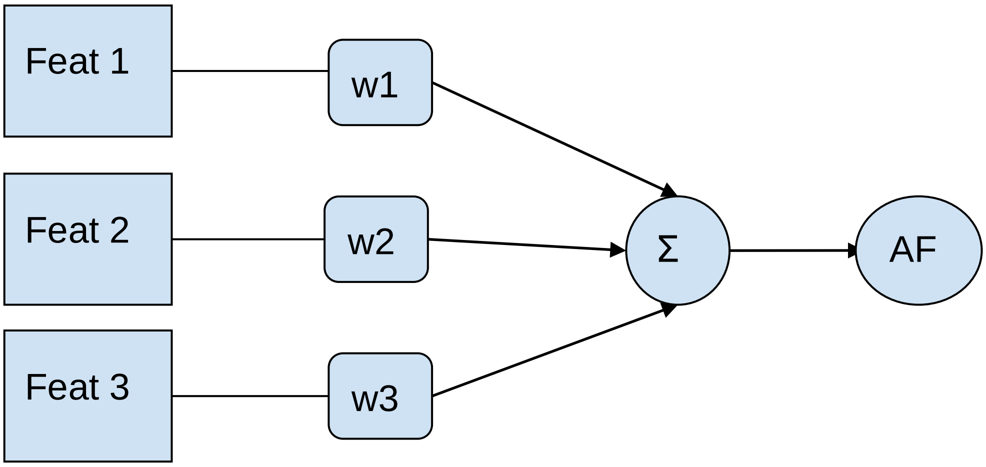
Procedure of changing the learning rate at each progression is called as recreated tempering. The capacity which decides the learning rate is called as learning plan.

7 Neural Networks and Sequential Model:



Multi-layered Neural Network

Neural Networks are the most widely used and powerful algorithms used in the field of artificial intelligence. Neural Networks emulates the working the human brain where one neuron is connected to thousands of neurons. Neural Networks emulates the working of hierarchical working of primary visual pathway of brain. Neural Networks consists of layered architecture which is used for predictions.



Img. structure of neural network.

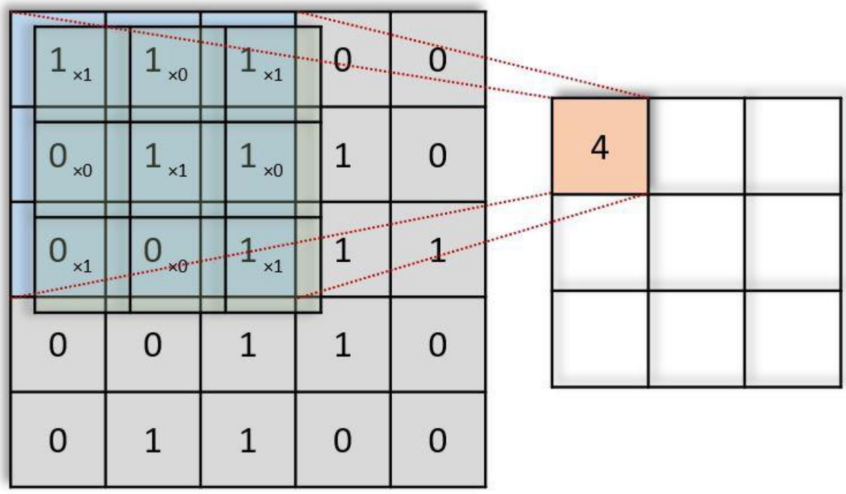
Neural Networks consists of two functions:

1. Feature extraction.
2. Classification.

Feature Extraction

The input to the feature extraction part is data that you want to classify and the output would be the features that would further be assigned to a classifier that would then be acted upon the weights to give the desired output.

In feature extraction we use filtres/kernels that would be convoluted on the input matrix that is made from the input data. Filters contains some values. Now filters are multiplied by the parts of the actual matrix to give us a feature map. This feature map contains values that we get after the multiplication of two matrices where the filter is moved upon various portions of input matrix which gives us values in feature maps. What filtering does is, it gives us the desired feature from the input data.



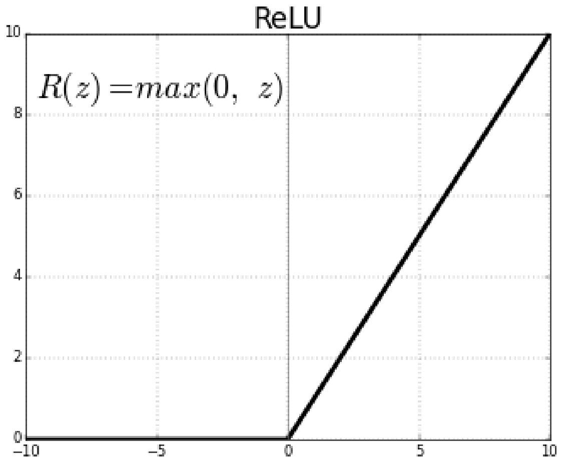
1 ×1	1 ×0	1 ×1	0	0
0 ×0	1 ×1	1 ×0	1	0
0 ×1	0 ×0	1 ×1	1	1
0	0	1	1	0
0	1	1	0	0

4		

Non Linearity in Feature Extraction

Neural networks consist of various layers which are linear combination of its inputs. We introduce non-linearity or an activation function allows us to classify our data even if it is not linearly separable.

Generally, in CNN feature extraction Linear Rectified Unit is used which basically convert all non-positive values to zero in our feature maps further enhancing the feature of an input based on filter.



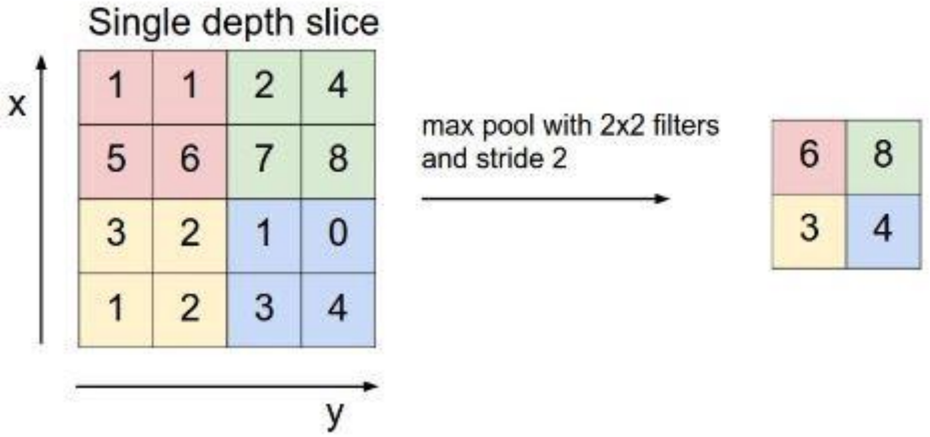
Pooling

Max pooling is an example based discretization process. The goal is to down-example an information portrayal (picture, shrouded layer yield framework, and so forth.), lessening its dimensionality and taking into account suspicions to be made about highlights contained in the sub-locales binned.

This is done to some extent to help over-fitting by giving a disconnected type of the portrayal. Too, it decreases the computational expense by diminishing the quantity of parameters to learn and gives fundamental interpretation invariance to the inner portrayal.

Max pooling is finished by applying a maximum channel to (for the most part) non-covering subregions of the underlying portrayal.

Let's say, as well, that we have a 2x2 filter that we'll run over our input. We'll have a stride of 2 (meaning the (dx, dy) for stepping over our input will be (2, 2)) and won't overlap regions.



1	1	2	4
5	6	7	8
3	2	1	0
1	2	3	4

6	8
3	4

Classification

After the extraction of feature maps these maps are input to a classifier which works in layers. Each layers has some neurons and each neuron is connected to other neurons in the next layer.

The connection between layers is assigned some weights which is multiplied with neuron values to give us a value of neuron in next layer.

Sequence Model

A sequent model treats the advice drawback as a sequent prediction drawback. Given the past interaction, we would like to grasp that item the user is presumably to love within the next time step. For example, assume user A has interaction with things whose ids are within the sequence of [2, 4, 17, 3, 5]. Then we'll have the subsequent increasing window prediction.

[2]->4
[2, 4]->17
[2, 4, 17]->3
[2, 4, 17, 3]-> 5

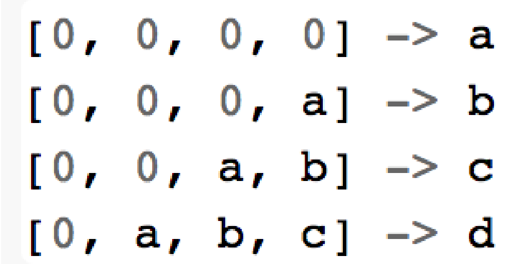
The array on the left stores the past interaction, whereas the whole number on the proper represents the item that user A can interacts next.

To train such model, we tend to merely flip the first interaction object into a sequent interaction object. The remaining are an equivalent.

```
fromspotlight.sequence.implicit import ImplicitSequenceModel
sequential_interaction = implicit_interactions.to_sequence()
implicit_sequence_model = ImplicitSequenceModel()
implicit_sequence_model.fit(sequential_interaction)
```

Note that to_sequence() perform pads zeros before of the sequence whose length isn't long enough to make sure that each sequence has an equivalent length.

Therefore, things with id zero got to be modified to alternative absolute unused id numbers so as for this perform to figure.



```
[0, 0, 0, 0] -> a
[0, 0, 0, a] -> b
[0, 0, a, b] -> c
[0, a, b, c] -> d
```

Choice of Loss function:

When specifying the model, we've the pliability to vary the loss perform. Model with completely different loss functions could have vital distinction within the performance. i will be able to concisely describe 2 main forms of loss perform outlined in Spotlight.

‘pointwise’: this is often the foremost easy kind of loss perform compared with others. because of the poorness of the sample (a heap of zero within the utility matrix), it's not computationally possible to require all the things into consideration. Thus, rather than computing loss for all the things given a user, we tend to solely think about a little of negative samples (item that user has not interacted with) haphazardly elect and every one the positive samples.
‘bpr’: Bayesian personalised Ranking (BPR) offers each item a rank for every of the user. It tries to create positive that rank for positive samples are above that of the negative samples with the subsequent formula.

$$L = 1.0 - \text{sigmoid}(\text{positive} - \text{negative})$$

Now, you've got learned the way to build a recommender system with Spotlight. It’s terribly easy to use with a good quantity of flexibility to meet would like. Although for a majority of the issues, the sequence models beat the resolving one, it takes a great deal longer to coach sequence models. Additionally, it might not be terribly useful if the info doesn't have a transparent sequent correlation once applying sequence model.

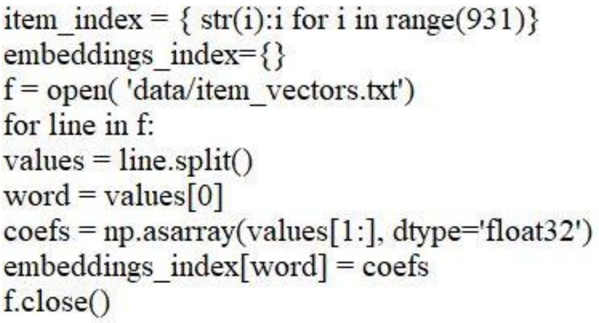
8 Item2Vec:

The rough plan behind Word2Vec is that we tend to leverage distributive illustration to cypher every word. Namely, every word is painted by a vector determined by alternative words close this word. We attempted to use distributive illustration to cypher every item supported the things that user interacted with before and when interacting with it.

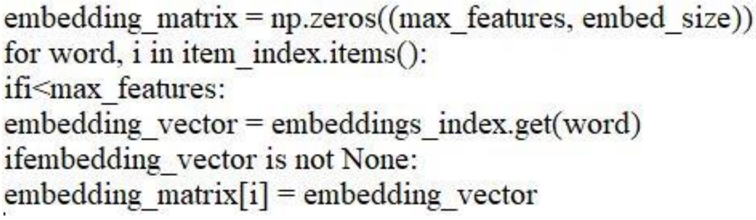
For each of the user, I 1st created associate item list in written record order. Then, Gensim’s Word2Vec model was trained on these item list. The trained item vectors were keep within the disk so we will load it for later use.

```
fromgensim.models import Word2Vec
model = Word2Vec(item_list, size=50, window=5, min_count=5, workers=10, sg=0)
model.wv.save_word2vec_format('data/item_vectors.txt')
```

After that, we tend to load the trained item vectors into associate embedding matrix.

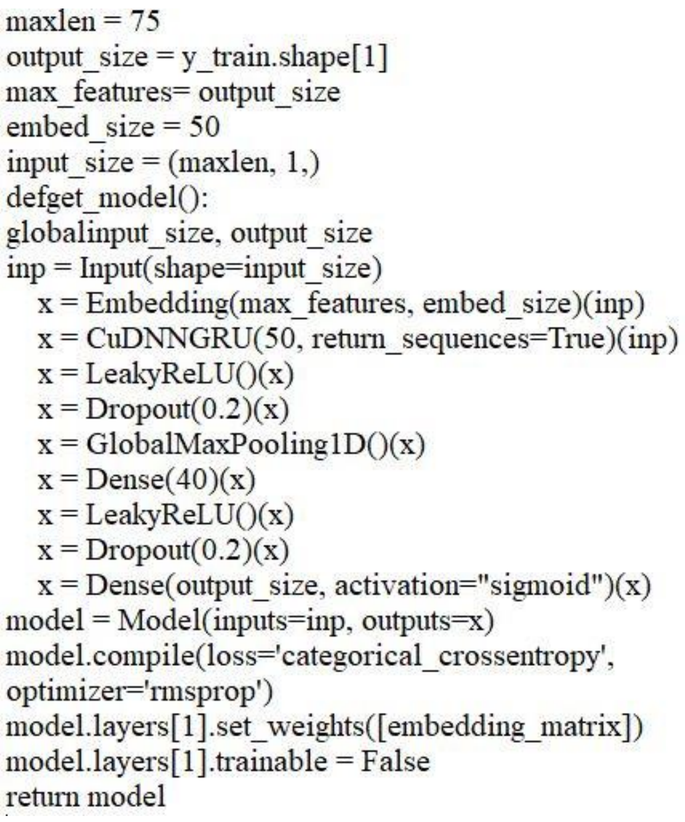


```
item_index = { str(i):i for i in range(931)}
embeddings_index={}
f = open( 'data/item_vectors.txt')
for line in f:
values = line.split()
word = values[0]
coefs = np.asarray(values[1:], dtype='float32')
embeddings_index[word] = coefs
f.close()
```



```
embedding_matrix = np.zeros((max_features, embed_size))
for word, i in item_index.items():
ifi<max_features:
embedding_vector = embeddings_index.get(word)
ifembedding_vector is not None:
embedding_matrix[i] = embedding_vector
```

We then outline our model for predicting the user’s future interaction. Basically, it's a GRU model accelerated by CuDNN. If an Nvidia GPU is not available then, don’t worry concerning it. You’ll be able to merely replace CuDNNGRU with GRU.



```
maxlen = 75
output_size = y_train.shape[1]
max_features= output_size
embed_size = 50
input_size = (maxlen, 1,)
defget_model():
globalinput_size, output_size
inp = Input(shape=input_size)
    x = Embedding(max_features, embed_size)(inp)
    x = CuDNNGRU(50, return_sequences=True)(inp)
    x = LeakyReLU()(x)
    x = Dropout(0.2)(x)
    x = GlobalMaxPooling1D()(x)
    x = Dense(40)(x)
    x = LeakyReLU()(x)
    x = Dropout(0.2)(x)
    x = Dense(output_size, activation="sigmoid")(x)
model = Model(inputs=inp, outputs=x)
model.compile(loss='categorical_crossentropy',
optimizer='rmsprop')
model.layers[1].set_weights([embedding_matrix])
model.layers[1].trainable = False
return model
```

Note that at we loaded the pre-trained embedding matrix into the embedding layer of the model and froze it by setting trainable to be false.

CHAPTER 5
TEST PLAN

The MovieLens 10M is used as dataset in our project. The MovieLens 10M dataset consists of 10,000,054 user ratings for 10681 shows and movies from 71,567 users. Each user has more than 20 ratings. The ratings for each movie are from 1 to 5. This dataset is randomly divided into 2 parts: the training set and the test set. For each user, the training set contains 90% of the user's ratings. The rest 10% ratings build up the test set. Collaborative filtering is trained based on the training set and algorithm evaluation is carried out based on the test set.

Recommendation systemsare currently very popular application of machine learning. In our project, based on the algorithms discussed we will build an recommender system and recommend the movies to the users and then compare the accuracy of the different models. We use the following abbreviations:

n = number of users

m = number of movies

p(i, j) = 1 if user j has rated movie i

y(i, j) = rating given by user j to movie i (defined only if r(i, j) = 1)

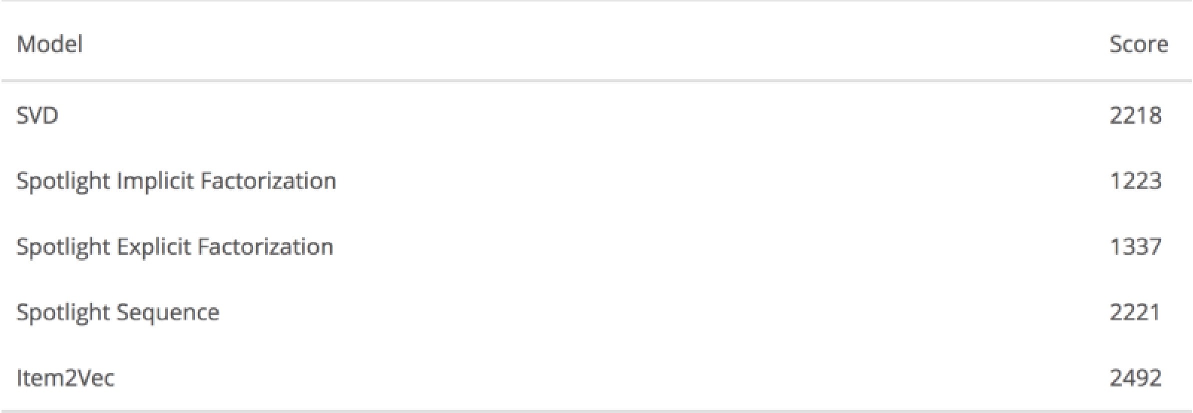
$\theta^{(j)}$ = parameter vector for user j

$x^{(i)}$ = feature vector for movie i

For user j, movie i, predicted rating: $(\theta^{(j)})^T (x^{(i)})$

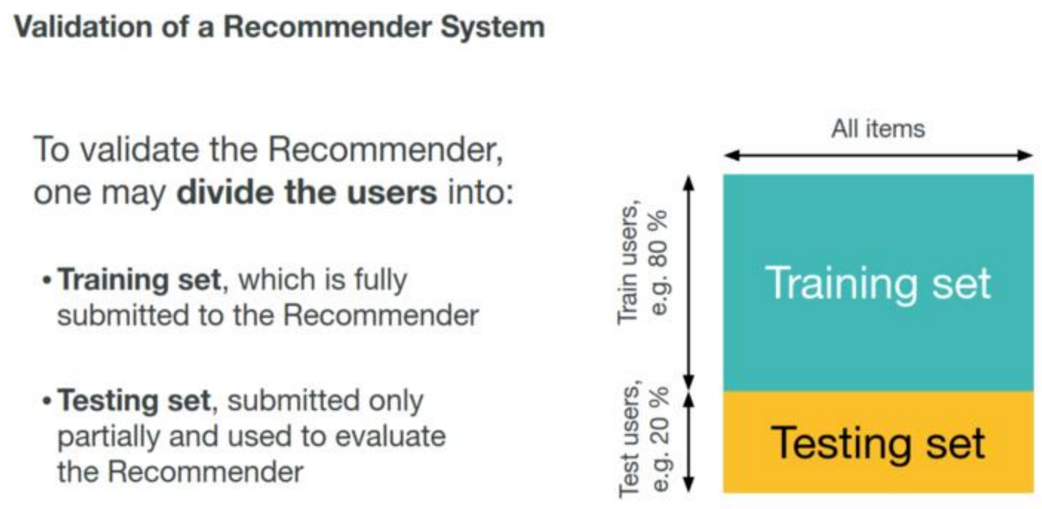
CHAPTER 6
RESULTS AND PERFORMANCE ANALYSIS

We have checked a number of the models. The general public board score for every of them is shown within the following table.



Model	Score
SVD	2218
Spotlight Implicit Factorization	1223
Spotlight Explicit Factorization	1337
Spotlight Sequence	2221
Item2Vec	2492

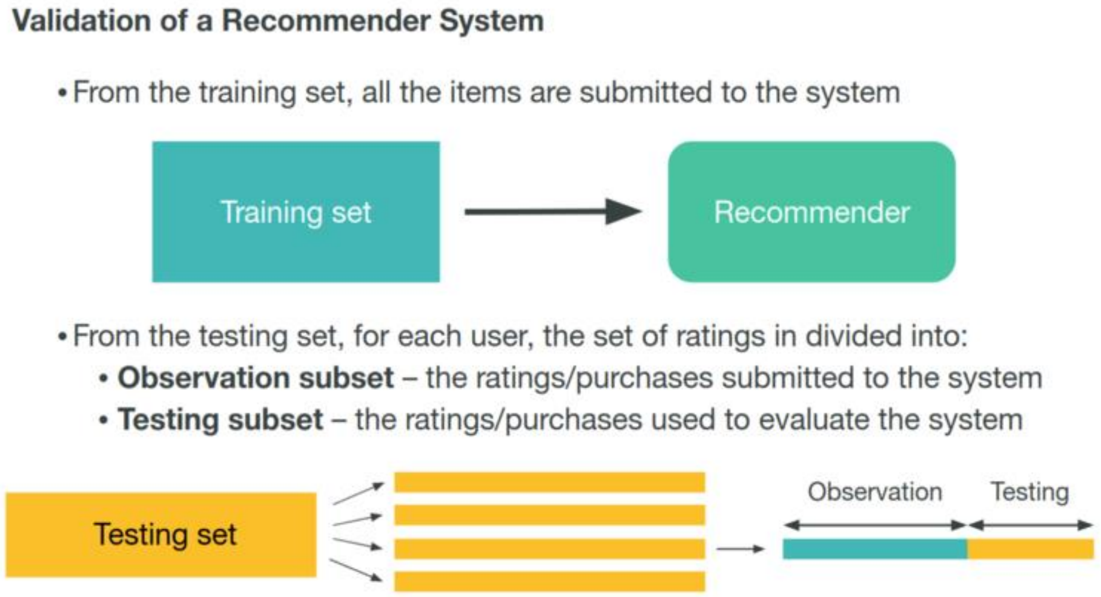
It looks that Neural Network-based models don't essentially beat ancient methodology for building a recommender system. Whereas being straightforward to know and implement, SVD’s score is on par with the Spotlight Sequence model that takes a far longer time to coach. Item2Vec, amazingly, is that the best model among all the models that I even have tested out. It is true that we tend to cannot choose of these models with just one check set. This provides you a rough plan of however sensible every model is.



Validation of a Recommender System

To validate the Recommender, one may **divide the users** into:

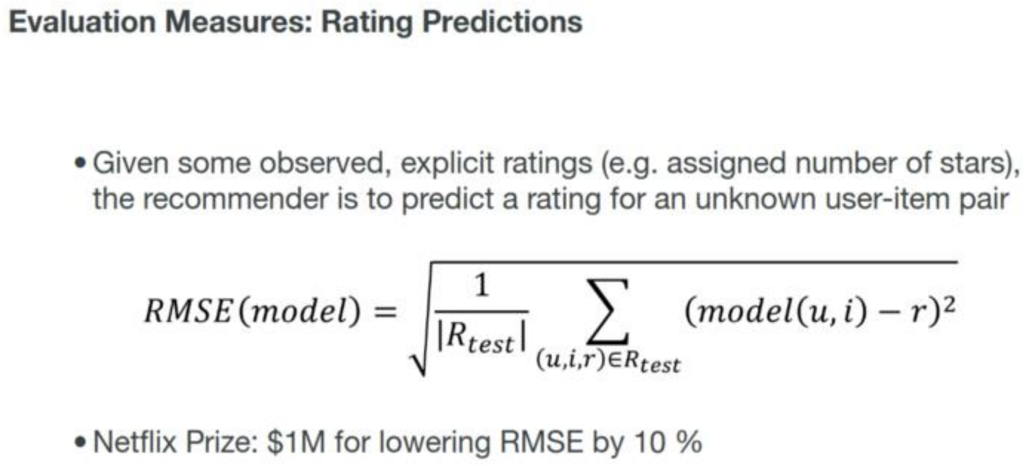
- **Training set**, which is fully submitted to the Recommender
- **Testing set**, submitted only partially and used to evaluate the Recommender



Validation of a Recommender System

- From the training set, all the items are submitted to the system

- From the testing set, for each user, the set of ratings in divided into:
  - **Observation subset** – the ratings/purchases submitted to the system
  - **Testing subset** – the ratings/purchases used to evaluate the system



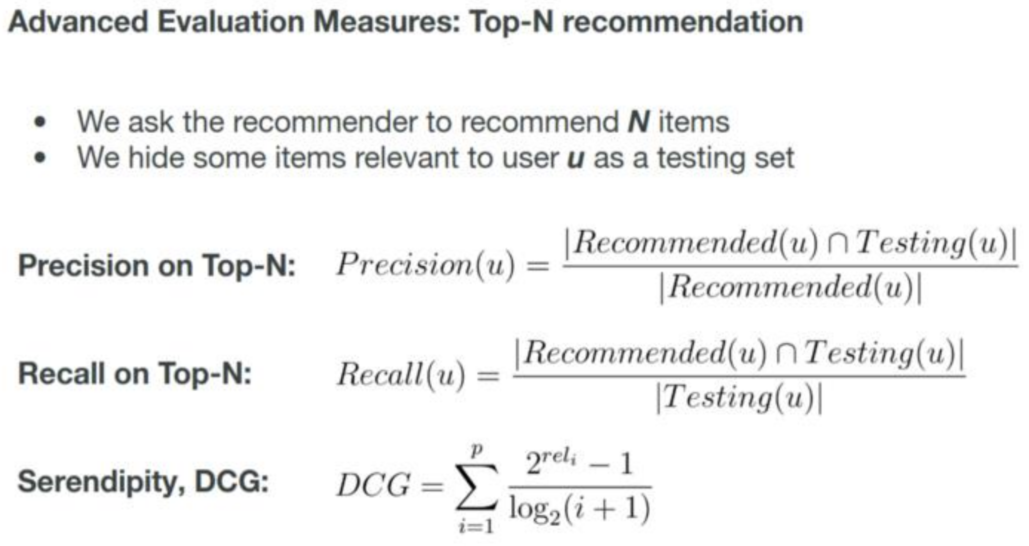
Evaluation Measures: Rating Predictions

- Given some observed, explicit ratings (e.g. assigned number of stars), the recommender is to predict a rating for an unknown user-item pair

$$RMSE(model) = \sqrt{\frac{1}{|R_{test}|} \sum_{(u,i,r)\in R_{test}} (model(u,i) - r)^2}$$

- Netflix Prize: $1M for lowering RMSE by 10 %

Root mean square error (RMSE) remains wide used despite several studies showed that RMSE is poor reckoner of on-line performance.



Advanced Evaluation Measures: Top-N recommendation

- We ask the recommender to recommend ***N*** items
- We hide some items relevant to user ***u*** as a testing set

Precision on Top-N: $$Precision(u) = \frac{|Recommended(u) \cap Testing(u)|}{|Recommended(u)|}$$

Recall on Top-N: $$Recall(u) = \frac{|Recommended(u) \cap Testing(u)|}{|Testing(u)|}$$

Serendipity, DCG: $$DCG = \sum_{i=1}^{p} \frac{2^{rel_i} - 1}{\log_2(i+1)}$$

Chapter 7
REFERENCES

[1] Bollacker, K.D., Lawrence, S., Giles, C.L.: CiteSeer: an autonomous web agent for automatic retrieval and identification of interesting publications. In: Proceedings of the 2nd international conference on Autonomous agents, pp. 116–123 (1998)

[2] Google Scholar, Scholar Update: Making New Connections, Google Scholar Blog. http://googlescholar.blogspot.de/2012/08/ scholar-updates-making-new-connections.html

[3] Airoldi, E.M., Blei, D.M., Fienberg, S.E., Xing, E.P., Jaakkola, T.:Mixed membership stochastic block models for relational data with application to protein–protein interactions. In: Proceedings of the International Biometrics Society AnnualMeeting, pp. 1–34 (2006)

[4] Arnold, A., Cohen, W.W.: Information extraction as link prediction: using curated citation networks to improve gene detection. In: Proceedings of the 4th International Conference on Wireless Algorithms, Systems, and Applications, pp. 541–550 (2009)

[5] Beel, J., Langer, S., Genzmehr, M.: Sponsored vs. Organic (Research Paper) Recommendations and the Impact of Labeling. In: Proceedings of the 17th International Conference on Theory and Practice of Digital Libraries (TPDL 2013), pp. 395–399 (2013)

[6] Beel, J., Langer, S., Genzmehr,M., Nürnberger, A.: Persistence in Recommender Systems: Giving the Same Recommendations to the Same Users Multiple Times. In: Proceedings of the 17th International Conference on Theory and Practice of Digital Libraries (TPDL 2013), vol. 8092, pp. 390–394 (2013)

[7] Beel, J., Langer, S., Genzmehr, M., Nürnberger, A.: Introducing Docear’s Research Paper Recommender System. In: Proceedings of the 13th ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL’13), pp. 459–460 (2013)

[8] Beel, J., Langer, S., Nürnberger, A., Genzmehr, M.: The Impact of Demographics (Age and Gender) and Other User Characteristics on Evaluating Recommender Systems. In: Proceedings of the 17th International Conference on Theory and Practice of Digital Libraries (TPDL 2013), pp. 400–404 (2013)

[9] Böhm, W., Geyer-schulz, A., Hahsler, M., Jahn, M.: Repeat-Buying Theory and Its Application for Recommender Services. In: Proceedings of the 25thAnnualConference of theGesellschaftfürKlassifikatione.V., pp. 229–239 (2003)

[10] Baez, M.,Mirylenka,D., Parra, C.:Understanding and supporting search for scholarly knowledge. In: Proceeding of the 7th European Computer Science Summit, pp. 1–8 (2011)

[11] Beel, J., Gipp, B., Langer, S., Genzmehr, M.: Docear: an academic literature suite for searching, organizing and creating academic literature. In: Proceedings of the 11th Annual International ACM/IEEE Joint Conference on Digital Libraries (JCDL), pp. 465–466 (2011)

[12] Beel, J., Gipp, B., Mueller, C.: SciPloreMindMapping'—a tool for creating mind maps combined with PDF and reference management. D-Lib Mag. 15(11) (2009)

[13] Bethard, S., Jurafsky, D.: Who should I cite: learning literature search models from citation behavior. In: Proceedings of the 19th ACM international conference on Information and knowledge management, pp. 609–618 (2010)

[14] Bogers, T., van den Bosch, A.: Recommending scientific articles using citeulike. In: Proceedings of the 2008 ACM conference on Recommender systems, pp. 287–290 (2008)

[15] Bollen, J., Van de Sompel, H.: An architecture for the aggregation and analysis of scholarly usage data. In: Proceedings of the 6th ACM/IEEE-CS joint conference on Digital libraries, pp. 298–307 (2006)

[16] CiteSeerX, T.: About RefSeer. http://refseer.ist.psu.edu/about (2012)CiteULike: My Top Recommendations. Website http://www.citeulike.org/profile/username/recommendations (2011)

[17] CiteULike: Science papers that interest you. Blog. http://blog.citeulike.org/?p=11 (2009)

[18] CiteULike: Data from CiteULike's new article recommender. Blog, http://blog.citeulike.org/?p=136 (2009)

[19] Caragea, C., Silvescu, A., Mitra, P., Giles, C.L.: Can't See the Forest for the Trees? A Citation Recommendation System. In: iConference 2013 Proceedings, pp. 849–851 (2013)

[20] Chandrasekaran, K., Gauch, S., Lakkaraju, P., Luong, H.: Concept-based document recommendations for citeseer authors. In: Proceedings of the 5th international conference on Adaptive Hypermedia andAdaptiveWeb-Based Systems, pp. 83–92 (2008)

[21] Choochaiwattana,W.: Usage of tagging for research paper recommendation. In: Proceedings of the 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE), vol. 2, pp. 439–442 (2010)

[22] Councill, I., Giles, C., Di Iorio, E., Gori, M., Maggini, M., Pucci, A.: Towards next generation CiteSeer: a flexible architecture for digital library deployment. In: Research and Advanced Technology for Digital Libraries, pp. 111–122 (2006)

[23] Dong, R., Tokarchuk, L.,Ma, A.: Digging Friendship: Paper Recommendation in Social Network. In: Proceedings of Networking and Electronic Commerce Research Conference (NAEC 2009), pp. 21–28 (2009)

[24] ExLibris: bX Usage-Based Services transform your discovery experience!, Web page, http://www.exlibrisgroup.com/category/bXUsageBasedServices (2013)

[25] Ekstrand, M.D., Kannan, P., Stemper, J.A., Butler, J.T., Konstan, J.A., Riedl, J.T.: Automatically building research reading lists. In: Proceedings of the 4th ACM conference on Recommender systems, pp. 159–166 (2010)

[26] Erosheva, E., Fienberg, S., Lafferty, J.: Mixed-membership models of scientific publications. Proc. Natl. Acad. Sci. U. S. Am. 101(Suppl 1), 5220–5227 (2004)

[27] Franke, M., Geyer-Schulz, A.: Using restricted random walks for library recommendations and knowledge space exploration. Int. J. Pattern Recognit. Artif. Intell. 21(02), 355–373 (2007)

[28] Ferrara, F., Pudota, N., Tasso, C.: A Keyphrase-Based Paper Recommender System. In: Proceedings of the IRCDL'11, pp. 14–25 (2011)

[29] Geyer-Schulz, A., Hahsler, M.: Comparing two recommender algorithms with the help of recommendations by peers. In: Proceedings of the WEBKDD 2002—Mining Web Data for Discovering Usage Patterns and Profiles, pp. 137–158 (2003)

[30] Geyer-Schulz, A., Hahsler, M.: Evaluation of recommender algorithms for an internet information broker based on simple association rules and on the repeat-buying theory. In: Proceedings of the 4th WebKDD Workshop: Web Mining for Usage Patterns and User Profiles, pp. 100–114 (2002)

[31] Geyer-Schulz, A., Hahsler, M., Jahn, M.: A customer purchase incidence model applied to recommender services. In: Proceedings of the 3rd InternationalWorkshop on MiningWeb Log Data Across All Customers Touch Points, pp. 25–47 (2002)

[32] Geyer-Schulz, A., Hahsler, M., Jahn, M.: Recommendations for virtual universities from observed user behavior. In: Proceedings of the 24th Annual Conference of the GesellschaftfürKlassifikatione.V., pp. 273–280 (2002)

[33] Geyer-Schulz, A., Hahsler, M., Jahn, M., Geyer, A.: WissenschaftlicheRecommendersysteme in VirtuellenUniversitäten. In: Proceedings of the Symposiom of UnternehmenHochschule, pp. 101–114 (2001)

[34] Geyer-Schulz, A., Hahsler,M., Neumann, A., Thede, A.: An integration strategy for distributed recommender services in legacy library systems. In: Between Data Science and Applied Data Analysis. Springer, pp. 412–420 (2003)

[35] Geyer-Schulz, A., Hahsler, M., Neumann, A., Thede, A.: Behavior-based recommender systems as value-added services for scientific libraries. Statistical Data Mining and Knowledge Discovery, pp. 433–454 (2003)

[36] Geyer-Schulz, A., Hahsler, M., Thede, A.: Comparing Simple Association-Rules and Repeat-Buying Based Recommender Systems in a B2B Environment. In: Proceedings of the 26th Annual Conference of the GesellschaftfürKlassifikatione.V., pp. 421–429 (2003)

[37] Geyer-Schulz, A., Neumann, A., Thede, A.: An architecture for behavior-based library recommender systems. Inf. Technol. Libr. 22(4), 165–174 (2003)

[38] Geyer-Schulz, A., Neumann, A., Thede, A.: Others also use: a robust recommender system for scientific libraries. In: Proceedings of the 7th European Conference on Digital Libraries, pp. 113–125 (2003)

[39] Gillitzer, B.: Der EmpfehlungsdienstBibTip - EinflächendeckendesAngebotimBibliotheksverbund Bayern. http://www.b-i-t-online.de/heft/2010-01/nachrichtenbeitrag3. pp. 1–4 (2010)

[40] Gottwald, S.: Recommender Systemefuer den Einsatz in Bibliotheken/ Survey on recommender systems. Konrad-Zuse-ZentrumfürInformationstechnik Berlin, ZIB-Report 11–30 (2011)

[41] Geyer-Schulz, A., Hahsler, M., Jahn, M.: Educational and scientific recommender systems: designing the information channels of the virtual university. Int. J. Eng. Educ. 17(2), 153–163 (2001)