Titre: Title:	Techniques to Infer the Number of Latent Dimensions
Auteur: Author:	Asana Neishabouri
Date:	2021
Type:	Mémoire ou thèse / Dissertation or Thesis
Référence: Citation:	Neishabouri, A. (2021). Techniques to Infer the Number of Latent Dimensions [Thèse de doctorat, Polytechnique Montréal]. PolyPublie. https://publications.polymtl.ca/9476/

POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Techniques to Infer the Number of Latent Dimensions

ASANA NEISHABOURI
Département de génie informatique et génie logiciel

Thèse présentée en vue de l'obtention du diplôme de *Philosophiæ Doctor*
Génie informatique

Octobre 2021

POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Cette thèse intitulée :

Techniques to Infer the Number of Latent Dimensions

présentée par **Asana NEISHABOURI**
en vue de l'obtention du diplôme de *Philosophiæ Doctor*
a été dûment acceptée par le jury d'examen constitué de :

Michel GAGNON, président
Michel DESMARAIS, membre et directeur de recherche
Foutse KHOMH, membre
Pier-Olivier CARON, membre externe

DEDICATION

To my beloved grandmother, in loving memory,

ACKNOWLEDGEMENTS

I would like to express my warmest thanks to my supervisor Professor Michel C. Desmarais for his great guidance, patience, and attitude throughout my Ph.D. journey which has been a long and sometimes frustrating process, and without his support and encouragement, I could not make it. It has been a privilege to work with you.

Next, I would like to express my deep appreciation to friend of my life, Mahdi, who is my husband. My husband and I, started our new path of life together since we started our study in Italy, and then we continued our adventure and growth in Canada which was beginning of my Ph.D journey. Completing this thesis took several years of my life and although the world of research had many great experiences and learning for me individually , it brought some limitations, deadlines, stresses and changes in our life. But he stood by me patiently in all the hard times that I had during my Ph.D period and I had his constant support and advise that prevented several wrong decisions that I am very grateful for that. My next big thanks go to my family in particular my mother and my lovely sister for their indirect contribution to this work by their continuous encouragement to achieve my goal. I am sincerely grateful for presence of my friends and their attention during my Ph.D. process.

RÉSUMÉ

Trouver le nombre de variables latentes/cachées, ou le nombre de facteurs/dimensions latentes est un problème omniprésent dans de nombreux domaines. Le seul fait de pouvoir référer à ce concept avec une telle variété de vocabulaire dénote de son importance à travers les disciplines. De manière générale, nous référons aux dimensions latentes comme des facteurs qui peuvent expliquer la structure des données mais qui ne sont jamais directement observables.

Le principe fondamental qui sous-tend l'estimation du nombre de dimensions latentes (DL, ou LD dans le texte anglais) est qu'on peut transformer un ensemble de données en une représentation plus compacte autour d'un nombre réduit de dimensions en minimisant la perte d'information. On présume que la représentation des données sous dimensions réduites contient moins de bruits et permet d'entraîner des modèles qui sont plus parcimonieux et souvent plus précis dans leurs prédictions. Le principe est fréquemment appliqué à domaines très variés tels la recherche d'information, la psychométrie et la psychologie, la modélisation des thématiques de textes, l'agglomération non supervisée, et les systèmes de recommandations pour ne nommer que ceux-ci.

Les techniques de factorisation matricielle sont de bons exemples où nous devons déterminer le nombre de dimensions latentes avant la phase d'apprentissage. Des modèles non linéaires tels que LDA et les réseaux de neurones sont également confrontés au problème de l'indication du nombre de sujets et de nœuds à inclure dans le modèle avant d'exécuter une analyse sur un ensemble de données, un problème qui revient à trouver le nombre de facteurs latents qui est ensuite utilisé pour diverses tâches de prédictions ou de modélisation.

Il est intéressant de noter que chaque domaine a ses propres méthodes de choix pour résoudre ce problème et peu d'études empruntent des méthodes d'autres domaines. Nous étudions l'efficacité de méthodes pour induire le nombre de DL provenant du domaine des statistiques, de la psychométrie et de l'apprentissage automatique. La performance de chaque méthode est analysée en fonction des caractéristiques des ensembles de données. Les résultats avec des données synthétiques et réelles révèlent non seulement de grandes différences en fonction de ces caractéristiques, mais en outre il n'existe pas de méthode universelle qui fonctionne le mieux dans toutes les conditions de données. Cette information est mise à profit afin (1) d'élaborer une approche qui fournit une estimation plus précise du nombre de DL et (2) un indicateur de la fiabilité de l'estimation obtenue.

Pour la première tâche, afin de considérer les caractéristiques des données ainsi que les estimations de LD de différentes méthodes, nous proposons une approche dite *ensembliste* pour

combiner les résultats de plusieurs méthodes et obtenir une estimation de LD, meilleure que n'importe quelle méthode unique.

Pour la seconde tâche, nous démontrons que la variance des valeurs estimées à travers les différentes méthodes est un indicateur de l'accuité de la valeur obtenue par la méthode ensembliste.

Enfin, notre étude étend la comparaison de méthodes au domaine de la modélisation des thématiques de textes. Nous proposons notamment une méthode pour induire le nombre de thèmes dans une structure de thématiques imbriquées, typiques de documents qui ont une base commune mais qui approfondissent un sujet progressivement en introduisant du vocabulaire plus spécialisé.

ABSTRACT

Finding the right number of latent variables/hidden variables, or latent factors/dimensions, is a ubiquitous problem. The various vocabulary that is used in many fields of study and refers to the concept of latent dimensions (LD), or to subtle nuances of this concept, is a clue to how widespread and important it is. LD relates to factors that cannot be observed directly but can only be inferred from the observed variables.

The fundamental aim behind the estimation of the number of LD in a data set is that the data can be transformed to the lower-dimensional representation with a minimal loss of information. The lower-dimensional data is assumed to contain less noise and allows to build models that can significantly improve their prediction. Such models can cover many tasks such as information retrieval, psychology and psychometrics, topic modeling, clustering, and recommender systems, among others.

Matrix factorization techniques are good examples where we need to determine the number of latent dimensions prior to the learning phase. Non-linear models such as LDA and neural networks also face the issue of stating the number of topics and nodes to include in the model before running an analysis over a data set, a problem that is akin to finding the number of latent factors which is then used for various predictions or modeling tasks.

Interestingly, each application domain has its own methods of choice to solve this problem and few studies borrow methods from outside their fields. We investigate the effectiveness of popular methods to induce the number of LD from the fields of factor analysis, quantitative psychology, and machine learning. The performance of each method is analyzed over datasets with different characteristics. Our experimental results over synthetic and real datasets reveals that data characteristics have a crucial effect on the methods performance and that there is no universal method that performs best in all data conditions. We leverage these information to (1) obtain a more accurate overall estimate and (2) as an indicator of the reliability of the LD estimate.

For the first task, we propose an ensemble method approach to combine the results from multiple methods and obtain an estimate of LD. Results show the approach performs better than any single method.

On the second task, we show that the variance across the method's estimates is a good indicator of the correctness of the obtained LD from the ensemble approach. We also investigate estimating the number of topics for topics modeling using multiple methods. In addition, we

compare popular methods of topic modeling for document clustering with an embedded topic structure, and propose novel methods to find the number of topics and their order, where topics are subsets of each other is other contributions of this work.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ACRONYMS

LD	Latent Dimension/Latent Variable/Latent Factors/Hidden Variable/Latent Topic
DS	Dataset
DT	Document-Topic
SVD	Singular Value Decomposition
RSVD	Randomized Singular Value Decomposition
RS	Recommender Systems PA
Parallel Analysis	
mPA	Parallel Analysis with average of matrices eigenvalues
cPA	Parallel Analysis with 95 percentile of matrices eigenvalues
MAP	Minimum Average Partial
MAP1	Minimum average partial with squared partial correlation
MAP2	Minumum average partial with forth power partial correlation
BCV	Bi-Cross Validation
BCV.G	Gabriel style of Bi-Cross Validation
BCV.W	Wold style of Bi-Cross Validation
EINLD	Ensemble Inference of the Number of Latent Dimensions
EINRLD	Ensemble Inference of the Reliability of the Number of Latent Dimensions
LDA	Latent Dirichlet Allocation
K	Number of topics
α	Document-topic density
β	topic-word density
HDP	Hierarchical Dirichlet Process
HTM	Hierarchical Topic Model
CTM	Correlated Topic Model
CE	Conditional Eigenvalues
condClust	Conditional Clustering

LIST OF APPENDICES

CHAPTER 1 INTRODUCTION

The problem of determining the number of latent dimensions is a common issue in non-supervised learning algorithms including neural networks. Many linear and non-linear models, from matrix factorization in recommender systems (RS), to clustering, and to topic modeling, face the issue of having to state the number of LD to include in the models before running an analysis over a dataset. All these tasks have the same issue of defining the right number of LD.

This work attempts to tackle this problem and we introduce a new method RSVD. We compare RSVD with the standard techniques of different fields of study which some of them are not known in the machine learning. We construct various experiments considering real and synthetic datasets with different characteristics to compare performance of the methods under different circumstances of data in RS and document-topic clustering domains.

Comparisons revealed that sometimes methods outside of a field outperform the commonly used methods within a field. This finding cast a new light on the task of finding right number of LD that in fact there is no universal method that performs best in all data conditions and the performance of the methods depends on data attributes. This evidence leads us to propose a novel multi-method approach EINLD that takes this finding to account and outperform other methods.

Another important advantage of using multiple methods is that we can infer the reliability of the obtained LD from the variance across the method's estimates which is not possible with any individual method. Although reliability of a method in finding the number of LD is very difficult to measure, we propose an ensemble technique EINRLD to assess the reliability of the obtained LD considering information gathered from multiple methods and the dataset attributes.

Afterward, we extend our study to evaluate the effectiveness of the methods from different fields of study in finding the number of topics where topics are subsets of each other and compare them with our proposed method CE. However, this task is mostly seen in Hierarchical Topic Modeling (HTM) or Correlated Topic Model (CTM), we aim to propose an automatic algorithm that does not need the consumer intervention to define any parameter to run an analysis.

The results of the designed experiments over synthetics and real datasets found clear boundaries for the effectiveness of the two methods of RSVD and CE. This finding employed to

find the order of topics after clustering documents with embedded topic structure using our new proposed method condClust. Although the generality of the current results must be established by future research, the present study provides clear support not only for finding the number of topics of documents with embedded topic structure, but furthermore it clusters documents based on that and return the order of topics/clusters.

Note that in this thesis we focus on some of the popular linear methods from different fields of study to find the number of LD in two domains of recommender systems (RS) and document-topic clustering for topic modeling using Latent Dirichlet Allocation (LDA) model. Each of these fields has its own criteria and measurements to address the problem of finding the number of LD. Although this work does not include all the existing methods for finding the number of LD and that set of methods that we considered are limited to the popular methods of each field, our goal is to show that, first, even the commonly used method of each field does not consistently perform better. Second, using multiple methods improve the results. Moreover, non-linear techniques including neural network methods are out of scope of this thesis and will remain for the future work.

1.1 Latent dimensions (LD) across different fields

Let us first briefly introduce the fields of study and the underlying methods that are the main focus of this thesis. And then later on in literature review we discuss more details about them.

1.1.1 LD in exploratory factor analysis

Exploratory factor analysis includes popular statistical techniques that are commonly used in the social and behavioral sciences and psychology to find latent factors for different tasks such as dimension reduction [2, 3]. The methods are widely used in statistical software such as SPSS. The most successful methods such as Parallel Analysis (PA) and Minimum Average Partial (MAP) are correlation based methods that each of which has two variations to find the number of LD. PA and MAP are the alternative methods that we borrow them from social science and psychology fields of study to include them in this study to find the number of LD/latent factors alongside the popular methods in recommender systems and topic modeling.

1.1.2 LD in recommender systems (RS):

Recommender system is an application of machine learning that aim to discover the user's preference on items using different algorithms. Collaborative filtering is one of the most popular approaches that is for suggesting appropriate items to users by predicting user's rating on items considering user's past behavior. The main problem of this method is the scalability and sparsity of datasets. One of the popular techniques to handle these issues is the Singular Value Decomposition (SVD) technique. SVD is a well known matrix factorisation technique from linear algebra that commonly is used as a dimensionality reduction technique in machine learning to reduces the number of variables of a dataset by reducing the space dimension from n-dimension to r-dimension where $r << n$. In the context of the recommender system, SVD is used in the collaborative filtering technique to improve prediction by transferring users and items to r-dimension as the latent dimension that can represent the relationship between users and items. However, finding the right number of r as the latent dimensions to truncate SVD is vital for this task.

Many variations of SVD are proposed in literature to define the lower rank or the number of latent dimensions of a matrix to truncate SVD. Truncated SVD only consider the first r singular values of data, where r is usually specified through *wrapper* approach to find the best rank to truncate the SVD via cross validation for minimizing prediction error. This technique shows effective and widely accepted in recommender systems [4].

Therefore, we review SVD based methods of Bi-Cross-Validation (BCV) and our proposed method Randomized Singular Value Decomposition (RSVD) as the techniques to find the number of LD in RS.

1.1.3 LD in document-topic (DT) clustering:

Latent Dirichlet Allocation (LDA) is an unsupervised learning method that was originally introduced by [5] and is arguably the most widely used method for document-topic clustering for the task of Topic Modeling. In the field of Topic Modeling, the problem of finding the number of latent dimensions/latent variables translates to the task of finding the number of latent topics (LT). In LDA, the number of topics that best represents a corpus of documents has to be determined in advance. However, the "true" number of underlying of topics is unknown, many studies [5, 6, for eg.] use perplexity which is the best known metric to identify the number of topics along with a wrapper technique that has the minimum perplexity [5, 7, 8, 9, 6, 10, for eg.]. Although other techniques such as Hierarchical Dirichlet Process are proposed to skip defining number of LT in LDA, some of the studies address the limitation

and shortcoming which we discuss it later. Thus, we consider perplexity as the standard technique to find the number of LT. Besides, we evaluate the performance of the SVD based and EFA methods for the task of finding the number of topics.

In this thesis, we aim to compare a range of linear methods to derive the number of LD considering datasets with different characteristics. Some of the methods are well known in other fields such as social science, yet almost unknown in the machine learning community. We investigate the popular methods of Exploratory Factor Analysis (EFA) in finding the number of LD in RS and topic modeling tasks. We show they perform to a level roughly similar to a popular machine learning approach, the wrapper technique [11], yet sometimes better under certain conditions of sparsity and data characteristics. Thus, we can state that one of the main factors to the success of a method depends on data representation and we aim to show that the performance of various methods in finding the number of LD depends on the different data characteristics such as sparsity, distribution, imputation and size and derive a method that takes this finding into account. Moreover, we assess the reliability of the obtained LD which is one of the main contributions of this work. We extend this work with introducing two novel methods to find the number and order of topics where topics have an embedded structure across documents.

1.2 Research Questions

The research questions addressed in this thesis are below:

1. What is the impact of data characteristics in finding the number of latent dimensions?
2. Can we derive a method that improves LD estimate by considering data specifications?
3. How to assess reliability of the obtained number of LD/LT?
4. Can we develop a method to improve LT estimation where topics have an embedded structure and find the order topics?

1.3 Contributions

The current thesis aims to bring a new approach to the problem of finding the number of latent dimensions/latent topics (LD). We first assess and compare the performance of multiple methods from different fields of study in finding the number LD over synthetic and real datasets. We investigate the effect of data characteristics comprehensively. Validation over synthetic data is our methodological choice because we know the ground truth of the generated data and we can control the different characteristics to investigate. These investigations lead us to the proposed approach of combining methods not only to better estimate the number of LD, but also to better evaluate the reliability of the estimated LD.

We explain more details of our contributions and proposed methods that are published and submitted in different conferences and journal below:

1.3.1 First article: An ensemble approach to determine the number of latent dimensions and assess its reliability

Neishabouri A, Desmarais MC, An ensemble approach to determine the number of latent dimensions and assess its reliability, submitted to Journal of Communications in Statistics-Simulation and Computation 2021.

In this study, we investigate the problem of finding the number of LD in real and synthetic datasets with different characteristics considering methods from different fields of study. The investigation mainly relies on synthetic generated datasets with normal and multinomial distributions that is a close distribution to a rating and document-term matrices. We aim to evaluate methods performance by assessing their capacity to infer LD considering dataset

attributes. Therefore, we address the research questions of what is the impact of data characteristics in finding the number of LD? And, Can we derive a method that improves LD estimate and evaluate the reliability of the obtained LD?

In this work, first, we extend a SVD based method and make it fully automatic and named it "Randomized Singular Value decomposition" (RSVD). It is a novel method that initially presented in [12]. Then we design different experiments to show the behavior of each method mentioned in the introduction including RSVD over datasets with different attributes. A comparison of the methods performance highlight that there is no universal method that can perform best in all data conditions. Moreover, it shows that when the methods have an accurate or a close estimate to the ground truth, there is a very small variance across the methods. Hence, considering several methods from different fields is more reliable than a single popular method within a certain field. We take advantage of this information to both obtain a more accurate overall estimate and as an indicator of its reliability.

Thus, we propose a new method "Ensemble Inference of the Number of Latent Dimensions" (EINLD) to exploit this information to derive a more accurate estimate using a multi-method approach.

Another significant contribution of this study is that we assess the reliability of the obtained number of LD which is a challenging task. We show that the common approach that is to use bootstrapping and look at the variance of the results could be misleading in this task because of the systematic error of each method. Our experimental results provide an insight that the variance across the method's estimates is a stronger indication to the reliability of the obtained number of LD than the variance across the multiple runs of the individual methods. Therefore, we propose a novel method "Ensemble Inference of the Reliability of the Number of Latent Dimensions" (EINRLD) that predict the expected loss category of the methods estimates to infer the reliability of the obtained LD [13].

1.3.2 Second article: Reliability of perplexity to find number of latent topics

Neishabouri A, Desmarais MC. Reliability of perplexity to find number of latent topics. In Proceedings of the Thirty-Third International Flairs Conference, 2020, 246–251.

We address the problem of finding the correct number of latent topics for document-topic clustering using Latent Dirichlet Allocation (LDA) over synthetic datasets with different characteristics. We investigate the performance of the methods outside of the typical topic modeling studies and compare them with the standard metric "Perplexity".

In this study, we address the research questions of what is the impact of data characteristics in

finding the number of topics? and what is the reliability of the popular method "Perplexity" in finding the number of topics? In fact, we claim that the commonly used method "perplexity" is not always reliable.

We explore the effect of dataset characteristics such as sparsity, number of topics (K), document-topic density (α) and topic-word density (β) in methods performance and we show that sparsity has a higher impact in the methods performance. Moreover, our experimental results stating the importance of considering multiple methods for this task. And, it shows that perplexity has an opposite error direction compared to the other methods, which can be an indicator of the reliability of the obtained number of topics [1].

1.3.3 Third article: Estimating the number of latent topics through a combination of methods

Neishabouri A., Desmarais M.C. Estimating the Number of Latent Topics Through a Combination of Methods. In 25th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems, Procedia Computer Science, 2021, 192, 1190-1197.

After demonstrating the most successful methods and boundaries of their effectiveness in finding the number of topics over synthetic datasets, we extend our study to evaluate the findings over real datasets. In this study, we address the main limitation of the studies over real datasets that is oftentimes the real ground truth of the underlying dataset is presumed and is not assessed.

Thus, we propose a novel technique "Cosine inter-intra topic" to assess the validity of the ground truth assumption of the underlying real datasets. Then, we simulate the experiments that were conducted over synthetic data over real datasets.

Another contribution of this study is to demonstrate that the variance across methods provides an indicator of the reliability of the estimates obtained by any method. This study suggests the possibility to not only provide more accurate estimates of the number of latent topics, but also assess the reliability of the estimates [14].

1.3.4 Fourth article: Inferring the number and order of embedded topics across documents

Neishabouri A, Desmarais MC, Inferring the Number and order of Embedded Topics Across Documents. In 25th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems, Procedia Computer Science, 2021, 192, 1198-1207.

After previous studies that were constructed over synthetic and real datasets to find the number of topics more accurately using multiple methods, we address the main limitation of them that they were conducted over documents that have a single dominant topic and the topics are not related to each other, while in reality documents mostly have more than one topic or mixture of topics that are not independent. In this study, we expand our work by addressing this limitation with the research question of can we develop a method to find the number of topics where topics have an embedded structure? and can we find the order of topics in such datasets?

Initially, we tackle this problem by proposing a novel method "Conditional Eigenvalue" (CE) to find the number of latent topics where topics are embedded within each other (topics are subsets of each other) and compare it with the best performed methods in the previous studies and find the boundary of their effectiveness.

In addition, we claim that for documents structured according to such embeddings, the tasks of inferring the number of clusters according to the underlying topics and identifying their embedding order are not handled effectively by the popular topic modeling techniques such as latent Dirichlet allocation (LDA) and Correlated Topic Model (CTM). Thus, we develop a new method "Conditional Clustering" (condClust) that contrary to LDA and CTM does not have any assumption of particular distribution of topics over documents and words per topics. And, it has a built-in function to find the number of topics/clusters that makes it fully automatic. Moreover, it can infer order of the topics using conditional probability after clustering documents [15].

1.4 Organization of Thesis

We first review the most successful method in finding the number of latent dimensions of different fields such as social science, recommender systems, and topic modeling. Then, we report the published and submitted articles $[1-4]$ to provide the details of our proposed methods and the related experimental results in subsequent chapters of $3:6.1$. Next, we provide general discussion of the results and limitations of this study before conclusion.

CHAPTER 2 LITERATURE REVIEW

The problem of finding the number of latent factors in a dataset dates back to early work by Guttman [16]. It extends to a large array of fields including psychology and social science [17], bioinformatics [18], information retrieval [19], and of course statistical learning [20, chapter 14]. Review of latent variables analysis and its applications can be found in [21, 22, 23].

Interestingly, not all methods are known across these fields. Methods such as Parallel Analysis are well known in quantitative psychology, psychometrics and the social sciences, but just about unknown in the machine learning field and considered underutilized [24]. Conversely, the method of choice for Machine Learning, the wrapper method [11], is rarely seen in social sciences.

We divide the literature review into three sections. In the first section, we investigate the most successful factor analysis techniques for finding the number of factors to retain from psychometric and social science namely Kaiser's eigenvalue-greater-than-one rule (K1), Parallel Analysis (PA), Minimum Average Partial test which is known as Velicer's MAP test.

In the second section, we investigate the commonly used techniques that are based on SVD such as Bi-Cross-Validation (BCV) of the SVD. SVD is a powerful matrix factorization technique to find the number of latent factors and is popular in machine learning and the recommender systems field.

Then, we review the Latent Dirichlet Allocation (LDA) technique for document-topic clustering and the issue to find the number of topics in the third section.

2.1 Factor Analysis

The main concept of factor analysis is to discover latent variables from the observed variables. Factor analysis is a a family of methods that can be used to find the latent factors driving observable variables. This is usually done through computing eigenvalues of the correlation matrix. The eigenvalues show how much a factor implies the variance of the observed variables. Here, we review some of the most popular techniques that we used in our experiments.

2.1.1 Kaiser's eigenvalue-greater-than-one rule (K1)

The K1 method was first introduced by Guttman [16] and later extended and popularized by Kaiser [25]. The method relies on the eigenvalues of the correlation matrix of the observed factors and stipulates that the number of eigenvalues greater than one corresponds to the number of latent factors to retain.

While this method is straightforward, some researchers consider it unreliable [26, 27, 28]. We will nevertheless include it in our comparison experiments, given that it is a classic method and the first that introduced the use of eigenvectors of the correlation matrix for determining the number of latent factors.

2.1.2 Parallel Analysis (mPA and cPA)

Akin to K1, Parallel Analysis is also based on the correlation matrix between the observed factors. It generates many random datasets with the same size as the original dataset . The eigenvalues of the correlation matrix of each of these matrices are computed and stored. The number of eigenvalues greater than the average random datasets eigenvalues indicate to the number of latent dimensions [27, 26]. This strategy was originally proposed by Horn [29]. Warne [30] showed that PA improves over the Eigenvalue-greater-than-one rule. Several researchers found this method appropriate and more accurate in determining the number of factors to retain [31, 27]. We will see that PA has a close relationship with PSVD below and this is corroborated by the closeness of the results.

The steps of the PA algorithm are:

1. Compute the eigenvalues of the correlation matrix of the original dataset .
2. Generate a set of randomized matrices with the same size as the original source matrix.
3. Compute the mean or 95th percentile of the correlation matrix eigenvalues obtained from the randomized datasets.
4. Compare the eigenvalues from the original matrix with the average eigenvalues obtained from the randomized datasets.
5. The number of latent dimensions corresponds to the number of the eigenvalues of the original dataset that are greater than the average eigenvalues obtained from the random generated datasets.

We will use two variants of PA, one that relies on the mean eigenvalue rule, mPA, vs the 95th percentile eigenvalue rule, cPA [32]. We call them "mPA" and "cPA" respectively.

2.1.3 Minimum Average Partial method (MAP1 and MAP2)

The Minimum Average Partial method is introduced by Velicer [33]. It is based on Principal Component Analysis (PCA) and relies on the series of partial correlation matrices to define the number of significant factors to retain [27, 26, 34, 35]. A Partial correlation is obtained by controlling the effect of other variables. For instance, to compute the correlation between x and y where we have three variables (x, y, z), the partial correlation between variables x and y is computed by removing the variance explained by the third variable z according to this formula:

$$r_{xy.z} = \frac{r_{xy} - (r_{xz} \times r_{yz})}{\sqrt{((1 - r_{xz}^2)(1 - r_{yz}^2))}}$$

The principle of this method is to derive correlations in a stepwise manner, removing the partial correlation of a co-variable. The correlation result is referred to as the squared partial correlation. The number of factors to retain is defined as the point where the minimum average of the squared partial correlations is obtained [33]. This method is revised in [36] and they suggest instead of squaring the partial correlation, compute the fourth power, which in our experiments we call them MAP1 and MAP2 respectively. More details about this method are given in [37, 36, 2]. In general, statisticians agree that the MAP and PA are the two most reliable techniques to extract the number of factors to retain with the reasonable result [26, 27, 35]. We will see that our results partially confirm these conclusions.

2.2 Matrix Factorization

Matrix factorization is an unsupervised learning technique to decompose a matrix into its constituent parts for finding the latent variables of data and dimensionality reduction. Survey [38] investigates the most established matrix factorization techniques and conclude that SVD based methods are the most accurate approaches in recommender systems. [4] also compare k-means-SVD-based recommendation with k-means-based recommendation and k-nearest neighbor-based recommendation, and show that k-means-SVD-based recommendation outperform the other two methods.

2.2.1 Bi-Cross-Validation of the SVD (BCV.W and BCV.G)

Singular Value Decomposition (SVD) is a well-known matrix factorization technique that decomposes the original matrix, $\mathbf{R}$, into the product of two eigenvector matrices, the eigenvectors of the cross-product of the rows and columns, and of the diagonal matrix of their

common singular values:

$$\mathbf{R} = \mathbf{U}_{m\times m}\ \mathbf{\Sigma}_{m\times n}\ \mathbf{V}^T_{n\times n}$$

where $\mathbf{U}$ and $\mathbf{V}$ are orthonormal, and $\mathbf{\Sigma}$ is a diagonal matrix with non-negative real values. Computing SVD involves of finding the eigenvalues and eigenvectors of $\mathbf{R}\mathbf{R}^T$ and $\mathbf{R}^T\mathbf{R}$. The eigenvectors of $\mathbf{R}^T\mathbf{R}$ and $\mathbf{R}\mathbf{R}^T$ make the columns of V and $\mathbf{U}$ respectively.

The singular values represent the importance of the eigenvectors, ordered by decreasing values.

The Bi-Cross-Validation (BCV) technique is a *wrapper* method to find the best rank to truncate the SVD via cross validation for minimizing prediction error [39].

The BCV technique has two variations: Gabriel-style (BCV.G) and Wold-style (BCV.W) cross validation. Both variations consists in splitting the data into a training and test sets. Prediction is done with a truncated (lower rank) product of the factorization. The prediction error is measured as the sum of squares of residuals between the truncated SVD and the original matrix. Determining the number of LD relies on a comparison of the residual error over a random set of values for the test set.

In the Wold-style cross validation, the test set is a random set of values in the matrix. The difference between Wold- and Gabriel-style is that, for the latter, the test set is "blocked". It holds out a certain number of rows and columns of a matrix simultaneously as a test set. BCV.G divides the rows of the matrix into k segments and the columns into h segments. The total number of folds are $k \times h$ which refer to the number of blocks. In each step, one of the blocks is considered as the test set and the remaining blocks are as the training set. More details about this method are given in [40, 41].

Owen et al. [40] report that BCV has a better result than other state-of-the-art methods without considering any missing value and imputation problem in the underlying data and where the size of dataset and the number of variables are large. It is worth mentioning that this study did not include the methods such as PA, K1 and MAP. Several studies indicate that the Wold-style cross validation performs better but is slower than Gabriel-style [41, 39].

2.3 Topic Modeling: Latent Dirichlet Allocation (LDA)

Topic modeling is essentially a technique to identify the major themes within a given corpus of documents. It applies to newspapers articles, reviews, tweets, etc. LDA is arguably the most common approach to topic modeling considering the very large body of research and publication it generated.

LDA is a generative, probabilistic model of documents that is introduced by [5]. It has been widely used for topic modeling in many fields, from recommender systems [42] to social science [43].

This technique uses word distribution to cluster documents and discover latent topics from them. It rests on the assumptions that each document consists of a mixture of topics and that each topic consists of a set of words, both of which follow a Dirichlet prior.

It is generative in the sense that a document corpus can be created from a process of randomly choosing topics and words from stastiscal distributions. A document is considered a bag of words, in the sense that order does not matter. Each word of this document is first chosen from a topic and, given that topic, from the probability distribution of words corresponding to the chosen topic. The Dirichlet priors of the topic and word distributions allow for imposing a general structure on these distributions, for example allowing for documents of one or two topics to be more likely than others, and forcing the distribution of words to approximate a Zipf law.

Figure 2.1 contains the plate diagram that is often used to provide an overview of LDA. We find D documents each composed of N words. Each document is assigned a vector that represents its topic assignements, θ_d, and each topic is also assigned a vector that represents its word assignments, ϕ_k. Both the topic and word assignments vectors have a Dirichlet prior, α for documents, and β for topics, which allows the control of the distribution of topics per document and words per topics. The generative process for a single document first consists in first generating a θ_d distribution of topics by sampling from the Dirichlet(α). Then, for each word of the document, a topic is sampled from θ_d, and a word is then sampled for the chosen topic k from ϕ_k.

LDA has the following hyperparameters that have to be determined before the training phase:

- Alpha (α), is the Dirichlet prior of document-topic density. Higher values of alpha implies documents composed of more topics and lower values implies fewer topics per document.
- Beta (β), is the Dirichlet prior of topic-word density. A higher beta indicates that topics are composed of a large vocabulary, and a lower value implies smaller vocabularies per topic.
- K is the number of topics.

[44] suggest a value of $50/T$ for α, where T is the number of topics, and 0.1 for β. And consequently many studies such as [45, 46] as well as the "topicmodels" package [47] in R

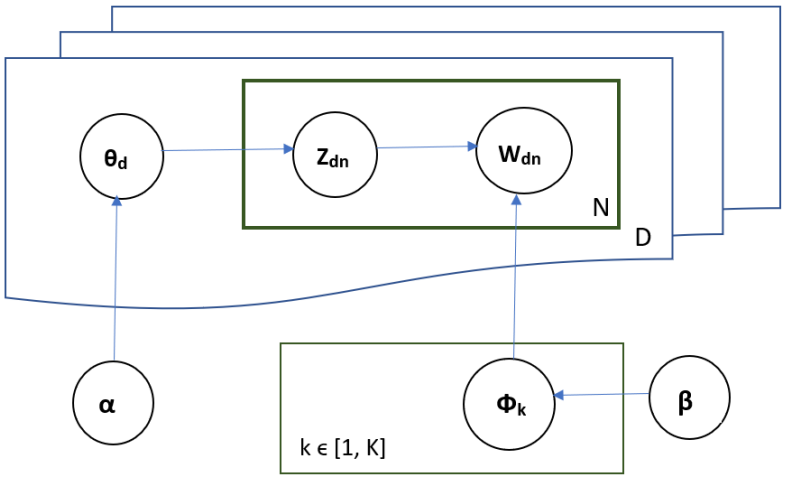


Figure 2.1 Graphical visualization of LDA. α: is a Dirichlet parameter that controls the number of topics expected in the document; β: is a Dirichlet parameter that controls the distribution of words per topic; θ_d: is document-topic distribution for document d; ϕ_k: is word distribution for topic k; z_{dn}: word-topic assignment for W_{dn}; W_{dn}: observed word (n-th word of the d-th document); K: defines the number of topics; N: vocabulary size; D: number of documents.

use of these values.

To define the number of topics, perplexity is arguably the most popular and standard metric in evaluating language modeling including LDA. For finding the number of topic parameters K, most studies [5, 7, 8, 9, 6, 10, for eg.] use perplexity along with a wrapper technique to find the number that will essentially outperform the accuracy of topic modeling.

The perplexity is defined as:

$$PP(\mathbf{W}) = P(\mathbf{W}_1\mathbf{W}_2, ..., \mathbf{W}_\mathrm{m})^{-1/m})$$

where PP and $\mathbf{W}$ refer to perplexity and word respectively, and $\mathbf{P}$ is the probability estimate assigned to document words. A model with a given number of topics that minimize perplexity value is considered an optimal model [48, 10] .

To skip defining the number of topics in advance, Hierarchical Dirichlet processes (HDP) is proposed which is a nonparametric Bayesian analysis known as the Dirichlet process (DP) mixture model [49]. It is an extension of LDA, designed for cases that the number of topics is not known a priori. Many researchers use HDP in the designed topic modeling to infer the number of topics automatically. They design a combination of HDP with LDA and another hierarchical topic modeling (HTM) to skip the number of topics such as [50, 51, 52, 53].

However, they could show a great improvement over LDA in some of the datasets but also some other researchers such as [54, 55, 56] also show that HDP based method has to deal with more hyperparameters and are not always stable which means in some other datasets LDA perform better than HDP model such as [57]. They use toy and NIPS datasets and show that in the toy dataset LDA and HDP perform similarly but in the NIPS dataset LDA performs better than HDP. However, they introduce another method called " DCNT" that outperforms both LDA and HDP but it needs a predefined number of topics as well. [58] is another research that introduces another hierarchical HDP based method called "hvHDP". They compare the behavior of different methods including hvHDP, htHDP, hierarchical Latent Dirichlet Allocation (hLDA), LDA, and Memory Model (MM) on the five different data sets (Genia, Seafood, LonelyPlanet, Elegance and NIPS data sets) and shows the performances in terms of perplexity. The results demonstrate that although hvHDP could achieve better accuracy mostly, sometimes the other methods perform better. So, we can conclude that none of the methods that learn the number of topics using HDP can systematically outperform LDA and each of them has their own limitations which could perform well under certain conditions which as mentioned in [57] could be because of the data distribution or the ability of the methods to capture non-Drichlet distribution, and that's why some other literature concludes that these methods are heuristic-based and computationally expensive which is also mentioned in [49] as the significant challenge.

So, finding the number of topics remains a necessary step to topic modeling and to document clustering. Accuracy improvements on this task represent a contemporary benefit. Although some of the researchers recently argue that the task is also relevant to more recent neural approaches as well, such as [59] who aim to find clusters within an embeddings vector space, [60] compares the different topic modeling including those with deep learning and emphasize the open problem of finding the number of topics and address the deep learning issues also. In addition, a new research [61] examines the combination of LDA and deep neural network with 2 and 3 layers (2NN DeepLDA and 3NN DeepLDA). It shows that using a deep neural network could improve the computational time of LDA but not the accuracy and in the best case, 2NN DeepLDA has almost the same accuracy as LDA. The choice of LDA in our case was motivated by its wide acceptance which is also shown in [62]. And also, we consider perplexity as a standard metric for the evaluation and finding the optimal number of topics based on the mentioned literature.

CHAPTER 3 ARTICLE 1: AN ENSEMBLE APPROACH TO DETERMINE THE NUMBER OF LATENT DIMENSIONS AND ASSESS ITS RELIABILITY

Neishabouri A., Desmarais M.C. An ensemble approach to determine the number of latent dimensions and assess its reliability. Submitted to Journal of Communications in Statistics-Simulation and Computation, 2021.(Hybrid)

Abstract

Determining the number of latent dimensions (LD) of a data set is a ubiquitous problem, for which numerous methods have been developed. We compare some of the most effective ones on synthetic data where the true number of LD is known, and show that their performance is sensitive to data set attributes such as sparsity, number of observations in relation to the number of features, and underlying feature distributions. Results also show this sensitivity is different across methods. This observation brings us to devise an ensemble technique to combine the results from multiple methods and achieve an estimate of LD, better than any single method. We also demonstrate that the variance of the methods is a good indicator of the expected loss of the LD estimate. This observation leads, in turn, to deriving a method for the assessment of the reliability of the estimate. Finally, we discuss the practical implications of the findings.

3.1 Introduction

The problem of determining the number of latent dimensions (LD), or latent factors, is ubiquitous in factor analysis and for non-supervised machine learning algorithms. It is closely associated to dimension reduction and feature selection techniques where we aim to find the optimal number to retain. Many linear and non-linear models, from matrix factorization, to clustering, and to LDA [5], face the issue of having to state the number of latent factors to include in the models before running an analysis over a data set.

In this paper, we aim to improve LD estimation by relying on the observation that the accuracy of methods to infer LD is affected by data set attributes. Some methods do better than others under certain data set attributes. Our experiments illustrate that the performance of different methods depends on the data set sparsity, distribution, imputation and the data set size. Therefore, the best choice is not universal and depends on these data set factors, which

in turn this leads to the idea of proposing a multi-method approach to improve estimating the number of LD. We hypothesize that a "meta-method" can take advantage of this information to derive a more accurate estimate. This will be referred to as an ensemble method, akin to ensemble techniques in machine learning.

Another goal of the paper is to address the question of LD estimation reliability. Obtaining a reliability estimate on the number of LD given by whatever method is used is a difficult problem. There are no specific methods to address this problem, but the generic approach is to use bootstrapping and look at the variance of the results. The wider the estimates distribution spread is, the larger is the confidence interval we would put on the LD estimate. However, if the method used has a bias error, this confidence interval will be misleading. An extreme case is a method that yields a fixed estimate from each resampling run. We would obtain a null variance and conclude in a perfectly reliable estimate. Yet, this conclusion neglects that it may have a systematic bias error that makes the estimate unreliable. In such case, estimates from a variety of methods may provide a better indicator of reliability.

We therefore investigate whether data set chacteristis and the variance of different methods are indicators of the reliability of the LD estimate. We observe that, for example, under high sparsity, neither a single method, nor an ensemble approach, can provide an accurate estimate of LD. Following principles similar to the ensemble technique that relies on multiple methods to better estimate the LD, we devise a method to estimate this reliability based on data set attributes and the distribution of error loss across different methods.

The rest of this paper is organized as follows: first, we describe the different state-of-the-art algorithms in §3.2. The experiments to analyze how the algorithms compare and behave under different data set attributes are described in section §3.3. In the following section, we propose an ensemble, multi-method approach to improve LD estimates over any single method. Next, we tackle the issue of providing a reliability assessment of the obtained LD in section 3.4.1. Finally, the conclusion and future work are discussed in section 8.

3.2 Literature Review: Latent factor analysis

The problem of finding the number of latent factors in a data set dates back to early work by Guttman [16]. It extends to a large array of fields including psychology and social science [17], bioinformatics [18], information retrieval [19], and, of course, statistical learning [20, chapter 14]. Review of latent variables analysis and its applications can be found in [21, 22, 23].

Interestingly, not all methods are known across every field. Methods such as Parallel Analysis are well known in psychometrics and the social sciences, but just about unknown in the

machine learning field and considered underutilized [24]. Conversely, the method of choice for Machine Learning, the wrapper method [11], is rarely seen in psychometrics and social science.

In the following sections, we briefly explain some of these techniques, namely the ones that we retain in our ensemble approach: Parallel Analysis (PA), Minimum Average Partial test, which is known as Velicer's MAP test, and the Bi-Cross-Validation (BCV) of the SVD, which is a popular wrapper method in machine learning, and finally RSVD which borrows a randomization process from PA and uses SVD. The choice is based on their effectiveness and popularity. However, it should be noted that principle of using a combination of techniques put forward in this paper, or "ensemble method", is not specific to any combination of method.

3.2.1 Parallel Analysis (mPA and cPA)

Parallel Analysis (PA) is part of a family of methods that rely on the analysis of the eigenvalues of the correlation matrix between factors to determine the number of LD. Guttman [16] did pionerring work in this line of research, which was and later extended and popularized by Kaiser [25] under the name K1. The K1 method stipulates that the number of eigenvalues of the correlation matrix greater than one corresponds to the number of latent factors to retain. However, while straightforward, some researchers consider the K1 method unreliable [26, 27]. This was confirmed in our own experiments and we chose to exclude it and instead focus on the more recent and effective PA method.

PA generates many random data sets of the same size as the original data set, also replicating the general distribution of the factors. The eigenvalues of the correlation matrix of each of these matrices are computed. The number of eigenvalues greater than the average random data sets eigenvalues indicate to the number of latent dimensions [27, 26]. This strategy was originally proposed by Horn [29]. Warne [30] showed that PA improves over the Eigenvalue-greater-than-one rule of the K1 method. Several researchers found this method appropriate and more accurate in determining the number of factors to retain [31, 27]. We will see that PA has a close relationship with RSVD below, and this is corroborated by the closeness of the results.

The steps of the PA algorithm are:

1. Compute the eigenvalues of the correlation matrix of the original data set .

2. Generate a set of randomized matrices with the same size as the original source matrix.

3. Compute the mean or 95th percentile of the correlation matrix eigenvalues obtained from the randomized data sets.

4. Compare the eigenvalues from the original matrix with the average eigenvalues obtained from the randomized data sets.

5. The number of latent dimensions corresponds to the number of the eigenvalues of the original data set that are greater than the average eigenvalues obtained from the random generated data sets.

We will use two variants of PA, one that relies on the mean eigenvalue rule, *mPA*, vs the 95th percentile eigenvalue rule, *cPA* [32]. We call them "mPA" and "cPA" respectively.

3.2.2 Minimum Average Partial method (MAP1 and MAP2)

The Minimum Average Partial method is introduced by Velicer [33]. It is based on Principal Component Analysis (PCA) and relies on the series of partial correlation matrices to define the number of significant factors to retain [27, 26, 34, 35]. A Partial correlation is obtained by controlling the effect of other variables. For instance, to compute the correlation between x and y where we have three variables (x, y, z), the partial correlation between variables x and y is computed by removing the variance explained by the third variable z according to the equation 6.1:

$$r_{xy.z} = \frac{r_{xy} - (r_{xz} \times r_{yz})}{\sqrt{((1 - r_{xz}^2)(1 - r_{yz}^2))}} \tag{3.1}$$

The principle of this method is to derive correlations in a stepwise manner, removing the partial correlation of a co-variable. The correlation result is referred to as the squared partial correlation. The number of factors to retain is defined as the point where the minimum average of the squared partial correlations is obtained [33].

This method is revised in [36] and they suggest instead another variant: the fourth power of the correlation, which in our experiments we call them MAP1 and MAP2 respectively. More details about this method are given in [37, 36, 2].

In general, statisticians agree that the MAP and PA are the two most reliable techniques to extract the number of factors to retain [26, 27, 35]. We will see that our results partially confirm these conclusions.

3.2.3 Bi-Cross-Validation of the SVD (BCV.W and BCV.G)

The Bi-Cross-Validation (BCV) technique is a *wrapper* approach to find the best rank to truncate the SVD via cross validation for minimizing prediction error [39]. Wrapper approaches are often used for the purpose of feature selection [21]. The general principle is to analyze the performance of a model given different subsets of factors and choose the subset that maximizes performance. In the context of dimension reduction, the optimal number of dimensions often occurs at the point where overfitting occurs the test performance starts degrading.

The underlying model for BCV is SVD. SVD is a well-known matrix factorization technique that decomposes the original matrix, $\mathbf{R}$, into the product of two eigenvector matrices, the eigenvectors of the cross-product of the rows and columns, and of the diagonal matrix of their common singular values.

$$\mathbf{R} = \mathbf{U}_{m\times m}\ \mathbf{\Sigma}_{m\times n}\ \mathbf{V}^T_{n\times n} \tag{3.2}$$

where $\mathbf{U}$ and $\mathbf{V}$ are orthonormal, and $\mathbf{\Sigma}$ is a diagonal matrix with non-negative real values. Computing SVD involves finding the eigenvalues and eigenvectors of $\mathbf{R}\mathbf{R}^T$ ($\mathbf{U}$) and $\mathbf{R}^T\mathbf{R}$ ($\mathbf{V}$). The singular values, $\mathbf{\Sigma}$, correspond to the eigenvalues of both $\mathbf{U}$ and $\mathbf{V}$ and they represent the importance of their respective eigenvector, ordered by decreasing values.

The BCV technique has two variations: Gabriel-style (BCV.G) and Wold-style (BCV.W) cross validation. Both variations consists in splitting the data into a training and test sets. Prediction is done with a truncated (lower rank) product of the factorization. The prediction error is measured as the sum of squares of residuals between the truncated SVD and the original matrix. Determining the number of LD relies on a comparison of the residual error over a random set of values for the test set.

In the Wold-style cross validation, the test set is a random set of values in the matrix. One of the differences between Wold- and Gabriel-style is that, for the latter, the test set is "blocked". It holds out a certain number of rows and columns of a matrix simultaneously as a test set. BCV.G divides the rows of the matrix into k segments and the columns into h segments. The total number of folds are $k \times h$ which refer to the number of blocks. In each step, one of the blocks is considered as the test set and the remaining blocks are as the training set. More details about this method are given in [40, 41].

Owen et al. [40] report that BCV has a better result than other state-of-the-art methods without considering any missing value and imputation problem in the underlying data and where the size of data set and the number of variables are large. It is worth mentioning

that this study did not include the methods PA and MAP. Several studies indicate that the Wold-style cross validation performs better but is slower than Gabriel-style [41, 39].

3.2.4 Singular Value Decomposition (RSVD)

This method was briefly introduced in [12] and we extend the algorithm here to make it fully algorithmic and name it Randomized Singular Value Decomposition (RSVD). This algorithm is shown below in Algorithm 3.1. We provide a full description here.

RSVD relies on SVD, as does Bi-Cross Validation. It also shares a randomization process with PA. The number of LD is determined through a comparison of the original singular values with the ones from a matrix where the original matrix values are randomly permuted columnwise.

Let us take the graph in Figure 3.2 to illustrate how the RSVD method determines the number of LD. A synthetic data set of dimension 250×150, created from 9 latent factors and with very little noise added in order to make these dimensions highly dominant, is decomposed with SVD and the singular values are plotted. One can see that the "elbow", often used to tell the number of LD, is apparent between the 8th and 9th singular values. However, the elbow method, or the "Scree plot" if PCS is used instead of SVD, are based on visual heuristics and deemed subjective. RSVD uses the *intersection* of the eigenvectors from the randomized matrix as an indicator of the number of LD. The simple rule is to choose the number LD that immediately follows the original SVD singular values curve that crosses the singular values of randomized data (red), at $LD = 9$.

3.3 Comparison of LD estimation methods under different data set attributes

We investigate how data set attributes affect the performance of different methods to estimate the number of LD. This investigation will lead us towards the goal of deriving an ensemble method to combine estimates from different individual methods under data with specific attributes, and eventually obtain a more accurate estimate.

While the performance of some of the methods covered in the literature review has been partially assessed and compared in previous works (see [21, 22, 23]), we aim here to show that data set attributes affect the estimates of each individual method. In particular, we wish to assess the methods' sensitivity to sparsity, feature distributions, imputation of missing values techniques, and the size of the data set.

Note that in order to generate experimental data set conditions that are comparable, the ex-

Algorithm 1 RSVD Algorithm

INPUT: data set (DS)
OUTPUT: the number of latent dimensions (LD)

```
1: procedure RSVD Algorithm
2:     R ← Normalize(DS);   ▷ centering and scaling by column on the (mean = 0 , sd = 1)
3:     R.svd ← SVD(R);
4:     R.d ← extract singular values(R.svd);
5:     R.randomized ← extract singular values of randomized(R);
6:     LD ← first index where R.d ≤ R.randomized;
7:     if(normalityTest(DS)==true) :
8:         return (LD − 1)
9:     else:
10:        return (LD);
```

Figure 3.1 Algorithm RSVD

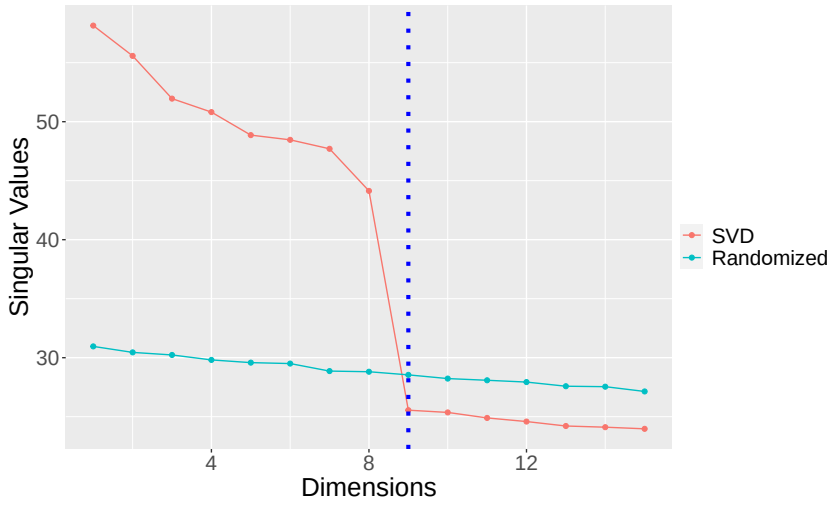


Figure 3.2 Singular values of a dense synthetic data set with a multinomial distribution and LD = 9 and its randomized. The intersection indicates the number of LD.

periments must rely on synthetic data. While real data would provide better generalizability to our conclusions, synthetic data remains the most reliable validation methodology given that we know the ground truth behind the synthetic data, and we can control the factors at play, which we cannot do with real data. We describe the different data set attributes under study below.

Note also that the choice and the range of possible values for each attribute is not exhaustive. The aim is to demonstrate the general principle that these attributes affect each method's performance differently and it is this difference that can be exploited to obtain a more accurate estimate of the number of LD as well as the reliability of this estimate. We later return to the issue of generalizing the conclusions of this investigation to a larger range of LD and data set attributes values.

3.3.1 Data Set attributes

The attributes of the data sets that affect each method's ability to correctly derive the underlying number of LD are listed and detailed below.

Size We explore the performance of the methods under data sets of four sizes:

Size 1 =	250×150
Size 2 =	250×250
Size 3 =	500×300
Size 4 =	943×1682

We will consider the dimensions $x \times y$ to respectively represent the number of observations by the number of factors, or features.

Sparsity Sparsity is related to the amount of data available, the imputation method, and the data matrix dimensions (number of observations by the number of factors). We study the performance of the methods from dense matrices to the highly sparse. For the data sets of size 1 to size 3, we consider twelve levels of sparsity: from 0% (dense), to 90%, by 10% increments, and an additional two levels: 96% and 98% which are representative of high sparsity data commonly find in numerous contexts such as recommender systems and term-document matrix.

Imputation Since all methods we investigate do not handle missing values, we examine the results after imputing the missing values with either the mean of the rows and columns,

or zero (fixed value).

Distribution Two types distributions are investigated, Normal and Multinomial, the latter being more representative of many real-world contexts, such as ratings data we find in recommender systems. Size 4 is specifically chosen to replicate this latter type of data, the MovieLens ratings data set [63]. This non-normal distribution mimics the 100k ratings distribution and sparsity level.

Number of LD Since we use synthetic data, the "true" number of LD is given and can be controlled. Of course, it cannot be used as a factor, but it remains useful to investigate how each method's performance is affected by the number of LD.

We use a range of LD = $[5, 10, 15]$. This choice is based on the assumption that the range is relatively common. However, this range excludes large number of LD that we can expect for contexts such as word embeddedings, where LD vectors of length 100 to 300 are found to be effective [64, 65, 66, 67]. This size of LD involves very large data sets that are beyond the scope of our study, and it would obviously be a subject for future study.

3.3.2 Experiment and implementation details

The implementations of the different methods are:

- Parallel Analysis (PA): We use the "paran" library [68] in the R statistical software application. We examine accuracy of the mean eigenvalue rule vs the 95th percentile eigenvalue rule. We call them "mPA" and "cPA" respectively.
- Minimum Average Partial (MAP): We use the "map" function from the "paramap"[69] package in R. We apply both the original (1976) MAP test (MAP1) and the revised (2000) one (MAP2).
- Bi-Cross Validation: We use the Wold-style cross validation with 5 folds and also Gabriel-style cross validation with 4 sub matrices which are the default of the library "bcv" [70]. We name them BCV.W and BCV.G respectively.
- Randomized SVD (RSVD): RSVD is straightforward to implement given the singular values (see section 3.2.4). The procedure is implemented in R.
- Proposed EINLD and EINRLD methods: These methods are ensemble techniques and use the "randomForest" library in R. They are described in section 3.4.1.

We refer to an *experiment* as the generation of a data set according to predefined attributes and running all methods over that data set. This allows for the comparison of the relative performance across methods over a single data set.

A new and independent data set is generated for each experiment. This allows independence across experiment results in order for later experiments with decision trees to learn from data that is independent from the test data. However, note that when inferring the number of LD with the individual methods, a cross validation does not apply since there is no learning and testing over labeled LD data. Training and testing only applies for ensemble techniques presented in section 3.4.1.

In the context of an experiment that is repeated with the same conditions, we will refer to this experiment as a *run*. We repeat each experimental condition 10 times, which we refer as 10 runs. We sometimes refer to the 10 runs as an experiment and the context will make it clear. Unless specifically stated, we report the mean of the results over these 10 runs. Workflow of our experiments is shown in Figure 3.3.

3.3.3 Generating data sets

The general principle for creating a synthetic data matrix, $\mathbf{R}$, is based on the product of two matrices, $\mathbf{P}_{m\times k}$ and $\mathbf{Q}_{k\times n}$, where k represents the latent factors and number of latent dimensions. Both matrices are generated by sampling from statistical distributions. More details are given below.

Noise is also added to $\mathbf{R}$ to mimic the stochastic nature of the data and make the task non-trivial. As can be seen in the example of Figure 3.2, the true number of latent dimensions is very well defined for this specific example where noise is low. The addition of a realistic amount of noise makes the task more challenging.

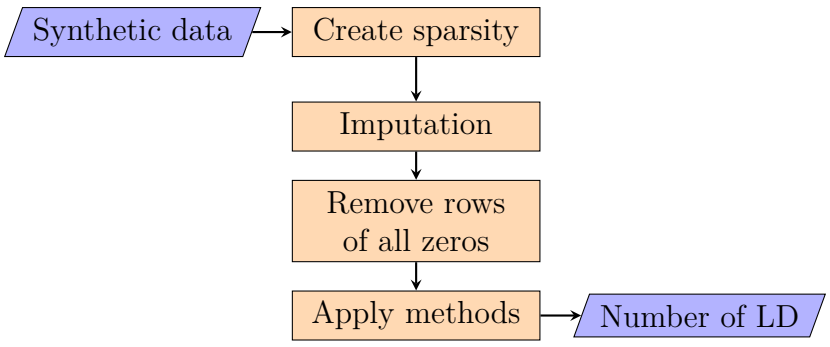


Figure 3.3 Workflow of experiments.

Note that factors are intentionally kept independent and uncorrelated in order for the ground truth to be non-ambiguous, because correlated factors would eventually merge into a single one as the strength of the correlation approaches 1. This would blur the ground truth and force us to set thresholds on correlation between variables that would determine if they are independent dimensions or not.

Table 3.1 gives a breakdown of the 9 data sets generated per distribution, Normal and Multinomial-Uniform.

Generating data sets with the Normal distribution:

The Normal (Gaussian) distribution data sets are generated according to the following equation:

$$\mathbf{R} = \mathbf{P} \cdot \mathbf{Q} + \epsilon \tag{3.3}$$

with $\epsilon \sim \mathcal{N}(0, 1)$. The matrices $\mathbf{P}$ and $\mathbf{Q}$ are also obtained by sampling in $\mathcal{N}(0, 1)$. $\mathbf{R}$ *is the data generated through the inner product of* $\mathbf{P}$ *and* $\mathbf{Q}$*, and the number of columns of* $\mathbf{P}$ *corresponds to the numer of latent dimensions of the data matrix* $\mathbf{R}$.[1]

Generating data sets with Multinomial-Uniform distribution

The multinomial data set is intended to reflect distributions such as ratings found in the recommender system field. The data matrix $\mathbf{R}$ is again obtained on the basis of the product $\mathbf{R_0} = \lfloor \mathbf{P} \cdot \mathbf{Q} + \epsilon \rceil$, where $\mathbf{P}$ contains a row of $0, 1$ unit vectors (*one-hot vectors*) and $\mathbf{Q}$ contains column unit vectors, and $\epsilon \sim \mathcal{N}(0, 1)$. $\mathbf{P}$ and $\mathbf{Q}$ are obtained by sampling from a uniform distribution. Values are also rounded to the nearest integer. The final result is $\mathbf{R} = 2 \times \mathbf{R_0} + 2$ and values less than 1 and greater than 5 are converted to 1 and 5 respectively (a value of 2 is also added to $\mathbf{R_0}$ for the sake of keeping the similarity with ratings, but this transformation is inconsequential for LD inference methods). The resulting distribution is close to ratings found in the MovieLens data set [63].

[1] The text in italic is not in the original paper.

Table 3.1 Nine synthetic data sets for the both normal and multinomial distributions of different sizes (size 1, size2, and size3)

	LD	P	Q	R
1)	5	250×5	5×150	250×150
2)	10	250×10	10×150	250×150
3)	15	250×15	15×150	250×150
4)	5	250×5	5×250	250×250
5)	10	250×10	10×250	250×250
6)	15	250×15	15×250	250×250
7)	5	500×5	5×300	500×300
8)	10	500×10	10×300	500×300
9)	15	500×15	15×300	500×300

3.3.4 Evaluation of the methods' LD estimates over different data sets attributes

A first set of experiments is conducted to assess and compare the performance the methods under different data sets characteristics. The goal is to show these attributes are influential and can be later exploited to combine methods into an ensemble technique and gain a performance increase.

We generate a combination of 9 data set parameters, as shown in table 3.1. This combination is replicated over three data set conditions: (1) normal and multinomial distributions, (2) mean- or zero-value imputation of missing values, and (3) different levels of sparsity.

Each method estimates LD and we compute the loss of each method after 10 runs using the Mean Absolute Error (L1 norm):

$$\text{loss}_{\text{MAE}} = 1/n \sum_{i}^{n} |\hat{d}_i - d_i| \tag{3.4}$$

Where n is the number of runs (10), d_i is the real number of latent dimensions of data set i and $\hat{d}_i$ is the estimated number.

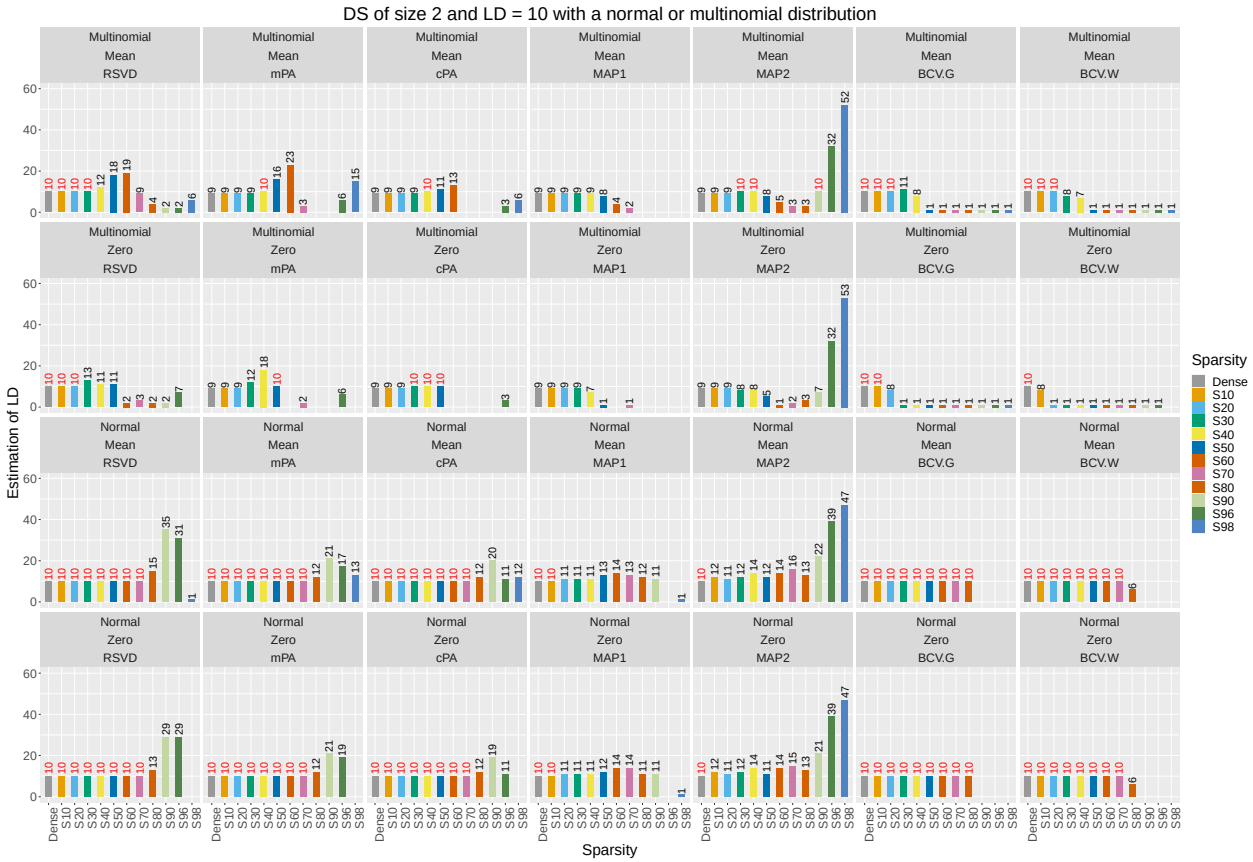


Figure 3.4 The results of the methods on the data set of size 2 and LD = 10 with normal and multinomial distributions. In this figure, absence of column bars meaning the method failed to yield any estimates. Correct estimates (10) are shown in red. The tendency of individual methods to overestimate and underestimate is apparent by looking at high sparsity conditions. For eg., MAP2 clearly overestimates for the Multinomial distribution (first two rows of column 5–third column from left), whereas the BCV.G and BCV. V methods underestimate (first two rows of the last two columns, 6 and 7). Note that absence of column bars means the method failed to have any estimate.

General performance

We start by looking at the general performance of methods as a function of the distribution and imputation conditions. Table 3.2 shows the loss, averaged over the levels of sparsity. It shows that for the mean-value and zero-value imputations and the normal distribution condition, mPA has the lowest loss. But in the data sets with a multinomial distribution, for the mean-value imputation, MAP2 has the lowest average loss, while for zero-value imputation MAP1 has the lowest loss. The results confirm that, for the simulated data sets, methods to estimate the number of LD are sensitive to the data distribution and imputation. The best method is not systematically the same over normal vs multinomial distributions. Imputation has a smaller effect.

Table 3.2 Average loss across imputation and distribution conditions, averaged over levels of sparsity and sizes of Table 3.1. Lowest losses are shown in bold.

	Mean		Zero	
	Normal	Multinom.	Normal	Multinom.
RSVD	5.86	3.14	5.80	4.38
BCV.W	5.91	2.74	5.71	3.90
BCV.G	5.64	2.70	5.47	3.94
mPA	**5.16**	3.10	**4.99**	4.34
cPA	5.59	2.96	5.42	4.20
MAP1	5.36	2.71	5.14	**3.84**
MAP2	6.15	**2.66**	6.44	4.45
Mean	6.11	4.5	5.86	5.74

Sparsity effect

Turning now to the effect of sparsity, Figure 3.4 breaks down the performance along sparsity levels and along the two distribution types. For the sake of brevity, only the data set with LD=10 and size 2 is taken as an example, but it is representative of the general pattern. Zero loss corresponds to bars with a label showing "10" in red. As expected, the greater the sparsity level is, the greater is the loss. We observe that at high sparsity, some methods overestimate while other underestimate LD. These differences in variance and bias error is what can lead to ensemble techniques that can assess the reliability of estimates. We return to these questions later and further investigate the sparsity and data set size in the next section.

Ratio of number of columns to number of rows on methods accuracy at different levels of sparsity

Another perspective of looking at patterns of method performances as a function of data set attributes is to take different ratio of columns (variables) to rows (observations).

Again, for brevity, we only retain single conditions for LD (15) and distribution (Normal), and vary this ratio of columns to row over four levels, 25%, 50%, 75% and 100%, and take three row sizes of matrices, 200, 250 and 350. Taking the 200 rows size example, the 100% condition will have 200 columns, whereas the 25% will contain 50 columns (1/4 of 200), and so on for the other sizes.

Figure 3.5 reports the average estimates over 10 runs for the different methods. Different patterns across methods do emerge, supporting the hypothesis that methods have a specific performance response to data set attributes.

High sparsity, larger data set

We end the analysis of the performance of methods over different data set characteristics by looking at larger data set.

In many real-world data sets, such as user ratings in recommender systems [63] and term-document matrices in language modeling, we find large data sets of high sparsity. We conduct an experiment to investigate the accuracy of each method on a data set that contains 100K votes, the Movielens 100K data set [63] of size 943×1682. We use both the real and a synthetic data set with LD = 19, where the sparsity is 0.937% and the missing values imputation are with zero.

The choice of 19 is based on the predefined categories of Movilens 100K, which we can assume is an indicator of ground truth of the real data. Table 3.3 reports the frequency of the 19 film genres. Considering the *unknown* category has a frequency of 2, 18 genres may be closer to the ground truth. Using an SVD wrapper based approach to predict votes, Sarwar et al. [71] found 14 to be optimal LD number.

We simulate the real 100K votes Movielens data set from the rating distributions of users-genres and movies-genres. To do so, we first generate Movie-Genre matrix with size of 1682×19 and with the same votes distribution as the real one. Figure 3.6a shows the real vs simulated Movie-Genre data set. Then, we generate User-Genre matrix with size of 943×19 and same voting distribution as the real one. Figure 3.6b shows the real vs simulated user-Genre data set. Finally, we multiply the User-Genre and Movie-Genre matrices to obtain a matrix with same size and distribution as the real movielense (100K vote) data set.

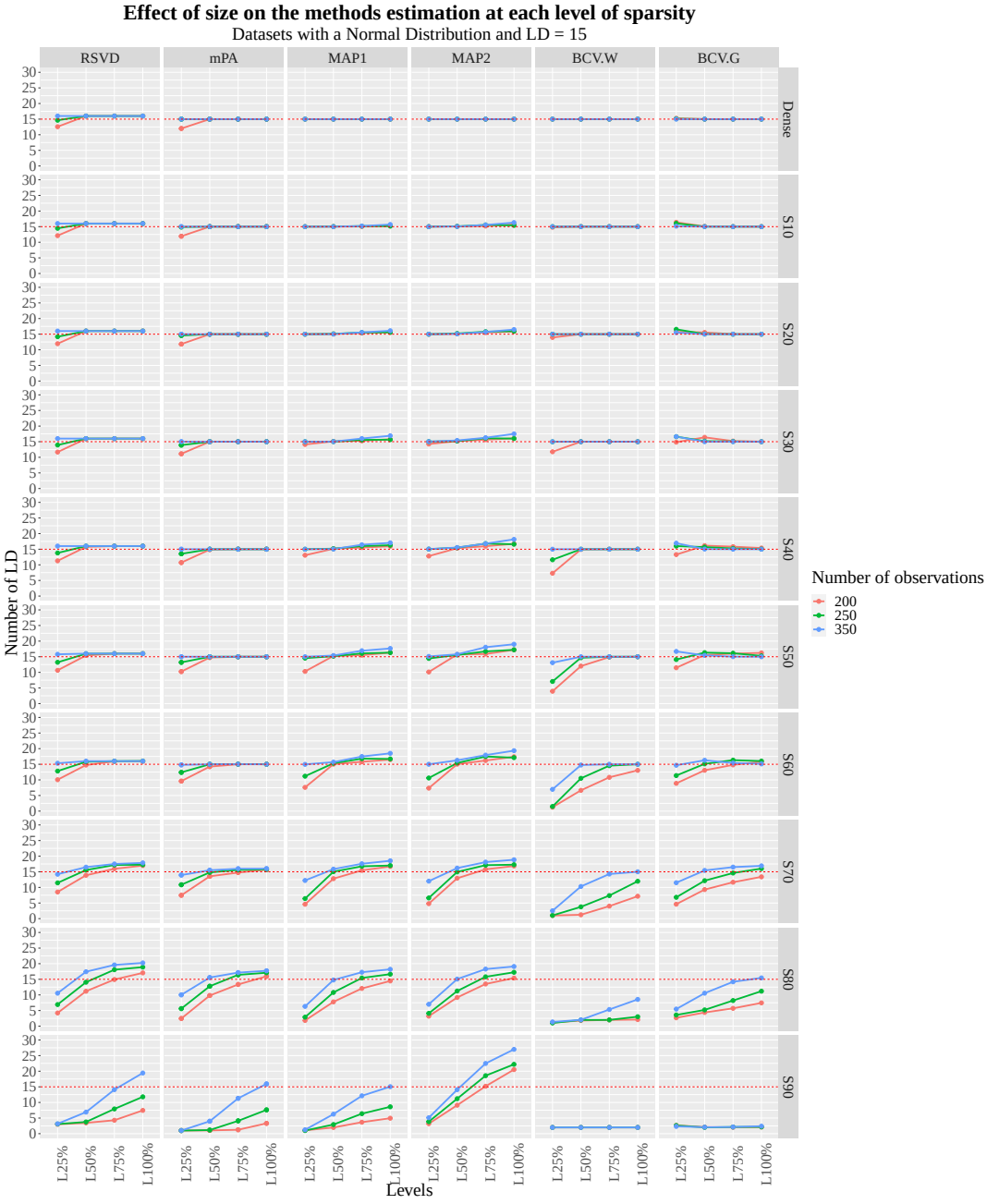
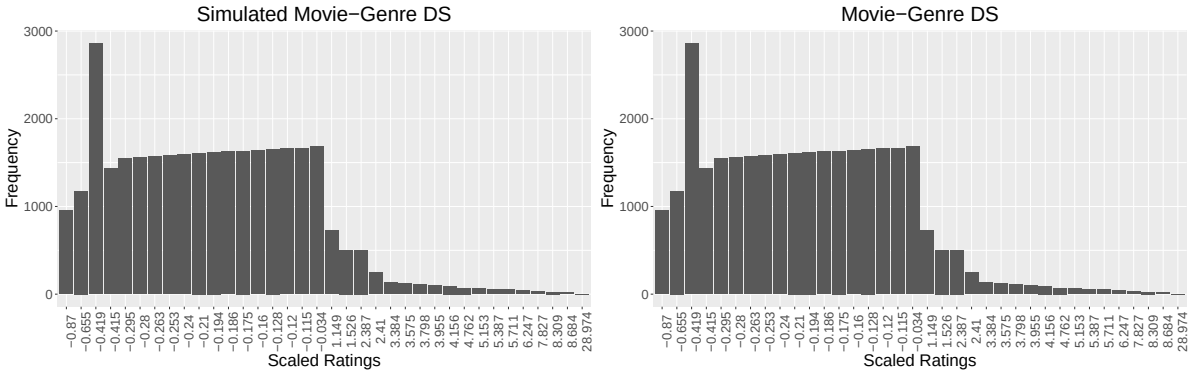
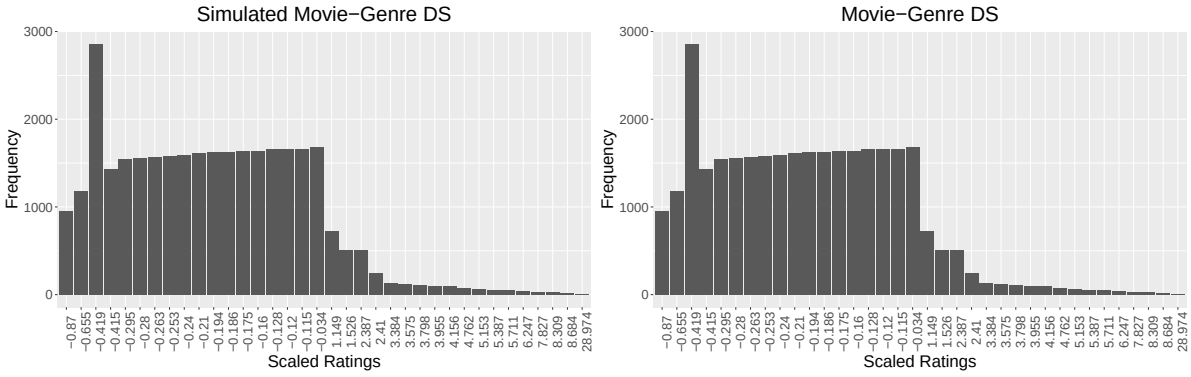


Figure 3.5 Levels $L25\%$: $L100\%$ indicate to ratio number of observations to number of variables.



(a) Movie-genre matrix distributions



(b) User-genre matrix distributions

Figure 3.6 Simulated and real distributions with LD = 19

Table 3.3 Frequency of genres

Genres	Freq	Genres	Freq
unknown	**2**	Sci-Fi	101
Fantasy	22	Crime	109
Film-Noir	24	Children's	122
Western	27	Adventure	135
Animation	42	Romance	247
Documentary	50	Action	251
Musical	56	Thriller	251
Mystery	61	Comedy	505
War	71	Drama	725
Horror	92		

Table 3.4 Estimated LD on a real and simulated data sets of size 943×1642 with a zero-imputation for sparsity of 0.937% (single run).

Data Sets	**RSVD**	**mPA**	**cPA**	**MAP1**	**MAP2**	**BCV.W**	**BCV.W**
Real	19	18	18	5	5	18	18
Simulated	17	16	16	5	2	18	19

Table 3.4 shows the results of each method over real and simulated data sets. They show all methods except "MAP" have a close estimate to the ground truth, or presumed ground truth for real data.

There are two takeaways from this experiment. One is that by keeping the similarity of the distributions of the latent matrices $\mathbf{P}$ and $\mathbf{Q}$, involved in the generation of the synthetic data ($\mathbf{R} = \mathbf{P} \cdot \mathbf{Q}$), with the matrices presumed to represent ground truth of real data, all methods behave similarly to infer the number of LD on both real and synthetic data. It is therefore possible to obtain synthetic data that is faithful to real data as far as LD estimation is concerned.

The second takeaway is that, akin to the previous experiments, there is a variability in the estimates from each method that can possibly provide a cue to their reliability. A question we address in section 3.3.6.

3.3.5 Choosing a method based on the data set attributes

While bias and variance error vary across methods, a relevant question to address is whether one method comes out as a best performer? To investigate this question, we identify the best performers across conditions.

Figure 3.7 reports which method is the best performer and what is its associated loss. For

example, the upper left barplot indicates that BCV.G has the best performance seven times, from sparsity levels "Dense" to the next 6 levels (S60 for 60% sparsity), and that the loss was 0 for the first 5, and (0.1, 1.3) for the next two (averages over 10 runs). Each sparsity level will have a single bar with a number indicating the loss, except for ties which will create duplicates. Without ties, a total of 12 values per row of barcharts are distributed. With ties, we can see that for the first bartchart row, multiple methods have a 0 loss, such that over 12 values are present for that row.

If a method performed better over all conditions, its corresponding barchart column would be filled with values. This is not the case. All methods are best performers over some conditions, but not across all conditions. Some do better with low sparsity, some with high sparsity, and so on for the other data sets attributes.

This gives further evidence that an ensemble technique may be able to exploit the specific advantages some methods have under a given set of data set characterics.

3.3.6 Variance across methods

Thus far, we find that data set attributes affect each method's accuracy differently, with systematic overestimation for some methods and underestimation for others as shown in Figure 3.4. Moreover, factors such as size and sparsity can systematically affect over/under-estimation as shown in Figure 3.5. Unsurprisingly, no single method is the best performer across all data set attributes as shown in Figure 3.7. We now turn to the analysis of the variance of estimates as a precursor to the estimation of the reliability of these estimates.

In order to show the effect of data set attributes on the variance of each method's estimates, we compute the 95% confidence interval for the 10 runs of each method's estimates considering distribution, size, LD and sparsity as the conditions. Figure 3.8 shows the method's estimates and the confidence interval where LD $=$ 15 at different conditions of distribution, size and sparsity levels.

It illustrates that some data set attributes can cause a higher uncertainty in the methods estimates across the multiple runs at a certain condition. Moreover, it shows that in some cases methods have a large bias error with a very small variance error, while in some other cases they have a larger variance error but small bias error. However, it clearly shows that when the methods have an accurate or a close estimate to the ground truth, there is a very small variance across the 10 runs at each condition for all of them. On the contrary, when methods have an inaccurate estimate they might still have a close estimate to one another but they all have a larger variance across the 10 runs.

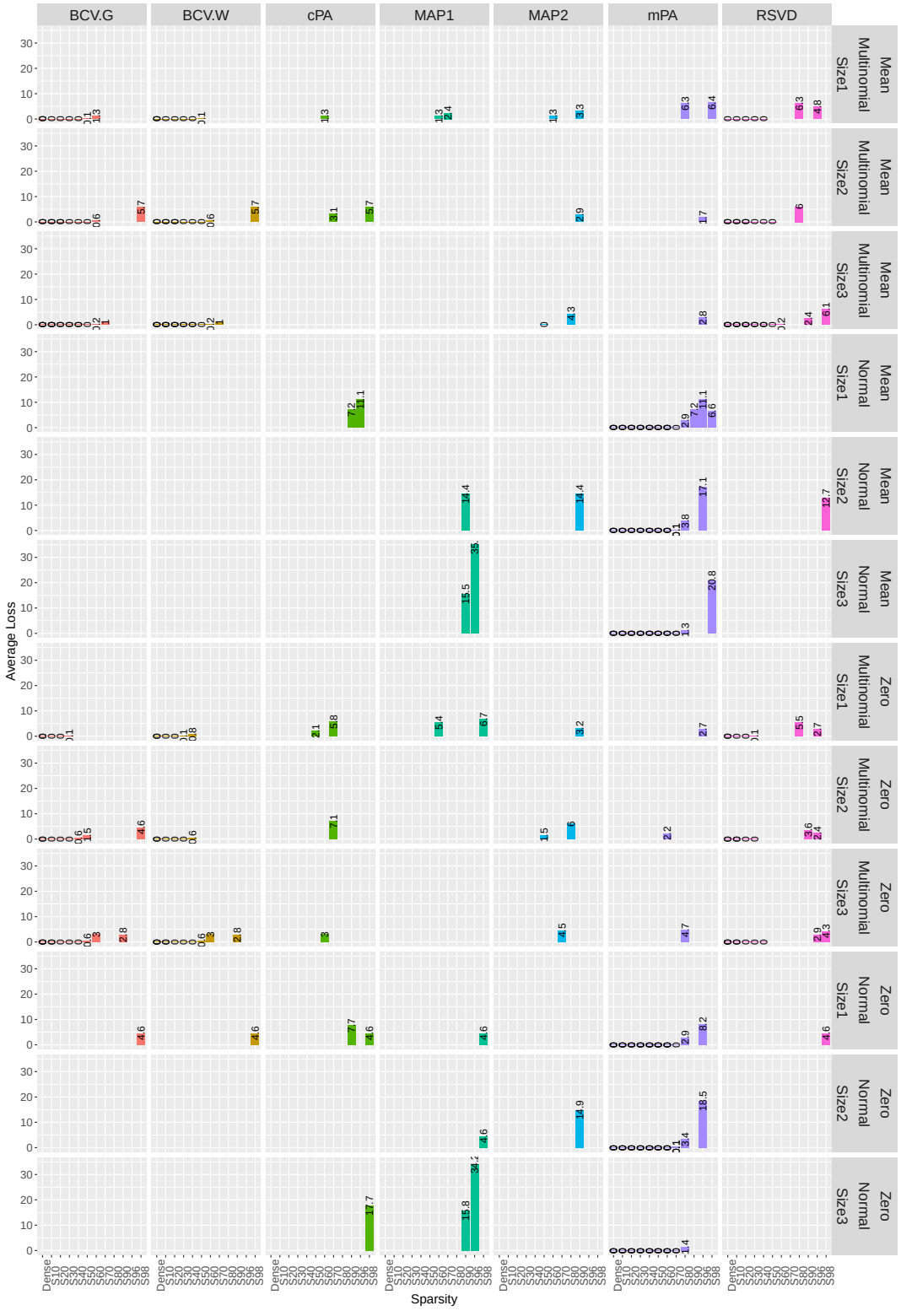


Figure 3.7 Best method to estimate LD considering data set attributes.

In addition, we observe that the presence of some data set attributes together causes more errors.

Table 3.5 Average loss of the methods over the conditions of distribution and imputation.

	Size 1	Size 2	Size 3
Multinomial	**4.98**	4.94	5.11
Normal	3.66	**5.72**	**8.90**

Table 3.5 shows an example of our observation. It illustrates the average loss of all the methods at each distribution and imputation. It shows that where data has a normal distribution, bigger size of data sets, significantly cause a higher loss for the methods on average.

In order to elaborate the behavior of the methods at each condition, we show the average of the methods estimates at each condition with the 95% confidence intervals in Figure 3.9. It illustrates that at higher sparsity, where data has a multinomial distribution, methods tend to underestimate the number of LD, while overestimate where data has a normal distribution. The error bars indicate to the uncertainty of methods estimates at each condition. We also observe that the estimates across methods tend to have low variance when these estimates are close to the true number of LD. We leverage this information to both obtain a more accurate overall estimate and as an indicator of its reliability.

3.3.7 No single best method and a variety of responses to data set attributes

We can conclude this section with the assertion that results of the experiments reported above show the methods to infer the number of LD under study have different performances under an array of data set characteristics. In particular, their relative success compared with each other is different, such that no single method comes out as best under all data set attributes covered. For eg., both Figure 3.4 and Table 3.2 demonstrate the bias error is considerably different between methods, and the spread of estimates is noticeable when large estimate errors are found.

These observations lead to the idea of using ensemble techniques that combine the input of multiple methods to not only yield a better LD estimate, but also a reliability assessment.

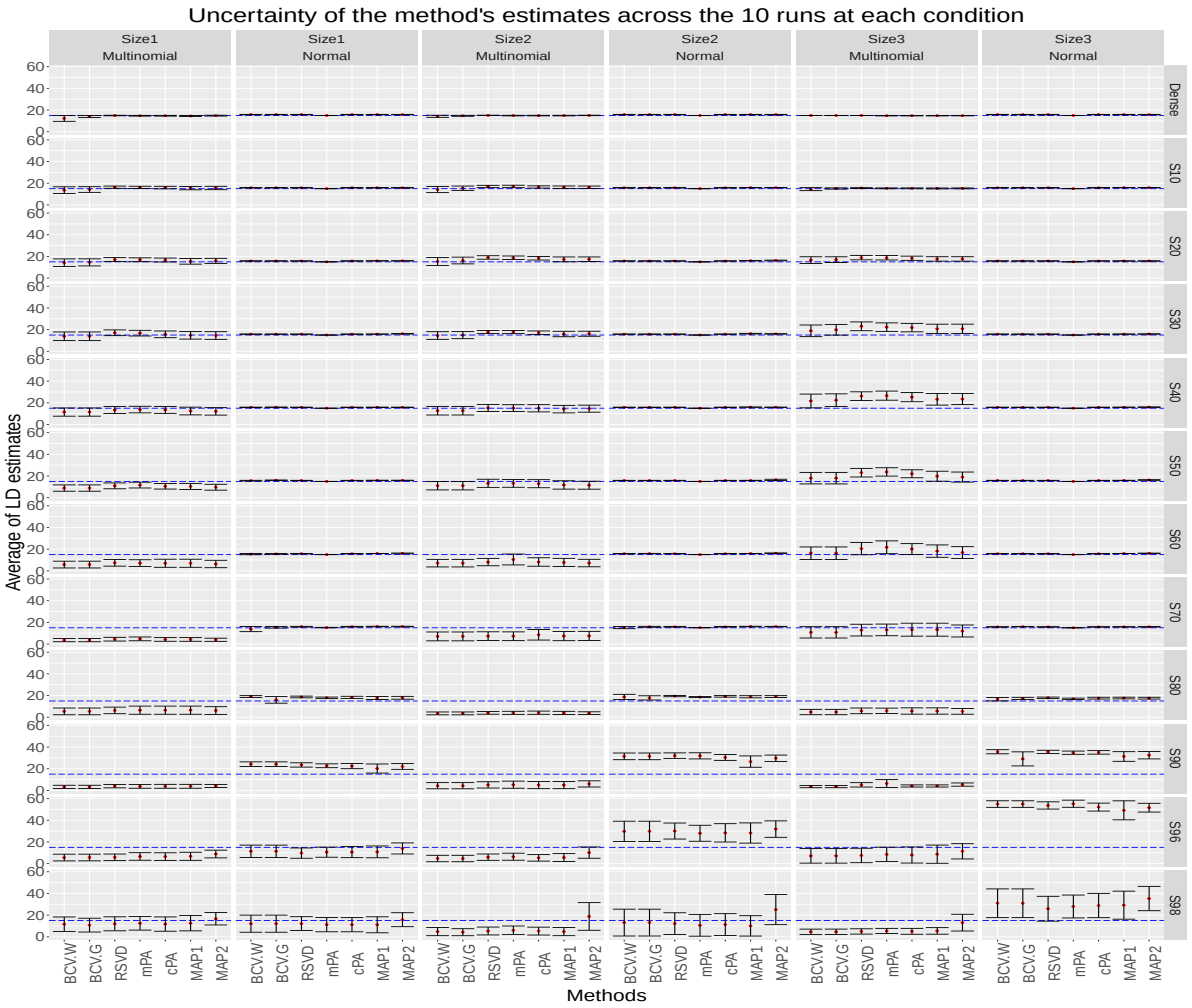


Figure 3.8 95% Confidence interval of each method's estimates over the 10 runs considering different data distribution, size and sparsity levels where LD =15.

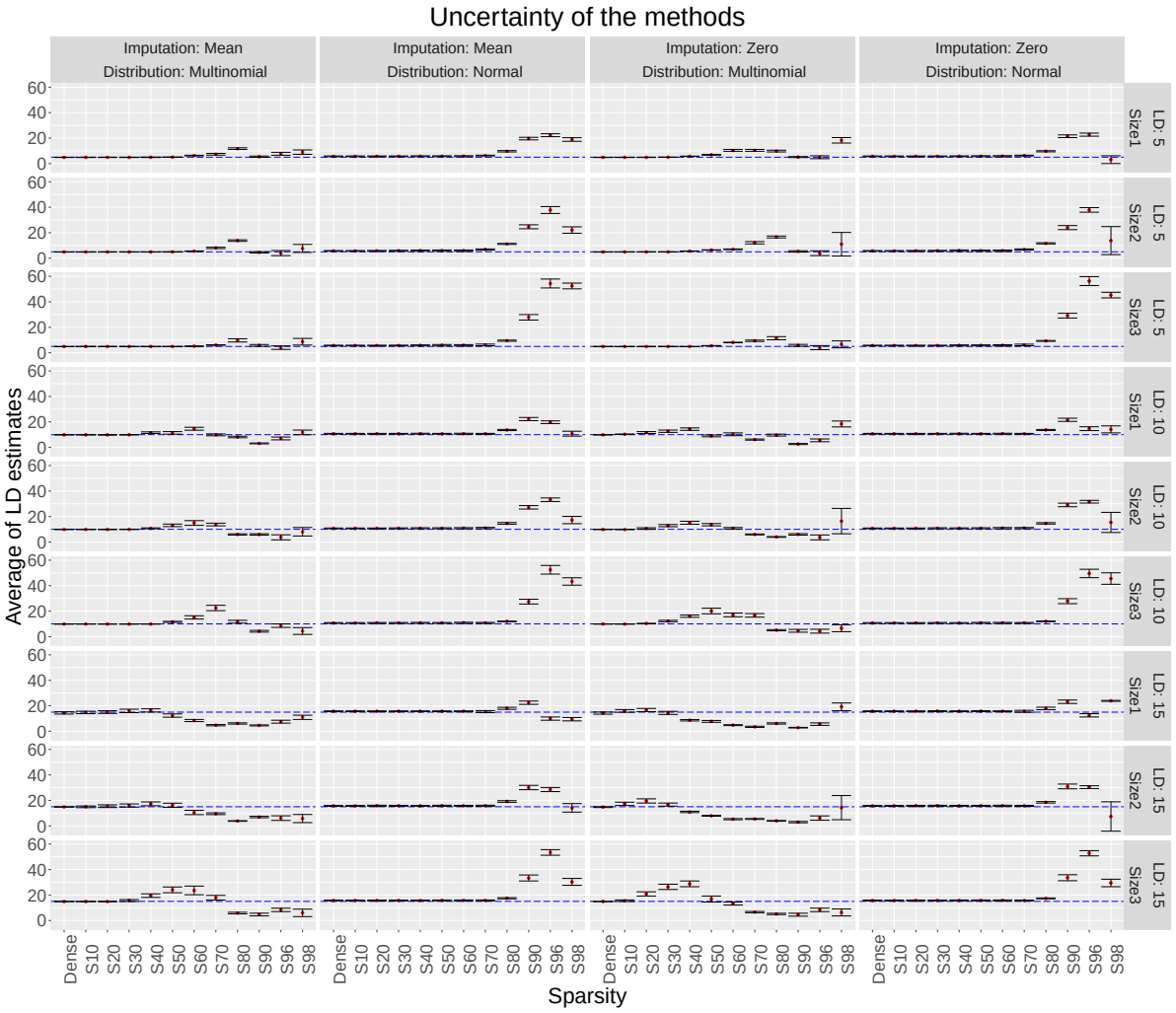


Figure 3.9 Confidence interval of the average of the method's estimates at each condition

3.4 Proposed ensemble methods

Experiments of the last section show that data set attributes affect each method's accuracy differently, with systematic overestimation for some and underestimation for others. We introduce a new method named "EINLD" (Ensemble Inference of the Number of Latent Dimensions) aimed at leveraging this information on which method works best under which conditions to improve estimating the number of LD. EINLD is a multi-method approach, or an *ensemble technique*, and it relies on random forest regression.

Beyond the improvement of accuracy, the variability of method estimates also leads to the opportunity to use this information to assess the reliability of overall estimated obtained with EINLD. The proposed method is named "EINRLD" (Ensemble Inference of the Reliability of the Number of Latent Dimensions) and aims to assess the reliability of EINLD's estimate, where the reliability is defined as a set labels that represents the expected loss category of the method's estimate. EINRLD also relies on an ensemble technique with a random forest classifier.

Since our two proposed methods relies on random forest, we briefly describe the main principles behind this algorithm before the two new methods, EINLD and EINRLD.

3.4.1 EINLD method: Improving estimating the number of LD through an ensemble technique

Both EINLD and EINRLD are ensemble techniques that rely on the Random forest algorithm (RF). RF is an ensemble learning algorithm that combines several decision tree for regression or classification to improve predictive performance over a single learner. The general principle is to train a set of decision trees from multiple resampling of the original data set and to aggregate the results of inferences with the trees, either by using a majority vote (for classification) or by averaging (for regression). Different sets of features are also explored during the training (feature bagging) to reduce the overtraining effect.

The EINLD method uses the following predictors and target variable:

Sparsity. Sparsity is a categorical variable corresponding to twelve levels, from Dense (0%) to 90%, and two more levels of high sparsity, 96% and 98%. The sparsity category labels are {Dense, S10, S20, ..., S98}.

Distribution. Normal and Multinomial. The distributions are generated as described in section 3.3.3.

Imputation. Imputation is either 0 or the row-column means.

Size. Sizes correspond to the 9 sizes described in section 3.3.3.

LD estimates (7 averages). LD estimates are the 10 run averages of each method's estimate listed in section 3.3.2.

Variance. Variance of estimates across the 10 runs.

Target variable. The target variable is the true number of LD of the experiment's data set (which is known because we use synthetic data).

The Random Forest algorithm to estimate LD in EINLD is a regression model trained with 7-fold cross-validation.

Figure 3.10 shows an instance of a decision tree. Each node contains a percentage representing the number of cases that fall under this node, and a number that represents the expected value, which corresponds to a prediction for the leaf nodes. The first decision branch is based on the average of the methods LD estimates, suggesting that some methods are more effective than others if the average falls below or above 12. The value of 10 at the top node indicates the expected estimate, which is a weighted sum of its children nodes ($0.62 \times 8.4 + 0.38 \times 13$) and it corresponds to the average of $\{5, 10, 15\}$ LD conditions in our experiments. At the next level down, the condition "Mean_Estimation < 12" (left) branches on the "Variance < 16" and represents to 62% of cases, whereas the negation of the condition (right) branches on a subset of sparsity levels and correspond to the remaining 38%. Akin to the value of 10 of the top node, the number 8.4 of the second level left branch is the estimate and corresponds to a weighted sum of the values and percentages of its children nodes. And so on for all other nodes and down to the leaf nodes representing the final estimates.

As explained above, the Random Forest algorithm for a regression task makes a set of predictions, one per decision tree, and takes their average as the predicted LD. Figure 3.10 is a single instance of a decision tree and some predictors are missing simply because the algorithm uses a subset of predictors for each instance, and therefore its interpretation can be misleading. This is unfortunately a drawback of RF, but the figure provides some insights on the model obtained.

Results

Table 3.6 reports the prediction results. It shows the MAE and Root Mean Square Error (RMSE) of EINLD compared to two baselines: the method that has the best estimate overall,

which is mPA, and the average estimate of the 7 methods. EINLD has a substantially smaller error compared to the best method, and to the average of all methods.

Table 3.6 MAE and RMSE of predicting LD. Bold numbers indicate to the lowest error and best method.

	Best method (mPA)	Average of the methods	EINLD
MAE	4.56	4.47	**1.30**
RMSE	9.01	8.89	**1.93**

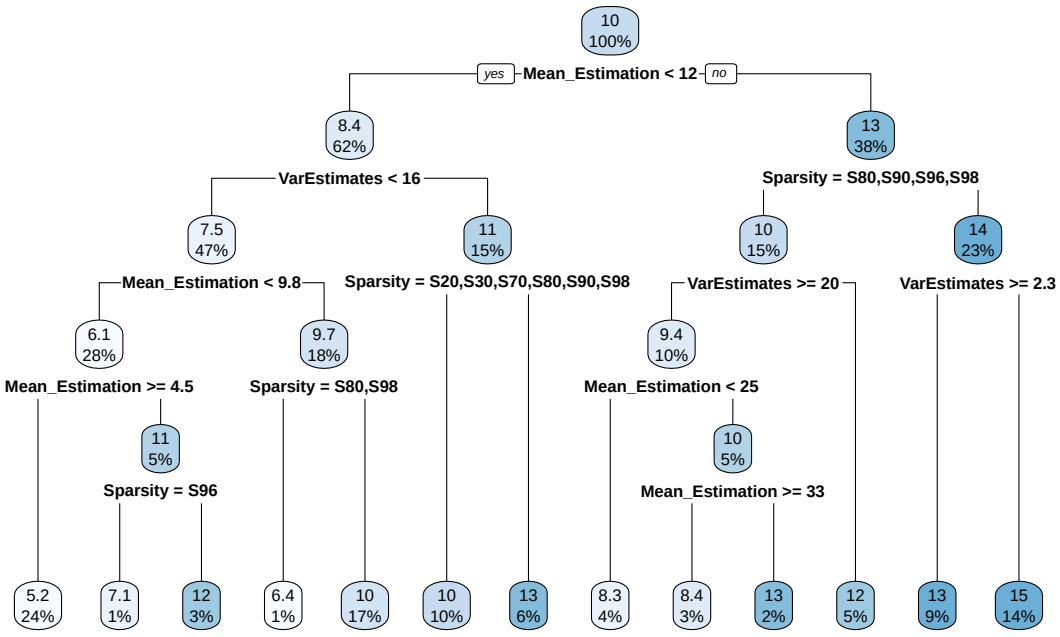


Figure 3.10 Predicting LD using the EINLD method while considering sparsity, distribution, imputation, size, average estimation of the methods after 10 runs, and variance of the method's estimates for 10 runs at each condition as the predictor variables. Lighter color gradients indicate lower LD values.

3.4.2 EINRLD: assessing the reliability of LD estimates

We now tackle the question of assessing the reliability of LD estimates, again using the information gathered from multiple methods and the data set attributes. The method EINRLD (Ensemble Inference of the Reliability of the Number of Latent Dimensions) is introduced for that purpose and, akin to estimating LD with EINLD, it relies on a Random Forest algorithm. However, instead of estimating the reliability as a regression, we use categories of reliability and revert to a classification task.

Estimating reliability can be considered an estimate of the expected loss. We determine three levels of losses: "Excellent" (maximum loss of 1), "Good" and "Unreliable". The thresholds for each category are defined as:

$$\text{reliability levels} = \begin{cases} \text{Excellent}, & \text{if } |loss| \leq 1 \\ \text{Good}, & \text{if } 1 < |loss| \leq 4 \\ \text{Unreliable}, & \text{otherwise} \end{cases}$$

where *loss* is the MAE as defined in Equation (6.2).

Next, we train a Random Forest classifier model with a 10−folds cross validation to predict the reliability level of the obtained LD.

We will study the reliability assessment with the ensemble approach using as a *basis* each of the 7 individual methods, as well as by considering EINLD an independent method to estimate the number of LD, even though it is in fact a method that relies on the 7 individual methods. This will provide a more comprehensive perspective on the ensemble approach to reliability assessment.

The predictor variables to develop EINRLD are listed below (see section 3.4.1 for details) and grouped around three cases:

- Case 1: sparsity + size + distribution + imputation + average of the method estimates after 10 runs.

  Case 1 can be considered as a baseline that takes into account the data set attributes and the average of a 10-runs sample.

- Case 2: Case 1 predictors + variance the individual method estimates across 10 runs.

  Case 2 considers the same predictors as Case 1, but it also uses the variance of the experiment's 10 runs to assess reliability. Taking the variance of estimates is a common practice for calculating a standard error with bootstrapping to determine a confidence

interval. This second baseline therefore takes data set attributes, the average estimated LD, and the variance of the 10 estimates.

- Case 3 (EINRLD): sparsity + size + distribution + imputation + average of EINLD estimate after 10 runs + variance across individual methods.

  Case 3 includes the variance across the 7 methods in addition to Case 2 predictors. We only compute Case 3 for the EINLD basis method, and name this as the EINRLD method. This method not only takes into account data set attributes, but also variance across methods to determine reliability. Although in principle we could compute Case 3 for each of the 7 individual methods and consider EINLD as an 8th method, the choice to retain only the EINLD basis is because EINLD is an ensemble technique that already relies on the 7 individual methods (considering EINLD as just another method would mean an invidual method estimate is used twice, initially in EINLD and later in EINRLD), and because EINLD is the most accurate choice for Case 3 and would be our best choice for EINRLD.

Performance metric

Because the reliability levels are highly imbalanced, we use micro-averaged F1-score metric:

$$\text{precision}_{\text{micro}} = \frac{\sum_{n=1}^{C} \text{TP}_n}{\sum_{n=1}^{C} \text{TP}_n + \sum_{n=1}^{C} \text{FP}_n} \tag{3.5}$$

$$\text{recall}_{\text{micro}} = \frac{\sum_{n=1}^{C} \text{TP}_n}{\sum_{n=1}^{C} \text{TP}_n + \sum_{n=1}^{C} \text{FN}_n} \tag{3.6}$$

$$\text{F1}_{\text{micro}} = 2 \times \frac{\text{precision}_{\text{micro}} \times \text{recall}_{\text{micro}}}{\text{precision}_{\text{micro}} + \text{recall}_{\text{micro}}} \tag{3.7}$$

where C is the number of target classes (Excellent, Good, Unreliable), and TP_n , FP_n and FN_n stand for the number of true positives, false positives and false negatives for the nth target class, respectively. Note that, micro precision, micro recall and micro F1-score values are all equal in micro average metric and they represent the accuracy of the classifier. So, we only present the $F1_{\text{micro}}$ for the results.

Table 3.7 F1-score of predicting reliability levels using each method through RF for Case 1 and Case 2.

	RSVD	mPA	cPA	MAP1	MAP2	BCV.W	BCV.G	EINLD
Case 1	0.836	0.789	0.808	0.764	0.836	0.803	0.796	**0.914**
Case 2	0.868	0.840	0.847	0.808	0.852	0.819	0.808	**0.940**
Case 3 (EINRLD)								**0.964**

Results

Table 3.7 shows the F1-score for the three cases. The performance of the 7 individual methods are reported for cases 1 and 2, along with the EINLD as a base method to the ensemble technique. For case 3, we only report the EINLD results and consider this as the EINRLD method, since it always outperforms the results obtained from the 7 individual methods.

For all three cases, we find that the Random Forest ensemble algorithm does systematically better at assessing reliability on the EINLD as a basis method for LD prediction than for the 7-individual methods.

As we could expect, the addition in Case 2 of the variance of the 10 runs (akin to bootstrapping in our case) provides a gain for all basis methods.

And finally, the addition of variance across methods in Case 3 brings the F1 score from 0.940 to 0.964 for the EINLD method. Considering that the maximum F1 score is 1, the improvement is substantial as it corresponds to a reduction of 40% of the remaining error compared to Case 2 ($1.0 - (1 - 0.964)/(1 - 0.940)$).

For the benefit of providing a more detailed perspective of the results, table 3.8 shows the confusion matrix of EINRLD. We find there were 11 cases misclassified out of a total of 303, of which the most off target were 5 true "Excellent" cases that were classified as "Unreliable".

Table 3.8 An instance for evaluating the performance of a classification using EINRLD for case 3 .

		Ground truth		
		Excellent	Good	Unreliable
Prediction	Excellent	165	1	0
	Good	0	93	5
	Unreliable	5	0	34

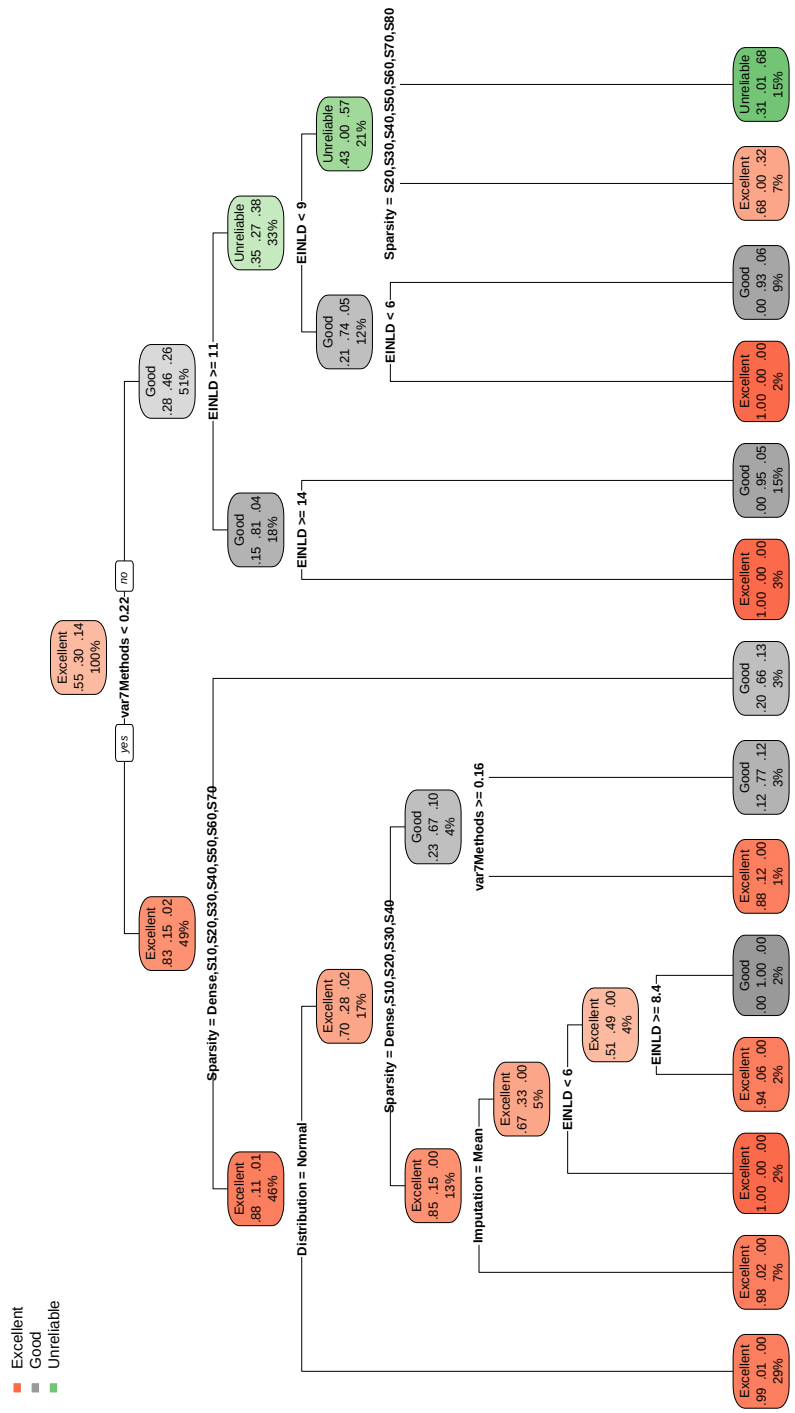


Figure 3.11 A sample of classification tree using EINRLD method to predict reliability level of the estimated LD at different conditions of data while the variables of methods, sparsity, size, distribution, imputation, variance and Diff are the predictor variables. Each node has three lines which the first line refers to the predicted class that is a class with the highest probability in a certain split, the second line shows the probability of the classes of Excellent, Good, Unreliable in each split and the third line shows the percentage of observations in the node.

3.5 Conclusion and Future Work

This study offers three main contributions to the task of finding the right number of LD. Our first contribution is to provide a comprehensive analysis of the impact of data set attributes on the methods performance. We design different experiments to compare the performance of popular methods from different fields of study in finding the number of latent dimensions (LD) over synthetic data sets by controlling different data set attributes. We demonstrate that in order to estimate the correct number of latent dimensions, we need to consider the data set attributes, and that highly popular approaches, such as a *wrapper* method, can break down quickly at high sparsity levels. We also show that well-known methods in the field of psychometrics and social science are more robust in some of the data conditions. Moreover, we show that data set attributes affect each method's accuracy differently, with systematic overestimation for some and underestimation for others.

The finding that the best choice is not universal, but instead it depends on the data set attributes, leads us to the second contribution which is a method to improve estimating the number of LD through an ensemble technique. We propose a new method EINLD which is a multi-method approach to estimate the number of LD that relies on a Random Forest regression. The result shows that it performs significantly better than any of the individual methods.

In addition to the observation that the performance of methods depends on data set attributes, we also observe that the variance within runs, and across methods, is an indicator of the general loss. The larger the variance, the more the methods will show loss. This observation leads us, in turn, to our third contribution, the EINRLD method to infer the reliability level of the estimated LD. This method represents a new approach in the context of finding the number of LD to infer the reliability of the obtained LD. We train a Random Forest classifier to evaluate the reliability of the methods estimates by predicting the three levels of reliability (Excellent, Good, Unreliable) based on the methods loss. The results show that including variance across the individual methods in the predictor variables of EINRLD improves the prediction performance of all the methods. Moreover, the results illustrate that our proposed method EINLD has a higher accuracy in predicting reliability using EINRLD compared to the other individual methods.

These findings can help us to design a tool that better assess the number of latent dimensions in data sets and can be useful for applications in topic modeling, factor analysis, recommender systems, among others. However, there remains significant limitations and future investigations towards making such tool a reality.

An important limitation is the range of LD that are investigated. As hinted above, some applications such as embeddings with neural networks often use vector representations that represent tens and up to many hundreds, such as the 1024-hidden layer of BERT-large [72]. Such LD sizes are currently not manageable with the proposed EINLD and EINRLD approaches, if only because most of the methods rely on eigenvector analysis that has a time complexity in the range of $\mathcal{O}(mn^2 + n^3)$ [73] and the matrices that have to be decomposed can have millions and more lines and many thousands of columns. While progress is being made to handle very large matrices, the proposed approaches remain appropriate for relatively small data sets where the expected number of latent dimensions is in the range of lower tens.

Another limitation is that the individual methods underlying EINLD and EINRLD are based on linear relationships among features. If, as in the case of embeddigns again, the models can capture non-linear relations, the number of LD will not accurately reflect such relations. We can surmise that using the proposed approaches in the context of neural models, for example, might be ill-advised, even with smaller data sets and low dimensionality.

In spite of these limitations, the EINLD and EINRLD techniques offer a novel ensemble approach that is applicable to any set of individual methods and data set attributes. Future work may consider new methods and attributes to extend the current findings. The approach is shown, in the context of this investigation, to substantially improve the accuracy of the number LD estimates, and improve the assessment of the reliability of an estimate. The reliability assessment may be the most important contribution considering that any method will necessarily yield some estimate, but rarely indicate whether the consumer of this estimate can have confidence it is close to the ground truth.

CHAPTER 4 ARTICLE 2: RELIABILITY OF PERPLEXITY TO FIND NUMBER OF LATENT TOPICS

Neishabouri A, Desmarais MC. Reliability of perplexity to find number of latent topics. In Proceedings of the Thirty-Third International Flairs Conference, 2020, 246–251.

Abstract

The problem of finding the correct number of latent topics in Latent Dirichlet Allocation is typically addressed by using a so-called *wrapper* approach and optimizing over the perplexity measure. This problem can be considered a dimensionality reduction task. We investigate how popular methods from different fields determine the right number of latent factors to retain. We address the reliability of these methods under different conditions and under different characteristics of datasets.

In particular, we show that although perplexity is the favorite statistical method to choose the number of latent topics, it does not systematically outperform other methods under different matrix sparsity levels. We show that SVD-based methods and a well-known methods in psychometrics sometimes yield the greatest performances. We also show that we can take advantage of antithetical results across methods to estimate the reliability of the estimated number of latent topics.

4.1 Introduction

Finding the number of hidden factors is a common problem for a number of statistical and machine learning techniques that are deployed in fields such as information retrieval, psychology, and recommender systems. Interestingly, each field of study has its own methods of choice to solve this problem. Few studies borrow methods from outside their fields to investigate the reliability and performance of these methods within a single field. In the field of Topic Modeling, this problem translates to the task of finding the number of topics.

We investigate if, and how methods outside of the typical topic modeling studies can tackle this task. Experiments are conducted with synthetic data where we know the ground truth (number of topics) and, as a generative model, LDA is well suited for that purpose.

The results of the experiments surprisingly show that, under certain conditions, the linear methods show better estimate the number of topics than perplexity. We also find that sparsity

has a key effect and even more important than α and β to find the correct number of topics (K) by the mentioned methods.

The rest of this paper is organized as follows. We first review LDA in more details and review the best known and most successful methods to find the correct number of latent factors in different fields. Then, we report the details of our experiments. Next, we discuss the results before concluding.

4.2 Latent Dirichlet Allocation

Latent Dirichlet Allocation (LDA) was originally introduced by [5] and is arguably the most widely used method for topic modeling. It is a generative, probabilistic model of documents. This technique uses words distribution to cluster texts and discover latent topics from it. It is based on the assumptions that each document consists of a mixture of hidden topics and that each topic consists of a set of words, both of which follow a Dirichlet prior.

LDA has the following hyperparameters that have to be determined before the training phase:

- Alpha (α), is the Dirichlet prior of document-topic density. Higher values of alpha implies documents composed of more topics and lower values implies fewer topics per document.
- Beta (β), is the Dirichlet prior of document-topic density. A higher beta indicates that topics are composed of a large vocabulary, and a lower value implies smaller vocabularies per topic.
- K is the number of topics.

[44] suggest a value of $50/T$ for α, where T is the number of topics, and 0.1 for β. And consequently many studies such as [45, 46] as well as the "topicmodels" package [47] in R use of these values.

For finding the number of topic parameters K, most studies [5, 6, for eg.] use perplexity along with a wrapper technique to find the number that will essentially yield the most likely probability of the data given the hyperparameters and the training parameters derived. A fair number of studies have shown its effectiveness [5, 10, 7, 9, 8, 74].

To skip defining the number of topics in advance, Hierarchical Topic Modeling (HTM) was proposed and studied by a number of researchers [50, 51, 52, 53]. However, other studies [54, 55, 56] have shown that hierarchical topic modeling does not yield better results and argue

that some of the cons of HTM methods are that they are heuristic-based and computationally expensive.

4.3 Dimension reduction methods

Let us now turn to alternative methods to address the problem of finding the number of latent topics that are inspired from dimension reduction and factor analysis.

In the social science and psychometrics areas, factor analysis techniques help decide the number of latent factors to retain from a dataset [26]. Among the best known, we find Kaiser's eigenvalue-greater-than-one rule, Parallel Analysis (PA), Cattell's Scree test, Minimum Average Partial test which is known as Velicer's MAP. They are reviewed below.

In areas of machine learning applications such as recommender systems, we find Singular Value Decomposition (SVD) based approaches such as Bi-Cross-Validation (BCV), which is known as a wrapper method, and the more recent Randomize-SVD (RSVD) [75]

In this paper we evaluate the method PA, which according to the psychometrics literature is a top performer [34], alongside with SVD-Based methods and perplexity in finding the correct number of topics under different conditions of datasets. This choice corresponds to our assessment of the most promising alternatives to perplexity [75]. Moreover, we investigate the reliability of these methods and compare them.

4.3.1 Parallel Analysis

Horn's Parallel analysis (PA) [29] is a well-known technique in psychometrics that is almost ignored by the machine learning community. It relies on the correlation matrix of the observed variables of the original datasets, and multiple random datasets generated having the same size as the original dataset. A factor is retained if its associated eigenvalues of the correlation matrix of the original dataset are bigger than the mean or 95 percentile eigenvalues that are derived from the correlation matrices of the random generated datasets. The remaining factors are considered random noise.

In this article, mPA and cPA refer to the two variations of PA method that correspond respectively to the average and 95 percentile of the eigenvalues of the random datasets respectively.

4.3.2 Bi-Cross-Validation (BCV) of the SVD

Gabriel cross-validation (BCV.G) and Wold style cross-validation (BCV.W) are wrapper methods that rely on cross validating the singular value decomposition (SVD) to find the best rank of a matrix to truncate the SVD [39].

The singular value decomposition of a matrix is a well-known matrix factorization technique. It decomposes the original matrix, $\mathbf{A}$, into three matrices as below.

$$\mathbf{A} = \mathbf{U}_{m\times m}\, \mathbf{\Sigma}_{m\times n}\, \mathbf{V}^{T}_{n\times n}$$

where U and V are two eigenvector matrices that are orthonormal and called the left-singular vectors and right-singular vectors of $\mathbf{A}$ respectively and the matrix $\mathbf{\Sigma}$ is diagonal with non-negative real values.

Both the Gabriel-style (BCV.G) and Wold-style (BCV.W) cross validation variations consists in dividing the data into a training and a test set. Prediction is done with a truncated (lower rank) product of the factorization. The prediction error is measured as the sum of squares of residuals between the truncated SVD and the original matrix. Determining the number of LD relies on a comparison of the residual error over a random set of values for the test set.

In the Wold-style cross validation, the test set is a random set of values in the matrix. The difference between Wold- and Gabriel-style is that, for the later, the test set is "blocked". It holds out a certain number of rows and columns of a matrix simultaneously as a test set. BCV.G divides the rows of the matrix into k segments and the columns into h segments. The total number of folds are $k \times h$ which refer to the number of blocks. In each step, one of the blocks is considered as the test set and the remaining blocks are as the training set to reconstruct the original matrix. More details about this method are given in [40, 41]

With large data sets free of missing values, [40] report that BCV has a better result than other methods in the state of the art. Several studies indicate that the Wold-style cross validation provides a better result but is slower than Gabriel-style [41, 39].

4.3.3 Randomized Singular Value Decomposition: RSVD

Randomized Singular Value Decomposition (RSVD) is briefly introduced in [12] and developed further in [75]. This method is similar to PA. It compares the singular values of the original matrix with the randomized matrix. The randomized matrix is a sample of the original matrix by selecting columns randomly with the same size.

The number of latent values is determined as the point of intersection of the two curves of

the randomized and original matrices SVD singular values.

4.3.4 Perplexity

Perplexity is arguably the most popular metric in language modeling. It is based on the probability of the unseen test set, normalized by the number of words to evaluate the goodness of LDA model. The perplexity is defined as:

$$PP(\mathbf{W}) = P(\mathbf{W}_1\mathbf{W}_2, ..., \mathbf{W}_\mathrm{m})^{-1/m})$$

where PP and $\mathbf{W}$ refer to perplexity and word respectively, and $\mathbf{P}$ is the probability estimate assigned to document words. A model with a given number of topics that minimize perplexity value is considered an optimal model [48, 10].

4.4 Experiments and results

We conduct experiments to compare the methods outlined above over the task of inferring the number of topics used to generate synthetic documents generated with the LDA model. Specifics of each method for the experiments are:

- Parallel Analysis (PA): we use the "paran" library [68] in R. We examine the accuracy of the mean eigenvalue rule and the 95th percentile eigenvalue rule. We call them "mPA" and "cPA" respectively.
- Bi-Cross Validation: Wold-style cross validation with 5 folds and also Gabriel-style cross validation with 4 sub matrices [70] which are the default of the library. We refer to them as BCV.W and BCV.G respectively.
- Randomized SVD (RSVD): implemented in R using algorithm 1 in [75].
- Perplexity: We use LDA with Gibbs sampling and perplexity functions in "topicmodels" library [47] in R.

PA, BCV and RSVD are linear methods to find the number of latent factors, while LDA is a non-linear method which uses perplexity to find the optimal number of latent factors to retain.

In our experiments we generate document-term matrices from a LDA model with known hyper-parameters. Using synthetic data is our choice of validation method because we know

the ground truth of the generated data and we can control the different characteristics to investigate. We design some experiments that simulate the short texts such as reviews, comments, abstracts, tweets, etc. We aim to find the correct number of topics behind the generated datasets using the mentioned methods in order to compare their performance.

We explore the performance of methods over document-term datasets of size $= 250 \times 1000$, where the rows and columns refer to the number of documents and vocabulary respectively. We also consider two sets of priors (alpha and beta) to generate document-term matrices, namely set $1 = [\alpha = 0.6, \beta = 0.1]$ and set $2 = [\alpha = 0.8, \beta = 0.6]$.

We study the methods' performance for different dataset characteristics such as the number of latent topics, level of sparsity (number of terms per document) for each set of priors. Hence, we generate 15 different datasets for each set of priors, and we further generate datasets for 3 sizes of latent topics, $K = [5, 10, 15]$ and that for different levels of sparsity using the number of terms per document such as $[50, 100, 200, 300, 400]$.

4.4.1 Datasets generation

To generate synthetic document-term matrices, we rely on the generative nature of LDA [5].

Algorithm 1 shows the procedure and steps to perform our experiment per each dataset.

Results of the experiments over the different data sets are reported below.

Algorithm 1 Experiment Procedure

- Define size of documents and vocabulary
- Define α, β and K
- DTM $\leftarrow$ Generate synthetic Document-Term dataset using LDA generative Process
- Estimate K using each of the linear method (DTM)
- Estimate K using Perplexity through following steps:
    1. Split DTM into five folds
    2. For each unique fold
        (a) Take the fold as a test
        (b) Take rest of the folds as a training set
        (c) For $K = 2 : 25$
            i. Fit LDA model using Gibbs sampling on the training set
            ii. Evaluate the fitted model on the test set using perplexity
            iii. Retain the evaluation score per each K
        (d) Take the average of the perplexity score for each K
    3. Return K with minimum average perplexity score

4.4.2 Experiment 1, $\alpha = 0.6$ and $\beta = 0.1$

In this experiment, we explore each method on the generated datasets with $\alpha = 0.6$ and $\beta = 0.1$. Figure 4.1 (top plot) displays the estimation of each method at different levels of sparsity (number of terms per documents) $= [50, 100, 200, 300, 400]$ and latent topics, $= [5, 10, 15]$.

For the further analysis, we compute loss of the estimated number of latent topics using the Mean Absolute Error (MAE):

$$\text{loss}_{\text{MAE}} = 1/n \sum_{i}^{n} |K_i - \hat{K}_i|$$

Where K_i is the real number of latent topics of data set i and $\hat{K}_i$ is the estimated number.

We discuss the results in the next section.

4.4.3 Experiment 2, $\alpha = 0.8$ and $\beta = 0.6$

This experiment illustrates the effect of higher priors in each of the mentioned method's estimation. We generate all the datasets with $\alpha = 0.8$ and $\beta = 0.6$. Figure 4.1 (bottom plot) shows the estimation of each method at different levels of sparsity and latent topics. The details are discussed in the following section.

4.5 Discussion and analysis of errors

According to Figure 4.1, we can conclude that dataset characteristics such as sparsity, number of topics, α and β play a crucial role for the capacity of the methods to find the number of latent topics.

As we could expect, the error estimation of K grows as the real value of K grows, and as the size of the documents drop from 400 to 50 terms. Unexpected is that the direction of errors is opposite when comparing perplexity with all other methods. These error trends are analyzed below.

Table 4.3 shows the ratio of correct and over/under estimation for each method averaged over all data sets. We find that the linear models RSVD and BCV often show a higher capability to estimate the right number of topics than perplexity. Moreover, we see that although BCV.W has the same accuracy as perplexity, it has a bias in the opposite direction. Perplexity overestimates the number of latent topics whereas BCV.W underestimate it. More on this trend below. Note that for the two plots in Figure 4.1, a bias correction is applied to

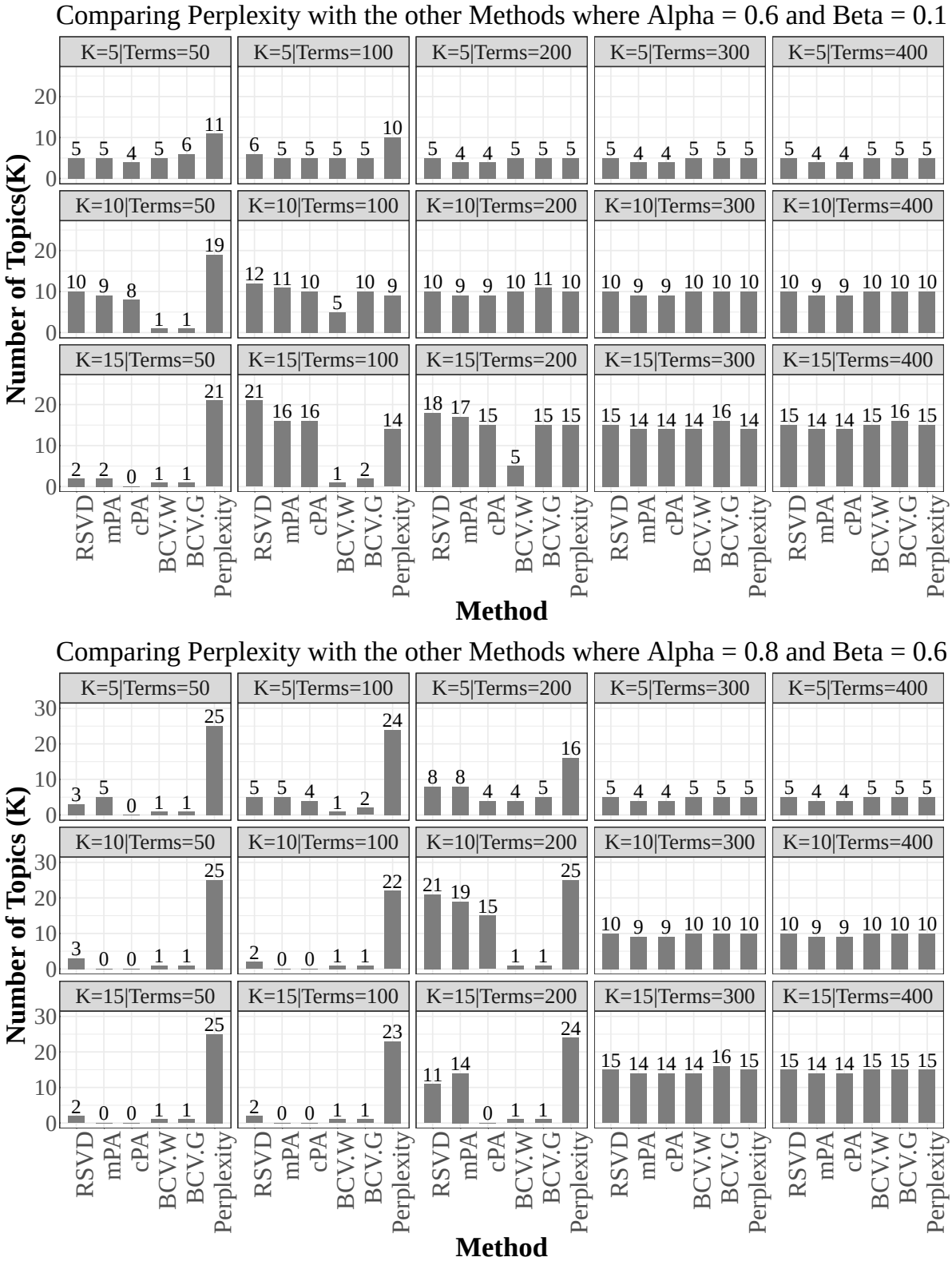


Figure 4.1 Estimation of the methods at each level of sparsity in the first (top) and second (bottom) experiments. In each panel, K and Terms in the figure refer to the number of topics and terms per document respectively.

mPA and cPA

Figure 4.1 also illustrates that all the methods have a higher loss with higher sparsity and higher priors (α, β).

In order to examine the effect of these parameters, we compute average loss of all the methods under each condition. Figure 4.2 displays the relation between average loss of all the methods and datasets characteristics such as sparsity (Terms per Documents), and each set of priors. We find that by increasing sparsity, all methods have a higher loss on average. Moreover, it shows that there is a higher average loss where priors have a higher value. We also can see that where there is a less sparsity, hyperparameters α and β do not affect the results significantly. Which means that if we could control the sparsity we can ignore the effect of Alpha and Beta values.

Another interesting point of the two plots in Figure 4.1 is that, with increased sparsity, the perplexity method overestimate whereas the other methods underestimate the number of topics. In order to investigate the behavior of perplexity with respect to the other methods, we define another variable as "status" that tells us whether each method over/underestimate if the estimations differ more than one from the correct latent topics. Table 4.1 shows the number of times perplexity overestimate in the same or opposite direction of the other methods.

To compare the behavior of perplexity with the other methods and assess if the overestimation of perplexity with respect to the other methods estimations is significant or not, we compute the odds ratio alongside the 95% confidence interval (CI) of perplexity. In order to avoid singular odds ratios, we make the Laplace correction, a kind of prior, and add 0.5 to all the values in table 4.1. Table 4.2 shows that the odds ratio of perplexity overestimation in opposite direction of the other method is 73.89 times greater than when is not overestimating and the CI also indicate that the odds ratio is statistically significant since it does not include 1.

Table 4.1 Perplexity Overestimation table

	Overestimate	Not_Overestimate
Opposite_Direction	9.00	0.00
Same_Direction	4.00	17.00

It also worth mentioning that perplexity can overestimate even more where $\alpha = 0.8$ and $\beta = 0.6$, but since to avoid time consumption we set a range of $K = 2 : 25$ to evaluate the LDA model using perplexity, it shows the maximum one.

Table 4.2 Perplexity Overestimation Odds Ratio and Confidence Interval

Odds Ratio	Lower Limit CI	Upper Limit CI
73.89	3.581	1524.226

Table 4.3 Accuracy and over/under estimation of each method

Methods	Correct	Overestimate	Underestimate
RSVD	0.57	0.20	0.23
mPA	0.53	0.30	0.17
cPA	0.57	0.17	0.27
BCV.W	0.47	0.00	0.53
BCV.G	0.47	0.17	0.37
Perplexity	0.47	0.43	0.10

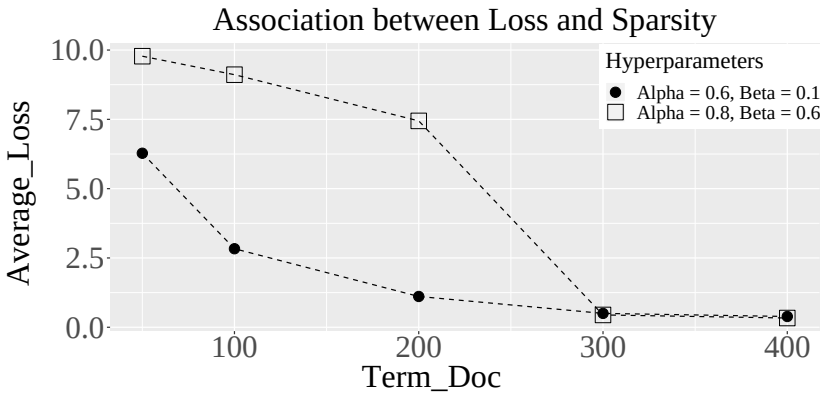


Figure 4.2 Association between sparsity, hyperparameters and loss

4.6 Conclusion and future work

We tackled the problem of finding the number of topics with well-known linear methods from the other fields that have not been utilized before. We showed that despite the fact these methods are within a linear framework and under certain conditions, some of them have a better performance than the commonly used perplexity measure to find the number of topics. We also show the boundaries where perplexity, as well as the other methods, become subject to unreliable estimations. We show that the performance deteriorates sharply as dataset sparsity and priors increase.

The boundaries have the interesting characteristic that the perplexity measure overestimates

the number of latent dimensions, whereas the other methods underestimates them. This leads to an interesting indicator that all methods are providing unreliable estimates.

The experiment results also corroborates the finding that LDA performs poorly with short texts [76], and in particular that the commonly used perplexity measure to derive the number of topics overestimates this parameter with high sparsity.

An important limitation of this work is that we do not have an analytical explanation for the different behavior of the methods.

Current research in each domain mostly focuses on specific technical approaches and evaluation that are known within the field which makes it difficult to conclude that if the achieved results are actually the best that could be and reliable. Our experiments show the necessity of engaging to more multidisciplinary methods in order to be aware of the reliability of an evaluation approach despite the popularity. Our experiments and contribution could lead to a more reliable and accurate estimation of the number of topics.

CHAPTER 5 ARTICLE 3: ESTIMATING THE NUMBER OF LATENT TOPICS THROUGH A COMBINATION OF METHODS

Neishabouri A., Desmarais M.C. Estimating the Number of Latent Topics Through a Combination of Methods. 25th International Hybrid Conference on Knowledge-Based Systems and Intelligent Information & Engineering, Procedia Computer Science, 2021, 192, 1190-1197.

Abstract

Estimating the number of latent topics is a prerequisite to using topic modeling algorithms such as LDA. A number of methods for this purpose are compared, including the *de facto* approach based on perplexity. Results from synthetic data show that a combination of methods yields better estimates than any single method, and in particular it provides an indicator of the reliability of the estimated number of topics, something no single method can do. In this paper, we extend the findings based on synthetic data over real data and introduce a technique, "Cosine inter-intra topics", to assess the validity of the ground truth of real data. Moreover, we measure the reliability of estimated number of topics through the variance of methods. Results corroborate the ones from synthetic data, suggesting the approach generalizes to real text corpus.

Keyword: Dimension reduction; Factor analysis; Latent topics; SVD; Perplexity; Parallel Analysis

5.1 Introduction

Topic modeling is most often conducted as an unsupervised learning task. Therefore, the number of topics that best represents a corpus of documents has to be determined in advance. The standard *wrapper* approach for this purpose in machine learning is to run the topic modeling algorithm over a range of values, and choose the value that has the greatest fit to the data. However, the "true" number of underlying of topics is unknown. To get around this issue, [1] used synthetic data, where the number of topics is known, and compared a variety of methods to determine if they correctly estimated the true number of topics. They showed that exploratory factor analysis methods and matrix factorization techniques often outperform the de facto method for LDA, perplexity. More importantly, they showed that the combination not only can yield an estimate better than any single method. The question we address in this study is whether this finding holds for real data. Moreover, we aim to measure

the reliability of estimated number of topics through variance of the methods estimations.

Since we do not know the "true" number of underlying topics of real datasets, We first introduce a new technique (Cosine inter-intra topics) to assess the validity of ground truth on the number of topics of a real dataset in which texts are categorized and whether the documents belong to just one topic or a combination of topics. Our objective is to know if the categories really represent the main latent topics of the corpus. The technique yields acceptable confidence in the underlying ground truth of the underlying dataset. Assuming the ground truth, we run a number of analysis similar to [1] and establish the convergence of all methods towards the ground truth at low sparsity of the data. Then, as sparsity increases, we show the effect on the performance of the different methods. Finally, we show how the combination of methods can lead to a better assessment of the number of latent topics, and also provide an indicator of the uncertainty of the result as sparsity increases.

5.2 Latent Topics and Perplexity

LDA is a generative, probabilistic model of documents that is introduced by [5]. This technique uses words distribution to cluster texts and discover latent topics from it. It is based on the assumptions that each document consists of a mixture of topics and that each topic consists of a set of words, both of which follow a Dirichlet prior.

This method has various hyperparameters that have to be defined before modeling:

1. Beta (β), is the Dirichlet prior of topic-word density.
2. Alpha (α), is the Dirichlet prior of document-topic density.
3. K is the number of topics.

K in LDA refers to the number of topics that needs to be defined in advance.

Latent factor analysis is a close cousin of the task of finding the number of topics in a corpus. Since the early work of [16] latent factors analysis has been widely studied in bioinformatics [18] and social sciences [17]. These approaches remain relatively unknown in the machine learning and NLP fields from which LDA emerged.

We briefly review the factor analysis methods that are relevant to topic analysis, along with the standard approach used with LDA, perplexity [5].

Parallel Analysis Parallel analysis (PA) was proposed by [29]. It is often considered to have a better accuracy than the other exploratory factor analysis methods [31, 27].

PA relies on the comparison of the eigenvalues of correlation matrices: the correlations of variables in the data matrix, and the correlations from multiple matrices of random values of the same dimensions as the data matrix, generated from the Normal distribution. It comes in two versions, cPA and mPA (see [75] for details).

Randomized Singular Value Decomposition, RSVD RSVD is similar to PA in that the technique relies on spectral analysis and comparison with randomized data [75]. The method first factorizes the data matrix, $\mathbf{A}$, with SVD into the product of the two eigenvectors matrices of $\mathbf{A}\mathbf{A}^{\mathrm{T}}$ and $\mathbf{A}^{\mathrm{T}}\mathbf{A}$, and a diagonal matrix containing the common singular values ordered by decreasing values, $\mathbf{\Sigma}$:

$$\mathbf{A} \;=\; \mathbf{U}_{m\times m}\mathbf{\Sigma}_{m\times n}\,\mathbf{V}^{T}_{n\times n} \tag{5.1}$$

Next, a column wise randomization of $\mathbf{A}$ is done and it is factorized with SVD to obtain $\mathbf{\Sigma}_r$, which is in turn compared with $\mathbf{\Sigma}$, the original singular values. The point on the diagonal where $\mathbf{\Sigma}_{(i,i)} < \mathbf{\Sigma}_{r(i,i)}$ indicates the number of latent dimensions (topics).

Perplexity Perplexity is the *de facto* metric to measure the fit with LDA [5, 10, 7]. It is the inverse probability of the data normalized by the number of words and is commonly used to assess the fit of an LDA model [5]. The optimal number of topics is chosen as the value that minimizes perplexity [48, 10, for eg.].

However, to skip defining the number of topics in advance, different versions of LDA topic modeling considering Hierarchical topic modeling (HTM) approaches including hierarchical Dirichlet process (HDP) based approaches which describes a Bayesian nonparametric method to clustering grouped data. In this method the number of topics discovered by the inference algorithm instead of being a parameter of the model to be defined in advance[50, 51, 52, 53, 77]. While, some studies have shown that although these methods could show an improvement over some datasets but they do not yield better results consistently [61]. They argue that some of the shortcomings of the methods are that they are not only heuristic-based and computationally expensive [54, 55, 56], but HDP-based methods also in practice need the number of allowed topics to be defined in advance [78, 79]. Over this list of studies, let us add that perplexity is also used to evaluate HDP and HTM based approaches and other derivatives of LDA models too [57].

The current study closely replicates the methodology of [1], but we use real data from preclassified text instead of synthetic data. Our objective is to compare the *de facto* method based on perplexity to determine the number of topics, with RSVD and factor analysis techniques

considering different data sparsity levels over real data.

While assessing the techniques over real data provides a more convincing argument of their performance in the field, the issue is whether we can trust we have a reliable structure of topics underlying this data. We test this assumption and then follow with an experiment to assess if the findings from synthetic data hold with real data below.

5.2.1 Datasets

We consider two real datasets in our experiment to assess performance of the methods over real data.

The first dataset is "Search Snippets" [64] which we will refer to as "DS 1". It has 12295 short texts from 8 different domains: Business, Computers, Culture-Art-Entertainment, Education-Science, Engineering, Health, Politics-Society and Sport. The average number of words per text is 14.42, and the sparsity of the term-document matrix is above 0.997, after cleaning the corpus by removing numbers, punctuation, stop words, and after stemming and lemmatization using the packages of "tm"[80] and "textstem" [81] in R.

The second dataset, named "DS2" is gathered from 32 texts that span 8 domains: Math, Web development, Machine Learning, Computer Science, History, Biology, Geography and Physics. The texts of this dataset are larger than DS 1. The average number of words per text in DS2 is 3018, and the sparsity of the term-document matrix is ≈ 0.93, after same text cleaning, stemming and lemmatization steps. [1]

We make the assumption that the domains of datasets constitute the ground truth by which we can define the topics.

5.2.2 Ground Truth Assumption Test: Cosine inter-intra topics

We consider the topics of these datasets as the ground truth. This assumption is put to test by comparing the inter-topic and intra-topic cosine similarity among words according to the following procedure.

First, we compute the centroid vector of word frequencies per topic:

$$\mathbf{c}_t = (\overline{w}_1, \overline{w}_2, ..., \overline{w}_n) \tag{5.2}$$

where $\mathbf{c}_t$ is the centroid of topic t and $\overline{w}_i$ is the average frequency of word i for that topic, and

[1] `https://github.com/asana-neishabouri/RealDS_LT` (Both data sets are available in the form of documents-term matrix in our github.)

the vocabulary size is n. Next, the cosine similarity between each document word frequency vector, $\mathbf{c}_d$ and each topic centroid, $\mathbf{c}_t$, is computed as:

$$\cos_{dt} \quad = \quad \mathbf{c}_d \cdot \mathbf{c}_t \; / \; ||\mathbf{c}_d|| \; ||\mathbf{c}_t|| \tag{5.3}$$

Finally, we compute the average of cosine similarities for a set of documents from a topic, $\mathcal{T}$, with the centroid of a topic t, $\mathbf{c}_t$:

$$\overline{\cos_{t\mathcal{T}}} = \sum_{d \in \mathcal{T}} \cos_{dt} \tag{5.4}$$

The results are shown in Figure 5.1. Rows represent the document topic sets ($\mathcal{T}$) and columns are the topic centroids, $\mathbf{c}_t$, and each cell of this matrix contains the corresponding value of $\overline{\cos}_{t\mathcal{T}}$. They illustrate that although some documents belong to more than one topic, the diagonal of each matrix shows greater average cosine similarity than any off-diagonal values, which supports the assumption that intra-document topics are more similar than inter-documents similarity.

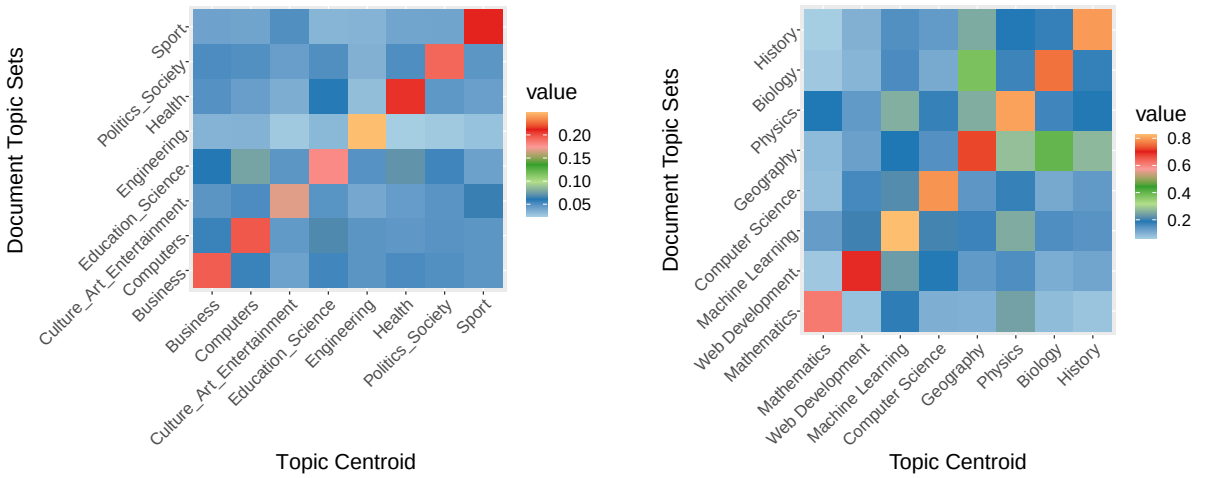
5.2.3 Experimental Setting

We run a number of experiments to evaluate each method's capacity to determine the "true" number of topics, as per our assumption that each document's category is its dominant topic.

The experiments are conducted over random samples of five domains from the total of eight domains for both DS1 (Search Snippets) and DS2 (32 textbooks). DS1 are short texts and 250 texts are sampled per domain and we control the sparsity by removing low frequency words, whereas we concatenate texts within each domain of DS2 to make different levels of sparsity. The size of documents and number of domains are comparable to the results of study [1] that was conducted over synthetic data. A summary of this study's results are reproduced in Table 5.1, for different levels of sparsity, as defined by the number of terms over vocabulary size which is 1000.

In the current study, we run experiments under conditions of sparsity: ≈ 0.90, ≈ 0.80 and ≈ 0.70.

We combine different means to manipulate the sparsity of the datasets. The first means is to randomly combine documents within a single domain. The effect is to create longer documents with larger vocabulary per document. The second means is to remove low frequency words.



(a) DS 1 (b) DS 2

Figure 5.1 Average cosine similarity between documents and domain centroids of datasets. Rows represent the document topic sets ($\mathcal{T}$) and columns are the topic centroids, $\mathbf{c}_t$.

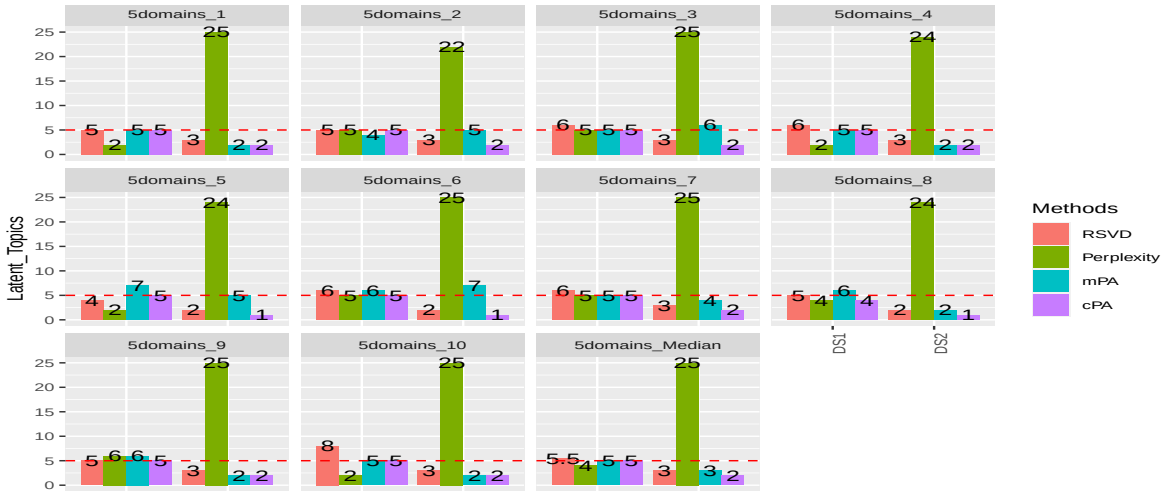
Table 5.1 Summary of the results of methods in [1] where number of topics is 5 at different levels of sparsity which is defined by the number of terms per document in the column of Terms/Doc.

Terms/Doc	Sparsity	RSVD	mPA	cPA	Perplexity
50	0.95	3	5	0	25
100	0.90	5	5	4	24
200	0.80	8	8	4	16
300	0.70	5	4	4	5
400	0.60	5	4	4	5

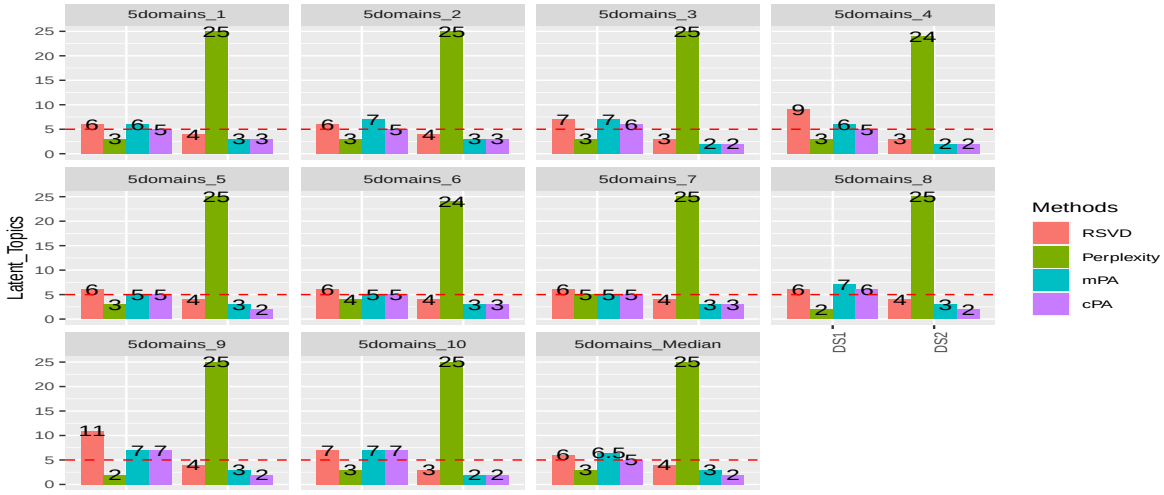
For a given sparsity level, we repeat each experiment ten times, each time sampling five domains out of the original eight in order to get better generalization across domains.

Then, we extend our experiment to assess the performance of the methods over all 8 topics of datasets by concatenating the texts of DS1 to create larger size document vocabulary and reduce the sparsity.

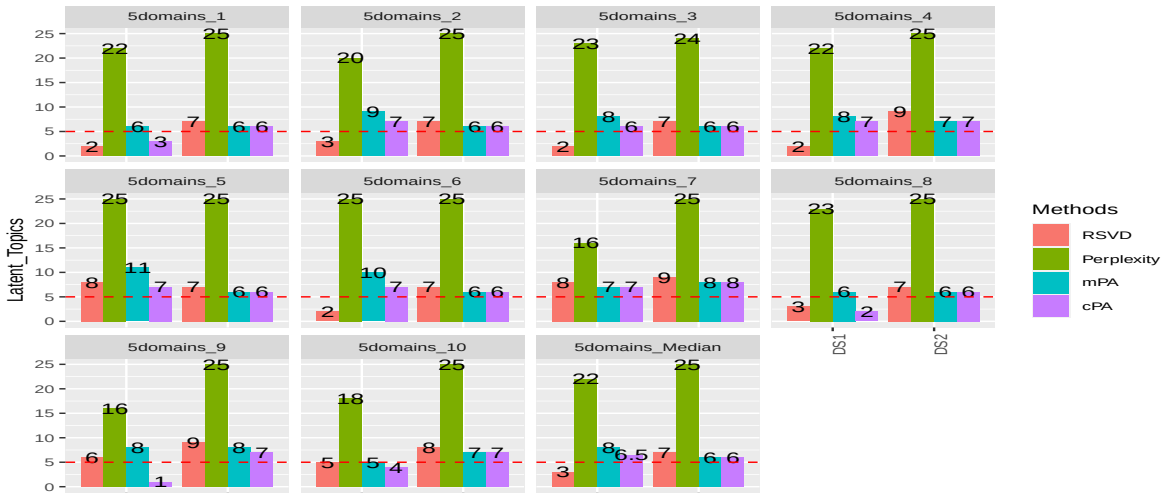
Note that the results for perplexity is topped at 25, and that the inferred number of topics is based on the results of 5-fold cross-validation to determine a suitable number of topics for LDA.



(a) Sparsity = 0.70



(b) Sparsity = 0.80



(c) Sparsity = 0.90

Figure 5.2 Estimating the number of latent topics over DS1 (four leftmost bars in each graph) and DS2 (four rightmost bars) where the number of topics = 5 and the levels of sparsity are ≈ 0.70, ≈ 0.80, and ≈ 0.90. Average dimensions of the term-document matrices of DS1 for the respective sparsity levels are 250×41 (a), 206×109 (b), and 164×80 (c), and for the DS2 are 5×1096 (a), 9×320 (b), and 20×4968 (c).

5.2.4 Results

The results from DS 1 and DS 2 are summarized in Figures 5.2 for three levels of sparsity and for which the ground truth is 5 topics according to the preclassification. The last panel at each sparsity level refers to the median of the methods after 10 runs.

Figure 5.2a reports the results at sparsity ≈ 0.70. For DS1, all methods have an estimate close to the "true" number of latent topics and the variance between the estimates is low. However, a larger error appears for DS2. Perplexity has a large overestimation (note that it is topped at 25) whereas other methods underestimate most of the time.

The general pattern is similar for sparsity ≈ 0.80 in Figure 5.2b. However, error estimates increase for DS1 larger and so does the variance across methods. As in Figure 5.2b, perplexity has a large overestimation on DS2, while the other methods have a closer estimation to the ground truth.

Finally, for sparsity ≈ 0.90 in Figure 5.2c, perplexity overestimate the number of topics in all the runs and the other methods mPA, cPA and RSVD also tend to overestimate the number of topics but have an estimate closer to the correct number of topics. Variance across methods is at its highest.

Figure 5.2 illustrates that, in general, the methods mPA, cPA and RSVD have a better performance than perplexity as sparsity increases. These results from real data sets are relatively consistent with the findings in [1] over synthetic data and are summarized in Table 5.1.

In addition, Table 5.2 shows the estimates of the methods on the DS1 and DS2 considering the 8 number of topics. From the general results, we can conclude that mPA and cPA have the best performance and RSVD is the second best, while perplexity over estimate the number of topics.

Table 5.2 Methods estimates over two datasets considering the 8 number of topics.

	Methods				Datasets characteristics			
	RSVD	mPA	cPA	Perplexity	size	N.topics	Sparsity	$\overline{N.words}/doc$
DS1	10	9	9	19	40 × 3225	8	0.77	3649.85
DS2	10	9	9	14	32 ×12543	8	0.93	3018

5.2.5 Reliability Assessment

Our hypothesis is that variance across method estimates at each condition is an indicator to the reliability of the inferred number of topics. The implication of this hypothesis is that there is an association between the accurate number of topics and the variance across method estimates.

In order to examine our hypothesis, we assess the odds ratio which is a ratio of correctness versus incorrectness of the method estimates considering variance across the method estimates as a case control.

To do so, we create a new feature of "status" and label the method estimates as "correct" if variance of the method estimate from the average of the method estimates $\leq$ 1 at each sparsity level of each dataset and "inCorrect" otherwise. Table 5.3 shows the frequency of the labels of "Correct" and "inCorrect" considering hypothetical group of variance.

Table 5.3 Frequency of "correct" versus "inCorrect" considering hypothetical group of variance.

	Correct	inCorrect
Variance ≤ 1	54	10
Variance > 1	31	145

The result shows that the odds ratio of correctness of the method estimates with variance $\leq$ 1 across the method is $\approx$ 25 times greater than when the variances are $>$ 1 with $CI = [2.451 - 4.007]$.

The fact that the performance varies across methods, indicates that the variance is a good indicator of the reliability of the estimate of the number of topics. This is an important finding since no other means existed to our knowledge to provide a reliability estimate of the number of topics obtained.

5.3 Conclusion

This paper extends the findings of [1] that were conducted over synthetic data and showed that linear methods such as mPA, cPA and RSVD have a similar to higher accuracy than perplexity to determine the true number of latent topics. The findings are confirmed over real data, where the ground truth assumption is unknown, but supported by a measure of Cosine inter-intra topic technique.

An important finding is that the variance across methods provides an indicator of the relia-

bility of the estimates obtained by any method. Perplexity often considerably overestimates the number of topics, especially at higher sparsity levels for which estimates are less reliable, whereas the tendency varies for the other methods.

This finding opens the possibility to not only provide more accurate estimates of the number of latent topics, but also assess the reliability of the estimates. Further research is under way to investigate this hypothesis.

A limitation of this study is that it is conducted over documents that have a single dominant topic. Yet documents often have more than one dominant topic and we can reasonably assume that, as a large proportion documents become mixtures of many topics, the performance would degrade. These limitations lead to further investigations that we plan to pursue.

CHAPTER 6 FINDING THE NUMBER OF TOPICS WHERE TOPICS ARE EMBEDDED WITHIN EACH OTHER

In the last two chapters, we explored methods to find the number of topics over documents that had one dominant topic where the topics were independent to one another. However, in many real world data sets such as text books, topics are not independent and are embedded within each other. In this chapter we aim to investigate the problem of finding the number of topics to cluster documents with an embedded topic structure. The text of the fourth paper starts in the next section 6.1.

6.1 ARTICLE 4: INFERRING THE NUMBER AND ORDER OF EMBEDDED TOPICS ACROSS DOCUMENTS

Neishabouri A., Desmarais M.C. Inferring the Number and order of Embedded Topics Across Documents. 25th International Hybrid Conference on Knowledge-Based Systems and Intelligent Information & Engineering, Procedia Computer Science, 2021, 192, 1198-1207.

Abstract

Documents are often organized according to an embedded structure, where a set of documents covers a topic and gets extended to a more specialized topic. We refer to this structure as embedded topics and address the issue of inferring the number and order of topics in a given corpus. While this problem is akin to finding clusters of documents and has been addressed in numerous studies in areas such as topic modeling, information extraction and knowledge discovery, we show that existing approaches are not effective in the specific context of embedded topic structures, and propose a novel technique for that purpose. We also propose an approach to uncover the order of such embedded topics. To determine the number of topics, the proposed method relies on the analysis of eigenvalues of a conditional probability matrix derived from the document-term matrix. We use Kmeans to determine the actual topic clusters, and conditional probability computation to determine the order. We compare the performance of our method to alternative methods for determining clusters and dimensionality. Results show that the proposed approach can effectively derive the right number of topics and embedding structure order.

Keywords: Dimension reduction; Conditional Topic Modeling; Topic modeling; Clustering;

6.2 Introduction

Many documents have a structure of embedded vocabulary. Textbook chapters often introduce basic concepts upon which later chapters expand. As such, early chapters will contain basic and topic specific vocabulary (concepts) that later chapters will reuse, as more advanced vocabulary is introduced. We claim that for documents structured according to such embeddings, the tasks of inferring the number of clusters, determining the cluster themselves and identifying their embedding order are not handled effectively by the popular topic modeling techniques such as latent Dirichlet allocation (LDA).

In this study we aim to address the issue of finding the number of topics where topics are embedded within each other and propose a new technique (*Conditional Eigenvalues*) using

eigenvalues and conditional probability to find the number of embedded topics/levels. We compare it with the best performing techniques to extract the number of topics.

Next, and given the number of embedded topics, we propose a novel clustering method named *condClust* that uses Kmeans and conditional probability to cluster documents-topics and find the order of topics.

While there are no methods to specifically identify the number and order of embedded topics, hierarchical topic modeling (HTM) and Correlation-based Topic Modeling (CTM) are the closest candidates [82, 83]. We retain CTM and standard LDA for the purpose of comparing the proposed clustering method with. The choice of LDA for the present work was motivated by the wide acceptance of this method, as evidenced in the survey by Jelodar et al. [62]. Since some articles, such as [82], show that CTM performs better than LDA and some show the opposite [84], we examine the clustering ability of both methods as standard methods for document clustering where topics have an embedding structure. We refer to such dataset as an "embedded topic" and the topics as "levels".

This paper is structured as follows. We begin with a review of existing works for clustering documents with an embedding structure, and a review of the best-known methods to find the correct number of topics. Then, we introduce our methods for the identification of the number of levels and to find order of the levels. Methodological details ensue and are followed by a discussion of results and some concluding remarks.

6.3 Related work

The problem of inferring embedded topic structure of documents is mostly seen as hierarchical topic modeling (HTM). Such topic modeling schemes have been proposed and studied by a number of researchers such as [52, 51, 53]. HTM applies a structured approach to topic identification. However, reports have also shown that HTM methods are not always stable and raised some of the shortcomings of HTM methods: they are heuristic-based and computationally expensive and do not reliably generate satisfactory outcomes [54, 55].

Issues with existing HTM methods were addressed by [85]. To improve these methods, they proposed knowledge-based hierarchical topic modeling (KHTM), where the topics are interrelated. They also used LDA as the base model. [77] also proposed a new hierarchical topic modeling (CluHTM) and showed improvement over other HTM methods including KHTM. They addressed the main challenges of HTM methods such as topic incoherence, unreasonable (hierarchical) structure, and issues related to the definition of the right number of topics and depth of the hierarchy. Although, their proposed method has shown improvement over state

of the art, it needs 5 parameters to be defined in advance manually including the minimum number of topics, maximum number of topics and depth of hierarchical structure, which we aim to avoid in our method.

To dispense with predefining the number of topics, the hierarchical Dirichlet process (HDP) is proposed. HDP is a nonparametric Bayesian analysis known as the Dirichlet process (DP) mixture model [49]. It is an extension of LDA, designed for cases where the number of topics is not known a priori. Moreover, a combination of HDP and LDA has been proposed to skip defining the number of topics [51, 52, 53]. However, researchers have shown that HDP-based methods in practice yield unrealistic results when faced with a large size of dataset [78, 79]. Researchers also discussed some of the drawbacks of these methods and report that they are not always stable [54, 55]. For example, Kim and Sudderth used toy and NIPS datasets and show that in the toy dataset LDA and HDP perform similarly but in the NIPS dataset LDA performs better than HDP. They also introduced a method called "DCNT" that outperformed both LDA and HDP but it requires topic number predefinition [57].

In view of the above works, we can conclude that none of the methods that learn the number of topics using HDP can systematically outperform LDA. As mentioned in [57], this could be due to the data distribution or the ability of the methods to capture non-Dirichlet distributions.

In all the topic modeling techniques reviewed above, including HDP-based topic modeling and neural network methods, many researchers still use either a predefined number of topics or trials to find the number of latent topics and they use perplexity, the best known and standard method for that purpose [5].

While perplexity is the most popular method to finding the number of topics, alternatives such as factor analysis methods and SVD-based methods prove to be successful. [1] show that perplexity fails to consistently outperform other methods under dataset different sparsity levels and distributions.

In the following we review alternative methods for finding the number of topics mentioned in [1]. Horn's parallel analysis (PA) [29] is one such technique, well known in psychometrics and social sciences. It is based on the correlation matrix of variables (e.g. terms-documents matrix in the context of topic modeling) and on its comparison with the correlation matrix of random data generated with the same dimensions as the original dataset. Factors for which the eigenvalues of the correlation matrix of the original dataset are larger than the mean or 95th percentile eigenvalues extracted from correlation matrices of the random generated datasets are considered as the essential factors. Another successful introduced method is Randomized singular value decomposition (RSVD) which is a SVD-based method for the identification of the number of latent factors as briefly introduced in [12] and extended in

[75]. This method is similar to PA but is not based on the correlation matrix of the dataset. It compares the singular values of the original matrix with a matrix of same size, generated with resample columns of the original matrix. The number of singular values of the original dataset that are greater than the resampled data demonstrate the significant factors to retain.

In the present paper, we evaluate and compare the above methods that were found to outperform the state of the art in finding the number of topics [1] alongside our new proposed method.

6.4 Proposed methods

In this section we introduce a new method titled "conditional eigenvalues" (CE) to identify the number of levels over synthetic and real data of embedded topic documents, where the levels (topics) are subsets of each other. Then, we propose another method, "conditional clustering" (condClust), to find the order of the discovered levels.

6.4.1 Conditional eigenvalues (CE)

Conditional eigenvalues (CE) is a proposed method for the identification of the number of levels of an embedded topic where the levels are subsets of each other. The method exploits conditional probability and the eigenvalues and is detailed in Algorithm 2. We first make a document term matrix after cleaning and stemming. After creating the document term matrix, we compute the conditional probability between the terms. Here, probability between two terms of T_i and T_j in the space of documents (D), is defined as the frequency of documents (D_{ij}) that contain T_i and T_j over the frequency of documents D_j that contain T_j :

$$\Pr\left(T_i \mid T_j\right) = \frac{|\{D_{ij} \mid (T_i \in D_{ij}) \wedge (T_j \in D_{ij})\}|}{|\{D_j \mid T_j \in D_j\}|} \tag{6.1}$$

Notice that line 3 of Algorithm 2 transforms the conditional probability from the $[0-1]$ range to $[0.5-1]$. This transformation reduces the variance of the conditional probability matrix. In order to compute the eigenvalues of the transformed conditional probability matrix ($\mathbf{D}$), we follow the procedure of [86] to obtain the symmetric matrix $\mathbf{M}$ from matrix $\mathbf{D}$ (line 4):

$$\mathbf{M} = \mathbf{D} + \mathbf{D}^{\mathrm{T}} \tag{6.2}$$

Finally, we compute the eigenvalues of **M** and consider the number of eigenvalues that are greater than the average eigenvalue as the number of levels of an embedded topic. These eigenvalues are known as characteristic roots, or latent roots [87, 88]. This method borrows from multiple factor analysis methods, including the PA method, that use eigenvalues to infer the number of latent variables. Based on literature, eigenvalues that are greater than the average eigenvalues are an indication to significant factors to retain [25].

Using the CE method, the number of levels of an embedded topic can be determined from the documents.

Algorithm 2 Conditional Eigenvalues (CE)

Inputs: Document-term matrix

Output: The number of levels of expertise

procedure CE
R = Compute conditional probability matrix; ▷ According to equation 6.1
D = Transform R using formula $1/(1+x)$;
M = Convert D to a symmetric matrix; ▷ According to equation 6.2
EigenM = Compute eigenvalues of M ;
CE = Count the number of eigenvalues that are greater than the average of EigenM;
return CE
end procedure

6.4.2 Conditional clustering (condClust)

Once we have obtained the number of embedded topics obtained from the procedure described above (section 6.4.1), we propose the conditional clustering method (condClust) method to discover their order (Algorithm 3).

This method includes the steps (line 2:5) to determine the number of levels of an embedded topic (see section 6.5.2 below) and *CE* method and employs this figure as k in Kmeans document clustering (line 7). After clustering the documents, we extract the total frequency of each term within each cluster. This operation produces one vector of term frequencies per cluster (line 10). Since in our method, each cluster corresponds to one level, we compute the conditional probability between the levels (l) considering all the terms (T) to find the dependency and order of the levels (line 12). To compute conditional probability between two levels of l_i and l_j in the space of terms (T), we use:

$$\Pr(l_i \mid l_j) = \frac{|\{T \mid (T \in l_i) \wedge (T \in l_j)\}|}{|\{T \mid (T \in l_j)\}|} \tag{6.3}$$

It is defined as the frequency of terms (T) that are in l_i and l_j over the frequency of terms T that are in l_j.

From the result of this operation, we can create a square conditional probability matrix for which rows and columns refer to different levels. The levels are then ordered in such a way that the sum of the conditional probability matrix rows lies in increasing order (lines 13 and 14). This implies that the first level has a smaller contribution from the other levels and likewise the last level that has the highest value and the largest contribution from the other levels.

Algorithm 3 Conditional Clustering (condClust)

Inputs: Document-term matrix(DTM)
T = empty matrix with k number of columns
Output: The ordered levels

1: **procedure** CONDCLUST
2: **if** $|subset| \leq 40\%$ ▷ $|subset|$ is an average combination of intersections between documents. **then**
3: (k = RSVD (**DTM**))
4: **else**
5: k = CE (**DTM**)
6: **end if**
7: C = Kmeans (**DTM**, k); ▷ Cluster documents into k clusters.;
8: **for** $(i = 1, i \leq k, i++)$ **do**
9: $\boldsymbol{C_i}$ = Document-Term matrix (**C[i]**);
10: **T[i]** = Column_Sums(**C[i]**); ▷ Computing sum of each term frequency withing each cluster and store in matrix **T** .
11: **end for**
12: **P** = Compute conditional probability between the columns of T; ▷ We creates matrix **P** which the rows and columns refer to the clusters/levels and the elements refer to the conditional probability.
13: $\boldsymbol{s}$ = Get_indices(sort(row_sums(**P**)), increasing = $True$); ▷ Finding indices of the rows of matrix **P** in such a way that the row sums are arranged in an increasing order.
14: **L** = **T**[, $\boldsymbol{s}$];
15: **return L**
16: **end procedure**

6.5 Experiments and results

We conduct experiments on real and synthetic data to assess the performance of the successful methods in state of the art for the task of inferring the number of levels of an embedded topic. We compare them with our proposed method CE. For this paper we use parallel analysis (PA) method of the "paran" library [68] in R. We assessed the accuracy of both variation of PA, which are mPA and cPA. And, we implement Randomized SVD (RSVD) in R using algorithm 1 in [75].

In the second experiment, we compare the clustering performance of LDA, CTM and Kmeans. We show that Kmeans has a higher accuracy where documents are multi topics with an embedding structure.

Moreover, we assess the accuracy of our proposed method *condClust* in ordering the topics/levels. The following outlines in more details the generation of synthetic data and the experimental results.

6.5.1 Datasets

Real Datasets

For our experiments, we consider that two varieties of possible topic embeddings. First, we consider a book contains an embedded topic structure, and the chapters as the levels and call it "Embedding 1". To do so, we examine two mathematics grade books (grades of [10-11]) considering the different number of chapters linearly such as $3, 5, 10$, and 15 as the number of levels to evaluate. Second, we investigate a knowledge domain of web development as an embedded topic and the different topics such as HTML, CSS, JavaScript, Bootstrap, and Jquery as the levels $[1-5]$ and call it "Embedding 2".

Synthetic datasets

In this study, we assume that an embedded topic which we call (E) consists of a set of terms $(T_1, T_2, .., T_n)$ that are grouped into different topics (k) while the topics are subsets of each other. This assumption can be formalized as:

$$\{k_1 \subset k_2 \subset ... \subset k_n | k_i \subset E\}$$

So, in order to generate synthetic data that could cover both variations of real data, we determine different sample sizes to subset from the documents of previous levels to include in

present level and we call it "subset size". Inheriting a larger subset size from a previous level to the next level creates a higher intersection between documents and can be considered as chapters of a book which corresponds to our first embedding type. And a smaller contribution from pre-levels to the next level makes the documents more orthogonal and independent, which refers to the second embedding type that is various topics within a knowledge area such as web development. We generate synthetic data to evaluate the performance of the methods for estimating the number of levels in an embedded topic. Using synthetic data was our choice of strategy because the ground truth of the generated data is known and we can control the data characteristics under investigation. To determine synthetic data characteristics, we examine the distribution of the real books under investigation and found that they follow log-normal distribution, and that each has its own mean and standard deviation. Moreover, we found that after cleaning and stemming the documents there is $\approx 10\%$ of shared words between all the levels which does not refer to any specific level. So, we simulate the synthetic data accordingly.

For our experiments we generate synthetic levels of different sizes with log-normal distribution inspired from real data as mentioned in algorithm 4, and then combine it with 10% of size of the vocabulary level of each level from shared words which are those words that do not belong to any specific level (line 6). We also considered different subset sizes for contribution of pre-levels to the next level to make different ratios of intersection between levels to cover both kinds of topic embeddings in our experiment. In algorithm 4 we specified $e = 50\%$ as the size of subsetting from the previous levels to generate the next level. However, we assess 6 different subset sizes of {10%, 20%, 40%, 50%, 70%, 90%} in our experiment. Moreover we consider different values for the mean and standard deviation of log-normal distribution as the inputs of our algorithm to generate the synthetic texts of each level to avoid bias.

6.5.2 Experiment 1 and results, inferring the number of levels of embedded topics

In this experiment, before evaluating the performance of the methods on real datasets, we assess the performance of the methods to infer the number of levels of embedded topics on synthetic datasets. To do so, we generate three synthetic embedded topics, namely "DS_nLev3", "DS_nLev5" and "DS_nLev10" with different numbers of levels, respectively 3, 5 and 10. The embedded topics are generated using Algorithm 4 considering different measures of subset size (line (e), Algorithm 4), in which a larger subset size can indicate chapters of a book while a smaller size can indicate books in a knowledge area. Moreover, we increase the number of documents of each level by performing two random samples with replacement from

Algorithm 4 Generate next level(l_i) with a given subset size of $e\%$ from the previous levels

Inputs:

(a) n = [200-700]; ▷ sizes to generate the levels vocabulary.

(b) m = $\{2, 6, 8, 13, 14\}$, s = $\{4, 6, 9, 10, 16, 19, 36, 38\}$; ▷ random value as mean(m) and standard deviation(SD) for log-normal(Lnorm) distribution

(c) T_c = define a set of shared terms of size 200 using sampling with replacement from 70 unique words generated from a log-normal distribution with $m = 2$ and $s = 4$;

(d) z = 10%; ▷ size for sampling from share terms

(e) e = 50%;

(f) lst = [] ; ▷ list of previous levels of level $\boldsymbol{l}_i$ such as ($\boldsymbol{l}_1, \cdots, \boldsymbol{l}_{i-1}$)if exists.

Output:

$\boldsymbol{l}_i(e)$ ▷ next level($\boldsymbol{l}_i$) considering the specified subset size (e) to inherent from the previous levels

procedure GENERATE NEXT LEVEL
  m_l = *sample*(m, 1); ▷ sample from the set of mean
  s_l = *sample*(s,1); ▷ sample from the set of (SD)
  T_r = *Generate_Terms*(n); ▷ generate distinct terms of size n as the vocabulary level.
  $\boldsymbol{p}_r$ = *Generate_Lnorm*(n, m_l, s_l) ▷ generate probability vector with log-normal distribution for the terms of T_r
  $\boldsymbol{l}_i$ = *sample*(T_r, n, $\boldsymbol{p}_r$, replacement = $True$) $\cup$ *sample*(T_c, $z \times (n + j)$) ;
    if $|lst| \geq 1$ **then**
      for $(q = 1, q \leq |\text{lst}|, q++)$ **do**
        $\boldsymbol{l}_i = \boldsymbol{l}_i \cup$ (*sample*(lst[q], size = e $\times$|lst[q]|, replacement = True));
      end for
    end if
    return $\boldsymbol{l}_i$
end procedure

each generated level of an embedded topic at certain subset size with sizes of $[40\% - 95\%]$. Once done, we will have 3 documents for each level of an embedded topic at each subset size.

However, per each sampling we add few new words as noise from a uniform distribution $[1 - 10]$ with replacement to increase differentiation between documents of a same level.

We generate an embedded topic for each of 6 subset sizes and then we create a document-term matrix (DTM) for each. We then apply the methods in the state of the art and our method to estimate the numbers of levels. We also repeat these experiments 10 times and report averages of bias error and RMSE of the methods in Table 6.1. The bias error and RMSE for each method is computed using:

$$\text{BiasError} = 1/n \sum_{i}^{n} (\hat{l} - l) \tag{6.4}$$

and,

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (l - \hat{l})^2} \tag{6.5}$$

Where l is the real number of levels and $\hat{l}$ is the estimated number.

Table 6.1 shows that in all synthetic datasets, for which subset size is greater than 40%, our proposed CE method has the most accurate estimations and outperforms the existing methods, while RSVD has a higher accuracy where subset size is less than or equal to 40%. It shows that as orthogonality between the levels decreases our method perform more reliably than the other methods.

Table 6.1 Bias and RMSE errors of the methods considering different number of levels and subset sizes on synthetic datasets. The bold numbers refer to the higher accuracy.

Error	Methods	**nLev = 3**						**nLev = 5**						**nLev = 10**					
		10%	**20%**	**40%**	**50%**	**70%**	**90%**	**10%**	**20%**	**40%**	**50%**	**70%**	**90%**	**10%**	**20%**	**40%**	**50%**	**70%**	**90%**
Bias	**RSVD**	**0**	**0**	**0**	-0.27	**0**	-0.30	**0**	**0**	**0**	-1.91	-1.9	-2	**0**	**-1.40**	-4	-6.82	-7	-7
	mPA	-1	-1	-1	-1.09	-1	-1.10	-1	-1	-1	-2.91	-2.9	-3	-1	-2.40	-5.20	-7.82	-8	-8
	cPA	-1	-1	-1	-1.18	-1	-1.20	-1	-1	-1	-2.91	-2.9	-3	-1.10	-2.40	-5.20	-7.82	-8	-8
	CE	1	-0.70	-0.90	**0**	0.1	**0**	1	**0**	1	**0**	**-1**	**0**	5.20	4.50	**3.60**	**-0.64**	**1.10**	**-1**
RMSE	**RSVD**	**0**	**0**	**0**	0.52	**0**	0.55	**0**	**0**	**0**	1.93	1.92	2	**0**	**1.48**	4	6.84	7	7
	mPA	1	1	1	1.13	1	1.14	1	1	1	2.92	2.92	3	1	2.45	5.22	7.84	8	8
	cPA	1	1	1	1.24	1	1.26	1	1	1	2.92	2.92	3	1.14	2.45	5.22	7.84	8	8
	CE	1	0.84	0.95	**0**	0.32	**0**	1	**0**	1	**1**	**1**	**0**	5.22	4.53	**3.63**	**1.31**	**1.14**	**1**

Then, in order to assess our method over the real data, we apply the methods on the real datasets from the mathematics and web development knowledge areas, referring to them as

Embedding 1 and Embedding 2. Table 6.2 shows the bias error of each method in estimating the number of levels for each mathematics grade book considering different number of chapters $(3, 5, 10, 15)$ as the number of levels and web development with five levels which refer to different topics of this domain such as "HTML", "CSS", "JavaScript","Bootstrap" and "JQuery". It shows that the CE method has a higher accuracy where the number of levels are greater than 3 and RSVD is the second-best method over real datasets considering different types of topic embedding.

Table 6.2 Bias error of methods on real datasets. The bold numbers refer to smaller bias errors.

	Embedding 1								Embedding 2
	G10				G11				Web
Levels	3	5	10	15	3	5	10	15	5
RSVD	**0**	**1**	5	7	**1**	2	5	8	-3
mPA	1	2	6	8	2	3	6	9	-4
cPA	1	2	6	8	2	3	6	9	-4
CE	-1	**-1**	**0**	**1**	**-1**	**0**	**0**	**6**	**1**

They show that the CE method has a higher accuracy where the number of levels are greater than 3 and RSVD is the second-best method over real datasets considering different types of topic embedding.

6.5.3 Experiment 2, clustering performance and order of levels of an embedded topic structure

We aim to assess the performance of our new proposed *condClust* method (method 3) in finding the order of the levels of an embedded topic structure. In order to find the order of the levels, first we need to do document-level clustering. So, in this experiment we evaluate clustering performance of popular methods of LDA and CTM in addition to Kmeans, which is used in our proposed *condClust* method over the synthetic generated embedded topic 2 (15 generated books with 5 levels) considering different subset sizes. Second, we assess the ability of our method to find the order of the clusters, which implies the order of the levels of an embedded topic. To do so, we assume to know the right number of clusters (levels) to evaluate the clustering methods. So, we employ the right number of clusters, which is 5 in this experiment for each method. We use of Kmeans function of "stats" library [89] in R with the default parameters and maximum number of iterations of 20 and also LDA and CTM functions of "topicmodels" library [47] with the default parameters. After that, to

find the clusters of documents using LDA and CTM, we used the "tidytext" library [90] and we assigned each word to a topic by greatest probability. So, as the result of this process, each topic contains a set of words which implies a cluster (level). On the other hand, we also extract the cluster label that assigned to each book using Kmeans clustering. Table 6.3 shows the ability of each method to discover clusters correctly considering different subset sizes.

Table 6.3 Results of clustering methods by subset size. The bold numbers refer to the higher accuracy.

	10%	20%	40%	50%	70%	90%
Kmeans	**0.82**	**0.91**	**0.94**	**0.96**	**0.95**	**0.96**
LDA	0.20	0.40	0.20	0.00	0.20	0.00
CTM	0.40	0.40	0.20	0.00	0.40	0.00

It illustrates that CTM has a higher accuracy than LDA in some of the subset size but could improve the Kmeans in none of the subset sizes.

Another interesting finding of this experiment is that it shows LDA and CTM cannot consistently cluster the documents in the predefined number of topics for all the subset sizes. And, it shows LDA and CTM have less than 5 clusters in 4/6 and 3/6 of the six subset sizes respectively. As the result of this experiment, we can confirm that Kmeans clustering has a higher accuracy in clustering of an embedded topic in all measures of intersection between documents. Therefore, in our proposed method *condClust* (method 3), according to our finding in experiment 1, we first find the level of intersection between documents to choose the right method to find the number of levels (lines 2: 5). Then, after finding the right number of levels, we follow steps 7 : 13 of the method to infer order of the levels. Then, to evaluate the accuracy of our proposed method, we compute the intersection between the discovered ordered levels and the actual levels. So, we compute the ratio of intersections between the set of terms of a sorted level using our method ($S = T_1, T_2, ..., T_n$) and terms of the original generated level ($L = T_1, T_2, ..., T_n$) as below:

$$\text{intersection} = |L \cap S|/|L| \tag{6.6}$$

Finally, we extend this experiment and examine the accuracy of our method in ordering the levels using the above formula for different subset sizes over the synthetic embedded topic 2 where the number of levels is 5. Table 6.4 shows the average accuracy of our method at each subset size after 10 repetitions. It illustrates that our method performs better as the measure of subset size increases.

Table 6.4 Average intersection between the original dataset and the ordered levels using *condClust* method considering different subset sizes after 10 repetitions.

	10%	20%	40%	50%	70%	90%
Accuracy	0.84	0.92	0.95	0.91	0.96	0.97

6.6 Discussion and analysis of errors

From the tables 6.1 we can infer that subset size of 40% is a boundary for the selection of the RSVD or the CE method to find the number of topics/levels.

Table 6.2 shows that the *CE* method outperforms the other methods over real data and the RSVD method is the second-best method. Tables 6.3 also confirms that as expected CTM outperforms LDA in datasets where topics are not independent. However, our experiments show that CTM and LDA failed to produce the known number of clusters in some of subset sizes. So, as [91] and [1] illustrate on LDA performance, our finding also confirms that these methods are very sensitive to the hyperparameters.

Moreover, because LDA and CTM follow a particular distribution (Dirichlet and logistic normal), they may not cluster the documents and topics precisely if the data does not follow the same distribution. Therefore, we can conclude that LDA and CTM methods are not always reliable where we face data with different distribution and follow an embedding structure and the results are not satisfactory compared to Kmeans in such datasets.

6.7 Conclusions and future work

In this study we tackled the problem of finding the number of topics and the order after clustering documents with an embedded topic structure. We conducted various experiments to address this issue and compared our proposed *CE* method with the state of the art. Moreover, we compared Kmeans clustering performance that is used in our second proposed method *condClust* with standard methods of LDA and CTM. First, we proposed the *CE* method to identify the number of topics/levels, which is the main issue for many clustering methods. We showed the accuracy and boundary of effectiveness of *CE* method in finding the number of topics/levels alongside the other methods in the state of the art over synthetic and real datasets.

Second, we proposed a new clustering method termed *condClust* that has a built-in function for identifying the number of topics. Contrary to many clustering methods, it has the benefit of being automatic and does not need any user intervention to define the number of topics and

it can be used in hierarchical topic modeling approaches to reduce the number of parameters. Moreover, it is a promising method for determining the order of topics where topics are subsets of each other. Although, we evaluate the performance of the ordering of our proposed method, but we could not find any relative alternative method to compare with. We also used synthetic data to show that LDA and CTM, which are the popular methods in document clustering and topic modeling, perform poorly over documents with an embedding structure. We plan to design more experiments and evaluate our method with data generated according to different distributions which is a limitation of this work. We also plan to apply our methods to more knowledge domains.

CHAPTER 7 GENERAL DISCUSSION

In the process of assessing the performance of a novel technique for LD estimation, RVSD, and comparing it with a number of alternative techniques, it became clear that data set attributes affected the performance of the various techniques in different ways. This observation led to the hypothesis that there is no single best method. It opened the opportunity to take advantage of this information to combine the different method estimates. Results supported the hypothesis that this combination is better than any single method and showed that using an ensemble algorithm yielded substantially more accurate estimates of the underlying number of LD.

Moreover, the observation of variance across methods leads to the idea of assessing the reliability of the estimate from the combination of methods. To appreciate the impact of variance on reliability, we run a hypothesis test on the methods loss by the variance across methods and report the results in table 7.1. A Chi-square test on the contingency table shows a p-value $p < 2.2e - 16$.

Table 7.1 Loss of methods by variance across methods

Variance	Loss (MAE)	
	<= 1 (low)	> 1 (high)
<= 1 (low)	1595	792
> 1 (high)	59	578

Results in table 3.7 showed that reliability assessment through this approach is better than through the standard approach of relying on the variance obtained from bootstrap experiments with a single method.

7.1 Best method per condition

A question we did not address in chapter 3 is whether the ensemble approach, EINLD, is better than the best method per condition? This would be the equivalent to using a lookup table to find the best method under a set of conditions and choose that single one[1].

Table 7.2's results show that while the best method per individual condition does outperform

[1]While the question was not addressed in the submitted paper of chapter 3, we will consider adding it in a resubmitted version.

Table 7.2 MAE and RMSE of predicting LD. Bold number indicate to the lowest error and how the best method. Results are the same as Table 3.6, except for those in italic.

	Best method's estimates per condition	Best method (mPA)	Average of the methods	EINLD
MAE	*3.30*	4.47	4.47	**1.30**
RMSE	*4.99*	9.01	8.89	**1.93**

the overall best method (mPA), the loss is still substantially lower for the EINLD method. These results confirm that the ensemble approach can leverage information from the performance of competing method to the best one to provide a better estimate of the number of LD.

7.2 Alternative approches to Random Forest

While the results obtained with Random Forest were clearly successful to demonstrate that an ensemble technique can outperform any single method, we did not compare alternatives to Random Forest.

One such alternative is gradient tree boosting [92]. It has been shown to be a method of choice for ensemble learning.

Another is a neural network approach. While it lacks the quality of being easily interpretable that decision trees have, neural networks can capture complex patterns and may be most appropriate given a large space of data set characteristics which would in turn result in a larger training data set.

7.3 Approches to practical applications

These investigations yield encouraging results. But can they lead to practical applications? We address this question and some of the approach's limitations and obstacles towards this goal.

To apply the general technique, one needs to create the set of conditions over which an ensemble algorithm can be trained. These conditions would be similar to the ones we saw in chapter 3, in particular data set sizes and sparsity which appears to be the most influencial

factors according to figures 3.11 and 3.10. The training data also needs to cover a range of LD that would presumably include the real number of LD.

The creation of training data that covers the above conditions is a relatively straightforward task, but the magnitude of the effort grows exponentially as the number and size of conditions increase. And it remains to be shown if the results hold as the space of conditions increase, in particular the number of LD. As we hinted already, exploring LD dimensions in the range of many tens and even hundreds, such as often found in embedding models, may prove unpractical.

7.4 Generalization to non linear relationships

An important limitation of this study stems from the use synthetic data created through a linear combination of latent factors. This limitation could be critical for deep learning models that are known to combine multiple layers of non linear relationships among factors. However, this limitation applies to most approaches in factor analysis. One path to resolve it would be the same as all approaches: find transformations that can bring factor effects closer to a linear space.

In addition, dimension reduction methods in neural network such as Boltzmann machine and autoencoder have their own hyperparameters that need to be defined in advance. We believe that investigating the performance and behavior of these methods under different condition of data and hyperparameters would be an interesting avenue to research for future work.

7.5 Multiplicity and orthogonality of topics

Moving to the investigation in finding the number of LD for topic modeling, we note that our investigations were limited to mostly single topic documents in our comparison with Perplexity. Should the documents be composed of multiple topics, it is unknown if the results reported would hold.

Furthermore, our investigation of the embedded topic structures did not cover the case of two or more orthogonal embedded structures, or more complex hierarchical or partial order structures. Yet, such structures are likely to be commonplace in reality. The embedded structure reflects the level of technicality or specialization within a topic (more specialized or involved topic covering the basic vocabulary of that topic and adding more technical terms, for example). This is undoubdly an interesting avenue of future research that we only briefly explored.

CHAPTER 8 CONCLUSION

8.1 Summary of the work

In this thesis, we tackle the issue of estimating the number of latent dimensions. We extend over current research in many ways. We study a large array of methods from multiple domains, from quantitative psychlogy to recommender sytems. We also go beyond normally distributed data, which is often the focus LD estimate methods, and study multinomial distributions that are typical of domains such as topic modeling. We bring techniques from Machine Learning to improve the estimate of LD and to provide an indication of the reliability of estimates. Using ensemble techniques is a novel approach to assessing the reliability of LD estimates that we believe offers a great potential.

An important contribution of our work is to show the impact of dataset attributes on the performance of the methods from different fields of study for finding the number of LD. Current research in each respective field has devoded little attention to how data set attributes affect the performance of different methods, such that it is difficult to conclude whether the achieved results are actually the best overall and reliable. Our experiments show that the best choice is not universal and depends on dataset attributes. Moreover, the results show that the identical estimations of the methods is an indicator to the reliability of the obtained number of LD. Despite the popularity of some methods within each field of study, these findings confirm the necessity of involving multidisciplinary methods and lead to two main contributions. We propose a novel multi-method approach called EINLD that take these finding into account and performs better than any single method in estimating the number of LD. This gain holds true not only if we take the best overall method across all conditions, but also the specific individual best method for each combination of conditions. Second, we tackle the problem of assessing the reliability of the obtained number of LD and propose a new method termed EINRLD using an ensemble technique that can assess the expected loss of the obtained LD, as defined by three main categories for the reliability (“Excellent”, “Good”, “Unreliable”).

In addition to the main contributions, EINLD and EINRLD, we explored the problem of assessing the number of LD in the particular field of topic modeling. The task of topic modeling is characterized by word clusters that can represent topics, and by documents that are composed of words from one or a few topics. We show that some of the methods we used in EINLD outperform the standard measure to establish the correct number of topics, Perplexity. We also explore the specific embedded topic structure, where the vocabulary

expands from one topic to another. We propose a new method "CE" to find the number topic subsets. The experimental results show that "CE" is better adapted than methods that assume an orthogonality of topics, or than CTM which drops that assumption to some extent. We elaborate a new clustering method termed "condClust" that has a built-in function to find the number of LD based on the boundary of effectiveness of the methods and it has a promising result in determining the order of topics.

8.2 Future work

While results from our experiments show convincing evidence that data set attributes affect the performance of the number of LD estimation methods, our investigation has been mostly empirical. We have not addressed the theoretical explanations the relationship between data set attributes and LD estimation, but this would certainly be a next step in future investigations. We expect that a better understanding and models of this relationship could in turn contribute to the issue of estimating the reliability of estimates.

On the practical side, the ensemble techniques developed could find their way into the design of a tool to better assess the number of LD, and provide its associated reliability level. Such tool would prove very useful for applications in topic modeling, factor analysis, recommender systems, among others.

On the question of inferring the number of topics under an embedded structure, this investigation has not addressed the case of multiple embedded topics. Yet, we can believet that this is a common occurence. For example, a textbook can cover a few general topics and contain chapters that dwell into more details for each of these topics. A physics book on waves and vibrations may start with generals chapters on periodic movements and vibrations, and later chapters will focus on more specific topics such as occillations, types of wave phenomena, interference and diffraction, etc. This problem is akin to the assessment of the level of technicality of a text segment. This is close to what HTM aims to do, but we have seen its limitations and the Conditional Eigenvalues could bring improvements over this difficult problem.

Finally, while the generic principle behind ensemble techniques is to combine multiple methods to better estimate or predict a variable, and our work falls in part within this framework, the assessment of the reliability of the prediction of the prediction through an ensemble technique has not received much attention and might be the most original contribution of this thesis. The idea coupld prove useful in multiple domains and applications. Any situation in which methods are affected in different manners under contextual factors such as data

set attributs can lend itself to this approach to assess reliability. This opens a large set of potential applications.

REFERENCES

[1] A. Neishabouri and M. Desmarais, "Reliability of perplexity to find number of latent topics," in *FLAIRS Conference*, 2020.

[2] M. G. R. Courtney and M. Gordon, "Determining the number of factors to retain in EFA: Using the SPSS R-Menu v2. 0 to make more judicious estimations," *Practical assessment, research & evaluation*, vol. 18, no. 8, p. 60, 2013.

[3] R. Cudeck and R. C. MacCallum, *Factor analysis at 100: Historical developments and future directions*. Routledge, 2007.

[4] H. Zarzour *et al.*, "A new collaborative filtering recommendation algorithm based on dimensionality reduction and clustering techniques," in *2018 9th international conference on information and communication systems (ICICS)*. IEEE, 2018, pp. 102–106.

[5] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation," *Journal of machine Learning research*, vol. 3, no. Jan, pp. 993–1022, 2003.

[6] J. Su and W.-P. Liao, "Latent Dirichlet allocation for text and image topic modeling," 2013.

[7] K. Henderson and T. Eliassi-Rad, "Applying latent Dirichlet allocation to group discovery in large graphs," in *Proceedings of the 2009 ACM symposium on Applied Computing*. ACM, 2009, pp. 1456–1461.

[8] M. Hoffman, F. R. Bach, and D. M. Blei, "Online learning for latent Dirichlet allocation," in *advances in neural information processing systems*, 2010, pp. 856–864.

[9] Y. Cha and J. Cho, "Social-network analysis using topic models," in *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2012, pp. 565–574.

[10] H. Q. Vu, G. Li, and R. Law, "Discovering implicit activity preferences in travel itineraries by topic modeling," *Tourism Management*, vol. 75, pp. 435–446, 2019.

[11] R. Kohavi, G. H. John *et al.*, "Wrappers for feature subset selection," *Artificial intelligence*, vol. 97, no. 1-2, pp. 273–324, 1997.

[12] B. Beheshti, M. Desmarais, and R. Naceur, "Methods to find the number of latent skills," in *EDM*, 2012.

[13] A. Neishabouri and M. C. Desmarais, "An ensemble approach to determine the number of latent dimensions and assess its reliability," 2021.

[14] A. Neishabouri and M. C. Desmarais, "Estimating the number of latent topics through a combination of methods," *Procedia Computer Science*, vol. 192, pp. 1190–1197, 2021.

[15] . Neishabouri and M. C. Desmarais, "Inferring the number and order of embedded topics across documents," *Procedia Computer Science*, vol. 192, pp. 1198–1207, 2021.

[16] L. Guttman, "Some necessary conditions for common-factor analysis," *Psychometrika*, vol. 19, no. 2, pp. 149–161, 1954.

[17] K. A. Bollen, "Latent variables in psychology and the social sciences," *Annual review of psychology*, vol. 53, no. 1, pp. 605–634, 2002.

[18] Z. M. Hira and D. F. Gillies, "A review of feature selection and feature extraction methods applied on microarray data," *Advances in bioinformatics*, vol. 2015, 2015.

[19] J. Venna *et al.*, "Information retrieval perspective to nonlinear dimensionality reduction for data visualization," *Journal of Machine Learning Research*, vol. 11, no. Feb, pp. 451–490, 2010.

[20] T. Hastie, R. Tibshirani, and J. Friedman, *The elements of statistical learning, second edition*. Springer series in statistics New York, 2009.

[21] P. Li and S. Chen, "A review on gaussian process latent variable models," *CAAI Transactions on Intelligence Technology*, 2016.

[22] A. Kelava *et al.*, "Nonparametric estimation of a latent variable model," *Journal of Multivariate Analysis*, vol. 154, pp. 112–134, 2017.

[23] A. Skrondal and S. Rabe-Hesketh, "Latent variable modelling: a survey," *Scandinavian Journal of Statistics*, vol. 34, no. 4, pp. 712–745, 2007.

[24] J. C. Hayton, D. G. Allen, and V. Scarpello, "Factor retention decisions in exploratory factor analysis: A tutorial on parallel analysis," *Organizational research methods*, vol. 7, no. 2, pp. 191–205, 2004.

[25] H. F. Kaiser, "The application of electronic computers to factor analysis," *Educational and psychological measurement*, vol. 20, no. 1, pp. 141–151, 1960.

[26] R. D. Ledesma and P. Valero-Mora, "Determining the number of factors to retain in EFA: An easy-to-use computer program for carrying out parallel analysis," *Practical assessment, research & evaluation*, vol. 12, no. 2, pp. 1–11, 2007.

[27] K. R. Mumford *et al.*, "Factor retention in exploratory factor analysis: A comparison of alternative methods." 2003.

[28] L. R. Fabrigar *et al.*, "Evaluating the use of exploratory factor analysis in psychological research." *Psychological methods*, vol. 4, no. 3, p. 272, 1999.

[29] J. L. Horn, "A rationale and test for the number of factors in factor analysis," *Psychometrika*, vol. 30, no. 2, pp. 179–185, 1965.

[30] R. Warne and R. Larsen, *Evaluating a proposed modification of the Guttman rule for determining the number of factors in an exploratory factor analysis.* SelectedWorks, 2014.

[31] R. G. Montanelli and L. G. Humphreys, "Latent roots of random data correlation matrices with squared multiple correlations on the diagonal: A monte carlo study," *Psychometrika*, vol. 41, no. 3, pp. 341–348, 1976.

[32] L. W. Glorfeld, "An improvement on Horn's parallel analysis methodology for selecting the correct number of factors to retain," *Educational and psychological measurement*, vol. 55, no. 3, pp. 377–393, 1995.

[33] W. F. Velicer, "Determining the number of components from the matrix of partial correlations," *Psychometrika*, vol. 41, no. 3, pp. 321–327, 1976.

[34] W. R. Zwick and W. F. Velicer, "Comparison of five rules for determining the number of components to retain." *Psychological bulletin*, vol. 99, no. 3, p. 432, 1986.

[35] B. P. O'connor, "Spss and sas programs for determining the number of components using parallel analysis and velicer's map test," *Behavior Research Methods*, vol. 32, no. 3, pp. 396–402, 2000.

[36] W. F. Velicer, C. A. Eaton, and J. L. Fava, "Construct explication through factor or component analysis: A review and evaluation of alternative procedures for determining the number of factors or components," in *Problems and solutions in human assessment*. Springer, 2000, pp. 41–71.

[37] P.-O. Caron, "Minimum average partial correlation and parallel analysis: The influence of oblique structures," *Communications in Statistics-Simulation and Computation*, pp. 1–8, 2018.

[38] D. Bokde, S. Girase, and D. Mukhopadhyay, "Matrix factorization model in collaborative filtering algorithms: A survey," *Procedia Computer Science*, vol. 49, pp. 136–146, 2015.

[39] A. B. Owen and P. O. Perry, "Bi-cross-validation of the svd and the nonnegative matrix factorization," *The annals of applied statistics*, pp. 564–594, 2009.

[40] A. B. Owen, J. Wang *et al.*, "Bi-cross-validation for factor analysis," *Statistical Science*, vol. 31, no. 1, pp. 119–139, 2016.

[41] B. Kanagal and V. Sindhwani, "Rank selection in low-rank matrix approximations: A study of cross-validation for nmfs," in *Proc Conf Adv Neural Inf Process*, vol. 1, 2010, pp. 10–15.

[42] Y. B. Touimi *et al.*, "Intelligent chatbot-lda recommender system," *International Journal of Emerging Technologies in Learning (iJET)*, vol. 15, no. 20, pp. 4–20, 2020.

[43] J. Zhang *et al.*, "Data-driven computational social science: A survey," *Big Data Research*, p. 100145, 2020.

[44] T. L. Griffiths and M. Steyvers, "Finding scientific topics," *Proceedings of the National academy of Sciences*, vol. 101, no. suppl 1, pp. 5228–5235, 2004.

[45] S. Jameel and W. Lam, "An n-gram topic model for time-stamped documents," in *European Conference on Information Retrieval*. Springer, 2013, pp. 292–304.

[46] L. Shang and K.-P. Chan, "A temporal latent topic model for facial expression recognition," in *Asian Conference on Computer Vision*. Springer, 2010, pp. 51–63.

[47] B. Grün and K. Hornik, "topicmodels: An R package for fitting topic models," *Journal of Statistical Software*, vol. 40, no. 13, pp. 1–30, 2011.

[48] Q. Chen, L. Yao, and J. Yang, "Short text classification based on LDA topic model," in *2016 International Conference on Audio, Language and Image Processing (ICALIP)*. IEEE, 2016, pp. 749–753.

[49] Y. W. Teh *et al.*, "Hierarchical Dirichlet processes," *Journal of the american statistical association*, vol. 101, no. 476, pp. 1566–1581, 2006.

[50] T. L. Griffiths *et al.*, "Hierarchical topic models and the nested chinese restaurant process," in *Advances in neural information processing systems*, 2004, pp. 17–24.

[51] D. M. Blei, T. L. Griffiths, and M. I. Jordan, "The nested chinese restaurant process and bayesian nonparametric inference of topic hierarchies," *Journal of the ACM (JACM)*, vol. 57, no. 2, p. 7, 2010.

[52] D. Mimno, W. Li, and A. McCallum, "Mixtures of hierarchical topics with pachinko allocation," in *Proceedings of the 24th international conference on Machine learning*. ACM, 2007, pp. 633–640.

[53] J. Paisley *et al.*, "Nested hierarchical Dirichlet processes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 2, pp. 256–270, 2014.

[54] J.-H. Kang, J. Ma, and Y. Liu, "Transfer topic modeling with ease and scalability," in *Proceedings of the 2012 SIAM International Conference on Data Mining*. SIAM, 2012, pp. 564–575.

[55] X.-L. Mao *et al.*, "SSHLDA: a semi-supervised hierarchical topic model," in *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning*. Association for Computational Linguistics, 2012, pp. 800–809.

[56] H. M. Wallach, D. M. Mimno, and A. McCallum, "Rethinking LDA: Why priors matter," in *Advances in neural information processing systems*, 2009, pp. 1973–1981.

[57] D. Kim and E. Sudderth, "The doubly correlated nonparametric topic model," *Advances in Neural Information Processing Systems*, vol. 24, pp. 1980–1988, 2011.

[58] E. Zavitsanos, G. Paliouras, and G. A. Vouros, "Non-parametric estimation of topic hierarchies from texts with hierarchical Dirichlet processes." *Journal of Machine Learning Research*, vol. 12, no. 10, 2011.

[59] G. Brunner *et al.*, "Disentangling the latent space of (variational) autoencoders for NLP," in *UK Workshop on Computational Intelligence*. Springer, 2018, pp. 163–168.

[60] J. Qiang *et al.*, "Short text topic modeling techniques, applications, and performance: a survey," *IEEE Transactions on Knowledge and Data Engineering*, 2020.

[61] M. R. Bhat *et al.*, "Deep LDA: A new way to topic model," *Journal of Information and Optimization Sciences*, vol. 41, no. 3, pp. 823–834, 2020.

[62] H. Jelodar *et al.*, "Latent Dirichlet allocation (LDA) and topic modeling: models, applications, a survey," *Multimedia Tools and Applications*, vol. 78, no. 11, pp. 15 169–15 211, 2019.

[63] F. M. Harper and J. A. Konstan, "The movielens datasets: History and context," *Acm transactions on interactive intelligent systems (tiis)*, vol. 5, no. 4, pp. 1–19, 2015.

[64] X.-H. Phan, L.-M. Nguyen, and S. Horiguchi, "Learning to classify short and sparse text & web with hidden topics from large-scale data collections," in *Proceedings of the 17th international conference on World Wide Web*, 2008, pp. 91–100.

[65] T. Mikolov *et al.*, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013, pp. 3111–3119.

[66] T. Ding, W. K. Bickel, and S. Pan, "Multi-view unsupervised user feature embedding for social media-based substance use prediction," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2017, pp. 2275–2284.

[67] E. Dobriban, "Permutation methods for factor analysis and PCA," *The Annals of Statistics*, vol. 48, no. 5, pp. 2824–2847, 2020.

[68] A. Dinno, *paran: Horn's Test of Principal Components/Factors*, 2018, R package version 1.5.2. [Online]. Available: https://CRAN.R-project.org/package=paran

[69] B. P. O'Connor, *paramap: Factor Analysis Functions for Assessing Dimensionality*, 2019, R package version 1.9.1.

[70] P. O. Perry, *bcv: Cross-Validation for the SVD (Bi-Cross-Validation)*, 2015, R package version 1.0.1.

[71] B. Sarwar *et al.*, "Application of dimensionality reduction in recommender system—a case study," Minnesota Univ Minneapolis Dept of Computer Science, Tech. Rep., 2000.

[72] J. Devlin *et al.*, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[73] X. Li, S. Wang, and Y. Cai, "Tutorial: Complexity analysis of singular value decomposition and its variants," *arXiv preprint arXiv:1906.12085*, 2019.

[74] H. Zhang *et al.*, "An LDA-based community structure discovery approach for large-scale social networks," in *2007 IEEE Intelligence and Security Informatics*. IEEE, 2007, pp. 200–207.

[75] A. Neishabouri and M. C. Desmarais, "Investigating methods to estimate the number of latent dimensions under different assumptions and data characteristics," Tech. Rep., 2019.

[76] J. Li *et al.*, "Key word extraction for short text via word2vec, doc2vec, and textrank," *Turkish Journal of Electrical Engineering & Computer Sciences*, vol. 27, no. 3, pp. 1794–1805, 2019.

[77] F. Viegas *et al.*, "Cluhtm-semantic hierarchical topic modeling based on cluwords," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 8138–8150.

[78] C. Wang, J. Paisley, and D. Blei, "Online variational inference for the hierarchical Dirichlet process," in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*. JMLR Workshop and Conference Proceedings, 2011, pp. 752–760.

[79] S. Koltcov *et al.*, "Analyzing the influence of hyper-parameters and regularizers of topic modeling in terms of renyi entropy," *Entropy*, vol. 22, no. 4, p. 394, 2020.

[80] I. Feinerer, K. Hornik, and D. Meyer, "Text mining infrastructure in r," *Journal of Statistical Software*, vol. 25, no. 5, pp. 1–54, March 2008. [Online]. Available: https://www.jstatsoft.org/v25/i05/

[81] T. W. Rinker, *textstem: Tools for stemming and lemmatizing text*, Buffalo, New York, 2018, version 0.1.4. [Online]. Available: http://github.com/trinker/textstem

[82] D. Blei and J. Lafferty, "Correlated topic models," *Advances in neural information processing systems*, vol. 18, p. 147, 2006.

[83] S. Lee, J. Song, and Y. Kim, "An empirical comparison of four text mining methods," *Journal of Computer Information Systems*, vol. 51, no. 1, pp. 1–10, 2010.

[84] H. Hruschka, "Linking multi-category purchases to latent activities of shoppers: analysing market baskets by topic models," *Marketing: ZFP–Journal of Research and Management*, vol. 36, no. 4, pp. 267–273, 2014.

[85] Y. Xu *et al.*, "Hierarchical topic modeling with automatic knowledge mining," *Expert Systems with Applications*, vol. 103, pp. 106–117, 2018.

[86] A. C. Aitken, *Determinants and matrices*. Read Books Ltd, 2017.

[87] G. Arfken, "Mathematical methods for physicists, volume third edition," 1985.

[88] M. Marcus and H. Minc, *Introduction to linear algebra*. Courier Corporation, 1988.

[89] R Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, 2013, ISBN 3-900051-07-0. [Online]. Available: http://www.R-project.org/

[90] J. Silge and D. Robinson, "tidytext: Text mining and analysis using tidy data principles in R," *JOSS*, vol. 1, no. 3, 2016. [Online]. Available: http://dx.doi.org/10.21105/joss.00037

[91] J. Tang *et al.*, "Understanding the limiting factors of topic modeling via posterior contraction analysis," in *International Conference on Machine Learning*, 2014, pp. 190–198.

[92] J. H. Friedman, "Stochastic gradient boosting," *Computational statistics & data analysis*, vol. 38, no. 4, pp. 367–378, 2002.

APPENDIX A DATA GENERATION PROCEDURE IN ARTICLE 1

Algorithms to generate data with normal and multinomial distributions are below:

Algorithm 5 Generating synthetic data of size $m \times n$ with a normal distribution

Inputs: n = {250,500} ; m = {150, 250, 300} ; k= {5, 10, 15}

Output: synthetic data set with a normal distribution

procedure GENERATING SYNTHETIC DATA SET WITH A NORMAL DISTRIBUTION

P = Generating a matrix of size $m \times k$; ▷ k represents the number of latent dimensions

Q = Generating a matrix of size $k \times n$; ▷ k represents the number of latent dimensions

ϵ = Generating a noise matrix of size $m \times n$; ▷ $\epsilon \sim \mathcal{N}(0, 1)$

R = $P \cdot Q + \epsilon$;

return R

end procedure

Algorithm 6 Generating synthetic rating data of size $m \times n$ with a multinomial distribution

Inputs: n = {250,500} ; m = {150, 250, 300} ; k= {5, 10, 15}

Output: synthetic rating data set with a multinomial distribution

procedure GENERATING SYNTHETIC RATING DATA SET WITH A MULTINOMIAL DISTRIBUTION

P = Generating a rating matrix of size $m \times k$; ▷ P contains row of unit vectors (one-hot vectors) ; k represents the number of latent dimensions;

Q = Generating a matrix of size $k \times n$; ▷ Q contains column unit vectors; k represents the number of latent dimensions

ϵ = Generating a noise matrix of size $m \times n$; ▷ $\epsilon \sim \mathcal{N}(0, 1)$

R_0 = $P \cdot Q + \epsilon$;

R = $\lfloor 2 \times R\_0 + 2 \rceil$;

return R

end procedure