UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE ENGENHARIA DE COMPUTAÇÃO

MATHEUS TOAZZA TURA

From Timeout-based to Item-by-Item Analysis: Investigating Methodologies for Splitting User Sessions Originated from Shared Accounts in Online Platforms

Work presented in partial fulfillment of the requirements for the degree of Bachelor in Computer Engineering

Advisor: Prof. Dr. Weverton Luis da Costa Cordeiro
Coadvisor: Prof. Dr. Renata de Matos Galante

Porto Alegre
May 2021

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
Reitor: Prof. Carlos André Bulhões Mendes
Vice-Reitora: Profª. Patricia Helena Lucas Pranke
Pró-Reitora de Ensino: Profª. Cíntia Inês Boll
Diretora do Instituto de Informática: Profª. Carla Maria Dal Sasso Freitas
Coordenador do Curso de Engenharia de Computação: Prof. Walter Fetter Lages
Bibliotecária-chefe do Instituto de Informática: Beatriz Regina Bastos Haro

ACKNOWLEDGEMENTS

First and foremost, I would like to dedicate this work to my parents Genir and Lurdes that supported me during my development as graduate student, and to my friend Dennis Balreira that helped me on the most difficult storms from hard and demanding disciplines and also to my homeless friend Paulo, that used to encourage me to finish my degree every time I used to see him on streets.

I am deeply grateful to my advisor, Weverton Cordeiro for his profissionalism and clarity on the way he guided me trough and also to my co-advisor Renata Galante that faded any misdirection with wisdom.

Special thanks to Carolina Nery that helped me with discussions and insights during the construction of this work.

“Nothing in life is to be feared, it is only to be understood.
Now is the time to understand more, so that we may fear less.”

— MARIE CURIE

ABSTRACT

Although some content providers register stream data from its users and can track their profile style for content recommendation, when two or more users share a same account, their true profile activity is obfuscated and fuzzed. This user behavior hinders the recommender systems from providers, moreover, the growing concerns on user privacy poses a risk to current models that rely on unconcealed user identity. This work proposes a way of classifying users' stream data trough sessions, based only on its media content, opening the possibility for breaking a same account profile within multiple user profiles and consequently identifying this activity. In this work dimensionality reduction and clustering methods are used to classify user stream data into sessions that correspond to each respective user profile. Experiments show that the event-driven nature of news content can challenge the construction of a session splitting method based exclusively on content-type without user profiling.

Keywords: Dimensionality reduction. clustering. user profiling. session identification. recommender systems.

RESUMO

Embora as provedoras de conteúdos registram dados de acessos de seus usuários e consigam analisar seus perfis para recomendações de conteúdo, quando duas ou mais pessoas compartilham da mesma conta a atividade e perfil original e individual de cada usuário é obfuscada e difusa por essas contas compartilhadas. Este comportamento confunde os sistemas de recomendação existentes, além disso, o aumento da preocupação com a privacidade dos usuários coloca em risco os modelos atuais que são dependentes de reconhecimento explícito dos usuários. Este trabalho propõe uma maneira de classificar o fluxo de dados dos usuários em sessões baseando-se apenas em seu conteúdo, abrindo portas para quebrar a mesma conta em múltiplos perfis de usuários e consequentemente identificando esta atividade. Neste trabalho técnicas de redução de dimensionalidade e métodos de clusterização são utilizados para classificar o fluxo de dados em sessões que correspondem respectivamente a cada perfil de usuário. Experimentos mostram que a natureza guiada a eventos dos conteúdos de notícias tornam desafiador a construção de um método de quebra de sessões exclusivamente baseado em categorização de conteúdo sem perfilização de usuário.

Palavras-chave: Redução de dimensionalidade, Clusterização, Perfis de usuários, Identificação de sessões, Sistemas de recomendações.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

PCA	Principal Component Analysis
SNE	Stochastic Neighbor Embedding
SVD	Singular Value Decomposition
tSNE	t-Distributed Stochastic Neighbor Embedding
EM	Expectation-Maximization Clustering
RNN	Recurrent Neural Networks
ALS	Alternating Least Squares
CF	Colaborative Filtering
CB	Content-based Filtering

CONTENTS

1 INTRODUCTION

Most platforms nowadays rely on recommender systems to create plug and play interactions between users and offered items, avoiding search time and making platform navigation minimal and crisp, however, some user behaviours such as account sharing or even the emerging privacy laws and growing concerns on data privacy that could hide user behaviour, recommender systems precision can get affected by profiling contamination on the user-item relations and its interpretation by the system.

One commonly known example responsible for grinding the recommender systems performance is the practice of account sharing that mixes and masks its account profiles against its real users. Account sharing behaviour is non-intentional in most cases, since device sharing is common, there could be a situation where some individual into his computer is searching videos in a logged-in streaming platform for specific subjects like for example C++ usage, but at some point, it leaves the computer empty without performing account logout, another individual then starts using the same device and platform but for a completely different subject, like children's cartoon.

This problem is evident on commonly used time-based session identification techniques such as Halfaker et al. (2015), the presented scenario would fail in identifying sessions, if chosen period threshold for breaking sessions is long enough, the elapsed time from user permutation usage on platform could be insignificant on the total session time, thus preserving the unity of the session, but in reality two completely sessions have happened during this period.

On Jiang et al. (2018) its main goal is identifying shared accounts in multimedia streaming and it addresses a clearly domain separation between *account-level recommendation* and *user-level recommendation*, but even so, they use a timeout along with their other techniques to break the session.

The proposed work consists in two global objectives: (i) Verifying the possibility of an algorithm for session identification based on item category instead of timeout, (ii) Create an evaluation method for reliably testing its accuracy and precision.

In order to execute those global objectives, *globo.com* dataset was preferred instead of others due to it's availability and data abundance, however, since they do not allow the use of news content, Moreira (2018) made available its embeddings that carries 250 dimensions of features for each of more than 300,000 articles, also *globo.com* dataset aggregates streams of user clicks with more than 2.8 millions of clicks with labeled information such as user_id and timestamp; the attribute of stream click is addressed at section 5.1. Another dataset for that could be used as subject of matter is the *lastFM millions song dataset* of Bertin-Mahieux et al. (2011) Dimensionality reduction technique was needed in order to reduce complexity of data and apply the $AffinityPropagation$ clustering algorithm.

This work is organized as follow: Initially on chapter 2 basic concepts and methods needed to fulfil the following reasoning are introduced, later related works on chapter 3 that backed the choices made on this project. On the chapter 4 the session problem modelling is introduced and described as well as known problems for multidimensional data and its manipulation. The section 4.3 that follows introduce solutions for the defined problems. Finally, on chapter 5 error measurement modeling is addressed along with it metrics results.

2 THEORETICAL BACKGROUND

Here you will find the foundations for this work, they will be introduced, those are crucial to assemble the complete picture.

At subsection 2.1, basic concepts and attributes are introduced and on subsection 2.3 and 2.4 methods and algorithms are introduced with its applicability and past usage. Related works are presented on subsection 3

2.1 Basic concepts

- $User$ is the person itself that is using the platform trough an account, most of times each user has its own account, but at sometimes, one account can have multiple users.
- $Account$ is a platform gateway and record that contains user detailed information and is used to register user history associating each requested item with its requester account id.
- $Item$ represents the consumed media, can be music, image, video or text in case of a news article.
- $Clickstream$ is a group of requested items ordered by request timestamp.
- $Session$ is a subset of a $clickstream$, the beginning and end of a session are commonly tagged considering user idleness, where the time difference between one requested item and the next one is higher than a certain period threshold, however, the proposed method is timeless and take in account only content categories as a session split threshold.

An account is the common gateway for an user to access the platform content, most of past works consider that each account is in fact one user which may happen on most scenarios because normally each user has its own account, but sometimes and even more on paid platforms, an account is shared between more users.

An item is a pair with a click timestamp and a media id identification. At Globo dataset, each item is a set $(timestamp, click\_id)$ but in any other dataset, the click id is a reference to news, music, image, video or any metadata associated. A session is a well defined segment from a stream of items from one account. There are many ways of defining session boundaries, on Arlitt (2000) the session is defined by using time threshold, same as Halfaker et al. (2015). For example when a user stays inactive for an amount of time, that can be considered an end of a session and start of another. In this proposal the sessions will be split considering the content type in order to identify account sharing or even user profiling for different user tastes.

2.2 Recommender system

The role of a *recommender system* is to make inferences of new *items* that the *user* would like and suggest them. It is a set of algorithms that together they can return for example an ordered list of probability/similarity from items that were previously compared against other items. These other items must be similar from the ones that the *account* have interacted or similar accounts have interacted with. Variables such as click-rate, visit frequency, time spent on page can be considered to calculate an affinity between interacted *items* and *account*.

Nowadays companies such as *Netflix* uses hybrid versions that combine multiple types of recommender systems, but commonly the recommender systems are classified between the two following type spectrums:

- Implicit vs Explicit
- Content-based vs Collaborative

Implicit vs Explicit definitions correspond to the way data is semantically related between the pair *account-item*. On *Implicit* recommender systems there is an implicit relationship between the used item and the expectation that the majority of those items are desirable by its user, on the other side of the spectrum, there are the *Explicit* recommender systems that usually uses a rate system given by the *user*, so there is not just a simple link between both entities, but also a subjective magnitude that could be either positive or negative, qualitatively speaking.

Content-based vs Collaborative classifies on an *account-account* relationship. On the first one the similarity between interacted items on a same account is used to retrieve similar new items that the user didn't yet explored, on the second one, the comparison is based upon other accounts that share similarity, so items from these accounts that are missing on the interaction history can be recommended. On state of art systems, both methods are combined in order to get the best results.

2.3 Dimensionality reduction

The more dimensions a data has the more diluted the statistical significance becomes, because data sparsity grows exponentially as the dimensions grow linearly, this phenomena is called *the curse of dimensionality* and is described on Marimont and Shapiro (1979) and one on way to finding statistical significance are techniques of *dimensionality reduction*, further discussions and solutions are discussed on 4.1.2.

One of firsts successful algorithms for this purpose is called *Principal Component Analysis* by Hotelling (1933) and it consists in finding linear combinations between pair of dimensions that corresponds to most statistical significance trough variance, where the two most significant linear combinations becomes the main dimensions and its points are plotted in a two dimensional Cartesian plane. Dimensionality reduction techniques are used before applying Clustering algorithms, because normally media contents are described as an array of multidimensional features.

Since media content normally arrives in image, sound or text, content classification is a multidimensional case, those types of data are rich in both syntactic and semantic information, the common pipeline of classification rely on techniques of feature extraction following dimensionality reduction, however, this type of content are not made of linear combinations problems, but non-linear relationships between dimensions, for this purpose several techniques rather than PCA (that is linear by its nature) were developed on the past. For the purpose of this work, given the non linearity of news embeddings and as mentioned on Moreira (2018), provider of the dataset embeddings used on this work, tSNE method from Maaten and Hinton (2008) was considered as the best approach; this technique uses *gradient descent* Bryson and Denham (1962) to minimize *relative entropy* Kullback and Leibler (1951) from distance distributions between all combinations of pairs from the data points; in fact, those distributions are the euclidean distance from each point against others and each distance projected into a distribution function, the innovative difference between tSNE and its predecessor SNE Hinton and Roweis (2002) is that it uses t-student distribution instead of gaussian, that is far more cheaper computationally and generates similar but denser results.

2.4 Cluster analysis

It is the method for grouping items that share a vicinity/similarity and then labelling each one of the groups, normally, clustering algorithms are applied on points plotted on two or three dimensional planes. There are many types of clustering algorithms but for the construction of four main used types of algorithms were taken into account describe below:

- **Centroid-based**
  One of the oldest clustering algorithms is k-means Steinhaus (1956), MacQueen et al. (1967), k is the number of clusters pick at the beginning of the algorithm, than k randomly cluster centers are introduced into the set of points, it is attributed each point to the closest cluster center in respect to it's euclidean distance, this process is repeated many times as necessary to reduce the variance between each cluster points, however, the problem of this kind of algorithm is that the user must test the k parameter multiple times till it find the best fit, *Affinity-propagation* algorithm Frey and Dueck (2007) on the other hand, it doesn't know previously the quantity of cluster centers, it is discovered during the calculation, for that reason *affinity propagation* was chosen for the proposed method.
- **Distribution-based**
  This clustering category works as a soft clustering, since during calculation it attributes for every point it is assigned a probability of ownership from each individual cluster, in every step of calculation, those probabilities are update and in the end, the point is claimed by the cluster which it has the biggest probability of ownership. The most used distribution-based algorithm is *EM clustering* from Dempster, Laird and Rubin (1977) where the mentioned steps are split into *Expectation (E)* steps followed by *Maximization (M)* steps in alternation between each other, it can be viewed also as an *relative entropy* reduction problem Kullback and Leibler (1951) between the expected distribution and the resulting distribution.
- **Density-based**
  It is based on the concept of "density-reachability"; the most used density-based method is DBSCAN Ester et al. (1996), where euclidean distance is used as a radius from the observed point, where every neighbor point inside its radius is "contaminated" by its cluster label, the problem of this kind of method is when you have homogeneous density of points or a smooth transition of density from on cluster to

another, this can mislead the algorithm. DBSCAN complexity has a $O(n\ log\ n)$ average complexity. It is commonly used for unsupervised learning and anomally detection.

- **Hierarchical-based**

  They are divided into *agglomerative* and *divisive* categories, on the first, it merges small clusters into big clusters, on the second one, it keep subdividing from one big starting cluster to more small clusters, they are mainly used for genetics and species classification and two famous methods are *single-linkage* and *complete-linkage* Johnson (1967), Sibson (1973) clustering. Hierarchical clustering algorithms are considered very slow since they have quadratic or cubic complexity, for that reason this type of clustering method was not chosen for the proposed work.

3 RELATED WORK

There is a distinction between timeout, time-decay and profile-based models for account sharing and user identification related works, for that reason, they are presented as follow divided by three addressed categories as timeout, time-decay and profile-based methods.

3.1 Timeout models

An attempt to categorize user sessions at Youtube platforms, Gill et al. (2008) made a distinction between user level and session level characterization, its definition of session beginning and end is a timeout fashion and the explanation for this is that analyzing traffic network is challenging because most of times there is no clear registry of login and logout. They used a benchmark technique to define what is the best time threshold that would really define a session; they found out that 40 minutes of inactivity as threshold is the spot because from this point on, the number of sessions start to level-off considerably. Nonetheless, they do not take into account multiple users swapping use like scenarios were a computer can have a very fast turn over, like lan-houses or public spaces with computers or even paid multi-platform account sharing such as <togetherprice.com> and <kotas.com.br>.

One example of account decomposition is Bajaj and Shekhar (2016) that individualize each user as a persona from online TV platforms accounts. Account sharing is very common between families in front of a TV. This persona-based individualization uses TV logs to generate user profiles; cosine similarity is applied on item frequency matrices that later are clustered using hierarchical clustering where each cluster is an account, lastly, they use their $Apriori$ algorithm to categorize each persona, however, their session categorization is based on hourly inactivity timeout and their technique is restricted to TV platforms.

The work Halfaker et al. (2015) consists in clickstream monitoring of different domains on online activity browsing (games, search engines, page views), thus achieving session identification using a fixed 30 minutes inactivity threshold as session identification, authors demonstrates high regularity between different sessions from a same user. Since they do not address account sharing, there is a chance that threshold based on item usage or user behaviour instead of time usage, would have more impact on identifying sessions related to different users.

A state of art contribution is described on Jiang et al. (2018) that using the *LastFM* and *KKBOX* datasets, they address a solution on user identification on shared accounts at streaming platforms given its history logs, proposing the novel unsupervised learning SHE-UI, they could be able to identify groups of users sharing a same account, bringing their solution to an user-level recommendation domain instead of account-level, however, there is a pre-processing step on the stream of items that split sessions based on 30 minutes user inactivity and not based on its content, so people using the same account on different computers with IP obfuscation techniques wouldn't be taken into account for example.

3.2 Time-decay models

A common method on recommender systems for defining subsets from stream data is introducing decaying weights on items in respect to time, Sottocornola, Symeonidis and Zanker (2018) addresses the absence of session delimitation, their technique introduce a time sliding window, they deal with data sparsity using the hibrid approach, combining both CF and CB; they delimit each session using fixed 30 minutes like other methods, this fixed threshold approach however, cannot detect users swapping the account usage in less than the fixed time threshold, even if there were two users are in separated ip addresses, taking this into account could be a privacy violation, or even in some places, using ip addresses to distinguish users can be an inoquous alternative when VPNs are overly used.

Some methods proposes RNNs in order to extract features from items, one of them is Zhang, Liu and Gulla (2019) that proposes dynamic attention-integrated neural network (DAINN) for news recommendations, DAINN network is used to extract item semantic embeddings and they join user long-term interests with user behavior sequence patterns, their method showed a slightly improvement against other methods. However, they use a fixed time window of 150 items to create the short-term user profile,

3.3 Profile based models

On Jindal and Sardana (2020) et al, they propose a dynamic threshold heuristic based on user behaviour history, their discoveries show that short time user navigation are more often and less correlated than longer ones, making them easier to categorize and achieving a higher accuracy. However, this method apply a threshold into the user behaviour pattern, and differently, this work relies on the distance between topics inside a session, also the proposed method uses fixed threshold for simplicity reasons.

Zhang et al. (2012) addressed the problem of account sharing and modeled an algorithm for identifying movie ratings from individual users sharing an account. It is used a model of union of linear subspaces for spotting shared accounts and a model of clustering for user identification per account. This algorithm has good accuracy on most cases, showing that it is possible to identify a distinction from each user based on item category and action, however, this algorithm is restricted for explicit recommender systems and movie datasets tend to be smaller in size and category amount in comparison to other media datasets.

A problem that e-commerce recommender systems face is when an advertisement of an exact size or individual characteristic for an user is required, because other users may end up contaminating the account history and spoiling the estimate of these unique attributes or sizes, on Sembium et al. (2018) the problem is discussed using Amazon.com e-commerce, instead of dealing with each account as a specific user and single profile, they propose a multi-persona approach boosting the accuracy, however, this method does not work well on products that have a high variability on its characteristics and it was only applied on an e-commerce platform, where there is a high repeatably of items.

Verstrepen and Goethals (2015) proposes a top-N recommender system for solving account sharing problems, three common problems are categorized and solved:

- **Generality problem** is when the recommended item is the mean of all user items, being uninteresting for none of the users.
- **Dominance problem**, happens when few users use the system more than the others, in that case most of items are related to those few users.
- **Presentation problem** addresses the difficulty of recommending an item for the current user and not the other idle users.

This model uses binary positive-only feedback as items and besides they used movie ratings datasets, their model probably can be used on implicit recommender systems because they share the binary feedback model between user and items. Their approach uses normalized cosine similarity with KNN Silverman and Jones (1989) to produce the account-item score recommendation matrix, then they introduce their novel

DAMIB-COVER algorithm that solves the generality problem by differentiating scores between a big sum of few similarities and a small sum of a lot similarities from users in a same account.

The problem of cross-domain along with account sharing is well addressed on Ma et al. (2019) that proposes a *Sequential recommender system*, they use a a sequence encoder in each domain to update the recommendation state when each item arrives, along with this there is a *shared account filter unit (SFU)* on each domain to recognize the weight of each user from an arrived item, those informations are exchangeable between each domain, where the final prediction is calculated by a *cross-domain transfer unit (CFU)* that takes into account other domains *SFUs*. Differently from described related works, pi-net is not session-based nor explicitly time-decay based, however, since the SFU unit and its sequence encoder is actually some kind of linked RNNs unit, there is an implicit time decay, since when new items arrive they tend over time to fade the impact of older items when modifying RNNs internal weights, along with this fact, their method is restricted to TV domains and they do not address news articles.

3.4 Comparative Analysis

The problem of time-based models like Halfaker et al. (2015), Gill et al. (2008), Sottocornola, Symeonidis and Zanker (2018), is that they are not able to split sessions based on individual users, since if many users share a device at the period of a defined session, they would be considered a single user, leading to the *generality problem* addressed by Verstrepen and Goethals (2015).

Profile-based recommender systems such as Zhang, Liu and Gulla (2019), Verstrepen and Goethals (2015) rely on account profiling based on item history, trading account privacy for accuracy, other hybrid models like Jindal and Sardana (2020), Jiang et al. (2018) that mixes both solutions also rely on user profiling and behaviour and user privacy is at risk. For instance even the time-based model as Gill et al. (2008) have its algorithm relying on internet traffic, posing risk to personal information.

4 METHODOLOGY

At the current chapter, problem definition, algorithms, tools used for the proposed methods and its validations are discussed and presented with its motivations. The current chapter is ordered as follow: at section 4.1, the problem definitions arise, the concept of session is defined as sets of items and the problem of simulating shared accounts from different account sessions is as well defined, mathematical and theoretical problems from data scarcity and sparcity are introduced and discussed.

On the section 4.3 the algorithms used for solving most of those problems are introduced and explained, furthermore, a comparison of those algorithms and the state-of-art solutions and motivations of choice are addressed at this section. At the last section 4.4, the technology used is mentioned as well as its application on the dataset.

4.1 Problem definition

Definition 1 *Let I be a set of items where $i \in I$ and an item i represents a news. Let a set of users U where $u \in U$ and each user u does not repeat itself into the set. A session s is defined by a list of items $s_1 = \{i_1, i_2, i_3\}$ and belongs to an individual user u_1, the items may repeat inside a session and also between sessions.*

It is considered by definition, that each User is an account itself, and that most of users doesn't share its account, and the amount of shared accounts have low statistical significance into the end result.

4.1.1 Simulated Accounts and Sessions

In order to simulate a shared account, two sessions are concatenated into a new session as $s_1 = \{i_1, i_2, i_3\}$ and $s_2 = \{i_7, i_8, i_9...\}$ and those sessions must belong to different users.

the new simulated session consists in the following representation

$s_{12} = s_1 \cup s_2 = \{i_1, i_2, i_3, i_7, i_8, i_9, ...\}$

The objective of the proposed method is to identify the exact frontier-item between sessions and revert this concatenation process separating the new session s_{12} into the original s_1 and s_2 sessions, in that scenario, item i_3 would be the frontier-item, since it is the last item of the first session s_1.

This process should use only the item content or category as parameters, being blind to the user ids or timestamps.

Time-based sessions are the most commonly used as consensus as a method for example at Sottocornola, Symeonidis and Zanker (2018) it is used a defined session trough a fixed time, but the way they make it flexible, were by using a decaying time-based weight into the session items, they used this approach to overcome the short life-span of news. At the proposed method, since not the news itself, but it's topic, the short life-span problem wasn't expected to be relevant enough.

4.1.2 Data scarcity and sparcity

Cold start problem. This flaw is addressed mostly to *collaborative filtering* recommender systems, since it is content-centric, when an user is fresh and he has little or no content history, the method lacks on precision resulting on spurious extrapolations.

A *content-based* system is no different in terms of this flaw, but it is more resistant since it uses content history, it can fill the gap against data scarcity from fresh users.

In order to avoid this dilemma, the proposed metrics are based on previously filtered sessions, where the first method use sessions with more than 10 items, and the second method, more than 200 items.

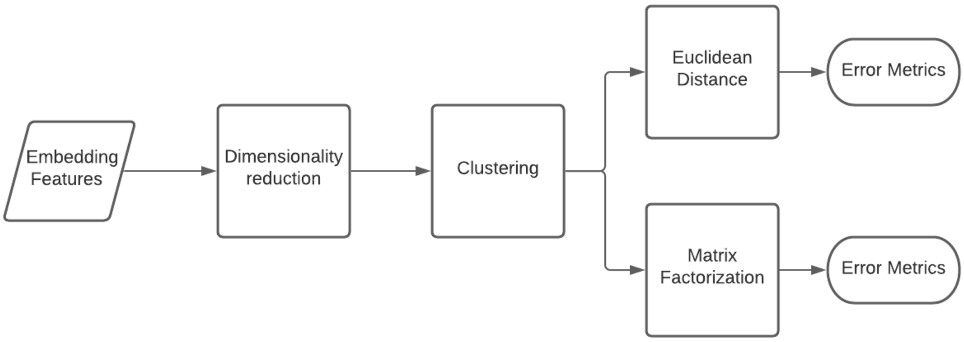
Curse of dimensionality Problem. Since content analysis and classification requires a complex analysis that generates great amounts of features/dimensions and usually recommender systems data are sparse, the curse of dimensionality Marimont and Shapiro (1979) is very relevant as a problem because usually any media content should be translated into a multi dimensional space before the classification, contributing to a non-ending crescendo *curse of dimensionality*.

4.2 Overview

Two main methods of evaluation are addressed, the first is an *euclidean distance* based method were the distance between each article content is considered; the second evaluation method relies on *matrix factorization* such as ALS technique Takane, Young and Leeuw (1977) where the read frequency from labeled news topics from each user acts as a form of implicit recommender system.

The image below describes in general terms the pipeline of both proposals

Figure 4.1: Proposed validation method



4.3 Used algorithms

On this section, all the used algorithms will be described as well as its historic evolution against their pioneers.

Principal Component Analysis Hotelling (1933) is the first known form of dimensionality reduction for multidimensional data. The pinnacle of the technique is mapping from all dimensions two linear combinations which have the biggest statistical significance, or in other terms, mapping on two dimensions which have maximized variance between all others.

If you have n dimensions and m items where each item has its own coordinate, PCA would translate this $m \times n$ space matrix into another rotated $m \times n$ matrix trough linear operations into each dimension, those new translated dimensions are called *principal components*. Their rotation is in fact a linear combination between two original dimensions which have its variance maximized using *Least Squares Regression*.

The *principal components* end up ordered based on their variance from its span, they are the eigen vectors positioned on the diagonal of the resulting matrix and the rank of *principal components* based on its variance is called *scree-plot*.

However, *Principal component analysis* is the naive approach for data dimensionality reduction, but it is very useful when data have information linked trough linear relationships. A successful and recent example of PCA use is this physical engine in Holden et al. (2019), that before a complex process of vector computation for physics simulation, a PCA was used to reduce the statistical irrelevance creating a meaningful and denser subspace in terms of significant information before training a neural network, making the algorithm 10 times faster and using less memory.

At Globo dataset, as Moreira, Ferreira and Cunha (2018) mentions, they could not provide the news content due to its closed source policy, however, they released a matrix of features where each news was represented by an array of 255 features, also they mention in their article that tSNE would translate to a better representation of the reduced data, since this matrix have non-euclidean information relationships, so in that case, PCA would not fit the best 2-dimensional surface in terms of statistical significance.

To think what would happen using a excessively linear algorithm like PCA to find the best surface, imagine a zig-zagged bent bread and you want to cut the middle of it with a knife, PCA would make a straight line cut ignoring the format of the bread. In that case, the algorithm they recommend is tSNE, a modern and state-of-art technique for dimensional reduction that uses machine learning and gradient descend techniques.

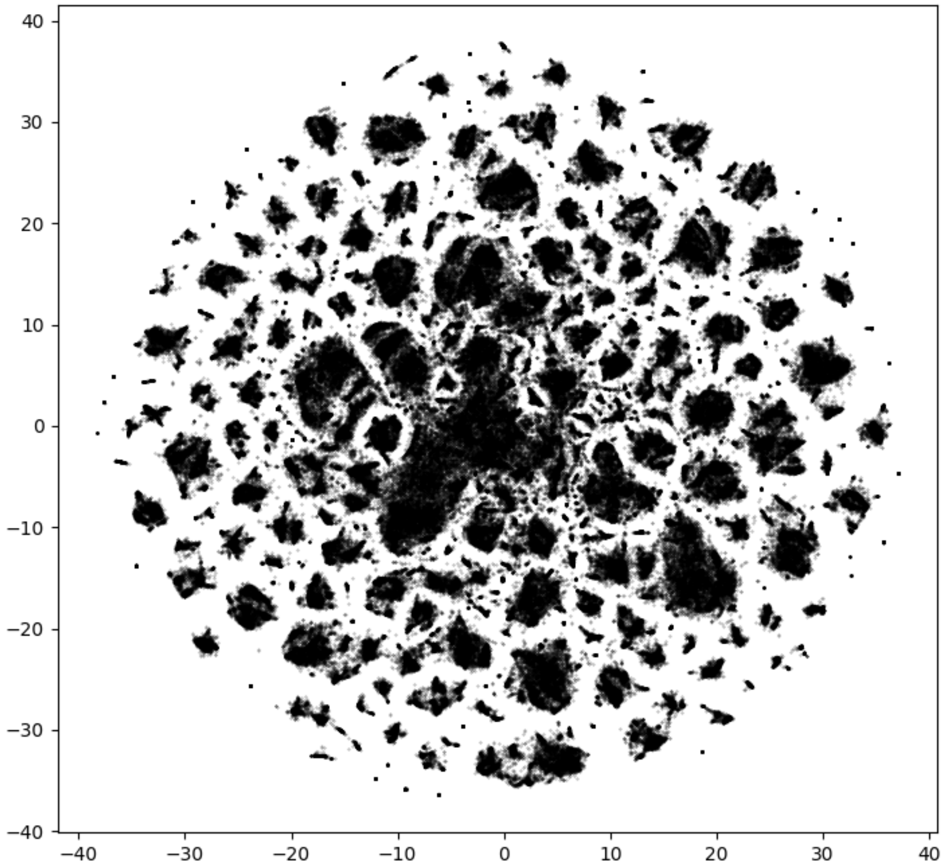
tSNE (t-distributed stochastic neighbor embedding) also maps highly dimensional data into a low dimensional representation resuming the data and resisting against dimensionality curse as mentioned Marimont and Shapiro (1979), but differently from PCA, tSNE Maaten and Hinton (2008) is a non-linear method, this method preserves local geometry and deform global geometry, more specifically, it tends to clump points that share a near neighborhood making the clusters more evident. Non-linear means that if you have highly dimensional data that has a geometric representation of a curvy non-convex surface of points, for example, PCA due to its linearity would make a straight cut into this curve finding straight planes and never really fitting into the curvy surface, PCA has its algorithms variances that reduce its problems, like the Kernel PCA, but tSNE is considered to be the state-of-art for reasons below.

tSNE It is a sequel of SNE (Stochastic Neighborhood embedding), but because the current distribution from SNE not only hardly clump the points close to each other but it is use to be computationally slow, because of the use of exponentiation when calculating the distribution equation.

The "t" letter stands for t-student distribution, it is computationally faster and has a less severe penalty on point agglutination due to its natural flattened curve, thus winning the "state-of-art" title due to its low cost and better precision.

On figure 4.2 there is a plot of tSNE applied on Globo dataset highly dimensional embedding that represents the content of each news, this exactly plot and its data was used on the proposed method, each point represented below is a relationship of a Xs, Ys coordinate with an article id

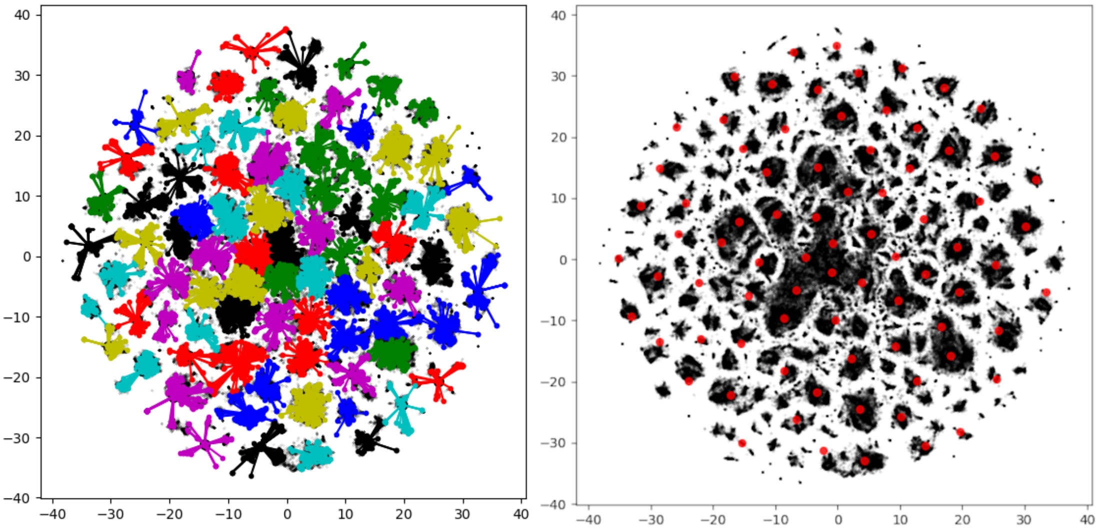
Figure 4.2: tSNE plot from globo dataset



Affinity propagation. The advantage of affinity propagation between other algorithms is that it does not need to specify an estimate of quantity of points found into the space, it naturally walks trough the points and finds the resulting clusters.

On figure 4.3 there is the visualization of the discovered clusters on the low dimensional embedding representation from tSNE result of globo dataset, the second image shows the center from each topic cluster, that later should be used to calculate the semantic difference from each topic based on the euclidean distance from the pair of topics being compared.

Figure 4.3: Affinity propagation clusters(left) and its centers(right)



Matrix Factorization. Netflix is known for it's competitions for collaborative filters algorithms, in 2008 *matrix factorization* Srebro and Jaakkola (2004) was used and achieved a huge breakthrough in collaborative filtering algorithms, and it is like the prime factors for grown ups, using matrices as factors instead of just scalar numbers, it is a useful mechanism to separate the users-items matrix into a items' matrix and a users' matrix, that their multiplication result in the original users-items matrix, results into Item X User original association matrix that each item is the relation of one user to a single item. The usefulness of this factorization, is that since the original matrix is sparse, you can fill the missing values by making inferences on the factorized data, because not every user in its history has real relationship with every item. An optimal way of doing so is by using a modern version of *Alternating Least Squares* algorithm Takane, Young and Leeuw (1977) explained below that uses *Gradient Descent* as convergence algorithm.

Alternating Least Squares. It is a faster variant of the Matrix Factorization that allows it to be computational distributed, the as matrix factorization, it has SVD Golub and Reinsch (1971) as central role on the factorization between item and user matrices, but in this case, a cost function with an error function is alternated between each matrix, updating the elements of one matrix at a time, this computational behaviour reduces the complexity of the cost function and allows it to be computed in a distributed fashion. Today a common implementation of ALS is from Spark $Hadoop$ in java, but also have facade libraries that allow calling ALS from spark $Hadoop$ from other languages like Scala and Python (Pyspark).

4.4 Technology usage

It was used $Python3$ with *pandas* team (2020) and *numpy* as main data manipulation, $Pyspark$ was used for ALS method recommender system. $Pyspark$ is a python library that is a facade for a java spark hadoop spark application server that uses map/reduce in order to make distributed computations.

For both tSNE and Affinity propagation *sklearn* library was the choice and the source code from the proposed work is available at github [1]

4.4.1 Applying Dimensionality reduction

Because *Globo.com* news text data is closed source, Moreira (2018) made available the trained article embeddings data [2] generated from its unique *CHAMALEON* RNN-based algorithm. The embedding file consists into a matrix of 250 dimensions of features and 364,047 news articles, those features are a representation of the content for each respective news article.

On the proposed method only the following data and respective embedding were considered: *click_timestamp, article_id* and *user_id.*

Since on their work tSNE was mentioned as a good method for dimensionality reduction, tSNE was applied on the embedding, resulting into a pair of coordinates X, Y for each article id as the dataframe sample below:

Table 4.1: tSNE coordinate results

article_id	x	y
...	...	...
69	10.950659	-26.211418
81	34.414822	-0.690890
84	35.335995	-1.578238
...	...	...

[1] github.com/zatura/clickstream-content-sessionization

[2] kaggle.com/gspmoreira/news-portal-user-interactions-by-globocom

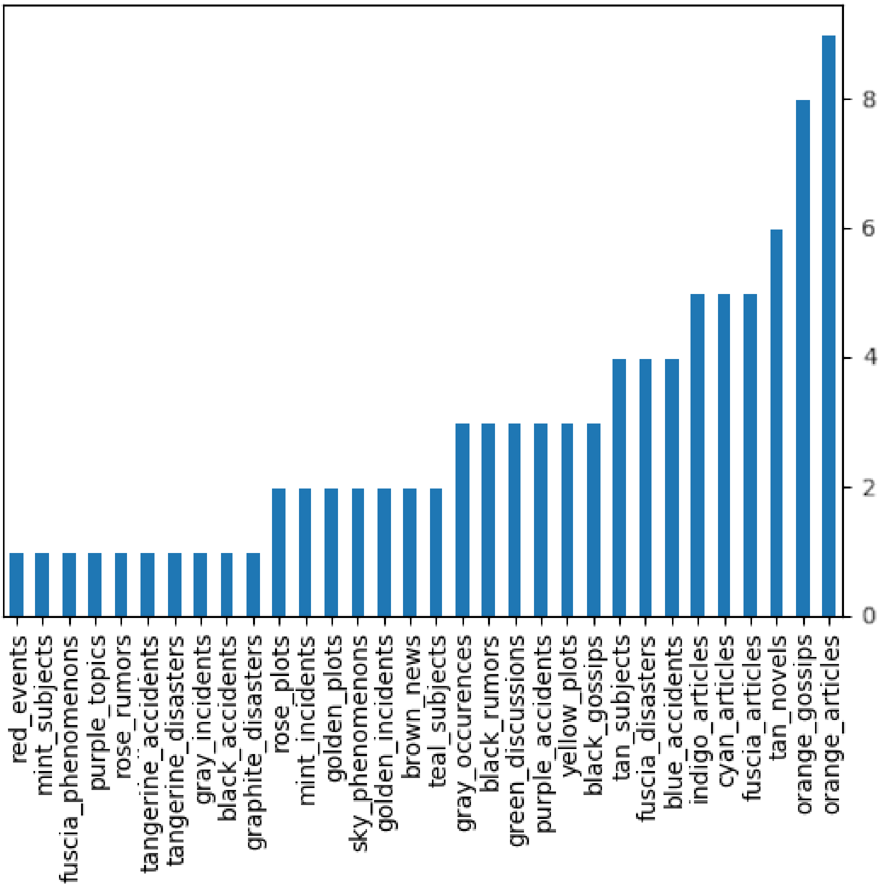
4.4.2 Applying Clustering

At this step, *Affinity propagation* is applied on the tSNE result data plane, several damping parameters were tested and presented further at the error metrics section. After the clustering, each cluster center point into the resulting plane receives a random generated human-readable label with the format <*random_color-random_substantive*>, as shown on table 4.2 example, also, chart 4.4 shows the distribution of consumed topics by a sampled user resembling a Benford's law distribution:

Table 4.2: Labeled cluster centers

x	y	topic
...	...	...
19.699291	-28.117729	tangerine-events
-8.711248	21.906485	purple-occurrences
11.325381	3.893677	yellow-episodes
...	...	...

Figure 4.4: Topic frequencies from a sampled user



5 EXPERIMENTS

The objective of this experiment is evaluating the accuracy of a content-only method of session splitting, without the need of user profiling, then contrasting to a technique that uses some sort of profiling, based on item similarity and matrix factorization method such as *alternate least squares*.

There will be a comparison between *euclidean distance* threshold against *alternating least square* methods and its parameters variations, different combinations of items and sessions were performed like the impact of temporal ordering or shuffling items by timestamp, different affinity propagation damping factors and different sizes of sessions.

On section 5.1 Globo dataset properties are presented, on 5.2, section 5.3 the simulation of sessions as metric method is adressed with its results and on 5.5 error heatmaps from iterations are shown.

5.1 Dataset

The dataset of choice was globo.com news dataset, it is a collection that consists on the following:

- approximately 2.8 millions news clicks
- more than 300 thousand users
- 45 thousand news articles
- 250 dimensional feature matrix

Since the availability of articles text data were not allowed by the company by commercial reasons, at Moreira (2018) presented CHAMELEON, a RNN algorithm for feature extraction, and together with the dataset, they made available a feature embeddings file extracted from its novel algorithm. Every news from the dataset have its own 250 feature dimensions on the embedding matrix

The dataset consists on the following data for each click as the columns:

$user\_id, session\_id, session\_start, session\_size, click\_article\_id,$

$click\_timestamp, click\_environment, click\_deviceGroup, click\_os,$

$click\_country, click\_region, click\_referrer\_type$

However session attributes such as $session\_id, session\_start$ and $session\_size$ are presented in the dataset, they were not considered on this session split method since they would conflict with the timeless nature approach, the only columns considered were $click\_timestamp$, $user\_id$ and $article\_id$ along with article embeddings; dataset sample is showed on the table below:

Table 5.1: Globo dataset sample as Dataframe

click_timestamp	user_id	article_id	click_country	click_region	...
...	...	...	...	...	
1506826800026	59	234853	1	21	
1506826801702	79	159359	1	13	...
1506826804207	154	96663	1	25	
...	...	...	...	...	

5.2 Evaluation metric

The chosen method for finding session boundaries was based on the will that users tend to encircle same or similar topics, similarly as shown on chart 4.4, on that sense, this hypothesis was used to create a model using euclidean distance from topics to measure how different they are.

Definition 2 *In a session S with a set of items $i_1, i_2, ...$ that each item has a topic t associated, where each topic t has a known coordinate x, y in a Cartesian plane. If the euclidean distance between the topic t_p of item i_n and the topic t_q of item i_{n+1} is higher than a chosen cutoff c, than it is considered an end of a session and beginning of another.*

The error metrics from cutoff heuristic is defined as follow:

- The cutoff c is a row index that belongs to one of the sessions, this index is found by the cutoff trigger heuristic at definition 2
- Because the two sessions doesn't have the same amount of items, thus asymmetric, a normalization mapping of the error metric is performed, the normalization effect is described on a gray color scale on table 5.2.
- It is calculated the difference between the number of items from the inferred index(found by heuristic) and the perfect index(the real cutoff) and divided by the total number of items at the artificial session (made by two real sessions), as represented above in the gray-colored table 5.2 by the error column.

So in this example, the index 3 is the perfect cutoff, representing the real threshold between the two sessions, the picked index by the cutoff heuristic will receive the respective error from its index row.

Table 5.2: Error normalization map from desired cutoff index

index	user_id	error
0	111	1
1	111	0.66
2	111	0.33
3	888	0
4	888	0.16
5	888	0.33
6	888	0.5
7	888	0.66
8	888	0.83
9	888	1

5.3 Session simulation

Because the dataset already have labeled the respective *user id* in each row, it was considered that the most part of accounts were in fact from individual users, in that case, it is assumed that even if some users really share the same account, this would be statistically irrelevant when analyzing the whole dataset. Considering this scenario true, the simulated sessions is constructed as a concatenation of two streams of two separated user ids as addressed on table 5.3 a sample from a simulated session.

Notice that the column *distance* stands for euclidean distance from current centroid of row index n and centroid from row index $n - 1$ in a simulated session table.

Table 5.3: Simulated session sample with desired cut in gray

	click_timestamp	user_id	x_centroid	y_centroid	distance
...	...	...	...	...	...
11	1507987156229	6207	-8.487595	3.206146	43.517998
12	1507988999717	6207	3.864952	-3.751176	27.925744
13	1507989029717	6207	5.004131	3.442250	7.283071
14	1507061447189	143259	25.503668	-1.358489	21.054171
15	1507061615464	143259	18.530073	-15.994876	16.212799
16	1507061615464	143259	18.530073	-15.994876	0.000000
...	...	...	...	...	...

5.4 Temporal desambiguation

Since time ordering can be sensitive for news articles a method to disambiguate the influence of temporal order was introduced because the objective of this work is to segregate sessions exclusively on its content excluding time factor for example.

There were two different types of measurements, one of them had time ordering were each user session had its items ordered by its timestamp, on the second one, the items were completely shuffled, removing time influence.

A floor cutoff euclidean distance was pick as 1, and the ceiling as 60, the origin of those numbers is related to the scale of de tSNE plot addressed on figure 4.2. With this limits defined, 15 linearly spaced cutoff distances were generated as input as shown by columns *cutoff dist* on table 5.4, for each session cut attempt, an error were calculated and attributed as section 5.2 describes, an array of error metrics was created and an *error mean* and also an *error standard deviation*, described on second and third column of table 5.4.

Looking on both mean and std metrics result, it has a neat meaning that time order does not affect the euclidean threshold heuristic, however, it is important to notice the high error mean (more than 0.5), furthermore, the results shows that the lower the error mean, the error standard deviation rises considerably, thus making it less reliable. For example, on table 5.4, notice that on the table with time ordering, the lowest error mean is 0.611 and its standard deviation mean is 0.3 or in other words, a deviation of 30 %, making this cutoff parameter unreliable for the session cutting purpose.

Table 5.4: Error metrics, session ordered by timestamp 5.4a vs shuffled 5.4b

(a)

cutoff dist.	error mean	error std
1.00	0.933	0.053
5.21	0.932	0.054
9.42	0.922	0.071
13.64	0.917	0.078
17.85	0.901	0.098
22.07	0.877	0.126
26.28	0.857	0.151
30.5	0.812	0.191
34.71	0.765	0.230
38.92	0.681	0.266
43.14	0.629	0.286
43.35	0.611	0.301
51.57	0.630	0.311
55.78	0.707	0.320
60.00	0.814	0.287

(b)

cutoff dist.	error mean	error std
1.00	0.939	0.045
5.21	0.938	0.048
9.42	0.933	0.057
13.64	0.927	0.663
17.85	0.912	0.088
22.07	0.893	0.114
26.28	0.867	0.144
30.5	0.827	0.184
34.71	0.778	0.223
38.92	0.710	0.264
43.14	0.646	0.289
43.35	0.615	0.310
51.57	0.631	0.322
55.78	0.721	0.325
60.00	0.813	0.298

5.5 Affinity propagation damping factor

Another possibility for interference on session cuts performance can be the granularity of clustering, or in other words, how specific or generic that a topic classification should be so that it is enough to recognize the difference between users sharing an account?

In a single topic like Math, many areas can be classified like algebra, calculus, statistic and so on, at the same time, you can be more generic, and simply address that all those areas are simply Math, and then you segregate other topics as Portuguese, Biology, Math and so on without opening into details.

Since the *damping factor* from the Affinity Propagation Algorithm described on section 4.3 changes the amount of classified clusters, another parameter was introduced on the error metrics calculations and that being the *damping factor*. For different damping factors, the amount of topics is addressed on table 5.5.

Table 5.5: damping factor effect on nº of clusters

damping	nº of clusters
0.5	4771
0.553	4716
0.606	2984
0.66	1146
0.766	91
0.82	152

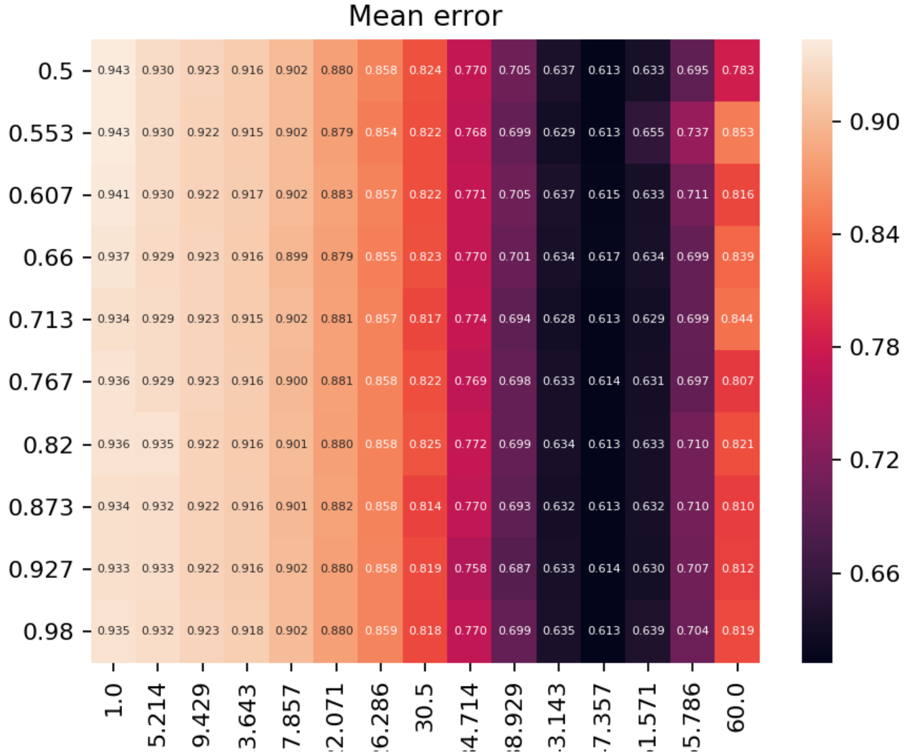
Similarly as section 5.4 a number of 15 cutoff values were tested between values 1 and 60, linearly spaced for each iteration with a different cutoff . The output was a vector of error values of the same size of the simulated sessions (one for each session, about 32 thousand sessions). Then, from this error vector, it was calculated its mean and standard deviation for each iteration (15 iterations), so that we could measure its reliability and precision.

The error heatmaps results are interpreted as the following:

- The Vertical axis of the represents the affinity propagation damping factor
- The Horizontal axis corresponds to the chosen cutoff distance threshold
- the value of each item on the heatmap corresponds to one of two metric errors: Mean error, Standard deviation error.

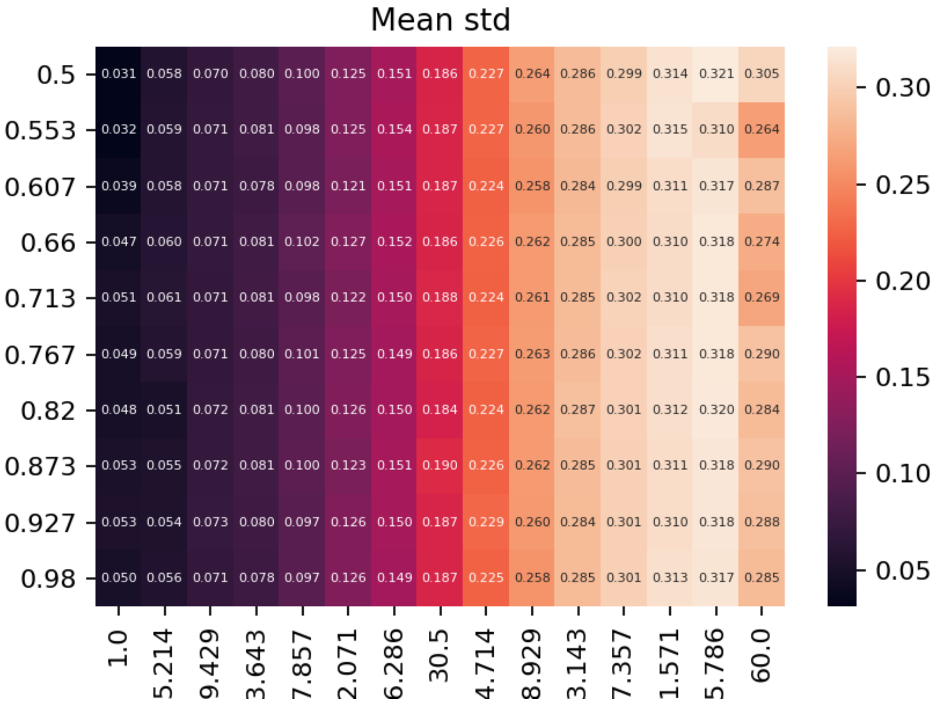
The objective of this metric is finding a lower than 50% mean error and the lowest $standard\ deviation$.

Figure 5.1: Error mean heatmap



	1.0	5.214	9.429	3.643	7.857	2.071	6.286	30.5	4.714	8.929	3.143	7.357	1.571	5.786	60.0
0.5	0.943	0.930	0.923	0.916	0.902	0.880	0.858	0.824	0.770	0.705	0.637	0.613	0.633	0.695	0.783
0.553	0.943	0.930	0.922	0.915	0.902	0.879	0.854	0.822	0.768	0.699	0.629	0.613	0.655	0.737	0.853
0.607	0.941	0.930	0.922	0.917	0.902	0.883	0.857	0.822	0.771	0.705	0.637	0.615	0.633	0.711	0.816
0.66	0.937	0.929	0.923	0.916	0.899	0.879	0.855	0.823	0.770	0.701	0.634	0.617	0.634	0.699	0.839
0.713	0.934	0.929	0.923	0.915	0.902	0.881	0.857	0.817	0.774	0.694	0.628	0.613	0.629	0.699	0.844
0.767	0.936	0.929	0.923	0.916	0.900	0.881	0.858	0.822	0.769	0.698	0.633	0.614	0.631	0.697	0.807
0.82	0.936	0.935	0.922	0.916	0.901	0.880	0.858	0.825	0.772	0.699	0.634	0.613	0.633	0.710	0.821
0.873	0.934	0.932	0.922	0.916	0.901	0.882	0.858	0.814	0.770	0.693	0.632	0.613	0.632	0.710	0.810
0.927	0.933	0.933	0.922	0.916	0.902	0.880	0.858	0.819	0.758	0.687	0.633	0.614	0.630	0.707	0.812
0.98	0.935	0.932	0.923	0.918	0.902	0.880	0.859	0.818	0.770	0.699	0.635	0.613	0.639	0.704	0.819

Figure 5.2: Error standard deviation heatmap



	1.0	5.214	9.429	3.643	7.857	2.071	6.286	30.5	4.714	8.929	3.143	7.357	1.571	5.786	60.0
0.5	0.031	0.058	0.070	0.080	0.100	0.125	0.151	0.186	0.227	0.264	0.286	0.299	0.314	0.321	0.305
0.553	0.032	0.059	0.071	0.081	0.098	0.125	0.154	0.187	0.227	0.260	0.286	0.302	0.315	0.310	0.264
0.607	0.039	0.058	0.071	0.078	0.098	0.121	0.151	0.187	0.224	0.258	0.284	0.299	0.311	0.317	0.287
0.66	0.047	0.060	0.071	0.081	0.102	0.127	0.152	0.186	0.226	0.262	0.285	0.300	0.310	0.318	0.274
0.713	0.051	0.061	0.071	0.081	0.098	0.122	0.150	0.188	0.224	0.261	0.285	0.302	0.310	0.318	0.269
0.767	0.049	0.059	0.071	0.080	0.101	0.125	0.149	0.186	0.227	0.263	0.286	0.302	0.311	0.318	0.290
0.82	0.048	0.051	0.072	0.081	0.100	0.126	0.150	0.184	0.224	0.262	0.287	0.301	0.312	0.320	0.284
0.873	0.053	0.055	0.072	0.081	0.100	0.123	0.151	0.190	0.226	0.262	0.285	0.301	0.311	0.318	0.290
0.927	0.053	0.054	0.073	0.080	0.097	0.126	0.150	0.187	0.229	0.260	0.284	0.301	0.310	0.318	0.288
0.98	0.050	0.056	0.071	0.078	0.097	0.126	0.149	0.187	0.225	0.258	0.285	0.301	0.313	0.317	0.285

5.6 Alternating least squares comparison

The last attempt to verify the reliability of the session cutoff method was using topic read frequency per user and feeding into an ALS implicit recommender system using *pyspark*, mentioned on section 4.4, creating a list of sets $(user\_id, topic\_alias, read\_frequency)$ as shown on table 5.6. Later, data was split into 25% / 75% train vs test data and the ALS model was trained.

Table 5.6: ALS: Read frequency per topic

user_id	topic_alias	read_frequency
...	...	...
26340	purple-articles	7
26340	rose-stories	5
58277	cyan-contents	1
...	...	...

In order to avoid cold start, only users with more than 200 items in total were chosen. Same as Euclidean Distance experiment on 5.2, fictitious sessions were created, but instead of a distance column as error metric, a new column 'prediction' were added that corresponds to the predicted probability that the probe $user\_id$ is likely to read that article.

The next step relies on making a cross probing between a pair of user sessions each time, so in a session from $user\_id = 413$, the predicted probability for each item is calculated for $user\_id = 45502$, than it is calculated the *mean* of all predictions as well as its *standard deviation* as shown on second line of table 5.7, lastly, the process is repeated with the $user\_id = 413$ against itself. The expectation of this experiment is that the prediction_mean of an user session comparing against itself should be high at majority of times with low *standard deviation* and for the second user session against the first one, the prediction_mean for the second user against the first should be at some median value with high *standard deviation*, meaning less correlation.

However, looking at table 5.7 there seems to be no relevant difference between the *prediction means* and *prediction standard deviations*, meaning that results from previous steps or even the embeddings could be faulty at some point.

The following table was created as result, it was extracted metrics from 413 users.

Table 5.7: Alternating Least Squares measurements

user_id	user_id_to_predict	prediction_mean	prediction_std
413	413	0.83287	0.2799
45502	413	0.86549	0.25900
12897	12897	0.84201	0.3291
16695	12897	0.91515	0.27717
2930	2930	0.79370	0.34552
9261	2930	0.87126	0.2551
20001	20001	0.8777	0.24580
6344	20001	0.91090	0.2383
62025	62025	0.75506	0.35445
19864	62025	0.47764	0.27196
43017	43017	0.83342	0.27467
21356	43017	0.81031	0.28030
3391	3391	0.85998	0.2949
681	3391	0.87366	0.30739
11521	11521	0.61621	0.3742
59193	11521	1.01384	0.5024
11359	11359	0.88445	0.2596
23036	11359	0.91379	0.24409
22301	22301	0.81135	0.30362
77985	22301	0.78393	0.32523
13885	13885	0.85638	0.264
9193	13885	0.80591	0.2890

6 CONCLUSION

Different previous session splitting heuristics were presented from time-based to profile-based and their privacy versus accuracy trade-offs along with its flaws were addressed, later an experiment of session splitting based on content-only against profile-based had their accuracy stressed.

The error standard deviation presented on the heatmaps with different input parameters reveals that the euclidean distance as threshold method does not have enough precision, but these results could have many explanations not only from the method itself. Since globo dataset is closed source and the embedding features available from Moreira (2018) were created from his own algorithm, a certain level of belief was implied that the embeddings really reflected its content and his algorithm worked well on quantitatively describing them.

The ALS algorithm method evaluation showed that there were no relevant item similarity between users itself against other users, meaning that rather this is a cause of the chaotic nature of news or tSNE as dimensionality reduction wasn't precise enough to classify the different news topics.

Finally, news as a type of media have event-driven and chaotic nature, even when readers have his preference for certain topics, unexpected and relevant news from other topics may arrive and catch the readers eyes, those phenomena are maybe common, thus, the proposed method of sessionization by euclidean distance is not a good candidate to split sessions based on news-content.

Since *tSNE* resulting plane dataset from *globo.com* seems to have data clumps and connections between some of them, a future work could imply in using density-based algorithms such as *DBSCAN* from Ester et al. (1996) on cluster analysis step instead of distance-based algorithms like affinity propagation. Considering the chaotic nature of news events and user choice over topics, another subject for further investigation is the likeliness of a user to change its topic into a single session and how that impacts the recommender system accuracy, a further comparison against other dataset such as *LastFM million song dataset* could be performed.

REFERENCES

ARLITT, M. Characterizing web user sessions. In: . [S.l.: s.n.], 2000. v. 28, p. 50–63.

BAJAJ, P.; SHEKHAR, S. Experience individualization on online tv platforms through persona-based account decomposition. In: **24th ACM International Conference on Multimedia**. New York, NY, USA: ACM, 2016. (MM '16), p. 252–256. ISBN 9781450336031. Available from Internet: <https://doi.org/10.1145/2964284.2967221>.

BERTIN-MAHIEUX, T. et al. The million song dataset. 2011.

BRYSON, A. E.; DENHAM, W. F. A Steepest-Ascent Method for Solving Optimum Programming Problems. **Journal of Applied Mechanics**, v. 29, n. 2, p. 247–257, 06 1962. ISSN 0021-8936. Available from Internet: <https://doi.org/10.1115/1.3640537>.

DEMPSTER, A. P.; LAIRD, N. M.; RUBIN, D. B. Maximum likelihood from incomplete data via the em algorithm. **Journal of the Royal Statistical Society: Series B (Methodological)**, Wiley Online Library, v. 39, n. 1, p. 1–22, 1977.

ESTER, M. et al. A density-based algorithm for discovering clusters in large spatial databases with noise. 12 1996. Available from Internet: <https://www.osti.gov/biblio/421283>.

FREY, B. J.; DUECK, D. Clustering by passing messages between data points. **Science**, v. 315, p. 2007, 2007.

GILL, P. et al. Characterizing user sessions on youtube. **Proceedings of SPIE - The International Society for Optical Engineering**, v. 6818, 01 2008.

GOLUB, G. H.; REINSCH, C. Singular value decomposition and least squares solutions. In: **Linear algebra**. [S.l.]: Springer, 1971. p. 134–151.

HALFAKER, A. et al. User session identification based on strong regularities in inter-activity time. In: **Proceedings of the 24th International Conference on World Wide Web**. Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee, 2015. (WWW '15), p. 410–418. ISBN 9781450334693. Available from Internet: <https://doi.org/10.1145/2736277.2741117>.

HINTON, G.; ROWEIS, S. Stochastic neighbor embedding. v. 15, 06 2002.

HOLDEN, D. et al. Subspace neural physics: Fast data-driven interactive simulation. In: **Proceedings of the 18th Annual ACM SIGGRAPH/Eurographics Symposium on Computer Animation**. New York, NY, USA: Association for Computing Machinery, 2019. (SCA '19). ISBN 9781450366779. Available from Internet: <https://doi.org/10.1145/3309486.3340245>.

HOTELLING, H. Analysis of a complex of statistical variables into principal components. In: BIOMETRIKA (Ed.). **Relations between two sets of variates**. Journal of Educational Psychology, 1933. p. 417–441, and 498–520. Available from Internet: <https://www.jstor.org/stable/2333955>.

JIANG, J.-Y. et al. Identifying users behind shared accounts in online streaming services. In: **The 41st International ACM SIGIR Conference on Research &#38; Development in Information Retrieval**. New York, NY, USA: ACM, 2018. (SIGIR '18), p. 65–74. ISBN 978-1-4503-5657-2. Available from Internet: <http://doi.acm.org/10.1145/3209978.3210054>.

JINDAL, H.; SARDANA, N. Web navigation prediction based on dynamic threshold heuristics. **Journal of King Saud University - Computer and Information Sciences**, 2020. ISSN 1319-1578. Available from Internet: <http://www.sciencedirect.com/science/article/pii/S1319157820303244>.

JOHNSON, S. C. Hierarchical clustering schemes. **Psychometrika**, Springer, v. 32, n. 3, p. 241–254, 1967.

KULLBACK, S.; LEIBLER, R. A. On information and sufficiency. **Ann. Math. Statist.**, v. 22, n. 1, p. 79–86, 1951.

MA, M. et al. π-net: A parallel information-sharing network for shared-account cross-domain sequential recommendations. In: **ACM SIGIR Conference on Research and Development in Information Retrieval**. New York, NY, USA: ACM, 2019. (SIGIR'19), p. 685–694. ISBN 9781450361729. Available from Internet: <https://doi.org/10.1145/3331184.3331200>.

MAATEN, L. van der; HINTON, G. Visualizing data using t-SNE. **Journal of Machine Learning Research**, v. 9, p. 2579–2605, 2008. Available from Internet: <http://www.jmlr.org/papers/v9/vandermaaten08a.html>.

MACQUEEN, J. et al. Some methods for classification and analysis of multivariate observations. In: OAKLAND, CA, USA. **Proceedings of the fifth Berkeley symposium on mathematical statistics and probability**. [S.l.], 1967. v. 1, n. 14, p. 281–297.

MARIMONT, R. B.; SHAPIRO, M. B. Nearest Neighbour Searches and the Curse of Dimensionality. **IMA Journal of Applied Mathematics**, v. 24, n. 1, p. 59–70, 08 1979. ISSN 0272-4960. Available from Internet: <https://doi.org/10.1093/imamat/24.1.59>.

MOREIRA, G. de S. P. Chameleon: A deep learning meta-architecture for news recommender systems. In: **Proceedings of the 12th ACM Conference on Recommender Systems**. New York, NY, USA: ACM, 2018. (RecSys '18), p. 578–583. ISBN 978-1-4503-5901-6. Available from Internet: <http://doi.acm.org/10.1145/3240323.3240331>.

MOREIRA, G. de S. P.; FERREIRA, F.; CUNHA, A. M. da. News session-based recommendations using deep neural networks. In: **Proceedings of the 3rd Workshop on Deep Learning for Recommender Systems**. New York, NY, USA: ACM, 2018. (DLRS 2018), p. 15–23. ISBN 978-1-4503-6617-5. Available from Internet: <http://doi.acm.org/10.1145/3270323.3270328>.

SEMBIUM, V. et al. Bayesian models for product size recommendations. In: **Proceedings of the 2018 World Wide Web Conference**. Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee, 2018. (WWW '18), p. 679–687. ISBN 9781450356398. Available from Internet: <https://doi.org/10.1145/3178876.3186149>.

SIBSON, R. Slink: an optimally efficient algorithm for the single-link cluster method. **The computer journal**, Oxford University Press, v. 16, n. 1, p. 30–34, 1973.

SILVERMAN, B.; JONES, M. C. E. fix and j.l. hodges(1951): an important contribution to nonparametric discriminant analysis and density estimation. **International Statistical Review**, v. 57, n. 3, p. 233–247, 1989. Available from Internet: <http://oro.open.ac.uk/28327/>.

SOTTOCORNOLA, G.; SYMEONIDIS, P.; ZANKER, M. Session-based news recommendations. In: **Companion Proceedings of the The Web Conference 2018**. Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee, 2018. (WWW '18), p. 1395–1399. ISBN 9781450356404. Available from Internet: <https://doi.org/10.1145/3184558.3191582>.

SREBRO, N.; JAAKKOLA, T. S. **Learning with Matrix Factorizations**. Thesis (PhD) — Massachusetts Institute of Technology, USA, USA, 2004. AAI0807530.

STEINHAUS, H. Sur la division des corps matériels en parties. **Bull. Acad. Polon. Sci**, v. 1, n. 804, p. 801, 1956.

TAKANE, Y.; YOUNG, F. W.; LEEUW, J. D. Nonmetric individual differences multidimensional scaling: An alternating least squares method with optimal scaling features. **Psychometrika**, Springer, v. 42, n. 1, p. 7–67, 1977.

TEAM, T. pandas development. **pandas-dev/pandas: Pandas**. Zenodo, 2020. Available from Internet: <https://doi.org/10.5281/zenodo.3509134>.

VERSTREPEN, K.; GOETHALS, B. Top-n recommendation for shared accounts. In: **Proceedings of the 9th ACM Conference on Recommender Systems**. New York, NY, USA: Association for Computing Machinery, 2015. (RecSys '15), p. 59–66. ISBN 9781450336925. Available from Internet: <https://doi.org/10.1145/2792838.2800170>.

ZHANG, A. et al. Guess who rated this movie: Identifying users through subspace clustering. **CoRR**, abs/1408.2055, 2012. Available from Internet: <http://arxiv.org/abs/1408.2055>.

ZHANG, L.; LIU, P.; GULLA, J. Dynamic attention-integrated neural network for session-based news recommendation. In: . [S.l.: s.n.], 2019. v. 108.