Qianqian Qi

Some studies in correspondence analysis of texts

Some studies in correspondence analysis of texts

Enkele studies in correspondentieanalyse van teksten
(met een samenvatting in het Nederlands)

Proefschrift

ter verkrijging van de graad van doctor aan de
Universiteit Utrecht
op gezag van de
rector magnificus, prof.dr. H.R.B.M. Kummeling,
ingevolge het besluit van het college voor promoties
in het openbaar te verdedigen op

vrijdag 18 oktober 2024 des ochtends te 10.15 uur

door

Qianqian Qi

geboren op 1 maart 1993

te Shandong, China

Promotoren:
Prof. dr. P.G.M. van der Heijden
Prof. dr. D.L. Oberski
Copromotor:
Dr. D.J. Hessen

Beoordelingscommissie:
Prof. dr. E.M.L. Dusseldorp
Prof. dr. M. Greenacre
Dr. P. Lugtig
Prof. dr. A.P.J.M. Siebes
Prof. dr. R. van de Schoot

This dissertation was accomplished with financial support from the China Scholarship Council (CSC).

Some studies in correspondence analysis of texts
Dissertation Utrecht University, Utrecht, the Netherlands
Met een samenvatting in het Nederlands

Print: Ridderprint | https://www.ridderprint.nl
ISBN: 978-90-393-7739-0
DOI: https://doi.org/10.33540/2453

Contents

Chapter 1

Introduction

Text data have found increasing interest in recent years because of the ubiquity of text data on the Web, social media, digital library, and others (Aggarwal, 2018). Some important applications of text data include searching the Web, filtering spam emails, and recommending articles or movies. Two main areas to study the applications of text data are text mining (also called machine learning from text) and natural language processing (NLP).

Text mining and NLP often have overlap of the tasks, methods, and goals, and the concepts are sometimes used interchangeably (Bagheri, 2021). On the one hand, text mining refers to the discovery by computer of new, previously unknown information, by automatically extracting information from different written resources (Hearst, 1999). On the other hand, NLP is any computer-based algorithm that handles, augments, and transforms natural language so that it can be represented for computation (Yim, Yetisgen, Harris, & Kwan, 2016).

This dissertation focuses on the comparison of popular techniques in text mining and NLP, and popular techniques in statistics. We explore whether a popular technique in statistics is also a good method in text mining and NLP, thus facilitating the development of text mining and NLP. Specifically, we compare the popular statistical technique correspondence analysis (CA) (Greenacre, 1984) with several popular text mining and NLP techniques, in particular with latent semantic analysis (LSA) (Dumais, Furnas, Landauer, Deerwester, & Harshman, 1988), PPMI-SVD (Levy & Goldberg, 2014), GloVe (Pennington, Socher, & Manning, 2014), and SGNS (Mikolov, Sutskever, Chen, Corrado, & Dean, 2013).

In text mining and NLP applications, a vector representation of text data is the key in designing an effective machine learning algorithm (Aggarwal, 2018; Le & Mikolov, 2014). For example, the nearest-neighbour algorithm for text categorization or the K-means algorithm for text clustering typically requires the text input to be represented as a vector. A commonly used vector representation for text data is the bag-of-words. In this case, the order of words is not considered. Many text document collections (such as web pages) are converted into document-term matrices, such that each document is represented by a row of the matrix. Terms (such as words) in the documents are represented by the columns of the matrix.

Document-term matrices are used in a large number of potentially important applications (Turney & Pantel, 2010; Aggarwal, 2018), including returning related information for a given query, classifying spam and non-spam emails, and finding a short answer to a question. In some applications, such as finding frequently co-occurring

groups of k words, a binary representation is sufficient where a cell is 1 if a word is present in a document, otherwise 0. However, a binary representation loses a lot of information because it does not contain the frequencies of terms. The frequency-based representation is more popular, where a cell contains information about the number of times a term occurs in a document.

In a document-term matrix, the representation of the document is often high-dimensional, sparse, and non-negative (Aggarwal, 2018). This is because the dimensionality depends on the number of words from all documents, which is typical large. Furthermore, a document contains a limited number of words. Consider the following document corpus with three documents and a vocabulary of eleven words that illustrates how to create a document-term matrix (Albright, 2004):

- d1: error invalid message file format,
- d2: error unable to open message file using message path,
- d3: error unable to format variable.

A document-term matrix is created for the above three documents as follows:

	error	invalid	message	file	format	unable	to	open	using	path	variable
d1	1	1	1	1	1	0	0	0	0	0	0
d2	1	0	2	1	0	1	1	1	1	1	0
d3	1	0	0	0	1	1	1	0	0	0	1

Table 1.1: Document-term matrix

In each cell the count represents the number of times a particular word is used in a particular document. For example, for document 2, *message* occurs twice, so the entry (*d2*, *message*) is 2, *invalid* does not occur in document *2*, so the entry (*d2*, *invalid*) is 0. Thus each document is represented as a vector of 11 values. For instance, *d2* is represented as $[1, 0, 2, 1, 0, 1, 1, 1, 1, 1, 0]$. Similarly, each word is represented as a column of the matrix. For example, *message* is represented as $[1, 2, 0]$.

An alternative way to represent words is to use a word-context matrix (Turney & Pantel, 2010; Aggarwal, 2018; Jurafsky & Martin, 2023). In a word-context matrix, rows are labelled by a set of words and columns by the contexts of these words, in which the contexts are given by words, phrases, or others. Each word is usually represented as a row of the matrix. The word-context matrix follows the distributional hypothesis in linguistics which is that words that occur in similar contexts tend to have similar meanings (Harris, 1954). Word-context matrices have many potentially important applications (Billhardt, Borrajo, & Maojo, 2002; Turney & Pantel, 2010; Hossain, Zahin Mauni, & Rab, 2022), including discovering different senses of polysemous words, automating thesaurus generation, and other downstream text mining and NLP tasks such as information retrieval and text categorization.

In a word-context matrix, a count in a cell can be the number of times the row (target) word and column (context) word co-occur in some context in a text. The

context generally refers to a window around a row word. For example, if the context is represented by 2 words to the left of a row word and 2 words to the right, a cell represents the numbers of times the column word occurs in such a ±2 word window around the row word. Consider the following text with a vocabulary of five words as an illustration of how to create a word-context matrix:

- a sunny day is a happy day

We use a window of size 2, i.e., 2 words to each side of a row word as its context words. The word-context matrix for the above text is

	a	sunny	day	is	happy
a	0	1	3	1	1
sunny	1	0	1	1	0
day	3	1	0	1	1
is	1	1	1	0	1
happy	1	0	1	1	0

Table 1.2: Word-context matrix

For example, for a row word *a*, the texts in its ±2 word window are

- sunny day
- day is
- happy day

We count the context words around *a*, and find that the entry (*a*, *a*) is 0, the entry (*a*, *sunny*) is 1, the entry (*a*, *day*) is 3, and so on. Each row word is represented as a vector of 5 values. Like a document-term matrix, a word-context matrix is high-dimensional, sparse, and non-negative.

This dissertation focuses on the document-term matrix and word-context matrix because these two matrices form the key building blocks for many text applications. The rest of this introductory chapter is organized as follows. In the next section, we will discuss popular techniques in text mining and NLP that are relevant for the analysis of document-term and word-context matrices. Section 1.2 introduces correspondence analysis. The questions we will study are given in Section 1.3. Section 1.4 introduces the contribution and outline of this dissertation. Section 1.5 presents possible future research.

1.1 Techniques in text mining and natural language processing

Document-term and word-context matrices are sparse and high-dimensional (Aggarwal, 2018). The process of creating low-dimensional representations of texts,

that reflect the information in the original matrix as good as possible, is referred to as dimensionality reduction. Dimensionality reduction is associated with the representation of text data and thus forms the key building block for other text applications such as clustering, categorization, and information retrieval.

In the machine learning literature, little to no attention has been paid to correspondence analysis (CA). Other popular dimensionality reduction methods receive more attention, like latent semantic analysis (LSA) (Aggarwal, 2018). LSA uses the singular value decomposition (SVD). SVD is also used in the calculation of the CA solution. Since CA also seems appropriate for the analysis of texts, the question arises whether CA performs well in analyzing texts compared with tools from text mining and NLP. So, in this dissertation, it is investigated whether CA is a good dimensionality reduction technique in text mining and NLP.

This section mainly introduces four popular techniques in text mining and NLP that are relevant for the analysis of document-term and word-context matrices. These four techniques involve two SVD type techniques: LSA and PPMI-SVD, and two gradient algorithm type techniques: GloVe and SGNS. More in-depth discussions of LSA are in the Chapter 2 and of PPMI-SVD, GloVe, and SGNS are in the Chapter 4.

1.1.1 Latent semantic analysis

LSA is a dimensionality reduction method used in the context of the document-term matrix (Aggarwal, 2018). Most text mining and NLP applications require the computation of similarities between pairs of documents, i.e., between the rows of the document-term matrix. When calculating similarities between documents, the length of a document may have undesirable effects, as a larger length of a document makes all the frequencies for that document larger. For example, when Euclidean distances are used for distance computation, the distance between two long documents tends to be very large, whereas the distance between two short documents tends to be much smaller. This is undesirable because the similarities between documents are severely affected by the lengths of the documents, whereas one is more interested in the *distribution* of counts per document. Moreover, not all words have equal importance. Low-frequency words are often more discriminative than high-frequency words. For example, the word "the" tends to occur in each document, which tends to be less discriminative than the word "good" occurring in less documents.

Weighting can be used to prevent differential lengths of documents from having differential effects on the representation, or be used to impose certain preconceptions of which terms are more important (Deerwester, Dumais, Furnas, Landauer, & Harshman, 1990). TF-IDF is a commonly used weighting scheme (Dumais, 1991; Aggarwal, 2018; Jurafsky & Martin, 2023). TF-IDF is intended to reflect how important a term is to a document in the entire collection. TF stands for the term frequency in a document. It can refer to elements of the raw document-term matrix. IDF stands for inverse document frequency. The inverse document frequency of a term is a decreasing function of the number of documents in which it occurs. This means that a term

that occurs in less documents receives more weight.

In most text mining and NLP applications, the focus is on whether two documents have similar meaning by calculating the similarity between the two documents. However, in a document-term matrix, individual words provide incomplete and unreliable evidence about the meaning of a document partly because of synonymy and polysemy (Dumais et al., 1988; Deerwester et al., 1990). Synonymy refers to equivalence in meaning of different words. For example, large and huge are almost synonymous but are handled as completely different words (two columns in the document-term matrix) in the similarity computations of documents. Polysemy refers to the fact that words can have more than one distinct meaning. For instance, mouse can refer to a computer device or to an animal but will be handled as the same word (one column in the document-term matrix) in the similarity computation of documents.

LSA tries to overcome the issues of synonymy and polysemy (Dumais et al., 1988; Deerwester et al., 1990; Aggarwal, 2018). In LSA, individual terms are replaced with derived latent semantic factors. The particular technique used is singular value decomposition (SVD). By SVD, a document-term matrix is decomposed into a set of orthogonal factors. Thus, by a smaller number of these orthogonal factors, low-dimensional representations of documents and terms are obtained.

SVD approximates a matrix by the product of three smaller matrices which provides the optimal approximation of the original matrix in a least-squares sense. The idea is that the product of these three smaller matrices captures the major association structure in the matrix and throws out noise (Dumais et al., 1988; Deerwester et al., 1990; Dumais, 1991; Aggarwal, 2018). SVD tries to pull out the latent semantic concepts in the data, and each document is represented as a low-dimensional, dense vector by a combinations of latent semantic concepts. The reduced representation is often able to improve semantic similarity. As a result, text mining and NLP applications can be improved by the reduced representation.

LSA combined with TF-IDF is popular. There are three steps to obtain the low-dimensional representations for documents and terms using the LSA with TF-IDF:

- Step 1: create the TF-IDF matrix;
- Step 2: compute the SVD of the matrix;
- Step 3: derive low-dimensional representations to obtain the coordinates of documents and terms.

The obtained low-dimensional representation of documents is often used in various text mining and NLP tasks.

1.1.2 Singular value decomposition of the positive pointwise mutual information matrix

While TF-IDF is a commonly used weighting scheme to study the similarity between documents in a document-term matrix, PPMI is a commonly used weighting scheme

for a word-context matrix (Jurafsky & Martin, 2023). PPMI stands for positive PMI (pointwise mutual information). PMI is an information-theoretic association measure which measures the association between a row word and a column word (Church & Hanks, 1990; Bullinaria & Levy, 2007; Turney & Pantel, 2010; Levy & Goldberg, 2014; Levy, Goldberg, & Dagan, 2015; Jurafsky & Martin, 2023). PMI is the log of the ratio of joint proportion of the row word and column word, and the product of marginal proportion of the row word and marginal proportion of the column word. Thus the joint proportion is divided by the proportion under statistical independence of rows and columns. The log transformation ensures that PMI values can in principle range from negative to positive infinity.

In the PMI matrix, if a row word and a column word co-occur very often compared to what is expected under independence, i.e., having a genuine association, their PMI value will be positive. If a row word and a column word co-occur exactly as often as under independence, i.e., having no relationship, then their PMI value is 0. If a row word and a column word co-occur rarely, then the PMI value is negative. The raw word-context matrix contains a large number of zeros. The log of 0 is undefined and in this situation, it is customary to set the PMI value to 0.

It is worth noting that the elements in the PMI matrix are not monotonic transformations of observed counts divided by counts under independence (Levy & Goldberg, 2014). This is because word-context pairs that co-occur rarely are negative, but word-context pairs that never co-occur (i.e., the values in the raw word-context matrix being 0) are set to 0. An alternative is the PPMI matrix. In the PPMI matrix all negative values are set to 0. In most applications, one makes use of the PPMI matrix instead of the PMI matrix (Salle, Villavicencio, & Idiart, 2016). Systematic comparisons of various word-context association metrics show that PPMI provides the best results overall in semantic similarity tasks (Bullinaria & Levy, 2007).

A common approach is to factorize the PPMI matrix using SVD (Bullinaria & Levy, 2012; Levy & Goldberg, 2014; Levy et al., 2015; Jurafsky & Martin, 2023), which we call PPMI-SVD, and thus, the low-dimensional representations of row words and column words are obtained by orthogonal factors of the SVD of the PPMI matrix. There are three steps to obtain the low-dimensional representations for row words and column words using PPMI-SVD:

- Step 1: create the PPMI matrix;
- Step 2: compute the SVD of the matrix;
- Step 3: derive low-dimensional representations to obtain the coordinates of row words and column words.

The obtained low-dimensional representations of row words are often used in various text mining and NLP tasks.

1.1.3 Global vectors for word representation

LSA and PPMI-SVD are techniques that can be used to obtain word representations via SVD. The resulting vector representations have the property of orthogonality. Two other popular techniques that can be used to obtain word representations relevant for the analysis of a word-context matrix are GloVe (the abbreviation for "global vectors for word representation") and SGNS (the abbreviation for "skip-gram with negative sampling"). GloVe and SGNS involve a gradient algorithm instead of SVD. Like SVD, gradient descent is a well-known optimization algorithm in text mining and NLP. Gradient descent obtains the values of parameters by minimizing the errors of an objective function, and the parameters are obtained along with the direction of the opposite gradient (Cauchy, 1847).

PPMI-SVD via SVD provides the optimal approximation of the PPMI in a least-squares sense and the resulting low-dimensional representations have the orthogonal property. In contrast, GloVe provides low-dimensional representations for words by an adaptive gradient algorithm which minimizes a weighted least-squares function (Pennington et al., 2014). GloVe has no orthogonal constraints on low-dimensional representations of row words and column words. The vector representations of row (target) words by GloVe can be useful in various text mining and NLP tasks (Levy et al., 2015).

In the weighted objective function of GloVe (Pennington et al., 2014), the matrix entries use a logarithmic function, and the error of the objective function corresponding to an entry is weighted as a function of the matrix entry with a maximum threshold. The use of a logarithmic function on the matrix entries and a maximum threshold on the error weight reduces the effect of words with very high frequencies. In the raw word-context matrix, the frequencies vary a lot, which may cause word representations to be dominated by huge values (Aggarwal, 2018). The results from Shi and Liu (2014) and Shazeer, Doherty, Evans, and Waterson (2016) indicate that GloVe factorizes a PMI matrix shifted by a fixed constant.

1.1.4 Skip-gram with negative sampling

SGNS stands for skip-gram with negative sampling of word2vec embeddings (Mikolov, Chen, Corrado, & Dean, 2013; Mikolov, Sutskever, et al., 2013). The algorithms used in SGNS are stochastic gradient descent and backpropagation (Rumelhart, Hinton, & Williams, 1986; Rong, 2014). Levy and Goldberg (2014) showed that SGNS implicitly factorizes a PMI matrix shifted by $\log n$, where n is the number of negative samples. The vector representations of target words by SGNS are often useful in various text mining and NLP tasks.

The three techniques PPMI-SVD, GloVe, and SGNS, that are relevant for the analysis of a word-context matrix, are all related to the PMI matrix. In the Chapter 4, we show a popular statistical tool CA, which we introduce next, is also related to the PMI matrix.

1.2 Correspondence analysis

A popular statistical technique to analyze contingency tables not often used in text mining and NLP is correspondence analysis (CA). CA provides a graphical display to visualize the association between two categorical variables (Greenacre, 1984; Greenacre & Hastie, 1987; Greenacre, 2017). From the plot, we obtain an understanding of how categories from the same variable or from different variables are related to each other (Beh & Lombardo, 2021). CA has received considerable attention in a variety of areas such as ecology (Greenacre, 2013) and marketing (Pitt, Bal, & Plangger, 2020). In the applications of CA, one tends to study contingency tables of two categorical variables by a two-dimensional plot rather than by criteria, such as the performance accuracy of a text classifier, in machine learning.

CA is highly flexible. It has no requirement for the matrix except that the entries of the matrix need to be non-negative. Like LSA and PPMI-SVD, CA is a dimensionality reduction technique that uses SVD to decompose the matrix of standardized residuals, which is obtained by double centering and rescaling the initial data matrix. There are three steps to obtain the representations of categories using the CA:

- Step 1: make the matrix of standardized residuals;
- Step 2: compute the SVD of the matrix;
- Step 3: derive low-dimensional representations to obtain the coordinates of row and column categories.

The obtained first two-dimensional representations for row categories and column categories are often used to make a two-dimensional plot. In this PhD dissertation, CA is used both for the analysis of the document-term matrix and the word-context matrix. More in-depth discussions of CA are in the Chapters 2, 4, and 5.

1.3 Research question

Document-term and word-context matrices are non-negative and can therefore be analyzed using CA. CA is used in text mining and NLP (Hou & Huang, 2020; Arenas-Márquez, Martinez-Torres, & Toral, 2021), but is not as popular in this area as LSA, PPMI-SVD, GloVe, and SGNS. Unlike LSA, PPMI-SVD, GloVe, and SGNS, where the derived vector representations are building blocks for a variety of text mining and NLP tasks, CA is often used to make a two-dimensional graphical display.

This restricts the popularity of CA in text mining and NLP to some extent. This raises the main research question of this dissertation:

In text mining and NLP, how does the performance of CA compare with the performance of LSA, PPMI-SVD, GloVe, and SGNS?

We compare CA with LSA, PPMI-SVD, GloVe, and SGNS from a theoretical and an empirical point of view. In empirical comparisons, we use the criteria often employed in machine learning, such as, the performance accuracy, to compare them. Specifically, we compare CA and LSA in text categorization and authorship attribution by computing the accuracy and in information retrieval by calculating the MAP (mean average precision). We compare CA with PPMI-SVD, GloVe, and SGNS in word similarity tasks by computing Spearman's correlation coefficient.

1.4 Contribution and outline of this dissertation

This dissertation consists of four main studies. The chapters can be summarized as follows.

Chapter 2 theoretically compares CA and LSA of a document-term matrix. In addition, the performance of CA is compared to the performance of different versions of LSA in the context of text categorization and authorship attribution. The criterion used to make comparisons is mainly a measure for accuracy. From a theoretical point of view it appears that CA has more attractive properties than LSA. For example, in LSA, the effect of the margins as well as the dependence between documents and terms is part of the matrix that is analyzed, while CA eliminates the effect of the margins and thus the solution only displays the dependence. The results for four empirical datasets show that CA can obtain higher accuracies on text categorization and authorship attribution than the different versions of LSA.

Chapter 3 also studies the performance of CA and LSA in the context of document-term matrices. CA and LSA are empirically compared in information retrieval by calculating the MAP (mean average precision). An attempt is made to improve CA by applying the two kinds of weighting, that are also used in LSA. These are weighting schemes for the elements of the document-term matrix and the adjustment of the singular value weighting exponent. The results for four empirical datasets show that CA always performs better than LSA. Weighting the elements of the raw data matrix can improve CA; however, it is data dependent and the improvement is small. Adjusting the singular value weighting exponent often improves the performance of CA; however, the extent of the improvement depends on the dataset and the number of dimensions.

Chapter 4 compares CA with PPMI-SVD, GloVe, and SGNS. Theoretically, like PPMI-SVD, GloVe, and SGNS, we are able to link CA to the factorization of the PMI matrix. An attempt is made to improve CA by making use of weighting schemes for the elements of the word-context matrix. An empirical comparison on word similarity tasks shows that the overall results for CA with the two weighting schemes are slightly better than those of PPMI-SVD, GloVe, and SGNS.

It is well known that CA is susceptible to outliers (Greenacre, 2013, 2017; Choulakian, 2020). In the last chapter of this dissertation, that is, Chapter 5, the so-called reconstitution algorithm is introduced to cope with outlying cells. This al-

gorithm can reduce the contribution of the outlying cells in CA. The reconstitution algorithm is compared with two alternative methods for handling outliers, the supplementary points method and MacroPCA. It is shown that the proposed strategy works well.

1.5 Future research

In statistics, and in text mining and NLP, similar techniques with different names may have different levels of development in these areas. A technique in one area may bring a new perspective to another area. This dissertation focuses on a comparison between the popular statistical technique CA and four popular text mining and NLP techniques: LSA, PPMI-SVD, GloVe, and SGNS. Extension to a comparison with other popular statistical techniques (such as latent class analysis) and popular text mining and NLP techniques (such as non-negative matrix factorization and probabilistic LSA) would be an interesting sequel.

The vector representations of documents or words from LSA, PPMI-SVD, GloVe, and SGNS are building blocks for text mining and NLP tasks. CA has a promising performance in text mining and NLP compared with LSA, PPMI-SVD, GloVe, and SGNS. We hope that the text mining and NLP applications can benefit from the low-dimensional representations of documents or words learned by CA. Thus, the derived vector representations for documents and words by CA as building blocks in various text mining and NLP tasks are interesting future study topics.

Finally, in what follows, we propose several other future study topics.

- In order to improve CA, we try different weighting schemes for a document-term matrix and for a word-context matrix in Chapter 3 and 4, respectively. For a word-context matrix, we use square-root and root-root weighting schemes and these two weighting schemes have positive effects on the performance of CA. It is worth studying what the effects of these two weighting schemes are on the performance of CA applied to a document-term matrix.

- In Chapter 4, a square-root and a root-root transformation are applied to a word-context matrix. It is interesting to generalize this power transformation of the elements of the matrix, such as applying 2/3 to the elements.

Summarizing, we have shown that CA is a technique that matches or outperforms techniques that are now commonly used in computing science. We think that the performance of CA in the studies of this dissertation shows that CA deserves more attention in this field.

Chapter 2

A comparison of latent semantic analysis and correspondence analysis of document-term matrices

Abstract

Latent semantic analysis (LSA) and correspondence analysis (CA) are two techniques that use a singular value decomposition (SVD) for dimensionality reduction. LSA has been extensively used to obtain low-dimensional representations that capture relationships among documents and terms. In this article, we present a theoretical analysis and comparison of the two techniques in the context of document-term matrices. We show that CA has some attractive properties as compared to LSA, for instance that effects of margins, that is, sums of row elements and column elements, arising from differing document-lengths and term-frequencies are effectively eliminated, so that the CA solution is optimally suited to focus on relationships among documents and terms. A unifying framework is proposed that includes both CA and LSA as special cases. We empirically compare CA to various LSA based methods on text categorization in English and authorship attribution on historical Dutch texts, and find that CA performs significantly better. We also apply CA to a long-standing question regarding the authorship of the Dutch national anthem *Wilhelmus* and provide further support that it can be attributed to the author Datheen, among several contenders.

This chapter is published in Natural Language Engineering as: Qi, Q., Hessen, D. J., Deoskar, T., & Van der Heijden, P. G. M. (2023). A comparison of latent semantic analysis and correspondence analysis of document-term matrices. Natural Language Engineering, 1-31. DOI: 10.1017/S1351324923000244. Author contributions: PvdH provided the idea. QQ worked out the idea, set up the experiments, and carried them out. TD provided expertise in suitable evaluation methods and literature in natural language processing. QQ, DH, TD, and PvdH discussed and edited the text. The code used in this study can be found at `https://github.com/qianqianqi28/calsa-tc`.

2.1 Introduction

Latent semantic analysis (LSA) is a method used in computational linguistics that uses singular value decomposition (SVD) for dimensionality reduction in order to extract usage-based representations of words from textual corpora (Landauer & Dumais, 1997; Jiao & Zhang, 2021). We focus here on LSA of document-term matrices; the rows of the document-term matrix correspond to the documents and the columns to the terms, and the elements are frequencies, that is, the number of occurrences of each term in each document. Documents may have different lengths and margins of documents refer to the marginal frequencies of documents, namely the sum of each row of the document-term matrix; also, terms may be more or less often used and margins of terms refer to the marginal frequencies of terms, namely the sum of each column of the document-term matrix.

Among many other tasks (Di Gangi, Bosco, & Pilato, 2019; Tseng, Chen, Chang, & Sung, 2019; Phillips et al., 2021; Hassani, Iranmanesh, & Mansouri, 2021; Ren & Coutanche, 2021; Gupta & Patel, 2021; Kalmukov, 2022), LSA has been used extensively for information retrieval (W. Zhang, Yoshida, & Tang, 2011; Patil, 2022), by using associations between documents and terms (Dumais et al., 1988; Deerwester et al., 1990; Dumais, 1991). The exact factorization achieved via SVD has been shown to achieve solutions comparable in some ways to those obtained by modern neural network based techniques (Levy & Goldberg, 2014; Levy et al., 2015), commonly used to obtain dense word representations from textual corpora (Jurafsky & Martin, 2023).

Correspondence analysis (CA) is a popular method for the analysis of contingency tables (Greenacre, 1984, 2017; Hou & Huang, 2020; Van Dam et al., 2021). It provides a graphical display of dependence between rows and columns of a two-way contingency table (Greenacre & Hastie, 1987). Like LSA, CA is a dimensionality reduction method. The methods have much in common as both use SVD. In both cases, after dimensionality reduction, many text mining tasks, such as text clustering, may be performed in the reduced dimensional space rather than in the higher dimensional space provided by the raw document-term matrix.

While a few empirical comparisons of LSA and CA, with mixed results, can be found in the literature, a comprehensive theoretical comparison is lacking. For example, Morin (1999) compared the two methods in the automatic exploration of themes in texts. Séguéla and Saporta (2011) compared the performance of CA and LSA with several weighting functions in a document clustering task, and found that CA gave better results. On the other hand, Séguéla and Saporta (2013) compared the performance of CA and LSA with TF-IDF on a recommender system, but found that CA performs less well.

The present article presents a theoretical comparison of the two techniques, and places them in a unifying framework. We show that CA has some favorable properties over LSA, such as a clear interpretation of the distances between documents and between terms of the original matrix, and a clear relation to statistical independence of documents and terms. Also, CA can eliminate the margins of documents and terms

simultaneously. Second, we empirically evaluate and compare the two techniques, by applying them to text categorization and authorship attribution in two languages. For text categorization, we use the BBCNews, BBCSport, and 20 Newsgroups datasets in English. In authorship attribution, we evaluate the two techniques on a large set of historical Dutch texts written by six well-known Dutch authors of the sixteenth century. Here, we additionally use CA to determine the unknown authorship of *Wilhelmus*, the national anthem of the Netherlands, whose authorship is controversial: CA attributes *Wilhelmus* to the author Datheen, out of the six contemporary contenders. To the best of our knowledge, this is the first application of CA to the *Wilhelmus*. In both cases, we find that CA performs better.

The rest of the article is organized as follows. Section 2.2 and Section 2.3 elaborate on the techniques LSA and CA in turn. A unifying framework is proposed in Section 2.4. In Section 2.5, we compare LSA and CA in text categorization using the BBCNews, BBCSport, and 20 Newsgroups datasets. Section 2.6 evaluates the performance of LSA and CA for authorship attribution of documents where the author is known, and uses CA to study the authorship of the *Wilhelmus*, whose author is unknown. The article ends with a conclusion.

2.2 Latent semantic analysis

LSA has been extensively used for improving information retrieval by using the associations between documents and terms (Dumais et al., 1988; Deerwester et al., 1990), among many other tasks. Since individual terms provide incomplete and unreliable evidence about the meaning of a document, in part due to synonymy and polysemy, individual terms are replaced with derived underlying (latent) semantic factors. Although LSA is a very well-known technique, we first present a detailed analysis of the mathematics involved in LSA here as this is usually not found in the literature, and in a later section, it will help in making the comparison between LSA and CA explicit. We start with LSA of the raw document-term matrix and then discuss LSA of weighted matrices. The weighted matrices we study here include (i) a matrix with row-normalized elements with L1, that is, for each row the elements are divided by the row sum (the L1 norm), so that the sum of the elements of each row is 1; (ii) a matrix with row-normalized elements with L2, that is, for each row the elements are divided by the square root of sum of squares of these elements (the L2 norm), so that the sum of squares of the elements of each row is 1; and (iii) a matrix that is transformed by term frequency-inverse document frequency (TF-IDF).

The discussion is illustrated using a toy dataset, with the aim to present a clear view of the properties of the dataset captured by LSA and CA; see Table 2.1. The toy dataset has 6 rows, the documents, and 6 columns, the terms, with the frequency of occurrence of terms in each document in the cells (Aggarwal, 2018). Based on term-frequencies in each document, the first three documents can be considered to primarily refer to *cats*, the last two primarily to *cars*, and the fourth document to both.

The fourth term, *jaguar*, is polysemous because it can refer to either a cat or a car. We will see below how the LSA approaches, and later CA, represent these properties in the data.

Table 2.1: A document-term matrix $\boldsymbol{F}$: size 6×6

	lion	tiger	cheetah	jaguar	porsche	ferrari
doc1	2	2	1	2	0	0
doc2	2	3	3	3	0	0
doc3	1	1	1	1	0	0
doc4	2	2	2	3	1	1
doc5	0	0	0	1	1	1
doc6	0	0	0	2	1	2

2.2.1 LSA of raw document-term matrix

LSA is an application of the mathematical tool SVD, and can take many forms, depending on the matrix analyzed. We start our discussion of LSA with the SVD of a raw document-term matrix $\boldsymbol{F}$, having size $m \times n$, with elements f_{ij}, $i = 1,...,m$ and $j = 1,...,n$ (Berry, Dumais, & O'Brien, 1995; Deisenroth, Faisal, & Ong, 2020). Without loss of generality we assume that $n \geq m$ and $\boldsymbol{F}$ has full rank.

SVD can be used to decompose $\boldsymbol{F}$ into a product of three matrices: $\boldsymbol{U}^f$, Σ^f, and $\boldsymbol{V}^f$, namely

$$\boldsymbol{F} = \boldsymbol{U}^f \Sigma^f (\boldsymbol{V}^f)^T \tag{2.1}$$

Here $\boldsymbol{U}^f$ is a $m \times m$ matrix with orthonormal columns called left singular vectors so that $(\boldsymbol{U}^f)^T \boldsymbol{U}^f = \boldsymbol{I}$, $\boldsymbol{V}^f$ is a $n \times m$ matrix with orthonormal columns called right singular vectors so that $(\boldsymbol{V}^f)^T \boldsymbol{V}^f = \boldsymbol{I}$, and Σ^f is a $m \times m$ diagonal matrix with singular values on the diagonal in descending order.

We denote the first k columns of $\boldsymbol{U}^f$ as the $m \times k$ matrix $\boldsymbol{U}^f_k$, the first k columns of $\boldsymbol{V}^f$ as the $n \times k$ matrix $\boldsymbol{V}^f_k$, and the k largest singular values on the diagonal of Σ^f as the $k \times k$ matrix Σ^f_k ($k \leq m$). Then $\boldsymbol{U}^f_k \Sigma^f_k (\boldsymbol{V}^f_k)^T$ provides the optimal rank-k approximation of $\boldsymbol{F}$ in a least-squares sense. That is, $\boldsymbol{X} = \boldsymbol{U}^f_k \Sigma^f_k (\boldsymbol{V}^f_k)^T$ minimizes Equation (2.2) among all matrices $\boldsymbol{X}$ of rank k:

$$\|\boldsymbol{F} - \boldsymbol{X}\|^2_F = \sum_i \sum_j (f_{ij} - x_{ij})^2 \tag{2.2}$$

The idea is that the matrix $\boldsymbol{U}^f_k \Sigma^f_k (\boldsymbol{V}^f_k)^T$ captures the major associational structure in the matrix and throws out noise (Dumais et al., 1988; Dumais, 1991). The total sum of squared singular values is equal to $\text{tr}((\Sigma^f)^2)$, where tr is the sum of elements on the main diagonal of a square matrix. The proportion of the total sum of squared singular values explained by the rank k approximation is $\text{tr}((\Sigma^f_k)^2)/\text{tr}((\Sigma^f)^2)$.

SVD can also be interpreted geometrically. As $\boldsymbol{F}$ is of size $m \times n$, each row of $\boldsymbol{F}$ can be represented as a point in an n-dimensional space with the row elements as coordinates, and each column can be represented as a point in an m-dimensional space with the column elements as coordinates. In a rank-k approximation, where $k < (m, n)$, each of the original m documents and n terms is approximated by only k coordinates. Thus SVD projects the sum of squared Euclidean distances from these row (column) points to the origin in the n (m)-dimensional space as much as possible to a lower, a k-dimensional space. The Euclidean distances between the rows of $\boldsymbol{F}$ are approximated by the Euclidean distances between the rows of $\boldsymbol{U}_k^f \Sigma_k^f$ from below, and the Euclidean distances between the rows of $\boldsymbol{F}^T$ are approximated by the Euclidean distances between the rows of $\boldsymbol{V}_k^f \Sigma_k^f$ from below.

The choice of k is crucial in many applications (Albright, 2004). A lower rank approximation cannot always express prominent relationships in text, whereas the higher rank approximation may add useless noise. How to choose k is an open issue (Deerwester et al., 1990). In practice, the value of k is selected such that a certain criterion is satisfied, for example, the proportion of explained total sum of squared singular values is at least a pre-specified proportion. Also, the use of a scree plot, showing the decline in subsequent squared singular values, can be considered.

As $\boldsymbol{F}$ is a non-negative matrix, the first column vectors in $\boldsymbol{U}$ and $\boldsymbol{V}$ have the special property that the elements of the vectors depart in the same direction from the origin (Perron, 1907; Frobenius, 1912; Hu et al., 2003). We give an intuitive geometric explanation for the m rows of $\boldsymbol{F}$. Each row is a vector in the non-negative n-dimensional subspace of R^n. As a result, the first singular vector, being in the middle of the m vectors, is also in the non-negative n-dimensional subspace of R^n. As each vector is the non-negative subspace, the angle between each vector with the first singular vector is between 0 and 90 degrees, and therefore the projection of each of the m vectors on the first singular vector, corresponding to the elements of $\boldsymbol{U}_1 \Sigma_1$, is non-negative (or each is non-positive, as we will discuss now). The same holds for the columns of $\boldsymbol{F}$ and the first singular vector $\boldsymbol{V}_1$. The reason that the elements of $\boldsymbol{U}_1$ and $\boldsymbol{V}_1$ are all either non-negative or non-positive is that $\boldsymbol{U}_1^f \Sigma_1^f (\boldsymbol{V}_1^f)^T = -\boldsymbol{U}_1^f \Sigma_1^f (-\boldsymbol{V}_1^f)^T$, as the singular values are defined to be non-negative. As the lengths of the row vectors in n-dimensional space to the origin are influenced by the sizes of the documents (i.e. the marginal frequencies), larger documents have larger projections on the first singular vector, and the first dimension mainly displays differences in the sizes of the margins.

As it turns out, the raw document-term matrix $\boldsymbol{F}$ in Table 2.1 does not have full

rank; its rank is 5. The SVD of $\boldsymbol{F}$ in Table 2.1 is

$$\begin{aligned}\boldsymbol{F} &= \boldsymbol{U}^f \Sigma^f (\boldsymbol{V}^f)^T \\ &= \begin{bmatrix} -0.411 & 0.175 & 0.825 & 0.252 & -0.239 \\ -0.646 & 0.314 & -0.562 & 0.301 & -0.279 \\ -0.232 & 0.127 & 0.034 & -0.099 & 0.503 \\ -0.562 & -0.203 & 0.044 & -0.603 & 0.333 \\ -0.099 & -0.456 & -0.024 & -0.404 & -0.672 \\ -0.186 & -0.778 & -0.034 & 0.556 & 0.223 \end{bmatrix} \begin{bmatrix} 8.425 & 0 & 0 & 0 & 0 \\ 0 & 3.261 & 0 & 0 & 0 \\ 0 & 0 & 0.988 & 0 & 0 \\ 0 & 0 & 0 & 0.574 & 0 \\ 0 & 0 & 0 & 0 & 0.272 \end{bmatrix} \\ &\quad \begin{bmatrix} -0.412 & 0.214 & 0.655 & -0.344 & 0.486 \\ -0.488 & 0.311 & 0.087 & 0.180 & -0.540 \\ -0.440 & 0.257 & -0.748 & -0.259 & 0.339 \\ -0.611 & -0.369 & 0.039 & 0.366 & -0.148 \\ -0.101 & -0.441 & -0.014 & -0.783 & -0.426 \\ -0.123 & -0.679 & -0.048 & 0.186 & 0.392 \end{bmatrix}^T \end{aligned} \tag{2.3}$$

For the raw matrix, LSA-RAW in Table 2.2 shows the singular values, the squares of the singular values, and the proportions of explained total sum of squared singular values (denoted as PSSSV). Together, the first two dimensions account for 0.855 + 0.128 = 0.983 of the total sum of squared singular values. Therefore, the documents and the terms can be approximated adequately in a two-dimensional representation using $\boldsymbol{U}_2^f \Sigma_2^f$ and $\boldsymbol{V}_2^f \Sigma_2^f$ as coordinates. As the Euclidean distances between the documents and between the terms in the two-dimensional representation, i.e., between the rows of $\boldsymbol{U}_2^f \Sigma_2^f$ and the rows of $\boldsymbol{V}_2^f \Sigma_2^f$, approximate the Euclidean distances between rows and between columns of the original matrix $\boldsymbol{F}$, such a two-dimensional representation simplifies the interpretation of the matrix considerably.

On the other hand, it is somewhat more difficult to examine the relation between a document and a term. The reason is that, by choosing a Euclidean distance-representation both for the documents and for terms, the singular values are used *twice* in the coordinates $\boldsymbol{U}_2^f \Sigma_2^f$ and $\boldsymbol{V}_2^f \Sigma_2^f$, and the inner product of coordinates of a document and coordinates of a term does not approximate the corresponding value in $\boldsymbol{F}$. Directions from the origin can be interpreted, though, as the double use of the singular values only leads to relatively reduced coordinates on the second dimension in comparison to the coordinates on the first dimension.

The two-dimensional representation of LSA-RAW is shown in Figure 2.1a. In Figure 2.1a Euclidean distances between documents, and between terms, reveal the similarity of documents, and terms, respectively. For example, documents 5 and 6 are close, and similar in the sense that their Euclidean distance is small. For these two documents the Euclidean distance in the matrix $\boldsymbol{F}$ is 1.414, and in the first two dimensions it is 1.279, so the first two dimensions provide an adequate representation of their similarity. The value 1.279 is much smaller than the Euclidean distances between Documents 5 and 1 (3.338), 5 and 2 (5.248), 5 and 3 (2.205), 5 and 4 (3.988) as

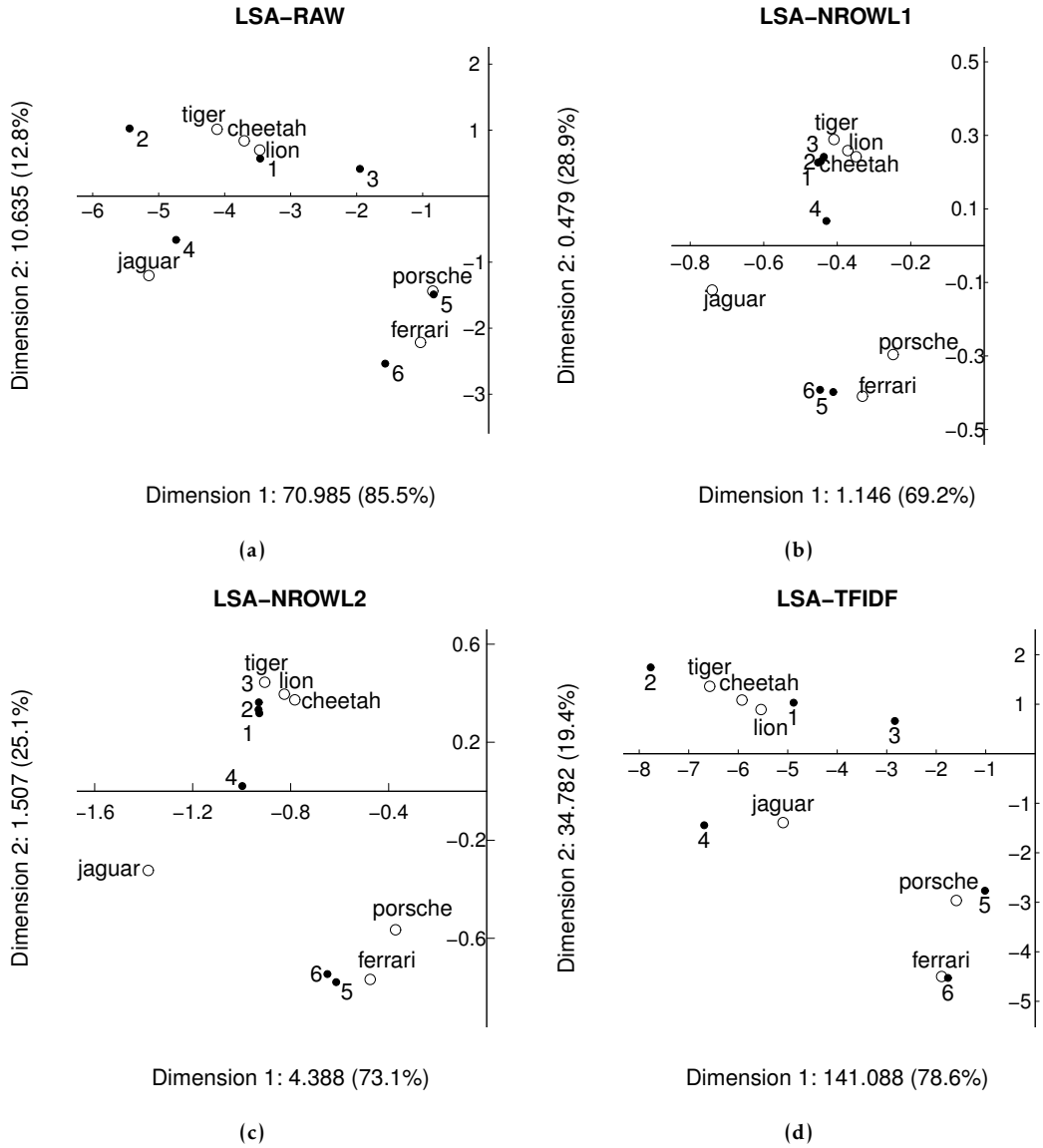


Figure 2.1: A two-dimensional plot of documents and terms (a) for raw matrix $\boldsymbol{F}$; (b) for row-normalized data $\boldsymbol{F}^{L1}$; (c) for row-normalized data $\boldsymbol{F}^{L2}$; (d) for matrix $\boldsymbol{F}^{\text{TF-IDF}}$.

Table 2.2: The singular values, the squares of singular values, and the proportion of explained total sum of squared singular values (PSSSV) for each dimension of LSA of $\boldsymbol{F}$, of $\boldsymbol{F}^{L1}$, of $\boldsymbol{F}^{L2}$, and of $\boldsymbol{F}^{\text{TF-IDF}}$.

methods	items	dim1	dim2	dim3	dim4	dim5
LSA-RAW	singular value	8.425	3.261	0.988	0.574	0.272
	square of singular value	70.985	10.635	0.976	0.330	0.074
	PSSSV	0.855	0.128	0.012	0.004	0.001
LSA-NROWL1	singular value	1.070	0.692	0.123	0.114	0.046
	square of singular value	1.146	0.479	0.015	0.013	0.002
	PSSSV	0.692	0.289	0.009	0.008	0.001
LSA-NROWL2	singular value	2.095	1.228	0.239	0.198	0.092
	square of singular value	4.388	1.507	0.057	0.039	0.009
	PSSSV	0.731	0.251	0.009	0.007	0.001
LSA-TFIDF	singular value	11.878	5.898	1.565	1.017	0.449
	square of singular value	141.088	34.782	2.451	1.034	0.202
	PSSSV	0.786	0.194	0.014	0.006	0.001

well as the Euclidean distances between Documents 6 and 1 (3.638), 6 and 2 (5.262), 6 and 3 (2.975), 6 and 4 (3.681). On the first dimension all documents and terms have a negative coordinate (see above). There is an order of 5, 6, 3, 1, 4, and 2 on the first dimension. This order is related to the row margins of Table 2.1, where 2 and 4 have the highest frequencies and therefore are further away from the origin. Overall, the two-dimensional representation of the documents reveals a mix of the sizes of the documents, the row margins $\Sigma_j f_{ij}$, and the relative use of the terms by the documents, i.e., for row i this is the vector of elements $f_{ij}/\Sigma_j f_{ij}$, also known as the *row profile* for row i. This mix makes the graphic representation difficult to interpret. Similarly, *porsche* and *ferrari* are lower left but close to the origin, *tiger*, *cheetah*, and *lion* are upper left and further away from the origin, and *jaguar* is far away at the lower left. Also there is a mix of the sizes of the terms, i.e., for column j this is column margin $\Sigma_i f_{ij}$, and the relative use of the documents by the terms, i.e., for column j this is the vector of elements $f_{ij}/\Sigma_i f_{ij}$, also known as the *column profile* for column j. The terms *porsche* and *ferrari* are related to documents 5 and 6 as they have the same position w.r.t. the origin, and similarly for *tiger*, *cheetah*, and *lion* to documents 1, 2, and 3, and *jaguar* to document 4.

Although the first dimension accounts for 85.5 per cent of the total sum of squared singular values, it provides little information about the relations among documents and terms. In particular, from Table 2.1 we expect that documents 1 to 3 are similar, documents 5 and 6 are similar, and document 4 is in-between; term *jaguar* is between cat terms (*tiger*, *cheetah*, and *lion*) and car terms (*porsche* and *ferrari*), but we cannot see that from the first dimension. This is because the margins of Table 2.1 play a dominant role in the first dimension.

2.2.2 LSA of weighted document-term matrix

Weighting can be used to prevent differential lengths of documents from having differential effects on the representation, or be used to impose certain preconceptions of which terms are more important (Deerwester et al., 1990). The frequencies f_{ij} in the raw document-term matrix $\boldsymbol{F}$ can be transformed with the aim to provide a better approximation of the interrelations between documents and terms (Nakov, Popova, & Mateev, 2001). The weight w_{ij} for term j in document i is normally expressed as a product of three components (Salton & Buckley, 1988; Kolda & O'leary, 1998; Ab Samat, Murad, Abdullah, & Atan, 2008)

$$w_{ij} = L(i,j) \times G(j) \times N(i) \tag{2.4}$$

where the local weighting $L(i,j)$ is the weight of term j in document i, the global weighting $G(j)$ is the weight of the term j in the entire document set, and $N(i)$ is the normalization component for document i.

When $L(i,j) = f(i,j)$, $G(j) = 1$, and $N(i) = 1$, the weighted $\boldsymbol{F}$ is equal to $\boldsymbol{F}$. In matrix notation, Equation (2.4) can be expressed as $\boldsymbol{W} = \boldsymbol{NLG}$, where $\boldsymbol{N}$ is a diagonal matrix with diagonal elements $N(i)$ and $\boldsymbol{G}$ is a diagonal matrix with diagonal elements $G(j)$. Notice that pre- or post-multiplying by a diagonal matrix leaves the rank of the matrix $\boldsymbol{L}$ intact.

We examine two common ways to weight f_{ij}. One is row normalization (Salton & Buckley, 1988; Ab Samat et al., 2008) with L1 and L2. The other is TF-IDF (Dumais, 1991).

2.2.2.1 SVD of matrix with row-normalized elements with L1

In row-normalized weighting with L1, we use Equation (2.4) with $L(i,j) = f_{ij}$, $G(j) = 1$, and $N(i) = 1/\sum_{j=1}^{n} f_{ij}$, and apply an SVD to this transformed matrix that we denote as $\boldsymbol{F}^{L1}$, which consists of the row profiles of $\boldsymbol{F}$. See Table 2.3. The last row, the average row profile, is the row profile of the column margins of Table 2.1.

Table 2.3: Row profiles of $\boldsymbol{F}$

	lion	tiger	cheetah	jaguar	porsche	ferrari	total
doc1	0.286	0.286	0.143	0.286	0.000	0.000	1.000
doc2	0.182	0.273	0.273	0.273	0.000	0.000	1.000
doc3	0.250	0.250	0.250	0.250	0.000	0.000	1.000
doc4	0.182	0.182	0.182	0.273	0.091	0.091	1.000
doc5	0.000	0.000	0.000	0.333	0.333	0.333	1.000
doc6	0.000	0.000	0.000	0.400	0.200	0.400	1.000
average row profile	0.171	0.195	0.171	0.293	0.073	0.098	1.000

We perform LSA of $\boldsymbol{F}^{L1}$ and find Table 2.2, part LSA-NROWL1. This shows that a rank 2 matrix approximates the data well as 0.692 + 0.289 = 0.981 of the total sum of

squared singular values is explained by these two dimensions. The first two columns of LSA of $\boldsymbol{F}^{L1}$ can be used to approximate $\boldsymbol{F}^{L1}$, see Equation (2.5).

$$\begin{aligned}\boldsymbol{F}^{L1} &\approx \boldsymbol{U}_2^{L1}\Sigma_2^{L1}(\boldsymbol{V}_2^{L1})^T \\ &= \begin{bmatrix} -0.423 & 0.327 \\ -0.415 & 0.332 \\ -0.408 & 0.349 \\ -0.401 & 0.097 \\ -0.384 & -0.575 \\ -0.417 & -0.567 \end{bmatrix} \begin{bmatrix} 1.070 & 0 \\ 0 & 0.692 \end{bmatrix} \begin{bmatrix} -0.347 & 0.374 \\ -0.382 & 0.417 \\ -0.326 & 0.350 \\ -0.692 & -0.174 \\ -0.232 & -0.428 \\ -0.310 & -0.592 \end{bmatrix}^T \end{aligned} \tag{2.5}$$

Documents and terms can be projected on a two dimensional space using $\boldsymbol{U}_2^{L1}\Sigma_2^{L1}$ and $\boldsymbol{V}_2^{L1}\Sigma_2^{L1}$ as coordinates, see Figure 2.1b. In this representation documents 1, 2, and 3 are quite close, and so are 5 and 6. Also, the terms *ferrari* and *porsche* are close and related to 5 and 6, *tiger*, *lion*, and *cheetah* are close and related to 1, 2, and 3.

Although the first dimension accounts for 69.2 per cent of the total sum of squared singular values, this dimension does not provide information about different use of terms by the documents as all documents have a similar coordinate. This is caused by the same marginal value 1 for each of the documents in $\boldsymbol{F}^{L1}$, which leads to almost the same distance from the origin. Also, we would expect *jaguar* to be in between cat terms (*tiger*, *cheetah*, and *lion*) and car terms (*porsche* and *ferrari*), but on the first dimension it appears as a separate, third group. This is caused by the high values in its column in $\boldsymbol{F}^{L1}$, which lead to a larger distance from the origin.

2.2.2.2 SVD of matrix with row-normalized elements with L2

In row-normalized weighting with L2, we use Equation (2.4) with $L(i,j) = f_{ij}$, $G(j) = 1$, and $N(i) = 1/\sqrt{\sum_{j=1}^{n} f_{ij}^2}$. The transformed matrix, denoted as $\boldsymbol{F}^{L2}$, is shown in Table 2.4. We then perform LSA on Table 2.4. Table 2.2, part LSA-NROWL2, indicates that a rank 2 matrix approximates the data well, as the sum of the PSSSV of the first two dimensions 0.731 + 0.251 = 0.982 contributes to 98.2 per cent of the total sum of squared singular values. The first two columns of LSA of $\boldsymbol{F}^{L2}$ can be used to approximate $\boldsymbol{F}^{L2}$, see Equation (2.6).

$$\begin{aligned}\boldsymbol{F}^{L2} &\approx \boldsymbol{U}_2^{L2}\Sigma_2^{L2}(\boldsymbol{V}_2^{L2})^T \\ &= \begin{bmatrix} -0.443 & 0.259 \\ -0.445 & 0.271 \\ -0.444 & 0.295 \\ -0.476 & 0.017 \\ -0.293 & -0.635 \\ -0.310 & -0.608 \end{bmatrix} \begin{bmatrix} 2.095 & 0 \\ 0 & 1.228 \end{bmatrix} \begin{bmatrix} -0.394 & 0.323 \\ -0.432 & 0.362 \\ -0.374 & 0.304 \\ -0.659 & -0.263 \\ -0.178 & -0.460 \\ -0.227 & -0.625 \end{bmatrix}^T \end{aligned} \tag{2.6}$$

Table 2.4: A row-normalized document-term matrix $\boldsymbol{F}^{L2}$

	lion	tiger	cheetah	jaguar	porsche	ferrari
doc1	0.555	0.555	0.277	0.555	0.000	0.000
doc2	0.359	0.539	0.539	0.539	0.000	0.000
doc3	0.500	0.500	0.500	0.500	0.000	0.000
doc4	0.417	0.417	0.417	0.626	0.209	0.209
doc5	0.000	0.000	0.000	0.577	0.577	0.577
doc6	0.000	0.000	0.000	0.667	0.333	0.667

Documents and terms can be projected on a two dimensional space using $\boldsymbol{U}_2^{L2}\boldsymbol{\Sigma}_2^{L2}$ and $\boldsymbol{V}_2^{L2}\boldsymbol{\Sigma}_2^{L2}$ as coordinates, see Figure 2.1c. In this representation documents 1, 2, and 3 are quite close, and so are 5 and 6. Also, the terms *ferrari* and *porsche* are close and related to 5 and 6, *tiger*, *lion*, and *cheetah* are close and related to 1, 2, and 3.

Although the first dimension accounts for 73.1 per cent of the total sum of squared singular values, and so, a major portion of the information in the matrix, we do not find the important aspect in the data that document 4 should be in between documents 1-3 on the one hand and documents 5-6 on the other hand on this dimension. This is caused by the high values in the row for doc4 in Table 2.4, which lead to a larger distance from the origin than the other documents have. Also, we would expect *jaguar* to be in between cat terms (*tiger*, *cheetah*, and *lion*) and car terms (*porsche* and *ferrari*), but on the first dimension it appears as a separate, third group. This is caused by the high values in its column in Table 2.4, which lead to a larger distance from the origin.

2.2.2.3 SVD of the term frequency-inverse document frequency matrix

TF-IDF is one commonly used transformation of text data. We use Equation (2.4) with $L(i,j) = f_{ij}$, $G(j) = 1 + \log(\frac{n\text{docs}}{df_j})$, and $N(i) = 1$, one form of TF-IDF, where $n$docs is the number of documents in the set and df_j is the number of documents where term j appears, and then apply an SVD to this transformed matrix that we denote as $\boldsymbol{F}^{\text{TF-IDF}}$, see Table 2.5. As is common in the literature, here we choose 2 as the base of the logarithmic function.

Table 2.5: A document-term matrix $\boldsymbol{F}^{\text{TF-IDF}}$

	lion	tiger	cheetah	jaguar	porsche	ferrari
doc1	3.170	3.170	1.585	2	0	0
doc2	3.170	4.755	4.755	3	0	0
doc3	1.585	1.585	1.585	1	0	0
doc4	3.170	3.170	3.170	3	2	2
doc5	0.000	0.000	0.000	1	2	2
doc6	0.000	0.000	0.000	2	2	4

We perform LSA of Table 2.5 and find Table 2.2, part LSA-TFIDF. This shows that a rank 2 matrix approximates the data well as 0.786 + 0.194 = 0.980 of the total sum of squared singular values is explained by these two dimensions. The matrix $\boldsymbol{F}^{\text{TF-IDF}}$ in Table 2.5 is approximated in the first two dimensions as follows:

$$\begin{aligned}\boldsymbol{F}^{\text{TF-IDF}} &\approx \boldsymbol{U}_2^{\text{TF-IDF}}\boldsymbol{\Sigma}_2^{\text{TF-IDF}}(\boldsymbol{V}_2^{\text{TF-IDF}})^T \\ &= \begin{bmatrix} -0.411 & 0.175 \\ -0.654 & 0.296 \\ -0.239 & 0.112 \\ -0.563 & -0.245 \\ -0.086 & -0.469 \\ -0.148 & -0.768 \end{bmatrix} \begin{bmatrix} 11.878 & 0 \\ 0 & 5.898 \end{bmatrix} \begin{bmatrix} -0.466 & 0.151 \\ -0.554 & 0.231 \\ -0.499 & 0.184 \\ -0.429 & -0.236 \\ -0.134 & -0.502 \\ -0.159 & -0.763 \end{bmatrix}^T \end{aligned} \tag{2.7}$$

Figure 2.1d is a two-dimensional plot of the documents and terms using $\boldsymbol{U}_2^{\text{TF-IDF}}\boldsymbol{\Sigma}_2^{\text{TF-IDF}}$ and $\boldsymbol{V}_2^{\text{TF-IDF}}\boldsymbol{\Sigma}_2^{\text{TF-IDF}}$ as coordinates for the 6×6 sample document-term matrix $\boldsymbol{F}^{\text{TF-IDF}}$. The configuration of documents in Figure 2.1d is very similar to that in Figure 2.1a. The configuration of terms in Figure 2.1d is different from that of terms in Figure 2.1a. In Figure 2.1d, there is an order of *porsche*, *ferrari*, *jaguar*, *lion*, *cheetah*, and *tiger* on the first dimension, whereas in Figure 2.1a, there is an order of *porsche*, *ferrari*, *lion*, *cheetah*, *tiger*, and *jaguar* on the first dimension. Compared with Figure 2.1a, the first dimension of Figure 2.1d shows that *jaguar* is in between cat terms (*tiger*, *cheetah*, and *lion*) and car terms (*porsche* and *ferrari*).

2.2.2.4 Out-of-sample documents

Representing out-of-sample documents in the k-dimensional subspace of LSA is important for many applications. Suppose an out-of-sample document $\boldsymbol{d}$ is a row vector. To represent $\boldsymbol{d}$ in lower dimensional space, first the out-of-sample document $\boldsymbol{d}$ can be transformed in the same way as the original documents (Dumais, 1991). Transformations for the above four applications of LSA are $\boldsymbol{d}_w^f = \boldsymbol{d}$, $\boldsymbol{d}_w^{L1} = \boldsymbol{d}/\sum_{j=1}^n d_j$, $\boldsymbol{d}_w^{L2} = \boldsymbol{d}/\sqrt{\sum_{j=1}^n d_j^2}$, and $\boldsymbol{d}_w^{\text{TF-IDF}} = [d_1 G(1), \cdots, d_n G(n)]$. The coordinates of the out-of-sample document $\boldsymbol{d}$ in LSA-RAW, LSA-NROWL1, LSA-NROWL2, and LSA-TFIDF are then calculated by $\boldsymbol{d}_w^f \boldsymbol{V}^f$, $\boldsymbol{d}_w^{L1}\boldsymbol{V}^{L1}$, $\boldsymbol{d}_w^{L2}\boldsymbol{V}^{L2}$, and $\boldsymbol{d}_w^{\text{TF-IDF}}\boldsymbol{V}^{\text{TF-IDF}}$, respectively (Aggarwal, 2018).

2.2.3 Conclusions regarding LSA of different matrices

In the raw document-term matrix the relationships among the documents and terms is blurred by differences in margins arising from differing document-lengths and marginal term-frequencies. Thus LSA of the raw matrix leads to a mix of margins, and relationships among documents and terms. In order to provide a better approximation of the interrelations between documents and terms, weighting schemes were used.

Normalizations of the documents have a beneficial effect. Yet, the properties of the frequencies that are evident from Table 2.1 where we expect, for example, that *jaguar* lies in between *porsche* and *ferrari* on the one hand and *tiger*, *cheetah*, and *lion* on the other hand, are not fully represented on the first dimension. This is due to the fact that the column margins of Tables 2.3 and 2.4 still play a role on the first dimension. The TF-IDF transformation also has a positive effect. Yet LSA is not successful. For example, we expect that documents 1 to 3 are similar, 5 and 6 are similar, and document 4 is in-between, but this order is not found in the first dimension. This is due to the fact that the row margins of Table 2.5 still play a role on the first dimension.

Generally, solutions of LSA have the drawback that they include the effect of the margins as well as the dependence. In the first dimension these margins play a dominant role as all points depart in the same direction from the origin. We can try to repair this property of LSA, by applying transformations of the rows and columns of Table 2.1 simultaneously. However, the transformations appear ad hoc. Instead we present in the next section a different technique, which better fits the properties of the data: CA.

2.3 Correspondence analysis

CA provides a low-dimensional representation of the interaction or dependence between the rows and columns of the contingency table (Greenacre & Hastie, 1987), which can be used to reveal the structure in the data (Hayashi, 1992). CA has been proposed multiple times, apparently independently, emphasizing different properties of the technique (Gifi, 1990). Some important contributions are provided in the Japanese literature, by Hayashi (1956, 1992), who emphasizes the property of CA that it maximizes the correlation coefficient between the row and column variable by assigning numerical scores to these variables; in the French literature, by Benzécri (1973), who emphasizes a distance interpretation, where Greenacre (1984) expressed Benzécri's work in a more convenient mathematical notation; and in the Dutch literature, by Gifi (1990) and Michailidis and De Leeuw (1998), who emphasize optimal scaling properties. We present CA here mainly from the French perspective.

The aim of CA as developed by Benzécri is to find a representation of the rows (columns) of frequency matrix $\boldsymbol{F}$ in such a way that Euclidean distances between the rows (columns) in the representation correspond to so-called χ^2-distances between rows (columns) of $\boldsymbol{F}$ (Gifi, 1990). We work with $\boldsymbol{P}$ with elements $p_{ij} = f_{ij}/f_{++}$, where f_{++} is the sum of all elements of $\boldsymbol{F}$. In the χ^2-distance profiles play an important role. The squared χ^2-distance between the kth row profile with elements p_{kj}/r_k and the lth row profile with elements p_{lj}/r_l is

$$\delta_{kl}^2 = \sum_j \frac{\left(p_{kj}/r_k - p_{lj}/r_l\right)^2}{c_j} \tag{2.8}$$

where r_i (also called the average column profile) and c_j (the average row profile) are the row and column sums of $\boldsymbol{P}$ respectively. Thus the difference between the jth elements of the two profiles is weighted by column margin (i.e. the last row of Table 2.3), c_j, so that this difference plays a relatively more important role in the χ^2-distance if it stems from a column having a small value c_j.

A representation where Euclidean distances between the rows of the matrix are equal to χ^2-distances is found as follows. In matrix notation, the matrix whose Euclidean distances between the rows are equal to χ^2-distances between rows of $\boldsymbol{F}$ is equal to $\boldsymbol{D}_r^{-1}\boldsymbol{P}\boldsymbol{D}_c^{-\frac{1}{2}}$, where $\boldsymbol{D}_r$ is a diagonal matrix with r_i as diagonal elements and $\boldsymbol{D}_c$ is a diagonal matrix with c_j as diagonal elements. Suppose we take the SVD of

$$\boldsymbol{D}_r^{-\frac{1}{2}}\boldsymbol{P}\boldsymbol{D}_c^{-\frac{1}{2}} = \boldsymbol{U}^{sp}\boldsymbol{\Sigma}^{sp}(\boldsymbol{V}^{sp})^T \tag{2.9}$$

Here $\boldsymbol{D}_r^{-\frac{1}{2}}\boldsymbol{P}\boldsymbol{D}_c^{-\frac{1}{2}}$ is a matrix with standardized proportions, hence the superscripts sp on the right hand side of the equation. Then, if we pre-multiply both sides of Equation (2.9) with $\boldsymbol{D}_r^{-\frac{1}{2}}$, we get

$$\boldsymbol{D}_r^{-1}\boldsymbol{P}\boldsymbol{D}_c^{-\frac{1}{2}} = \boldsymbol{D}_r^{-\frac{1}{2}}\boldsymbol{U}^{sp}\boldsymbol{\Sigma}^{sp}(\boldsymbol{V}^{sp})^T \tag{2.10}$$

Thus a representation using the rows of $\boldsymbol{D}_r^{-\frac{1}{2}}\boldsymbol{U}^{sp}\boldsymbol{\Sigma}^{sp}$ as row coordinates leads to Euclidean distances between these row points being equal to χ^2-distances between rows of $\boldsymbol{F}$. Similar to Equation (2.8) we can also define χ^2-distances between the columns of $\boldsymbol{F}$, and in matrix notation this leads to the matrix $\boldsymbol{D}_r^{-\frac{1}{2}}\boldsymbol{P}\boldsymbol{D}_c^{-1}$. Then, in a similar way as for the χ^2-distances for the rows, Equation (2.9) can be used as an intermediate step to go to a solution for the columns. Post-multiplying the left and right hand sides in Equation (2.9) by $\boldsymbol{D}_c^{-\frac{1}{2}}$ provides us with the coordinates for a representation where Euclidean distances between the column points (the rows of $\boldsymbol{D}_c^{-\frac{1}{2}}\boldsymbol{V}^{sp}\boldsymbol{\Sigma}^{sp}$ as coordinates for these columns) are equal to χ^2-distances between the columns of $\boldsymbol{F}$. Notice that Equation (2.9) plays the dual role of an intermediate step in going to a solution both for the rows and the columns.

The matrices $\boldsymbol{D}_r^{-\frac{1}{2}}\boldsymbol{U}^{sp}\boldsymbol{\Sigma}^{sp}$ and $\boldsymbol{D}_c^{-\frac{1}{2}}\boldsymbol{V}^{sp}\boldsymbol{\Sigma}^{sp}$ have a first column being equal to 1, a so-called artificial dimension. This artificial dimension reflects the fact that the row margins of the matrix $\boldsymbol{D}_r^{-1}\boldsymbol{P}$ with the row profiles of Table 2.1 are 1 and the column margins of the matrix $\boldsymbol{P}\boldsymbol{D}_c^{-1}$ with the column profiles of Table 2.1 are 1. This artificial dimension is eliminated by not taking the SVD of $\boldsymbol{D}_r^{-\frac{1}{2}}\boldsymbol{P}\boldsymbol{D}_c^{-\frac{1}{2}}$ but of $\boldsymbol{D}_r^{-\frac{1}{2}}(\boldsymbol{P}-\boldsymbol{E})\boldsymbol{D}_c^{-\frac{1}{2}}$, where the elements of $\boldsymbol{E}$ are defined as the product of the margins r_i and c_j. Due to subtracting $\boldsymbol{E}$ from $\boldsymbol{P}$, the rank of $\boldsymbol{D}_r^{-\frac{1}{2}}(\boldsymbol{P}-\boldsymbol{E})\boldsymbol{D}_c^{-\frac{1}{2}}$ is $m-1$, which is 1 less than the rank of $\boldsymbol{F}$. Notice that the elements of $\boldsymbol{D}_r^{-\frac{1}{2}}(\boldsymbol{P}-\boldsymbol{E})\boldsymbol{D}_c^{-\frac{1}{2}}$ are standardized residuals under the independence model, and the sum of squares of these elements yields the so-called

total inertia, which is equal to the Pearson χ^2 statistic divided by sample size f_{++}. By taking the SVD of the matrix of standardized residuals, we get

$$\boldsymbol{D}_r^{-\frac{1}{2}}(\boldsymbol{P}-\boldsymbol{E})\boldsymbol{D}_c^{-\frac{1}{2}} = \boldsymbol{U}^{sr}\Sigma^{sr}(\boldsymbol{V}^{sr})^T \tag{2.11}$$

and

$$\boldsymbol{D}_r^{-1}(\boldsymbol{P}-\boldsymbol{E})\boldsymbol{D}_c^{-1} = \boldsymbol{\Phi}^{sr}\Sigma^{sr}(\boldsymbol{\Gamma}^{sr})^T \tag{2.12}$$

where $\boldsymbol{\Phi}^{sr} = \boldsymbol{D}_r^{-\frac{1}{2}}\boldsymbol{U}^{sr}$ and $\boldsymbol{\Gamma}^{sr} = \boldsymbol{D}_c^{-\frac{1}{2}}\boldsymbol{V}^{sr}$. We use the abbreviation *sr* for the matrices on the right hand side of Equation (2.11) to refer to the matrix of standardized residuals on the left hand side of the equation. CA simultaneously provides a geometric representation of row profiles and column profiles of Table 2.1, where the effects of row margins and column margins of Table 2.1 are eliminated. $\boldsymbol{\Phi}^{sr}$ and $\boldsymbol{\Gamma}^{sr}$ are called standard coordinates of rows and columns, respectively. They have the property that their weighted average is 0 and weighted sum of squares is 1:

$$\mathbf{1}^T\boldsymbol{D}_r\boldsymbol{\Phi}^{sr} = \mathbf{0}^T = \mathbf{1}^T\boldsymbol{D}_c\boldsymbol{\Gamma}^{sr} \tag{2.13}$$

and

$$(\boldsymbol{\Phi}^{sr})^T\boldsymbol{D}_r\boldsymbol{\Phi}^{sr} = \mathbf{I} = (\boldsymbol{\Gamma}^{sr})^T\boldsymbol{D}_c\boldsymbol{\Gamma}^{sr} \tag{2.14}$$

Equation (2.13) reflects the fact that the row and column margins of $\boldsymbol{P}-\boldsymbol{E}$ vanish (Van der Heijden, De Falguerolles, & De Leeuw, 1989).

We can make graphic displays using $\boldsymbol{\Phi}_k^{sr}\Sigma_k^{sr}$ and $\boldsymbol{\Gamma}_k^{sr}\Sigma_k^{sr}$ as coordinates, which has the advantage that Euclidean distances between the points approximate χ^2-distances both for the rows of $\boldsymbol{F}$ and for the columns of $\boldsymbol{F}$, but it has the drawback that Σ_k^{sr} is used twice. We can also make graphic displays using $\boldsymbol{\Phi}_k^{sr}\Sigma_k^{sr}$ and $\boldsymbol{\Gamma}_k^{sr}$, or $\boldsymbol{\Phi}_k^{sr}$ and $\boldsymbol{\Gamma}_k^{sr}\Sigma_k^{sr}$. Thus, from Equation (2.12), this has the advantage that the inner product of the coordinates of a document and the coordinates of a term approximates the corresponding value in $\boldsymbol{D}_r^{-1}(\boldsymbol{P}-\boldsymbol{E})\boldsymbol{D}_c^{-1}$.

If we choose $\boldsymbol{\Phi}^{sr}\Sigma^{sr}$ for the row points and $\boldsymbol{\Gamma}^{sr}$ for the column points, then CA has the property that the row points are in weighted average of the column points, where the weights are the row profile values. Actually, $\boldsymbol{\Gamma}^{sr}$ can be seen as coordinates for the extreme row profiles projected onto the subspace. The extreme row profiles are totally concentrated into one of the terms. For example, $[0,0,1,0,0,0]$ represents the row profile of a document that is totally concentrated into *cheetah*. At the same time, if we choose $\boldsymbol{\Phi}^{sr}$ for the row points and $\boldsymbol{\Gamma}^{sr}\Sigma^{sr}$ for the column points, column points are in weighted average of row points, where the weights are the column profile values. In a similar way as for the rows, $\boldsymbol{\Phi}^{sr}$ provide coordinates for the extreme column profiles projected onto the subspace. The relationship between these row points and column points can be shown by rewriting Equation (2.11) and using Equation (2.13) as

$$\boldsymbol{D}_r^{-1}\boldsymbol{P}\boldsymbol{\Gamma}^{sr} = \boldsymbol{\Phi}^{sr}\Sigma^{sr} \tag{2.15}$$

and

$$\boldsymbol{D}_c^{-1}\boldsymbol{P}^T\boldsymbol{\Phi}^{sr} = \boldsymbol{\Gamma}^{sr}\Sigma^{sr} \tag{2.16}$$

These equations are called the transition formulas. In fact, using transition formulas is one of the ways in which the solution of CA can be obtained: starting from arbitrary values for the columns, one first centers and standardizes the column coordinates so that the weighted sum is 0 and the weighted sums of squares is 1, next places the rows in the weighted average of the columns, then places the columns in the weighted average of the rows, and so on, until convergence. This is known as reciprocal averaging (M. O. Hill, 1973, 1974). Using the transition formula (2.15), the coordinates of the out-of-sample document $\boldsymbol{d}$ is $(\boldsymbol{d}/\sum_{j=1}^{n} d_j)\boldsymbol{\Gamma}^{sr}$ (Greenacre, 2017).

The origin in the graphic representation for the rows stands for the average row profile, which can be seen as follows. Let $\boldsymbol{D}_r^{-1}\boldsymbol{P}\boldsymbol{D}_c^{-\frac{1}{2}}$ be the matrix where Euclidean distances between the rows are χ^2-distances between rows of $\boldsymbol{F}$. Assume we plot the rows of this matrix using the n elements of each row as coordinates. Then, eliminating the artificial dimension in $\boldsymbol{D}_r^{-1}\boldsymbol{P}\boldsymbol{D}_c^{-\frac{1}{2}}$ leads to the subtraction of the average row profile from each row, as $\boldsymbol{D}_r^{-1}\boldsymbol{E}$ is a matrix with the average row profile in each row. In other words, the cloud of row points is translated to the origin, with the average row profile being exactly in the origin (compare Equation (2.13): $\boldsymbol{0}^T = \boldsymbol{1}^T\boldsymbol{D}_c\boldsymbol{\Gamma}^{sr}$). When two row points are departing in the same way from the origin, they depart in the same way from the average profile, and when two row points are on opposite sides of the origin, they depart in opposite ways from the average profile. If the documents and terms are statistically independent, then $p_{ij}/r_i = c_j$, and all document profiles would lie in the origin. Thus comparing row profiles with the origin is a way to study the departure from independence and to study the relations between documents and terms. Similarly, the origin in the graphic representation for the columns stands for average column profile.

We now analyze the example discussed in the LSA section. There are three steps to obtain the CA solution. Step 1: make the matrix $\boldsymbol{D}_r^{-\frac{1}{2}}(\boldsymbol{P}-\boldsymbol{E})\boldsymbol{D}_c^{-\frac{1}{2}}$ of standardized residuals; Step 2: compute the SVD of the matrix; Step 3: derive $\boldsymbol{\Phi}^{sr} = \boldsymbol{D}_r^{-\frac{1}{2}}\boldsymbol{U}^{sr}$ and $\boldsymbol{\Gamma}^{sr} = \boldsymbol{D}_c^{-\frac{1}{2}}\boldsymbol{V}^{sr}$, and post-multiply $\boldsymbol{\Phi}^{sr}$ and $\boldsymbol{\Gamma}^{sr}$ by Σ^{sr} to obtain the coordinates. Table 2.6 shows the matrix $\boldsymbol{D}_r^{-\frac{1}{2}}(\boldsymbol{P}-\boldsymbol{E})\boldsymbol{D}_c^{-\frac{1}{2}}$ of standardized residuals (in lower-case notation, the elements of the matrix are $(p_{ij}-e_{ij})/\sqrt{e_{ij}}$).

Table 2.6: The matrix $\boldsymbol{D}_r^{-\frac{1}{2}}(\boldsymbol{P}-\boldsymbol{E})\boldsymbol{D}_c^{-\frac{1}{2}}$ of standardized residuals

	lion	tiger	cheetah	jaguar	porsche	ferrari
doc1	0.115	0.085	-0.028	-0.005	-0.112	-0.129
doc2	0.014	0.091	0.128	-0.019	-0.140	-0.162
doc3	0.060	0.039	0.060	-0.025	-0.084	-0.098
doc4	0.014	-0.016	0.014	-0.019	0.034	-0.011
doc5	-0.112	-0.119	-0.112	0.020	0.260	0.204
doc6	-0.144	-0.154	-0.144	0.069	0.164	0.338

Table 2.7: The singular values, the inertia, and the proportions of explained total inertia for each dimension of CA.

	dim1	dim2	dim3	dim4
singular value	0.689	0.131	0.124	0.044
inertia	0.475	0.017	0.015	0.002
the proportion of inertia	0.932	0.034	0.030	0.004

We perform an SVD of $\boldsymbol{D}_r^{-\frac{1}{2}}(\boldsymbol{P}-\boldsymbol{E})\boldsymbol{D}_c^{-\frac{1}{2}}$ in Table 2.6 and find Table 2.7. Due to subtracting $\boldsymbol{E}$ from $\boldsymbol{P}$, the rank of the matrix in Table 2.6 is 4, which is 1 less than that in Table 2.1. The proportion of the total inertia explained by only the first dimension accounts for 0.932 of the total inertia. The matrix $\boldsymbol{D}_r^{-\frac{1}{2}}(\boldsymbol{P}-\boldsymbol{E})\boldsymbol{D}_c^{-\frac{1}{2}}$ in Table 2.6 is approximated in the first two dimensions as follows:

$$\begin{aligned}\boldsymbol{D}_r^{-\frac{1}{2}}(\boldsymbol{P}-\boldsymbol{E})\boldsymbol{D}_c^{-\frac{1}{2}} &\approx \boldsymbol{U}_2^{sr}\Sigma_2^{sr}(\boldsymbol{V}_2^{sr})^T \\ &= \begin{bmatrix} -0.286 & 0.789 \\ -0.368 & -0.517 \\ -0.231 & -0.025 \\ 0.007 & -0.138 \\ 0.547 & -0.206 \\ 0.656 & 0.220 \end{bmatrix} \begin{bmatrix} 0.689 & 0 \\ 0 & 0.131 \end{bmatrix} \begin{bmatrix} -0.301 & 0.544 \\ -0.338 & 0.090 \\ -0.303 & -0.761 \\ 0.102 & 0.152 \\ 0.512 & -0.275 \\ 0.656 & 0.136 \end{bmatrix}^T \end{aligned} \tag{2.17}$$

Figure 2.2a is the map with a symmetric role for the rows and the columns, having $\boldsymbol{\Phi}_2^{sr}\Sigma_2^{sr}$ and $\boldsymbol{\Gamma}_2^{sr}\Sigma_2^{sr}$ as coordinates. The larger the deviations from document (term) points to the origin are, the larger the dependence between documents and terms. Looking only at the first dimension and document profiles' positions, we can see that the groups furthest apart are documents 1-3 on the left-hand side, opposed to documents 5-6 on the right-hand side. They differ in opposite ways from the average row profile that lies in the origin. For the term points on the first dimension, the cat terms (*tiger*, *cheetah*, and *lion*) lie on the left, and car terms (*porsche* and *ferrari*) on the right. They differ in opposite ways from the average column profile. Importantly, CA clearly displays the properties we see in the data matrix, as document 4 lies between documents 1-3 and documents 5-6, and the term *jaguar* lies between cat terms and car terms, unlike all four of the LSA-based analyses presented in Figure 2.1.

Figure 2.2b is the asymmetric map with documents in the weighted average of the terms ($\boldsymbol{\Phi}_2^{sr}\Sigma_2^{sr}$ and $\boldsymbol{\Gamma}_2^{sr}$ as coordinates, notice that the position of the documents is identical as in Figure 2.2a). From this graphic display we can study the position of the documents as they are in the weighted average of the terms, using the row profile elements as weights. For example, document 1 is closer to *lion* and *tiger* than to *porsche* and *ferrari*, because it has higher profile values than average values on terms *lion* and *tiger* (both 0.286 in comparison with the average profile values 0.171 and 0.195) and lower profile values on the terms *porsche* and *ferrari* (both 0.000 in comparison to

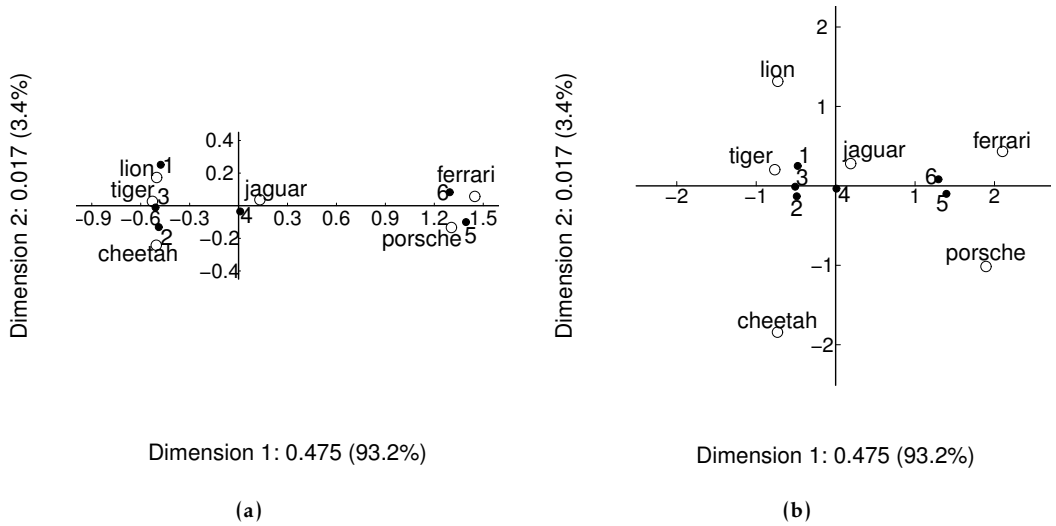


Figure 2.2: The data of Table 2.1 using CA for (a) symmetric map; (b) asymmetric map.

0.073 and 0.098), see Table 2.3. Thus document 1 is pulled into the direction of *lion* and *tiger*.

2.3.1 Conclusions regarding CA

In CA, an SVD is applied to the matrix $\boldsymbol{D}_r^{-\frac{1}{2}}(\boldsymbol{P}-\boldsymbol{E})\boldsymbol{D}_c^{-\frac{1}{2}}$ of standardized residuals. Due to $\boldsymbol{E}$, in CA the effect of the margins is eliminated—a solution only displays the relationships among documents and terms. In CA all points are scattered around the origin and the origin represents the profile of the row and column margins of $\boldsymbol{F}$.

In comparison, LSA also tries to capture the relationships among documents and terms, which is not easy. The reason is that these relations are blurred by the effect of the margins that are also displayed in the LSA solution. CA does not have this property. Therefore it appears that CA is a better tool for information retrieval, natural language processing, and text mining.

2.4 A unifying framework

Here we present a unifying framework that integrates LSA and CA. This section also serves the purpose of showing their similarities and their differences.

To first summarize LSA (see section 2.2.2 for details), a matrix is weighted, and the weighted matrix is decomposed. Assume we start off with the document-term matrix $\boldsymbol{F}$, the row weights of $\boldsymbol{F}$ are collected in the diagonal matrix $\boldsymbol{N}$, the column weights in the diagonal matrix $\boldsymbol{G}$, and there may be local weighting of the elements f_{ij} of $\boldsymbol{F}$ leading to a locally weighted matrix $\boldsymbol{L}$. Thus the weighted matrix $\boldsymbol{W}$ can be written as

the matrix product

$$\boldsymbol{W} = \boldsymbol{NLG} \tag{2.18}$$

Subsequently, in LSA the matrix $\boldsymbol{W}$ is decomposed using SVD into a product of three matrices: the matrix $\boldsymbol{U}$ with orthonormal columns, the diagonal matrix Σ with singular values in descending order, and the matrix $\boldsymbol{V}$ with orthonormal columns, namely

$$\boldsymbol{W} = \boldsymbol{U}\Sigma\boldsymbol{V}^T \tag{2.19}$$

with

$$\boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{I} = \boldsymbol{V}^T\boldsymbol{V} \tag{2.20}$$

Graphic representations are usually made using $\boldsymbol{U}\Sigma$ as coordinates for the rows and $\boldsymbol{V}\Sigma$ for the columns.

In contrast, in CA we take the SVD of the matrix of standardized residuals. Let $\boldsymbol{P}$ be the matrix with proportions $p_{ij} = f_{ij}/f_{++}$, where f_{++} is the sum of all elements of $\boldsymbol{F}$; let $\boldsymbol{E}$ be the matrix with expected proportions under independence $e_{ij} = r_i c_j$, where r_i and c_j are the row and column sums of $\boldsymbol{P}$ respectively; let $\boldsymbol{D_r}$ and $\boldsymbol{D_c}$ be diagonal matrices with row and column sums r_i and c_j respectively. Thus the matrix of standardized residuals is $\boldsymbol{D}_r^{-\frac{1}{2}}(\boldsymbol{P}-\boldsymbol{E})\boldsymbol{D}_c^{-\frac{1}{2}}$. If we take the SVD of this matrix we get (2.11),

$$\boldsymbol{D}_r^{-\frac{1}{2}}(\boldsymbol{P}-\boldsymbol{E})\boldsymbol{D}_c^{-\frac{1}{2}} = \boldsymbol{U}\Sigma\boldsymbol{V}^T \tag{2.21}$$

In CA the matrices $\boldsymbol{U}$ and $\boldsymbol{V}$ are further adjusted by

$$\boldsymbol{\Phi} = \boldsymbol{D}_r^{-\frac{1}{2}}\boldsymbol{U}, \boldsymbol{\Gamma} = \boldsymbol{D}_c^{-\frac{1}{2}}\boldsymbol{V} \tag{2.22}$$

so that we can write

$$\boldsymbol{D}_r^{-1}(\boldsymbol{P}-\boldsymbol{E})\boldsymbol{D}_c^{-1} = \boldsymbol{\Phi}\Sigma\boldsymbol{\Gamma}^T \tag{2.23}$$

with

$$\boldsymbol{\Phi}^T\boldsymbol{D_r}\boldsymbol{\Phi} = \boldsymbol{I} = \boldsymbol{\Gamma}^T\boldsymbol{D_c}\boldsymbol{\Gamma} \tag{2.24}$$

Graphic representations are usually made using $\boldsymbol{\Phi}\Sigma$ and $\boldsymbol{\Gamma}\Sigma$ as coordinates for the rows and columns respectively.

This brings us to the point where we can formulate a unifying framework. We distinguish the matrix to be analyzed and the decomposition of this matrix. For the matrix to be analyzed the weighted matrix defined in (2.18) can be used by LSA as well as by CA. Equation (2.18) is sufficiently general for LSA. For CA, using (2.21), we set $\boldsymbol{N} = \boldsymbol{D_r}^{-\frac{1}{2}}$, $\boldsymbol{L} = \boldsymbol{P}-\boldsymbol{E}$, and $\boldsymbol{G} = \boldsymbol{D_c}^{-\frac{1}{2}}$. This shows that the matrix decomposed in CA in (2.21) can be formulated in the LSA framework in (2.18).

The decomposition used in LSA leads to matrices $\boldsymbol{U}$ with orthonormal columns and $\boldsymbol{V}$ with orthonormal columns used for coordinates, see (2.20), whereas in CA the decomposition leads to matrices $\boldsymbol{\Phi}$ with weighted orthonormal columns and $\boldsymbol{\Gamma}$ with weighted orthonormal columns, see (2.24). If we rewrite (2.20) as $\boldsymbol{U}^T\boldsymbol{I}\boldsymbol{U} = \boldsymbol{I} = \boldsymbol{V}^T\boldsymbol{I}\boldsymbol{V}$,

we see this is a difference between using an identity metric $\boldsymbol{I}$ and a metric defined by the margins that are collected in $\boldsymbol{D_r}$ and in $\boldsymbol{D_c}$. The influence of this metric used in CA is most clearly visible in the definition of the chi-squared distances (2.8), that makes that, for example, for row profiles i and i', equally large differences between columns j and j' are weighted by the margins of j and j' in such a way that a column with a smaller margin takes a larger part in the chi-squared distance between i and i'.

2.5 Text categorization

LSA is widely used in text categorization (W. Zhang et al., 2011; Elghazel, Aussem, Gharroudi, & Saadaoui, 2016; Dzisevič & Šešok, 2019; Phillips et al., 2021). However, to our best knowledge, few papers on text categorization use CA, even though CA is similar to LSA. In this section, we compare the performance of LSA and CA in text categorization of three English datasets: BBCNews, BBCSport, and 20 Newsgroups. These datasets have recently been studied in the evaluation of text categorization, for example, Barman and Chowdhury (2020).

2.5.1 Datasets and methods

The BBCNews dataset (Greene & Cunningham, 2006) consists of 2,225 documents that are divided into five categories: "Business" (510 documents), "Entertainment" (386), "Politics" (417), "Sport" (511), and "Technology" (401). The BBCSport dataset (Greene & Cunningham, 2006) consists of 737 documents that are divided into five categories: "athletics" (101), "cricket" (124), "football" (265), "rugby" (147), and "tennis" (100). The 20 Newsgroups dataset, i.e. the 20news-bydata version (Rennie, 2005), consists of 18,846 documents that are divided into 20 categories. The dataset is sorted into a training (60 per cent) and a test set (40 per cent). We use a subset of these documents. Specifically, we choose 2,963 documents from three categories: "comp.graphics" (584 documents for training set and 389 documents for test set), "rec.sport.hockey" (600 and 399), and "sci.crypt" (595 and 396). The reason we choose a subset (three categories) of 20 Newsgroups is that we want to explore text categorization for datasets with a different but similar number of categories: six (for *Wilhelmus* dataset in Section 2.6), five (for BBCNews), five (for BBCSport), and three (for a subset of 20 Newsgroups).

To pre-process these datasets we project all characters to lower case, remove punctuation marks, numbers, and stop words, and apply lemmatization. Subsequently, terms with frequencies lower than 10 are ignored. In addition, following Silge and Robinson (2017), we remove unwanted parts of the 20 Newsgroups dataset such as headers (including fields like "From:" or "Reply-To:" that describe the message), because these are mostly irrelevant for text categorization.

We use two approaches to compare LSA and CA. One is visualization, where we use LSA and CA to visualize documents by projecting them onto two dimensions. The

other is to use distance measures to quantitatively evaluate and compare performance in text categorization. We use four different methods based on Euclidean distance for measuring the distance from a document to a set of documents (Guthrie, 2008; Koppel & Seidman, 2013; Kestemont, Stover, Koppel, Karsdorp, & Daelemans, 2016). We choose the Euclidean distance because it plays a central role in the geometric interpretation of LSA and CA (see section 2.2 and 2.3).

Centroid Euclidean distance between the document and the centroid of the set of documents. The centroid for a set of documents is calculated by averaging the coordinates across all these documents.

In the other three methods we first calculate the Euclidean distance between the document and every document of the set of documents.

Average average of these Euclidean distances

Single the minimum Euclidean distance among the Euclidean distances

Complete the maximum Euclidean distance among the Euclidean distances.

These four methods are similar to the procedures of measuring the distance between clusters in hierarchical clustering analysis, using the centroid, average, single, and complete linkage method respectively (Jarman, 2020).

In line with the foregoing sections, we denote the raw document-term matrix by $\boldsymbol{F}$. In the case of LSA we examine four versions: LSA of $\boldsymbol{F}$ (LSA-RAW), LSA of the row-normalized matrices $\boldsymbol{F}^{L1}$ (LSA-NROWL1) and $\boldsymbol{F}^{L2}$ (LSA-NROWL2), and LSA of the TF-IDF matrix $\boldsymbol{F}^{\text{TF-IDF}}$ (LSA-TFIDF). In addition, we also compare performance with the raw document-term matrix, denoted as RAW, where no dimensionality reduction has taken place.

2.5.2 Visualization

The 2,225 documents of the BBCNews dataset lead to a document-term matrix of size 2,225 × 5,050. Figure 2.3 shows the results of an analysis of this document-term matrix by the four LSA methods (LSA-RAW, LSA-NROWL1, LSA-NROWL2, LSA-TFIDF) and CA. On this dataset, we find that, although the percentage of the total sum of squared singular values in the first two dimensions for CA is lower than the four LSA methods, the four LSA methods do not separate the classes well but CA does a reasonably good job. This is because the margins play an important role in the first two dimensions for the four LSA methods and the relations between documents are blurred by these margins.

The 737 documents of BBCSport dataset lead to a document-term matrix of size 737 × 2,071. Figure 2.4 shows the results of an analysis of this document-term matrix. Again, we find that the LSA methods do not separate the classes well, but CA does a reasonably good job.

The 2,963 documents of 20 Newsgroups dataset lead to a document-term matrix of size 2,927 × 2,834.[1] Figure 2.5 shows the results of an analysis of this document-term matrix. On this dataset, we find that CA is doing a reasonably good job, and so do LSA-NROWL1 and LSA-NROWL2.

2.5.3 Distance measures

For the 20 Newsgroups dataset, there is a training and a test set, and we assess the accuracy as a measure for the correct classification of the documents of the test set. For the 20 Newsgroups data set there are four steps. First, we apply all four varieties of LSA and CA to all documents of the training set. The documents of the test set are projected into the reduced dimensional space, see Section 2.2.2.4 and Section 2.3. Second, using the centroid, average, single, and complete method, for each document of the test set, the distance between the document and a set of documents for each of three categories ("comp.graphics", "rec.sport.hockey", "sci.crypt") in the training set is computed. The predicted category for the document is the category with the smallest distance. Third, we compare the predicted category with the true category of the document. Finally, the accuracy is the proportion of correct classifications of all documents of the test set. For BBCNews and BBCSport datasets, in order to evaluate LSA methods and CA, we use five-fold cross validation (Gareth, Daniela, Trevor, & Robert, 2021). That is, the dataset is randomly divided into five folds. The four folds (80 per cent of the dataset) are used as training set and the remaining one fold (20 per cent of the dataset) is as validation set. The accuracy of each fold is obtained as in the 20 Newsgroups dataset. Then the accuracy is averaged across five folds.

For each form of LSA and for CA, there is an accuracy for each number of dimensions (for five-fold cross validation, the accuracy is averaged across five folds). The maximum accuracy is the maximum value across these accuracies. Table 2.8 shows the maximum accuracy for LSA-RAW, LSA-NROWL1, LSA-NROWL2, LSA-TFIDF, and CA for the four distance measures[2], along with the *minimum* optimal dimension k where this maximum accuracy is reached [3]. First, if we ignore the complete distance method, considering that it has low accuracy overall, CA yields the maximum accuracy compared to the RAW method (i.e. without dimensionality reduction) as well as all four LSA methods for each combination of dataset and other distance measurement method, except for the BBCSport dataset with the average method, where CA has the second largest accuracy. Second, for each dataset CA is doing best overall. Specifically, CA with the centroid, the single, and the centroid distance method provides the best accuracy for BBCNews, BBCSport, and 20 Newsgroups datasets, respectively.

In order to further explore different dimensionality reduction methods under opti-

[1] After preprocessing, 36 documents out of 2,963 became empty documents and were removed.

[2] For BBCSport dataset, we explore the number of all dimensions of dimensionality reduction methods. For BBCNews and 20 Newsgroups datasets, we vary the number of dimension k from 1 to 450.

[3] There is not one single optimal number of dimensions that provides the maximum accuracy; for reasons of space, we show only the lowest in Tables 2.8, 2.9.

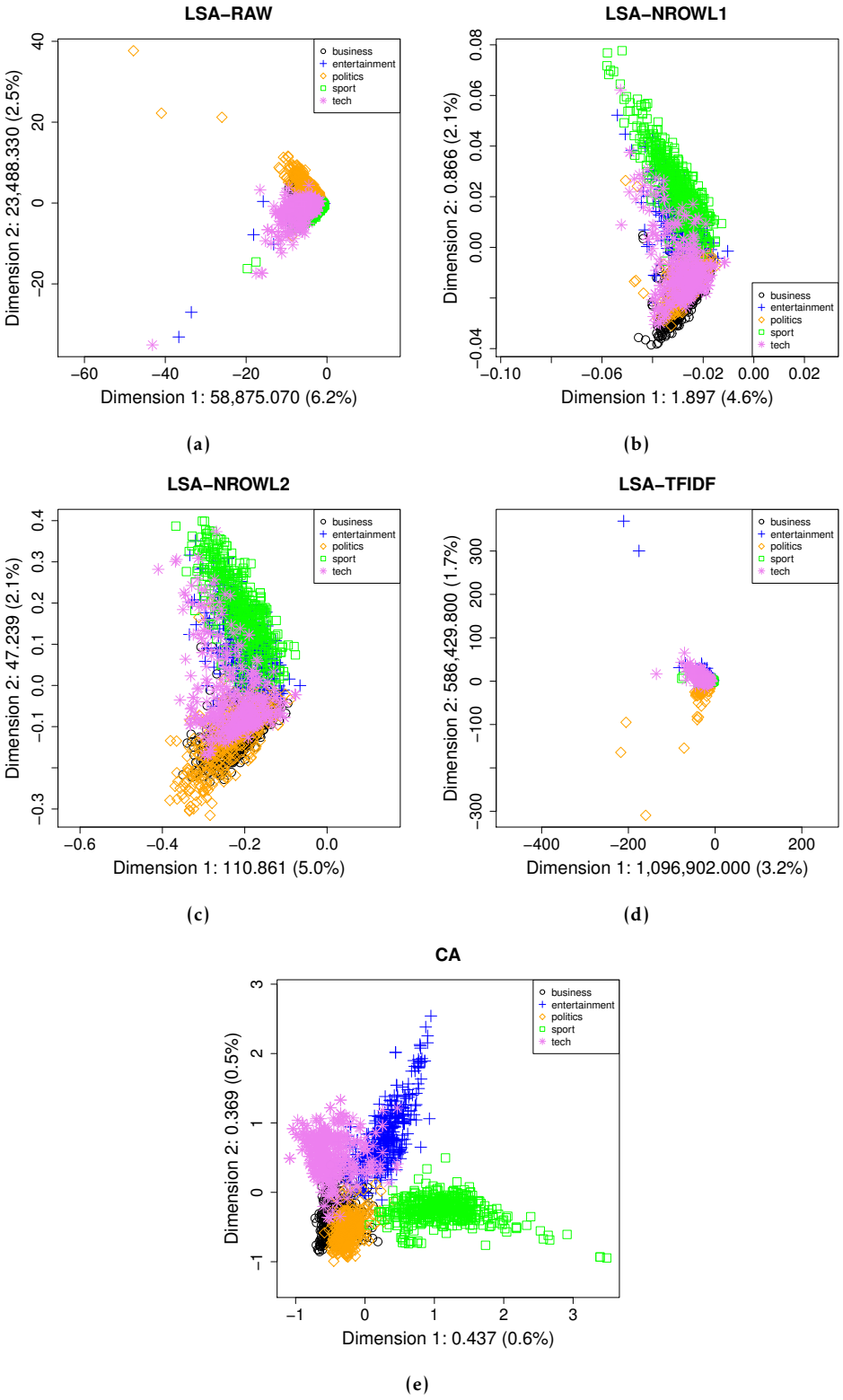


Figure 2.3: The first two dimensions for each document of BBCNews dataset by (a) LSA-RAW; (b) LSA-NROWL1; (c) LSA-NROWL2; (d) LSA-TFIDF; (e) CA.

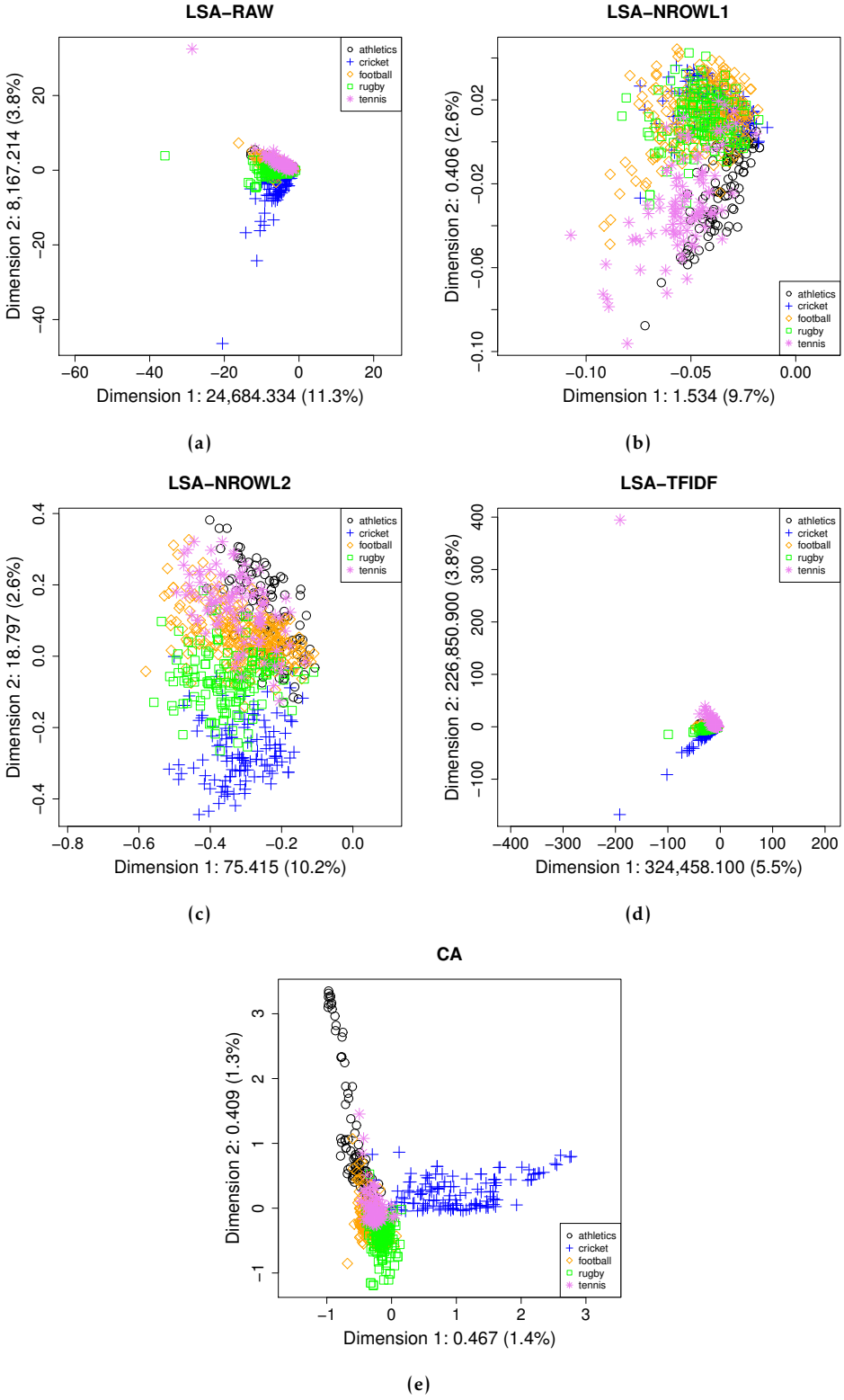


Figure 2.4: The first two dimensions for each document of BBCSport dataset by (a) LSA-RAW; (b) LSA-NROWL1; (c) LSA-NROWL2; (d) LSA-TFIDF; (e) CA.

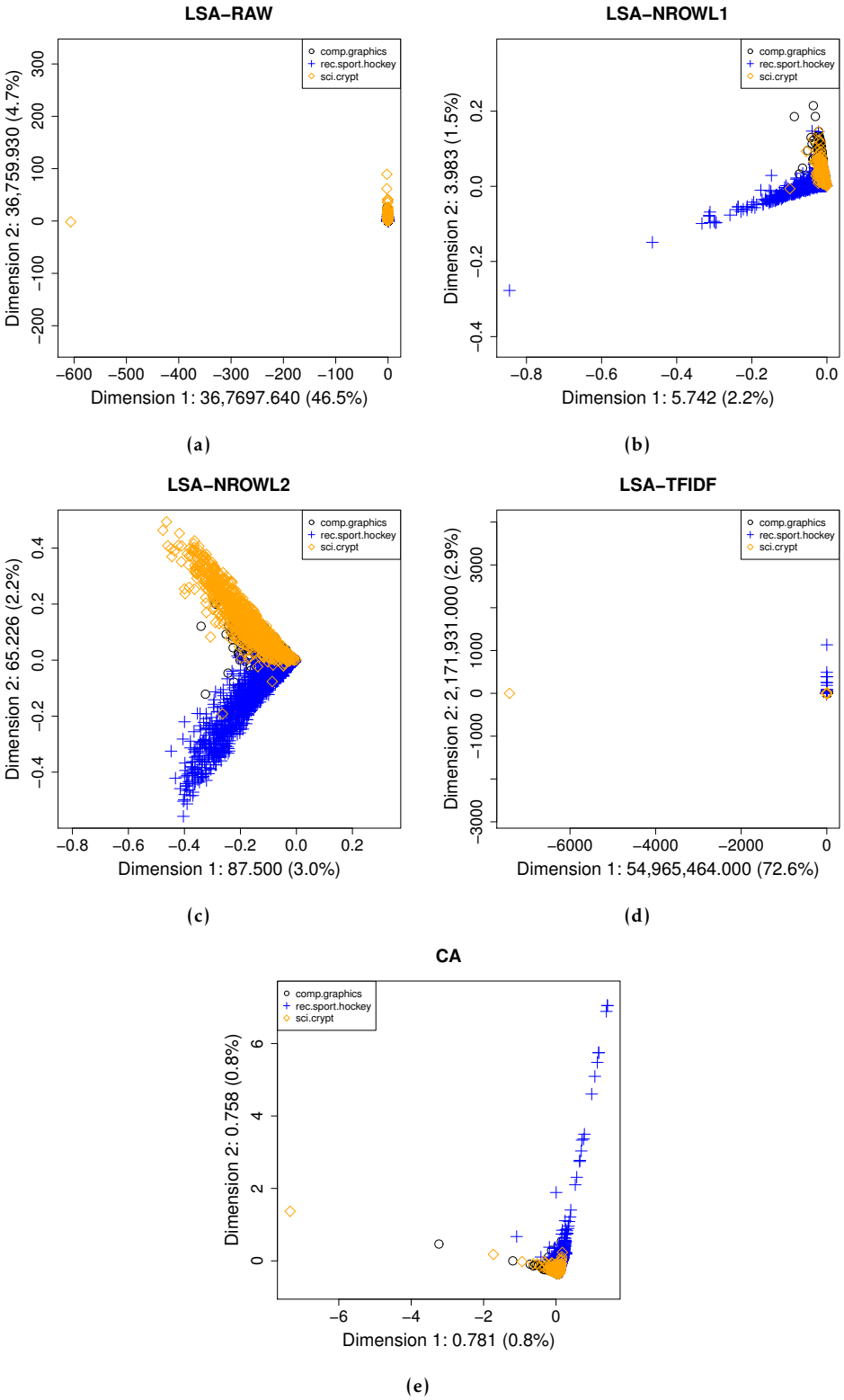


Figure 2.5: The first two dimensions for each document of 20 Newsgroups dataset by (a) LSA-RAW; (b) LSA-NROWL1; (c) LSA-NROWL2; (d) LSA-TFIDF; (e) CA.

mal distance measurement method which provides highest accuracy, Figure 2.6 shows the accuracy as a function of the numbers of dimensions under centroid, single, and centroid methods for BBCNews, BBCSport, and 20 Newsgroups datasets, respectively. CA in combination with the optimal distance measurement method performs better than the other methods over a large range, especially for BBCNews dataset, almost irrespective of dimension.

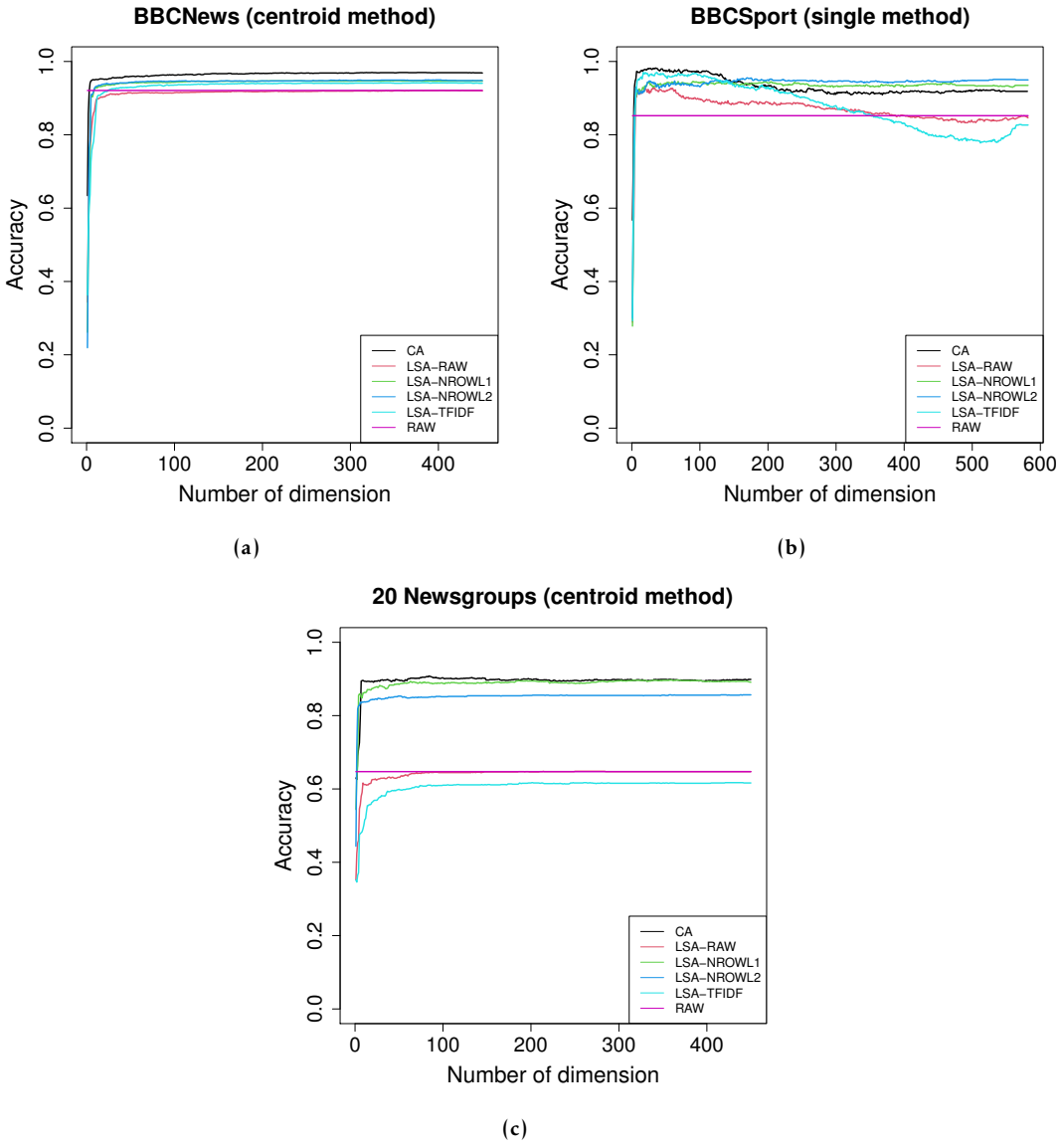
Table 2.8: The minimum optimal dimensionality k and the accuracy (Acc) in k for LSA-RAW, LSA-NROWL1, LSA-NROWL2, LSA-TFIDF, and CA, and the Acc for RAW using different distance measurement methods with the BBCNews, BBCSport, and 20 Newsgroups datasets.

Datasets	Methods	Centroid		Average		Single		Complete	
		k	Acc	k	Acc	k	Acc	k	Acc
BBCNews	RAW		0.921		0.339		0.791		0.229
	LSA-RAW	401	0.921	7	0.714	24	0.942	1	0.237
	LSA-NROWL1	339	0.947	5	0.898	30	0.948	5	0.723
	LSA-NROWL2	385	0.950	23	0.930	450	0.951	5	**0.829**
	LSA-TFIDF	381	0.942	13	0.725	32	0.953	13	0.253
	CA	318	**0.970**	5	**0.943**	22	**0.961**	4	0.647
BBCSport	RAW		0.917		0.418		0.852		0.193
	LSA-RAW	72	0.919	9	0.843	33	0.930	9	0.332
	LSA-NROWL1	275	0.950	10	0.928	129	0.946	5	0.613
	LSA-NROWL2	96	0.952	103	**0.950**	175	0.955	5	**0.873**
	LSA-TFIDF	486	0.931	9	0.806	20	0.970	7	0.241
	CA	565	**0.978**	24	0.936	35	**0.982**	4	0.420
20 Newsgroups	RAW		0.647		0.330		0.688		0.328
	LSA-RAW	214	0.648	9	0.409	26	0.847	2	0.342
	LSA-NROWL1	358	0.897	4	0.847	306	0.852	83	0.412
	LSA-NROWL2	357	0.857	54	0.885	6	0.858	3	**0.735**
	LSA-TFIDF	201	0.617	1	0.347	70	0.863	1	0.340
	CA	84	**0.908**	7	**0.888**	27	**0.902**	11	0.465

2.6 Authorship attribution

In this section we examine the performance of LSA and CA on a dataset originally set up for authorship attribution. We first use the dataset to see how well LSA and CA are able to assign documents with a known author to the correct author. Second, we assign a document with unknown author to one of the known authors.

Authorship attribution is the process of identifying the authorship of a document; its applications include plagiarism detection and resolving of authorship disputes (Bozkurt, Baghoglu, & Uyar, 2007), and are particularly relevant for historical texts, where other historical records are not sufficient to determine authorship. Both LSA



(a) (b)

(c)

Figure 2.6: Accuracy as a function of dimension for CA, LSA-RAW, LSA-NROWL1, LSA-NROWL2, LSA-TFIDF, and RAW

and CA have been used for authorship attribution before. For example, Soboroff, Nicholas, Kukla, and Ebert (1997) applied LSA with n-grams as terms to visualize authorship among biblical Hebrew texts. McCarthy, Lewis, Dufty, and McNamara (2006) applied LSA to lexical features to automatically detect semantic similarities between words (Stamatatos, 2009). Satyam, Dawn, and Saha (2014) used LSA on a character n-gram based representation to build a similarity measure between a questioned document and known documents. Mealand (1995) studied the Gospel of Luke using a visualization provided by CA. Mealand (1997) also measured genre differences in Mark by CA. Mannion and Dixon (2004) applied CA to study authorship attribution of the case of Oliver Goldsmith by visualization.

The *Wilhelmus* is the national anthem of the Netherlands and its authorship is unknown and much debated. There is a substantive amount of qualitative research attempting to determine the authorship of the *Wilhelmus*, with quantitative or statistical methods being used relatively recently. To the best of our knowledge, the authorship of the *Wilhelmus* was first studied by statistical methods and computational means in Winkel (2015), whose results on authorship attribution were inconclusive. After that, Kestemont, Stronks, De Bruin, and Winkel (2017a, 2017b) studied the question using principal component analysis and the General Imposters (GI) method, attributing the *Wilhelmus* to the writer Datheen. Vargas Quiros (2017) used the data of Kestemont et al. (2017a, 2017b), and applied the KRIMP compression algorithm (Van Leeuwen, Vreeken, & Siebes, 2006) and Kullback-Leibler Divergence — they tended to agree with Kestemont et al. (2017a, 2017b), even though the KRIMP attributed the *Wilhelmus* to another author when a different feature selection method was used. Thus, the results were inconclusive, with a tendency to prefer Datheen. Our paper provides further evidence in favor of attributing the authorship to *Datheen*.

2.6.1 Data and methods

We use a total of 186 documents by six writers, consisting of 35 documents written by Datheen, 46 by Marnix, 23 by Heere, 35 by Haecht, 33 by Fruytiers, and 14 by Coornhert. These documents contain tag-lemma pairs as terms, obtained through part-of-speech tagging and lemmatizing of the texts, and are made publicly available by Kestemont et al. (2016, 2017a, 2017b). The average marginal frequencies range from 406 for documents by Fruytiers to 545 for documents by Haecht. See Kestemont (2017) for more details regarding the dataset. Similar to Section 2.5, in this section we also use visualization and distance measures to compare LSA and CA.

2.6.2 Visualization

We first examine all documents of two authors Marnix and Datheen[4], using the 300 most frequent tag-lemma pairs. These form a document-term matrix of size 81×300. Figure 2.7 shows the results of analyzing this document-term matrix using the four LSA methods (LSA-RAW, LSA-NROWL1, LSA-NROWL2, LSA-TFIDF), and CA. The *Wilhelmus* document is not included in the data matrix but it is projected into the solutions for illustrative purposes by W, in red, see Section 2.2.2.4 and Section 2.3. As seen in Figure 2.7, all four varieties of LSA fail to show a clear separation, while CA separates documents by the two authors clearly, even though the first 2 dimensions for CA account for a much smaller percentage of the total sum of squared singular values than the first 2 dimensions for the four LSA methods. This is because the margins play an important role in the first two dimensions for the four LSA methods and the relations between documents are blurred by these margins. We also see that in CA the *Wilhelmus* is clearly attributed to Datheen.

Given the effectiveness of CA and the attribution of the *Wilhelmus* to Datheen in the above analysis, we now show visualizations of CA for documents by Datheen and four other authors in turn (Figure 2.8). For three out of four authors, there is a clear separation between that author and Datheen. In the case Haecht however (sub-figure (b)), there is no clear separation from Datheen. In all three cases where there is a clear separation, *Wilhelmus* is attributed to Datheen, as before.

Finally, we apply all four varieties of LSA and CA to all documents of the six authors, which form a document-term matrix of size 186×300. Figure 2.9 shows the results of the analysis of this matrix by LSA-RAW, LSA-NROWL1, LSA-NROWL2, LSA-TFIDF, and CA. The *Wilhelmus* is projected into the solutions afterwards. Again we find that, although the percentage of the total sum of squared singular values in the first two dimensions for CA is lower than the four LSA methods, CA separates the documents quite well compared with the four LSA methods. For instance, documents written by Marnix are effectively separated from the documents written by other authors. The documents of the other authors also seem to form much more distinguishable clusters, as compared to LSA, except for Datheen and Haecht.

2.6.3 Distance measures

To evaluate LSA methods and CA, we use leave-one-out cross-validation (LOOCV) (Gareth et al., 2021) with the 186 documents of six authors. Using LOOCV, each time we discern the following four steps. At the first step, a single document of the 186 documents is used as the validation set and the remaining 185 documents make up the training set. The 185 documents of training set form a document-term matrix with 185 rows and 300 columns. At step two, we perform LSA-RAW, LSA-NROWL1,

[4]We chose these two authors specifically, out of our dataset, as they are the two main contenders for the authorship of *Wilhelmus* – Marnix has been the most popular candidate from qualitative analysis, and since the work of Kestemont et al. (2017a, 2017b) Datheen is also a serious candidate.

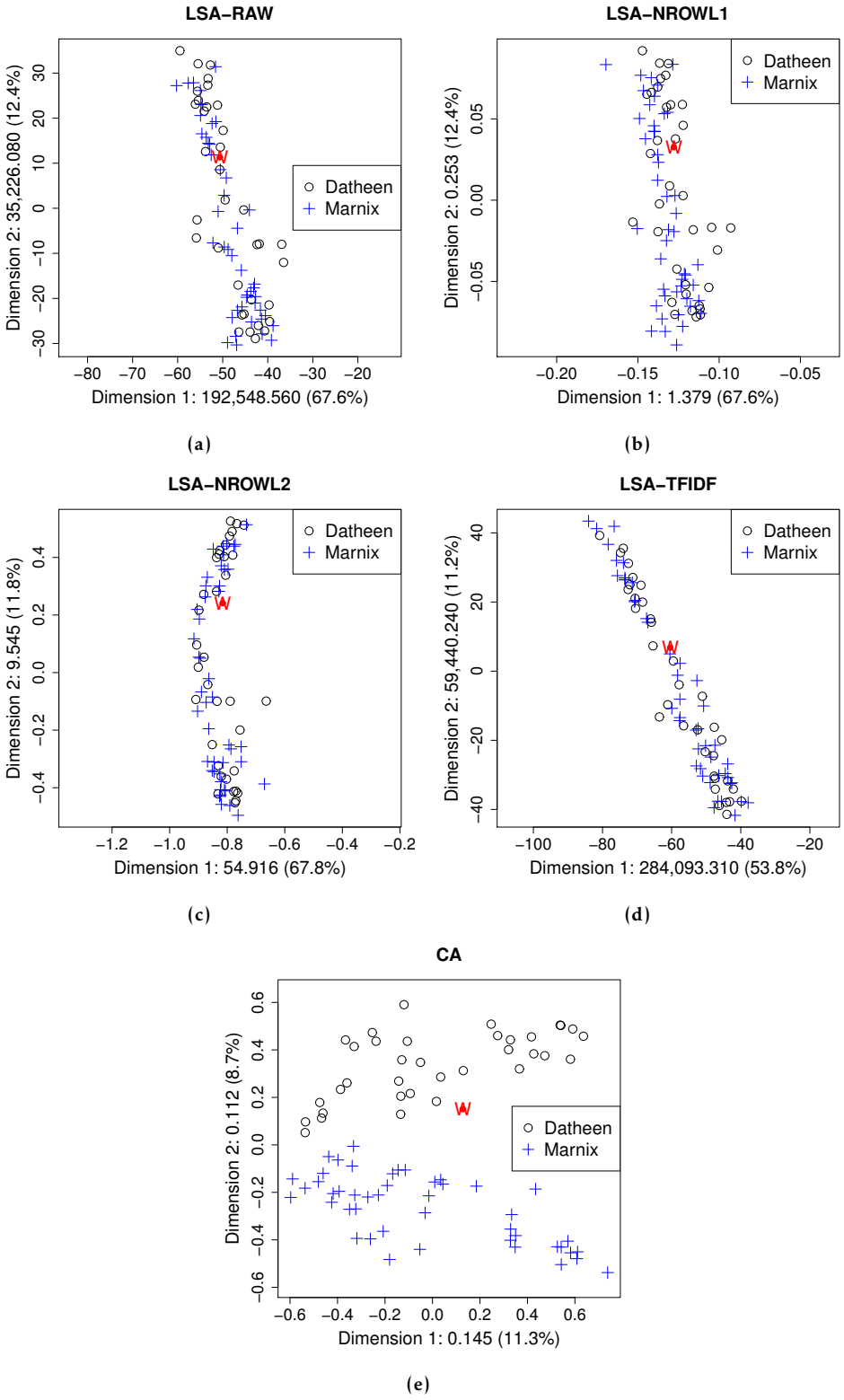


Figure 2.7: The first two dimensions for each document of author Datheen and author Marnix, and the *Wilhelmus* (in red) by (a) LSA-RAW; (b) LSA-NROWL1; (c) LSA-NROWL2; (d) LSA-TFIDF; (e) CA.

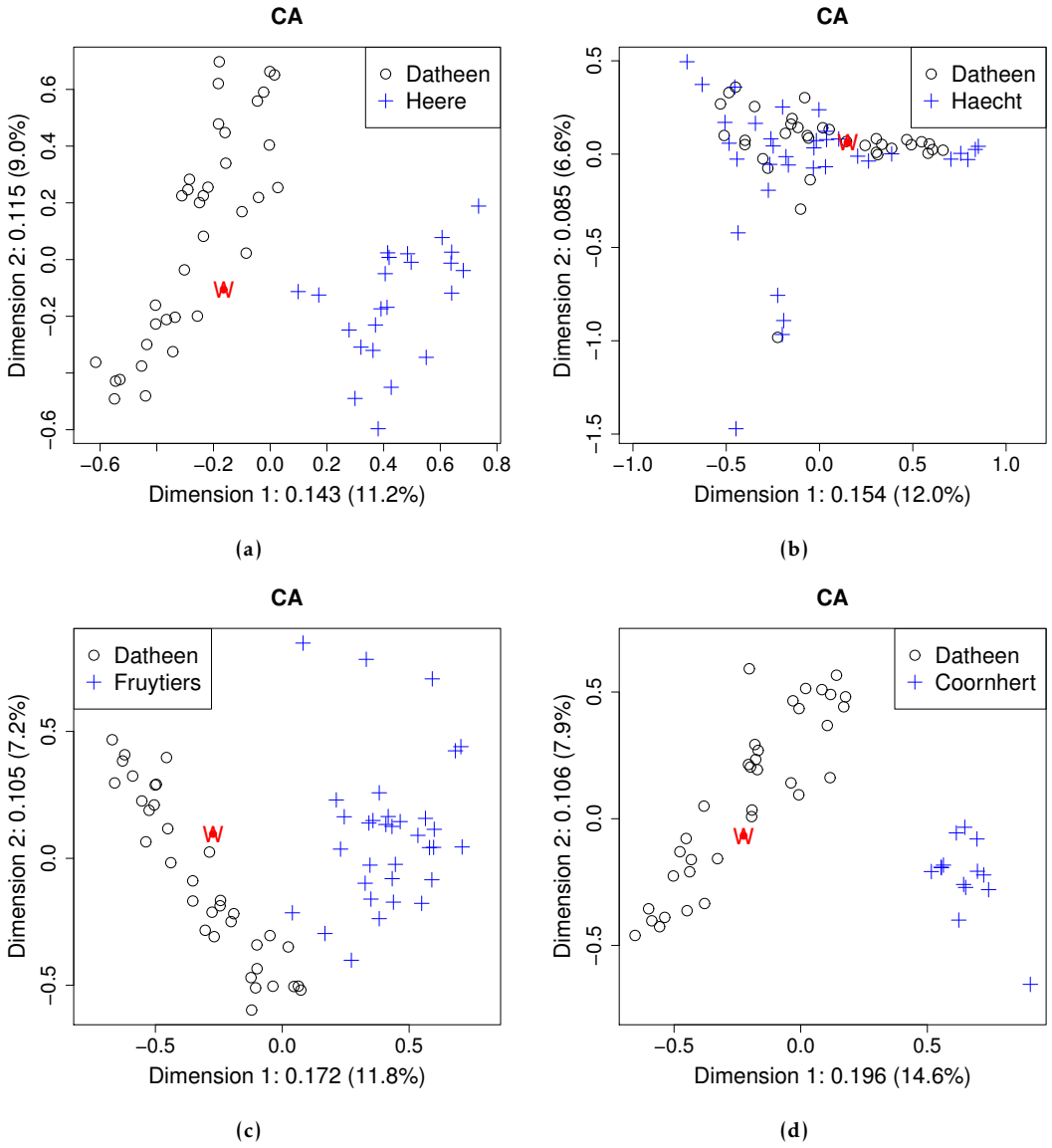


Figure 2.8: The first two dimensions for each document of author Datheen and another author, and the *Wilhelmus* (in red) using CA: (a) Heere; (b) Haecht; (c) Fruytiers; (d) Coornhert.

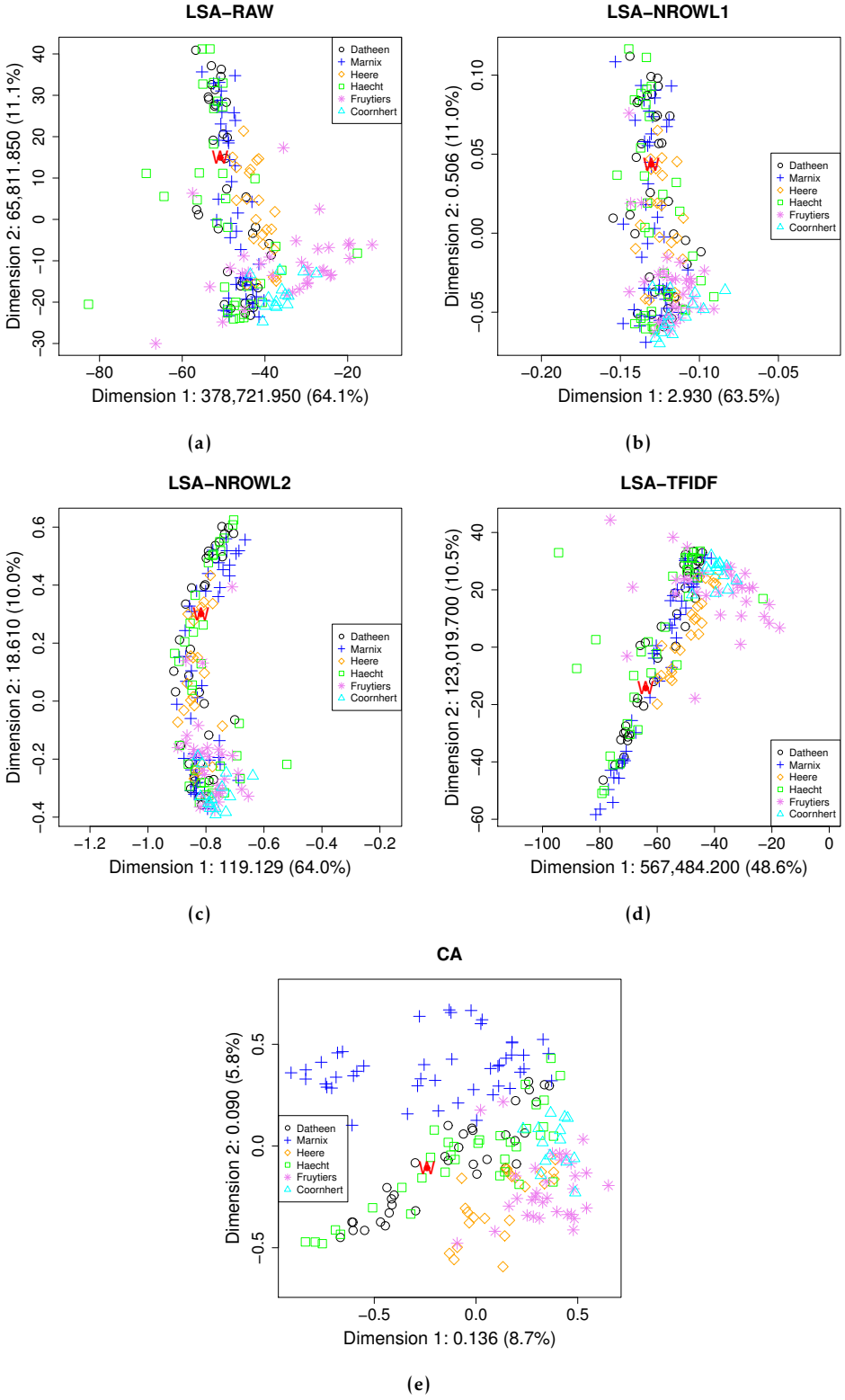


Figure 2.9: The first two dimensions for each document of six authors, and the *Wilhelmus* (in red) by (a) LSA-RAW; (b) LSA-NROWL1; (c) LSA-NROWL2; (d) LSA-TFIDF; (e) CA.

LSA-NROWL2, LSA-TFIDF, and CA on this document-term matrix to obtain the coordinates of the 185 documents. The single document of validation set is projected into the solutions, see Section 2.2.2.4 and Section 2.3. At step three, using the centroid, average, single, and complete method, the distance is computed between the single document and the six author groups of documents. For this single document, the predicted author of the document is the author with the smallest distance. At the final step, we compare the predicted author with the true author of the single document. We repeat this 186 times, once for each single document. The accuracy is calculated by the ratio: number of times an author is correctly predicted divided by 186.

Table 2.9: The minimum optimal dimensionality k and the accuracy in k for LSA-RAW, LSA-NROWL1, LSA-NROWL2, LSA-TFIDF, and CA, and the accuracy for RAW using different distance measurement methods with *Wilhelmus* dataset.

Methods	Centroid		Average		Single		Complete	
	k	Accuracy	k	Accuracy	k	Accuracy	k	Accuracy
RAW		0.720		0.522		0.672		0.177
LSA-RAW	51	0.720	70	0.554	14	0.720	1	0.296
LSA-NROWL1	93	0.731	116	0.645	22	0.710	75	0.226
LSA-NROWL2	59	0.742	41	0.699	21	0.715	77	0.301
LSA-TFIDF	84	0.720	90	0.538	23	0.731	1	0.231
CA	151	**0.930**	12	**0.790**	19	**0.785**	95	**0.452**

Table 2.9 shows the maximum accuracy for LSA-RAW, LSA-NROWL1, LSA-NROWL2, LSA-TFIDF, and CA for the four distance measures [5], along with the minimum optimal dimension k. First, CA yields the maximum accuracy for all distance measurement methods as compared to the RAW method as well as all four LSA methods. Second, CA with the centroid method provides the highest accuracy.

In order to further explore the centroid method, Figure 2.10 shows the accuracy with different numbers of dimensions for LSA-RAW, LSA-NROWL1, LSA-NROWL2, LSA-TFIDF, and CA. Figure 2.10a displays all dimensions on the horizontal axis, and Figure 2.10b focuses on the first 10 dimensions. CA in combination with the centroid method performs better than the other methods almost irrespective of dimension, except for the very first ones. Also, the accuracy of CA in combination with the centroid method is very high over a large range.

2.6.4 Authorship attribution of the *Wilhelmus*

Since CA in combination with the centroid method appears to be the best overall, we use them to determine the authorship of the *Wilhelmus*. In the 34 optimal dimensions (dimensions 151-184), we find that the *Wilhelmus* is attributed to the author Datheen, while Haecht is the second most likely candidate. The distance of the *Wilhelmus* to the

[5] For *Wilhelmus* dataset, we explore the number of all dimensions of dimensionality reduction methods

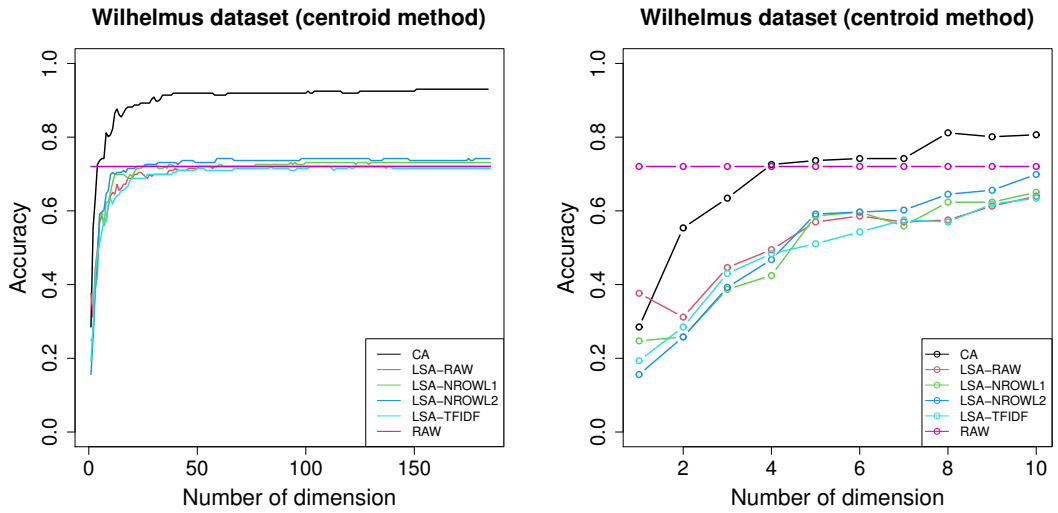


Figure 2.10: Accuracy versus the number of dimensions (centroid method) for CA, RAW, LSA-RAW, LSA-NROWL1, LSA-NROWL2, and LSA-TFIDF with *Wilhelmus* dataset.

centroid of documents of Datheen averaged across 34 optimal dimensions is 0.825, to Haecht 0.880, to Marnix 0.939, to Heere 1.015, to Fruytiers 1.064, and to Coornhert 1.253. Thus, CA attributes *Wilhelmus* to Datheen, and provides more weight using an independent statistical technique, to prior results by Kestemont et al. (2017a, 2017b) in resolving this debate.

2.7 Conclusion

LSA and CA both allow for dimensionality reduction by the SVD of a matrix; however, the actual matrix analyzed by LSA and CA is different, and therefore LSA and CA capture different kinds of information. In LSA we apply an SVD to $\boldsymbol{F}$, or to a weighted $\boldsymbol{F}$. In CA, an SVD is applied to the matrix $\boldsymbol{D}_r^{-\frac{1}{2}}(\boldsymbol{P}-\boldsymbol{E})\boldsymbol{D}_c^{-\frac{1}{2}}$ of standardized residuals. The elements in $\boldsymbol{D}_r^{-\frac{1}{2}}(\boldsymbol{P}-\boldsymbol{E})\boldsymbol{D}_c^{-\frac{1}{2}}$ display the departure from the margins, that is, departure from the expected frequencies under independence collected in $\boldsymbol{E}$. Due to $\boldsymbol{E}$, in CA the effect of the margins is eliminated — a solution only displays the dependence between documents and terms. Concluding, in LSA, the effect of the margins as well as the dependence is part of the matrix that is analyzed and these margins usually play a dominant role in the first dimension of the LSA solution as usually on the first dimension all points depart in the same direction from the origin. On the other hand, in CA all points are scattered around the origin and the origin represents the profile of the row and column margins of $\boldsymbol{F}$.

In summary, although LSA allows a study of the relations between documents, between terms, and between documents and terms, this study is not easy. The reason

is that these relations are blurred by the effect of the margins that are also displayed in the LSA solution. CA does not have this property. Therefore it appears that CA is a better tool for studying the relations between documents, between terms, and between documents and terms. Also, discussed in Section 2.3, CA has many nice properties like providing a geometric display where the Euclidean distances approximate the χ^2-distances between the rows and between the columns of the matrix, and the relation to the Pearson χ^2 statistic. Overall, from a theoretical point of view it appears that CA has more attractive properties than LSA. Empirically, we evaluated and compared the two methods on text categorization in English and authorship attribution in Dutch, and found that CA can both separate documents better visually, and obtain higher accuracies on text categorization and authorship attribution as compared to LSA techniques.

A document-term matrix is similar to a word-context matrix, commonly used to represent word meanings, in the sense that it is also a matrix of counts. However, in the context of word-context matrices the ways in which the counts are transformed are usually different from the way they are transformed for document-term matrices, and therefore, due to space limitations, we defer a comparison of CA and LSA of word-context matrices to future work. In the future, it is also interesting to compare word embeddings learned by LSA based methods and CA to more recent static word embedding approaches such as Word2Vec and GloVe, or even contextualized word embeddings models like BERT. And it is interesting to compare LSA based methods and CA on recent classifiers, such as neural network models.

Chapter 3

Improving information retrieval through correspondence analysis instead of latent semantic analysis

Abstract

The initial dimensions extracted by latent semantic analysis (LSA) of a document-term matrix have been shown to mainly display marginal effects, which are irrelevant for information retrieval. To improve the performance of LSA, usually the elements of the raw document-term matrix are weighted and the weighting exponent of singular values can be adjusted. An alternative information retrieval technique that ignores the marginal effects is correspondence analysis (CA). In this paper, the information retrieval performance of LSA and CA is empirically compared. Moreover, it is explored whether the two weightings also improve the performance of CA. The results for four empirical datasets show that CA always performs better than LSA. Weighting the elements of the raw data matrix can improve CA; however, it is data dependent and the improvement is small. Adjusting the singular value weighting exponent often improves the performance of CA; however, the extent of the improvement depends on the dataset and the number of dimensions.

This chapter is published in Journal of Intelligent Information Systems as: Qi, Q., Hessen, D. J., & Van der Heijden, P. G. M. (2024). Improving information retrieval through correspondence analysis instead of latent semantic analysis. Journal of Intelligent Information Systems 62, 209–230. https://doi.org/10.1007/s10844-023-00815-y. Author contributions: QQ posed the problem and set up the experiments. QQ, DH, and PvdH discussed and edited the text. The code used in this study can be found at https://github.com/qianqianqi28/calsa-ir.

3.1 Introduction

In information retrieval, the similarity between a given user query and each document in a document-term matrix is calculated and documents with high similarity are returned (Kolda & O'leary, 1998; W. Zhang et al., 2011; Al-Qahtani, Amira, & Ramzan, 2015; J. Guo et al., 2022). Latent semantic analysis (LSA) has been used as a common baseline for information retrieval (Parali, Zontul, & Ertuğrul, 2019; Duan, Gao, Ni, & Wang, 2021; Chang, Lee, Wu, Liu, & Liu, 2021). Compared to Word2Vec (Skip-Gram model) LSA showed a better performance in extracting relevant semantic patterns in dream reports (Altszyler, Sigman, Ribeiro, & Slezak, 2016). LSA also outperformed neural network methods (such as ELMo word embeddings) in text classification tasks for educational data (Phillips et al., 2021).

New methods that rely on LSA have been proposed (Azmi, Al-Jouie, & Hussain, 2019; Gupta & Patel, 2021; Hassani et al., 2021; Suleman & Korkontzelos, 2021; Horasan, 2022; Patil, 2022). For example, Gupta and Patel (2021) proposed an algorithm for text summarization that uses LSA, TF-IDF keyword extractor, and BERT encoder model. The algorithm performed better than latent Dirichlet allocation. Horasan (2022) proposed a collaborative filtering-based recommendation system using LSA and achieved good performance. Patil (2022) developed a new promising procedure for information retrieval using LSA and TF-IDF.

Weighting the elements of the raw document-term matrix is a common and effective method to improve the performance of LSA (Dumais, 1991; Horasan, Erbay, Varçın, & Deniz, 2019; Bacciu et al., 2019). LSA usually involves the SVD of a raw or pre-processed document-term matrix. In addition, Caron (2001) proposed changing the weighting exponent of the singular values in LSA to improve information retrieval. His results showed that adjusting the weighting exponent of singular values improves the performance of information retrieval. Since Caron (2001), singular value weighting exponents have been studied and applied in word embeddings generated from word-context matrices (Bullinaria & Levy, 2012; Österlund, Ödling, & Sahlgren, 2015; Drozd, Gladkova, & Matsuoka, 2016; Yin & Shen, 2018). Other variants that change the singular value weighting exponent have been studied in word embeddings created by Word2Vec and GloVe (Mu & Viswanath, 2018; Liu, Ungar, & Sedoc, 2019).

The larger the weighting exponent of the singular values, the higher is the emphasis given to the initial dimensions. According to the experimental results of Caron (2001), giving more emphasis to initial dimensions can often improve the performance of information retrieval on standard test datasets, whereas giving more emphasis to initial dimensions can decrease the performance on question/answer matching. Papers about word embeddings tend to reduce the contribution of initial dimensions to improve performance (Bullinaria & Levy, 2012; Österlund et al., 2015; Drozd et al., 2016; Yin & Shen, 2018; Mu & Viswanath, 2018; Liu et al., 2019), although the optimal value of the singular value weighting exponent is task dependent (Österlund et al., 2015). Bullinaria and Levy (2012) reported that assigning less weight to initial dimensions leads to improved performance for TOEFL, distance comparison, semantic

categorization, and clustering purity tasks on a word-context matrix created from the ukWaC corpus (Baroni, Bernardini, Ferraresi, & Zanchetta, 2009). They argued that the general pattern appears to be that the initial dimensions tend not to contribute the most useful information about semantics and have a large "noise" component that is best removed or reduced.

Capturing associations between documents and terms appears necessary for the success of LSA in computing science; however, the solution of LSA is a mix of the associations between documents and terms, and marginal effects arising from the lengths of documents and marginal frequencies of terms (Qi, Hessen, Deoskar, & Van der Heijden, 2023). Hu et al. (2003) and Qi et al. (2023) showed that margins play an important role in the first dimensions extracted by LSA.

Correspondence analysis (CA) is another information retrieval technique that uses SVD (Greenacre, 1984; Morin, 2004; Greenacre, 2017; Beh & Lombardo, 2021). In computing science, CA has not been explored as much as LSA. CA is usually used to make two-dimensional graphical displays (Hou & Huang, 2020; Arenas-Márquez et al., 2021; Van Dam et al., 2021). For example, Arenas-Márquez et al. (2021) depicted a biplot using CA to show that the document encoding of convolutional neural encoder can emphasize the dissimilarity between documents belonging to different classes. Unlike LSA, CA ignores the information on marginal frequency differences between documents and between terms from the solution by preprocessing the data, and it only focuses on the relationships between documents and terms (Qi et al., 2023). Thus, CA seems more suitable for information retrieval.

Séguéla and Saporta (2011) and Qi et al. (2023) experimentally compared LSA and CA for text clustering and text categorization, respectively, and they found that CA performed better than LSA. Although LSA was originally proposed for information retrieval, an empirical comparison between LSA and CA continues to remain lacking in this field. In this paper, therefore, three English datasets and one Dutch dataset are used to compare the performance of LSA and CA in information retrieval.

Whereas LSA owes its popularity to its applicability to different matrices, in CA, it is unusual to weight the elements of the raw document-term matrix. Processing the raw document-term matrix is an integral part of CA (Greenacre, 1984, 2017; Beh & Lombardo, 2021). CA is based on the SVD of the matrix of standardized residuals. Here, however, we study the CA of document-term matrices whose entries are weighted to see if this has an impact on the performance of CA. In addition, based on the success of adjusting the weighting exponent of singular values in LSA, we will explore whether this is also successful in CA.

In summary, this work makes three contributions. First, to compare LSA and CA in information retrieval. Second, to explore whether weightings, including the weighting of the elements of the raw document-term matrix and the adjusting of the singular value weighting exponent, can improve the performance of CA. Third, to study what the initial dimensions of LSA correspond to and whether CA is effective in ignoring the useless information in the raw or pre-processed document-term matrix that contributes a large part of the initial dimensions extracted by LSA. We extensively

compare the performances of LSA and CA applied to four datasets using Euclidean distance, dot similarity, and cosine similarity.

The paper is organized as follows. In Section 3.2, LSA and CA are described in brief. Section 3.3 presents the methodology used in this paper. The results for Euclidean distance are presented in Section 3.4, and the results for dot similarity and cosine similarity are presented in Section 3.5. Finally, Section 3.6 concludes and discusses the results.

3.2 LSA and CA

In this section, we briefly describe LSA and CA. We refer the readers to Qi et al. (2023) for a more detailed presentation of the methods.

3.2.1 LSA

Consider a raw document-term matrix $\boldsymbol{F} = [f_{ij}]$ with m rows ($i = 1,...,m$) and n columns ($j = 1,...,n$), where the rows represent documents and the columns represent terms. Weighting might be used to prevent the differential lengths of documents from considerably affecting the representation, or to impose certain preconceptions about which terms are more important (Deerwester et al., 1990). The weighted element a_{ij} for term j in document i is

$$a_{ij} = L(i,j) \times G(j) \times N(i), \tag{3.1}$$

where the local weighting term $L(i,j)$ is the weight of term j in document i, $G(j)$ is the global weight of term j in the entire set of documents, and $N(i)$ is the weighting component for document i. The popular TF-IDF can be written in the form $L(i,j) = f_{ij}, G(j) = 1 + \log_2(n\text{docs}/df_j), N(i) = 1$, where $n$docs is the number of documents in the set and df_j is the number of documents where term j appears (Dumais, 1991). The SVD of $\boldsymbol{A} = [a_{ij}]$ is

$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T \tag{3.2}$$

where $\boldsymbol{U}^T\boldsymbol{U} = \mathbf{I}$, $\boldsymbol{V}^T\boldsymbol{V} = \mathbf{I}$, and $\boldsymbol{\Sigma}$ is a diagonal matrix with singular values on the diagonal in the descending order. We denote matrices that contain the first k columns of $\boldsymbol{U}$, first k columns of $\boldsymbol{V}$, and k largest singular values of $\boldsymbol{\Sigma}$ by $\boldsymbol{U}_k$, $\boldsymbol{V}_k$, and $\boldsymbol{\Sigma}_k$, respectively. Then, $\boldsymbol{U}_k\boldsymbol{\Sigma}_k(\boldsymbol{V}_k)^T$ provides the optimal rank-k approximation of $\boldsymbol{A}$ in a least-squares sense, which shows that SVD can be used for data reduction. In LSA, the rows of $\boldsymbol{U}_k\boldsymbol{\Sigma}_k$ and $\boldsymbol{V}_k\boldsymbol{\Sigma}_k$ provide the coordinates of row and column points, respectively. Euclidean distances between the rows of $\boldsymbol{U}_k\boldsymbol{\Sigma}_k$ ($\boldsymbol{V}_k\boldsymbol{\Sigma}_k$) approximate those between the rows (columns) of $\boldsymbol{A}$.

Representing out-of-sample documents or queries in the k-dimensional subspace of LSA is important for many applications including information retrieval. Suppose

that the new weighted document is a row vector $\boldsymbol{d}$. Since $\boldsymbol{V}^T\boldsymbol{V} = \mathbf{I}$ and $\boldsymbol{U}^T\boldsymbol{U} = \mathbf{I}$, we have

$$\boldsymbol{A}\boldsymbol{V}_k = \boldsymbol{U}_k\Sigma_k \tag{3.3}$$

and

$$\boldsymbol{A}^T\boldsymbol{U}_k = \boldsymbol{V}_k\Sigma_k \tag{3.4}$$

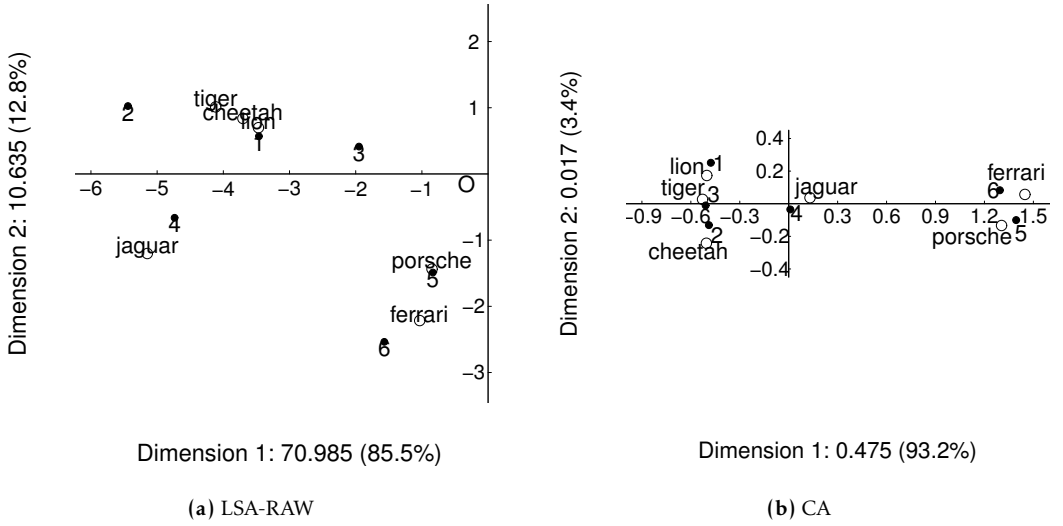
Therefore, using Equation (3.3), the coordinates of the out-of-sample document $\boldsymbol{d}$ in the k-dimensional subspace of LSA is $\boldsymbol{d}\boldsymbol{V}_k$. Similarly, using Equation (3.4), the coordinates of the out-of-sample term $\boldsymbol{t}$ (represented as row vector) in the k-dimensional subspace of LSA is $\boldsymbol{t}\boldsymbol{U}_k$.

As in Qi et al. (2023), we first use a small dataset to illustrate LSA. This small dataset is introduced in Aggarwal (2018) (see Table 3.1), and it contains 6 documents. For each document, we are interested in the frequency of occurrence of six terms. The first three documents primarily refer to cats, the last two primarily to cars, and the fourth to both. The fourth term, jaguar, is polysemous because it can refer to either a cat or a car.

Table 3.1: A document-term matrix $\boldsymbol{F}$: size 6×6

	lion	tiger	cheetah	jaguar	porsche	ferrari
doc1	2	2	1	2	0	0
doc2	2	3	3	3	0	0
doc3	1	1	1	1	0	0
doc4	2	2	2	3	1	1
doc5	0	0	0	1	1	1
doc6	0	0	0	2	1	2

In the LSA of the raw document-term matrix (LSA-RAW), the rows and columns of $\boldsymbol{F}$ are not weighted, and therefore, we can replace $\mathbf{A}$ in Equation (3.2) by $\boldsymbol{F}$. The coordinates of the documents and of the terms for LSA-RAW in the first two dimensions are $\boldsymbol{U}_2\Sigma_2$ and $\boldsymbol{V}_2\Sigma_2$, respectively. Figure 3.1a shows the two-dimensional plot of the documents and terms. Cat terms (*lion*, *cheetah*, and *tiger*) are close together; car terms (*porsche* and *ferrari*) are close together; car documents (5 and 6) are close together. However, the cat documents (1, 2, and 3) are not close together, neither is document 4 in between cat documents and car documents, and neither is *jaguar* in between cat terms and car terms. This can be attributed to the fact that LSA displays both the relationships between documents and terms and the sizes of the documents and terms: for the latter, *jaguar*, for example, is used most often in the documents and is furthest away from the origin.



(a) LSA-RAW **(b)** CA

Figure 3.1: A two-dimensional plot of documents and terms for (a) LSA-RAW, (b) CA (Qi et al., 2023).

3.2.2 CA

In CA, an SVD is applied to the matrix of standarized residuals given by Greenacre (2017)

$$\boldsymbol{S} = \boldsymbol{D}_r^{-\frac{1}{2}}(\boldsymbol{P}-\boldsymbol{E})\boldsymbol{D}_c^{-\frac{1}{2}} \tag{3.5}$$

where $\boldsymbol{P} = [p_{ij}]$ is the matrix of joint observed proportions with $p_{ij} = f_{ij}/\sum_i \sum_j f_{ij}$, $\boldsymbol{D}_r$ is a diagonal matrix with $r_i = \sum_j p_{ij}$ $(i = 1, 2, \cdots, m)$ on the diagonal, $\boldsymbol{D}_c$ is a diagonal matrix with $c_j = \sum_i p_{ij}$ $(j = 1, 2, \cdots, n)$ on the diagonal, and $\boldsymbol{E} = [r_i c_j]$ is the matrix of expected proportions under the statistical independence of the documents and the terms. The elements of $\boldsymbol{D}_r^{-\frac{1}{2}}(\boldsymbol{P}-\boldsymbol{E})\boldsymbol{D}_c^{-\frac{1}{2}}$ are standardized residuals under the statistical independence model. The sum of squares of these elements yields the total inertia, i.e., the Pearson χ^2 statistic divided by sample size $\sum_i \sum_j f_{ij}$. By taking the SVD of the matrix of standardized residuals, we get

$$\boldsymbol{D}_r^{-\frac{1}{2}}(\boldsymbol{P}-\boldsymbol{E})\boldsymbol{D}_c^{-\frac{1}{2}} = \boldsymbol{U}\Sigma\boldsymbol{V}^T \tag{3.6}$$

In CA, the rows of $\boldsymbol{\Phi}_k\Sigma_k$ and $\boldsymbol{\Gamma}_k\Sigma_k$ provide the coordinates of row and column points, respectively, where $\boldsymbol{\Phi}_k = \boldsymbol{D}_r^{-\frac{1}{2}}\boldsymbol{U}_k$ and $\boldsymbol{\Gamma}_k = \boldsymbol{D}_c^{-\frac{1}{2}}\boldsymbol{V}_k$. The weighted sum of the coordinates is 0: $\sum_i r_i\phi_{ik} = 0 = \sum_j c_j\gamma_{jk}$. Euclidean distances between the rows of $\boldsymbol{\Phi}_k\Sigma_k$ ($\boldsymbol{\Gamma}_k\Sigma_k$) approximate χ^2-distances between the rows (columns) of $\boldsymbol{F}$, where the squared χ^2-distance between rows k and l is

$$\delta_{kl}^2 = \sum_j \frac{\left(p_{kj}/r_k - p_{lj}/r_l\right)^2}{c_j} \tag{3.7}$$

In Equation (3.7), the rows are transformed into vectors of conditional proportions adding up to 1 for each row, such as the kth row: p_{kj}/r_k, $j = 1, 2, \cdots, n$, and the differences between the column elements for column j in the transformed rows are corrected for c_j, which represents the size of column j.

The transition formulas are

$$\boldsymbol{D}_r^{-1}\boldsymbol{P}\boldsymbol{\Gamma}_k = \boldsymbol{\Phi}_k\Sigma_k \tag{3.8}$$

and

$$\boldsymbol{D}_c^{-1}\boldsymbol{P}^T\boldsymbol{\Phi}_k = \boldsymbol{\Gamma}_k\Sigma_k \tag{3.9}$$

Equation (3.8) shows that the row points are in the weighted averages of the column points when rows of $\boldsymbol{D}_r^{-1}\boldsymbol{P}$ are used as weights, and Equation (3.9) shows that the column points are in the weighted averages of the row points simultaneously.

According to Equation (3.8), a new document $\boldsymbol{d}$, represented by a row vector, can be projected onto the k-dimensional subspace by placing it in the weighted average of the column points using $(\boldsymbol{d}/\sum_{j=1}^{n} d_j)\boldsymbol{\Gamma}_k$. This can be similarly done for a new term $\boldsymbol{t}$.

For the CA of Table 3.1, the coordinates of the documents and terms for CA in the first two dimensions are $\boldsymbol{\Phi}_2\Sigma_2$ and $\boldsymbol{\Gamma}_2\Sigma_2$, respectively. Figure 3.1b shows a two-dimensional plot of the documents and terms. Cat terms (*lion*, *cheetah*, and *tiger*) are close together; car terms (*porsche* and *ferrari*) are close together; *jaguar* is in between cat and car terms; car documents (5 and 6) are close together, cat documents (1, 2, and 3) are close together; and document 4 is in between cat and car documents. All data properties are found in Figure 3.1b. A comparison of Figures 3.1b and 3.1a suggests that CA provides a clearer visualization of the important aspects of the data than LSA. This is because the coordinates of each dimension are orthogonal to the margins due to $\sum_i r_i\phi_{ik} = 0 = \sum_j c_j\gamma_{jk}$, and CA focuses only on the relationship between the documents and the terms.

3.3 Methodology

In this section, we introduce the CA of a document-term matrix whose entries are weighted. We also discuss how the influence of the initial dimensions can be studied. Subsequently, we describe the study design, datasets, and evaluation methods used.

3.3.1 CA of a document-term matrix of weighted frequencies

Weighting the entries of the raw document-term matrix is an effective method for improving the performance of LSA, and this motivates us to study the weighting of the elements of the input matrix of CA. So, we try to improve the performance of CA by using the same weighting methods as in LSA.

The processing of the raw data matrix by $\boldsymbol{D}_r^{-\frac{1}{2}}(\boldsymbol{P}-\boldsymbol{E})\boldsymbol{D}_c^{-\frac{1}{2}}$ (see Equation (3.5)) is considered an integral part of CA. This processing step effectively eliminates the margins, which allows CA to focus on the relationships between documents and terms.

The weighting of the entries of the raw document-term matrix in Equation (3.1), such as by TF-IDF, can be used to assign higher values to terms with more indicative of the meaning of documents. Thus, the weighting of the entries of the raw document-term matrix may also be an effective method for improving the performance of CA.

To perform the CA of a document-term matrix of weighted frequencies, we first use Equation (3.1) to obtain a document-term matrix $\boldsymbol{A}$ of weighted frequencies, and then, we perform CA on this matrix $\boldsymbol{A}$ instead of $\boldsymbol{F}$.

3.3.2 Changing the contributions of the initial dimensions in SVD

Caron (2001) proposed adjusting the relative strengths of vector components in LSA using $\boldsymbol{U}_k\Sigma_k^\alpha$ or $\boldsymbol{V}_k\Sigma_k^\alpha$ as coordinates instead of $\boldsymbol{U}_k\Sigma_k$ or $\boldsymbol{V}_k\Sigma_k$, where α is the singular value weighting exponent that adjusts the importance of the dimensions. The weighting exponent α determines how components are weighted relative to the standard $\alpha = 1$ case described in Section 3.2.1. In comparison to $\alpha = 1$, $\alpha < 1$ gives less emphasis to initial dimensions, and $\alpha > 1$, more emphasis.

Bullinaria and Levy (2012) used both weighting exponent $\alpha < 1$ and the exclusion of initial dimensions, which led to performance improvements of a similar degree. They argued that the general pattern appears to be that the dimensions with the highest singular values tend not to contribute the most useful information about semantics and have a large "noise" component that is best removed or reduced. However, it is unclear what the initial dimensions actually correspond to. Given this context, we change the contributions of the initial dimensions extracted by both LSA and CA and compare their performances. We explore whether the performance of CA can be improved by adjusting the singular value weighting exponent using $\boldsymbol{\Phi}_k\Sigma_k^\alpha$ or $\boldsymbol{\Gamma}_k\Sigma_k^\alpha$ as coordinates instead of $\boldsymbol{\Phi}_k\Sigma_k$ or $\boldsymbol{\Gamma}_k\Sigma_k$. That is, we try to improve the performance of CA by using the method (adjusting the singular weighting exponent) used in LSA.

We use Table 3.1 to illustrate the impact of α on singular values and coordinates. We use $\alpha = 0.5$, $\alpha = 1$, and $\alpha = 1.5$. In the literature, we regularly encounter $\alpha = 0.5$ because it relates to

$$\boldsymbol{F} = \boldsymbol{U}\Sigma\boldsymbol{V}^T = \left(\boldsymbol{U}\Sigma^{1/2}\right)\left(\Sigma^{1/2}\boldsymbol{V}^T\right) \tag{3.10}$$

which can then be used for making biplots (Gabriel, 1971) using coordinate pairs $\boldsymbol{U}_2\Sigma_2^{1/2}$ and $\boldsymbol{V}_2\Sigma_2^{1/2}$. In practice, one often sees the use of the coordinate pair $\boldsymbol{U}_2\Sigma_2$ and $\boldsymbol{V}_2\Sigma_2$; however, this is not a biplot representation as Σ_2 is used twice. In a biplot, if the row points are $\boldsymbol{U}_2\Sigma_2^a$, then the column points are $\boldsymbol{V}_2\Sigma_2^{1-a}$, i.e., any entry of the matrix is approximated by the inner product of the corresponding row and column vectors. Hereafter, we do not make a biplot; instead, we make a symmetric plot where documents and terms have the same value of α because symmetric coordinates are usually used in experiments (Dumais et al., 1988; Deerwester et al., 1990; Berry et al., 1995; Levy et al., 2015).

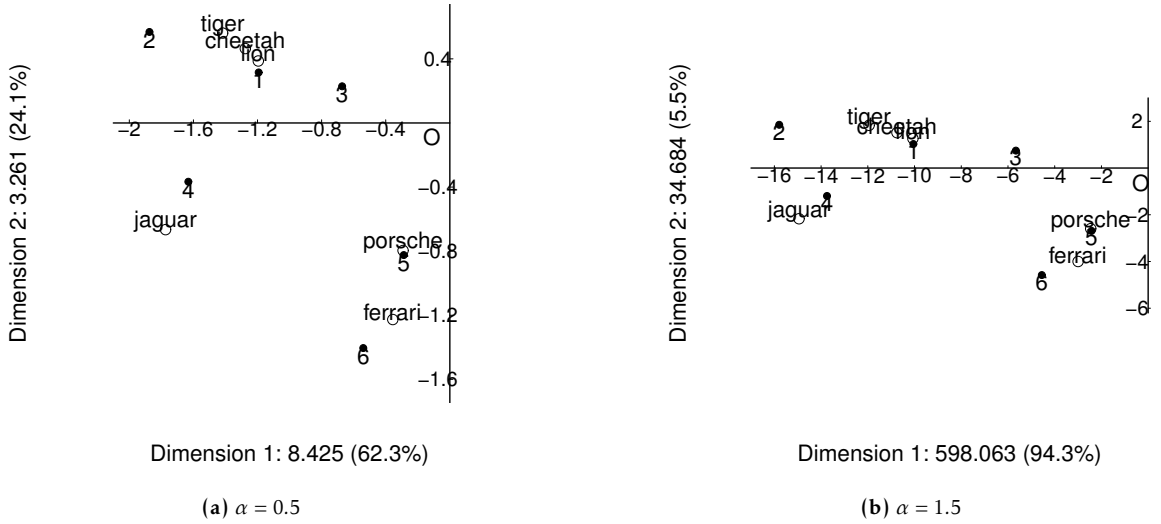
Table 3.2 lists the singular values to the power α: σ^α, the squared singular values to the power α: $\sigma^{2\alpha}$, and proportions $\sigma^{2\alpha}/\sum_\sigma \sigma^{2\alpha}$, where we refer to the total sum of squared singular values to the power of α, $\sum_\sigma \sigma^{2\alpha}$, as α–inertia. These proportions

show how the sum of the Euclidean distances of all components to the origin is distributed over the components. The greater α is, the more emphasis is given to the initial components and less emphasis to the latter ones. The first dimension accounts for 0.623, 0.855, and 0.943 of α-inertia, while the fifth dimension accounts for 0.020, 0.001, and 0.000, with α being 0.5, 1, and 1.5, respectively. The standard LSA solution has $\alpha = 1$.

Table 3.2: The σ^{α}, $\sigma^{2\alpha}$, and the proportion of explained α-inertia $\sigma^{2\alpha}/\sum_{\sigma}\sigma^{2\alpha}$ for each dimension of LSA-RAW.

	dim1	dim2	dim3	dim4	dim5
$\sigma^{0.5}$	2.903	1.806	0.994	0.758	0.522
σ^{1}	8.425	3.261	0.988	0.574	0.272
$\sigma^{1}/\sum_{\sigma}\sigma^{1}$	0.623	0.241	0.073	0.042	0.020
σ^{1}	8.425	3.261	0.988	0.574	0.272
σ^{2}	70.985	10.635	0.976	0.330	0.074
$\sigma^{2}/\sum_{\sigma}\sigma^{2}$	0.855	0.128	0.012	0.004	0.001
$\sigma^{1.5}$	24.455	5.889	0.982	0.435	0.142
σ^{3}	598.063	34.684	0.964	0.189	0.020
$\sigma^{3}/\sum_{\sigma}\sigma^{3}$	0.943	0.055	0.002	0.000	0.000

Figure 3.2 shows the two-dimensional plots of documents and terms for LSA-RAW with $\alpha = 0.5, 1.5$. The standard coordinates with $\alpha = 1$ was shown in Figure 3.1a. As α increases, the Euclidean distances between row points (column points) on the first dimension increase relative to the second dimension.



(a) $\alpha = 0.5$ **(b)** $\alpha = 1.5$

Figure 3.2: A two-dimensional plot of documents and terms for LSA-RAW with (a) $\alpha = 0.5$ and (b) $\alpha = 1.5$.

3.3.3 Design

We compare the performances of LSA and CA for information retrieval, where two kinds of weightings are studied in LSA: the elements of the raw document-term matrix are weighted and the weighting exponent α is varied. We also explore the impact of these weightings in CA. We vary the number of dimension k from 1, 2, $\cdots$, 20, 22, $\cdots$, 50, 60, $\cdots$ to 100 and the value of α from -6, -5.5, $\cdots$, -2, -1.8, $\cdots$, 4, 4.5, $\cdots$ to 8; we explore all $40 \times 47 = 1,880$ combinations of parameter values.

In the study of weighting the elements of the raw document-term matrix, we perform the LSA and CA of

- raw matrix $\boldsymbol{F}$, denoted by RAW,
- L1 row-normalized matrix $\boldsymbol{F}^{L1}$ with $L(i,j) = f_{ij}$, $G(j) = 1$, and $N(i) = 1/\sum_{j=1}^{n} f_{ij}$, NROWL1,
- L2 row-normalized matrix $\boldsymbol{F}^{L2}$ with $L(i,j) = f_{ij}$, $G(j) = 1$, and $N(i) = 1/\sqrt{\sum_{j=1}^{n} f_{ij}^2}$, NROWL2, and
- TF-IDF matrix $\boldsymbol{F}^{\text{TF-IDF}}$ described in Section 3.2.1, TFIDF.

We refer to the combination of the CA and TF-IDF matrix as CA-TFIDF. Similarly, we obtain LSA-RAW, LSA-NROWL1, LSA-NROWL2, LSA-TFIDF, CA-RAW, CA-NROWL1, and CA-NROWL2. For performance comparison, RAW is used for term matchings without dimensionality reduction.

3.3.4 Datasets

LSA and CA are compared using three English datasets and one Dutch dataset. The three English datasets are the BBCSport (Greene & Cunningham, 2006), BBCNews (Greene & Cunningham, 2006), and 20 Newsgroups datasets (20-news-18846 bydata version) (Rennie, 2005). The Dutch dataset is the *Wilhelmus* dataset (Kestemont, 2017). The three English datasets have recently been used in information retrieval studies (Bounabi, Moutaouakil, & Satori, 2019; Bianco, Duarte, & Gonçalves, 2023). The *Wilhelmus* dataset is produced for studying authorship attribution of the song *Wilhelmus*, which is the national anthem of the Netherlands. The author of the song is unknown.

Some statistics of the four datasets used are presented in Table 3.3. The BBCNews dataset includes 2,225 documents that fall into one of five categories. The BBCSport dataset includes 737 documents that fall into one of five categories. The 20 Newsgroups dataset includes 18,846 documents that fall into one of 20 categories. This dataset is sorted into a training (60%) and a test (40%) set. We use a subset of this dataset to evaluate information retrieval. We randomly choose 600 documents from the training set of four categories (comp.graphics, rec.sport.hockey, sci.crypt,

Table 3.3: Characteristics of datasets.

Categories	Data
business	510
entertainment	386
politics	417
sport	511
technology	401

(a) BBCNews dataset.

Categories	Data
athletics	101
cricket	124
football	265
rugby	147
tennis	100

(b) BBCSport dataset.

Categories	Training data	Test data
comp.graphics	141	100
rec.sport.hockey	164	99
sci.crypt	161	106
talk.politics.guns	134	95

(c) 20 Newsgroups dataset.

Categories	Data
datheen	35
marnix	46
heere	23
haecht	35
fruytiers	33
coornhert	14

(d) Wilhelmus dataset.

and talk.politics.guns) and 400 documents from the test set of these four categories. The *Wilhelmus* dataset includes 186 documents divided into six categories.

To pre-process the three English datasets, we change all characters to lower case, remove punctuation marks, numbers, and stop words, and apply lemmatization. Subsequently, terms with frequencies lower than 10 are ignored. In addition, we remove unwanted parts of the 20 Newsgroups dataset, such as the header (including fields like "From:" and "Reply-To:" followed by email address), because these are almost irrelevant for information retrieval. The Dutch *Wilhelmus* dataset is already pre-processed into tag-lemma pairs. Following Kestemont (2017) and Qi et al. (2023), in *Wilhelmus* dataset, we use the 300 most frequent tag-lemma pairs.

Since the *Wilhelmus* and BBCSport datasets have a relatively low number of documents, we use leave-one-out cross-validation (LOOCV) for the *Wilhelmus* dataset and five-fold cross-validation for the BBCSport dataset to evaluate LSA and CA (Gareth et al., 2021). The BBCNews dataset is randomly divided into training (80%) and validation (20%) sets.

In the information retrieval part of the study, each document in the validation set is used as a query, where the category of the document is known. The documents in the training set that fall in the same category as the query are the relevant documents for this query.

3.3.5 Evaluation

We compare the MAP of each of the four versions of LSA and CA to explore the performance of these methods in information retrieval under changes in the contributions of initial dimensions (Kolda & O'leary, 1998). The MAP is calculated as follows:

- The similarity is assessed between a query vector and each document vector of a document collection. We use three similarity metrics: Euclidean distance, dot similarity, and cosine similarity. As Euclidean distance is a key motivation for CA, we report results on Euclidean distance, and only report partial results for dot and cosine similarity in the main paper and the other results in the supplementary materials.
- For Euclidean distance, the documents are ranked in an increasing order based on their similarity with the query vector (for dot and cosine similarity, the ranking is in the decreasing order); therefore, the first document has the highest similarity.
- Precision-recall points are derived from the ordered list of documents. For a given query, Table 3.4 defines four types of documents in the ordered list based on whether a document is relevant and retrieved:

  $\mathbf{C}$ = the set of relevant documents from the ordered list, i.e., documents that fall in the same category as the query

  $\mathbf{D}$ = the set of retrieved documents from the ordered list., i.e., when 10 documents are returned, the set of retrieved documents consists of the first 10 documents in the ordered list.

Table 3.4: Retrieved and relevant documents.

	Relevant	Non-Relevant
Retrieved	$\mathbf{C} \cap \mathbf{D}$	$\overline{\mathbf{C}} \cap \mathbf{D}$
Not Retrieved	$\mathbf{C} \cap \overline{\mathbf{D}}$	$\overline{\mathbf{C}} \cap \overline{\mathbf{D}}$

Let $|.|$ denote the number of documents in a set. Then, precision and recall are defined as

$$\text{precision} = \frac{|\mathbf{C} \cap \mathbf{D}|}{|\mathbf{D}|} \tag{3.11}$$

and

$$\text{recall} = \frac{|\mathbf{C} \cap \mathbf{D}|}{|\mathbf{C}|}. \tag{3.12}$$

Thus, precision is defined as the ratio of the number of relevant documents retrieved over the total number of retrieved documents, and recall is defined as the ratio of the number of relevant documents retrieved over the total number of relevant documents. For a given query, the set $\mathbf{C}$ is fixed. The set $\mathbf{D}$ is not fixed; if we return the first i documents, then $\mathbf{D}$ consists of the first i documents in the ordered list. Thus, for a given i, we can obtain a precision (see Equation (3.11)) and recall (see Equation (3.12)) pair. We run values of i from 1 to l (the number of documents in the ordered list), and obtain l precision-recall pairs.

- Then, 11 pseudo-precisions are calculated under 11 recalls (0, 0.1, $\cdots$, 1.0), where a pseudo-precision at recall x is the maximum precision from recall x to recall 1. For example, pseudo-precision at recall 0.2 is the maximum precision from recall 0.2 to recall 1.
- The average precision for the query is obtained by averaging the 11 pseudo-precisions.
- The MAP is the mean across all queries.

Greater MAP values indicate a better performance.

3.4 Results for Euclidean distance

3.4.1 Comparing LSA and CA for information retrieval

3.4.1.1 MAP as a function of the number of dimensions for the four versions of LSA with the standard weighting exponent $\alpha = 1$ and for CA

We first investigate the performance of LSA and CA in terms of MAP, in their standard use, i.e., without varying the weighting exponent α, i.e., $\alpha = 1$. Term matching without the preliminary use of LSA and CA, i.e., directly on the document-term matrix, is denoted by RAW. We expect that, in line with Qi et al. (2023), the performance of LSA and CA will be better than that of RAW, and the performance of CA will be better than that of the four versions of LSA.

Figure 3.3 shows MAP as a function of the number of dimensions k for different weighting schemes of LSA, and for CA. We display only the first 20 dimensions, as all lines usually decrease after dimension 20. Figures with dimensionality up to 100 can be found in the supplementary materials. For the four versions of LSA, and for CA, Table 3.5 presents the dimension number for which the optimal MAP is reached, as well as the MAP values, in each of the four datasets. We conclude the following from Figure 3.3 and Table 3.5:

- Both LSA and CA result in better MAP than RAW, which results in a straight line when the full dimensional matrix is used.
- For both LSA and CA, performance is a function of the number of dimensions k. Overall, MAP rises as a function of k to reach a peak, and then, it goes down. For CA, the peak is reached at $k = 4$. In CA, the information used to calculate MAP increases in the first four dimensions in comparison to the noise. In the components of $k \geq 5$, the noise dominates the useful information, which results in the MAP going down from this point.
- CA results in a considerably better MAP than the four versions of LSA: LSA-RAW, LSA-NROWL1, LSA-NROWL2, and LSA-TFIDF, which is in line with Qi et

al. (2023), who showed that the performance of CA is better than that of LSA for document-term matrices. This is because of the differential treatment of margins in LSA and CA. The margins provide irrelevant information for making queries. In CA, the margins are removed, and therefore, the relative amount of information in comparison to the noise, which we informally refer to as the information - noise ratio, is considerably larger in CA than in LSA. This explains the better MAP in CA.

- The peaks for the four versions of LSA are usually found at higher dimensionality k than the peaks for CA. This is because margins are noise for queries when we fix $\alpha = 1$; in LSA, this noise plays an important role in the first few dimensions. Hence, this earlier peak in CA is also explained by its better information - noise ratio.

- The four LSA methods are not equally effective. In all four datasets, the performance of LSA can be significantly improved using weighting schemes. The improvements over LSA-RAW are data dependent. On average, across the four datasets, LSA-NROWL2 is the best, but for the *Wilhelmus* dataset, LSA-NROWL1 and LSA-NROWL2 result in a somewhat worse MAP than that with LSA-RAW.

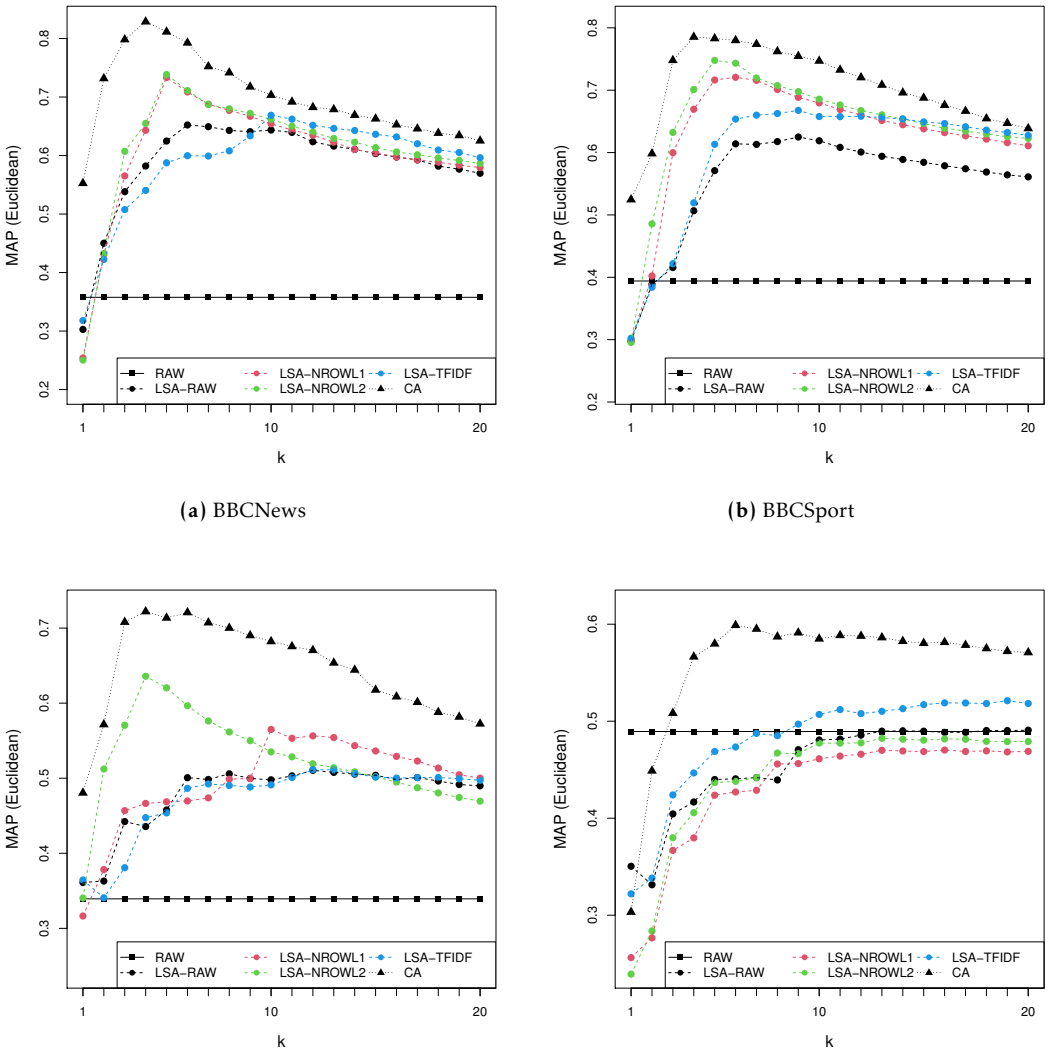
Table 3.5: MAP with the optimal number of dimensions k. Bold values are best.

	BBCNews		BBCSport		20 Newsgroups		Wilhelmus	
	k	MAP	k	MAP	k	MAP	k	MAP
RAW		0.358		0.394		0.339		0.489
LSA-RAW	6	0.652	9	0.625	12	0.510	24	0.492
LSA-NROWL1	5	0.733	6	0.721	10	0.565	16	0.470
LSA-NROWL2	5	0.738	5	0.748	4	0.636	13	0.482
LSA-TFIDF	10	0.669	9	0.668	12	0.512	19	0.521
CA	4	**0.829**	4	**0.785**	4	**0.722**	6	**0.599**

3.4.1.2 MAP as a function of the weighting exponent α for LSA compared with MAP for CA under varying numbers of dimensions

In Section 3.4.1.1, we found that CA outperforms the four versions of LSA in terms of MAP, where LSA had the usual weighting exponent $\alpha = 1$. In this section, we study whether the performance of LSA-RAW improves when we vary α.

Figure 3.4 shows MAP as a function of α for LSA-RAW with the number of dimensions $k = 4, 6, 9, 12,$ and 24. For comparison, we also report the MAP values for CA found in Section 3.4.1.1 under these dimensions. We choose these values of k because these dimensions are optimal for LSA-RAW and CA in Table 3.5. Table 3.6 shows the optimal α and corresponding MAP, which is a condensed version of Figure 3.4. We conclude the following from Figure 3.4 and Table 3.6:



(a) BBCNews

(b) BBCSport

(c) 20 Newsgroups

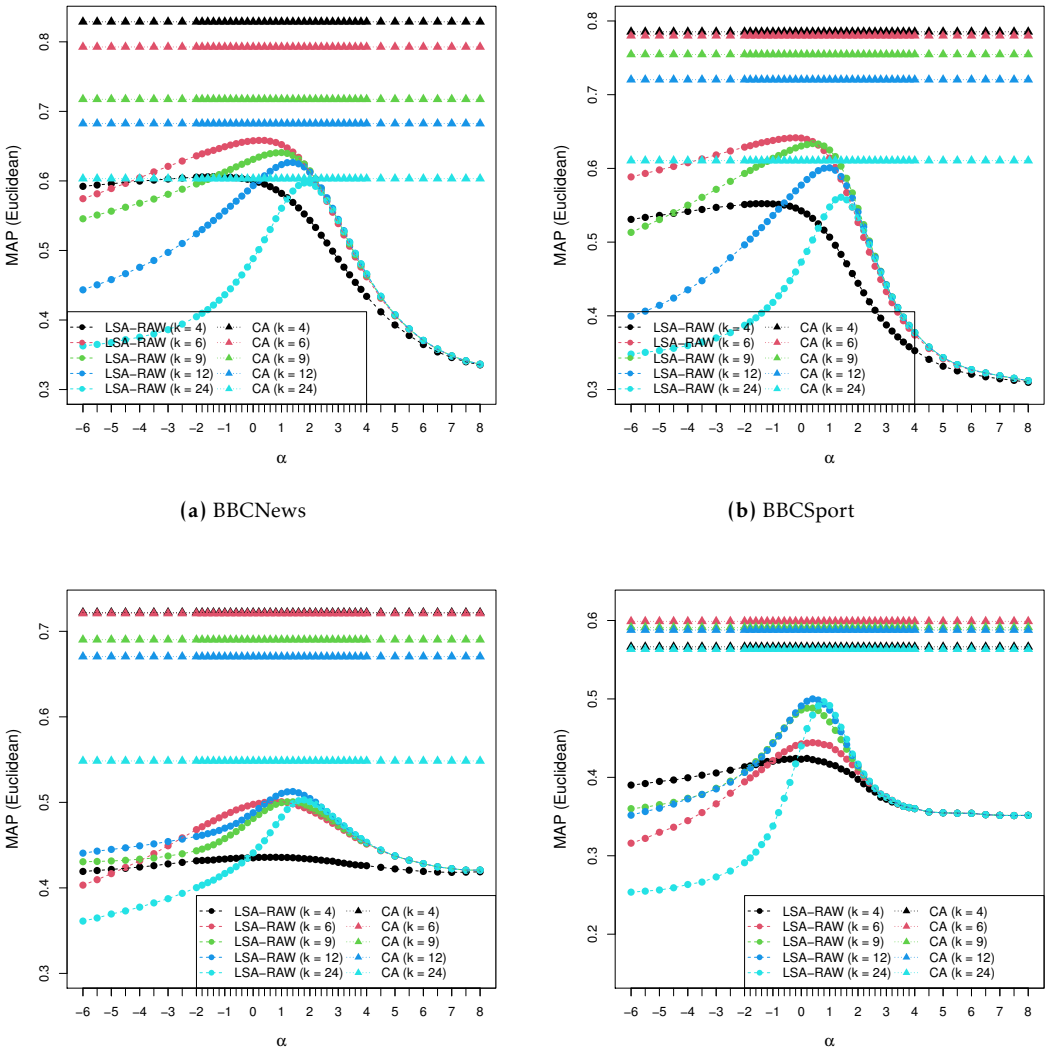
(d) Wilhelmus

Figure 3.3: MAP as a function of the number of dimensions k under standard coordinates.

- Although the performance of LSA-RAW improves by varying α, CA still outperforms LSA-RAW.
- For LSA-RAW, the overall MAP first increases and then decreases as a function of α. This means that varying α can potentially improve the performance of LSA-RAW.
- The increase in MAP is minor. Consider, for example, the BBCNews dataset. In Section 3.4.1.1, we found that the MAP was optimal with a value of 0.652 for $\alpha = 1$, when $k = 6$. Table 3.6 shows that for $\alpha = 0.2$, the MAP increases to 0.658. Apparently, for 6 dimensions, when $\alpha = 0.2$, the information - noise ratio is optimal in terms of MAP. For $\alpha = 0.2$, the distances on later dimensions (of the 6 dimensions) are increased and those on initial dimensions are reduced. This means that, with $\alpha = 0.2$, the impact of the initial dimensions affected most by the margins is reduced. This is consistent with the results of Bullinaria and Levy (2012), which indicates that reducing the initial dimensions improves performance.
- Moreover, the optimal α for LSA-RAW is data dependent and generally increases with k. This replicates results of Caron (2001). As the number of dimensions varies, the change in the optimal α is the result of the information - noise ratio for the specific number of dimensions studied. For example, for the BBCNews dataset, the optimal number of dimensions is 6; for larger numbers of dimensions, the optimal α increases. An increasing α indicates that distances at earlier dimensions are more important for information retrieval, and therefore, the role of the later dimensions is played down.

Table 3.6: MAP with the optimal weighting exponent α for LSA-RAW and MAP for CA under $k = 4, 6, 9, 12,$ and 24. Bold values are best.

	BBCNews		BBCSport		20 Newsgroups		Wilhelmus	
	α	MAP	α	MAP	α	MAP	α	MAP
LSA-RAW ($k = 4$)	-1.4	0.606	-1.4	0.552	0.8	0.436	0.2	0.424
LSA-RAW ($k = 6$)	0.2	0.658	-0.2	0.642	0.8	0.501	0.4	0.444
LSA-RAW ($k = 9$)	1	0.641	0.4	0.634	1.2	0.501	0.4	0.488
LSA-RAW ($k = 12$)	1.4	0.627	1	0.601	1.4	0.513	0.4	0.500
LSA-RAW ($k = 24$)	1.8	0.597	1.4	0.561	1.8	0.503	0.8	0.496
CA ($k = 4$)		**0.829**		**0.785**		**0.722**		0.566
CA ($k = 6$)		0.793		0.780		0.721		**0.599**
CA ($k = 9$)		0.717		0.755		0.690		0.591
CA ($k = 12$)		0.682		0.720		0.670		0.588
CA ($k = 24$)		0.603		0.611		0.548		0.563



(a) BBCNews

(b) BBCSport

(c) 20 Newsgroups

(d) Wilhelmus

Figure 3.4: MAP as a function of α for LSA-RAW and MAP for CA under varying k.

3.4.2 Adjusting CA using weighting

3.4.2.1 Weighting the elements of the raw document-term matrix for CA

Weighting the elements of the raw document-term matrix is an effective way to improve the performance of LSA for information retrieval. Here, we explore whether this holds for CA. Similar to Figure 3.3, Figure 3.5 shows MAP as a function of k for different weighting schemes of CA. CA in Figure 3.3 is referred to as CA-RAW in Figure 3.5; for CA/CA-RAW, the results in these two figures are identical. For the four versions of CA, Table 3.7 shows the dimensionality for which the optimal MAP is reached, as well as the MAP value. We conclude the following from Figure 3.5 and Table 3.7:

- Overall, the weighting of the elements of the raw matrix sometimes improves the performance of CA, but these improvements over CA-RAW are small and data dependent.
- Comparing Table 3.5 with Table 3.7, the performance of CA-NROWL1 is better than that of LSA-NROWL1, the performance of CA-NROWL2 is better than that of LSA-NROWL2, and the performance of CA-TFIDF is better than that of LSA-TFIDF.

Relative to LSA, it is harder to improve the performance of CA in information retrieval by weighting the elements of the raw matrix because (1) the MAP of CA-RAW is already relatively high, and (2) CA-RAW has weighted the elements of the raw document-term matrix as it is an integral part of this technique (Equation (3.5)).

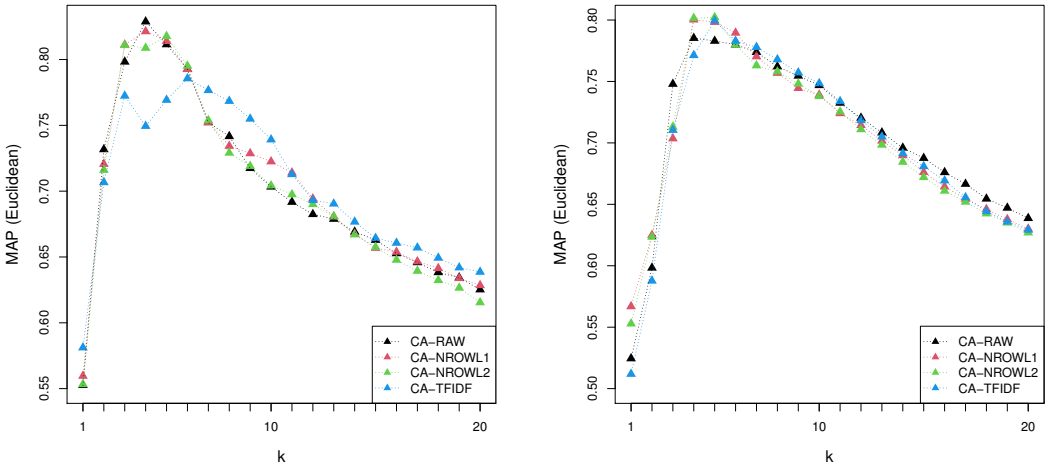
Table 3.7: MAP with the optimal number of dimensions k for the four versions of CA. Bold values are best.

	BBCNews		BBCSport		20 Newsgroups		Wilhelmus	
	k	MAP	k	MAP	k	MAP	k	MAP
CA-RAW	4	**0.829**	4	0.785	4	**0.722**	6	0.599
CA-NROWL1	4	0.821	4	0.800	7	0.631	6	0.603
CA-NROWL2	5	0.818	5	**0.802**	6	0.695	6	0.604
CA-TFIDF	6	0.786	5	0.800	4	0.704	5	**0.618**

3.4.2.2 MAP as a function of the weighting exponent α for CA

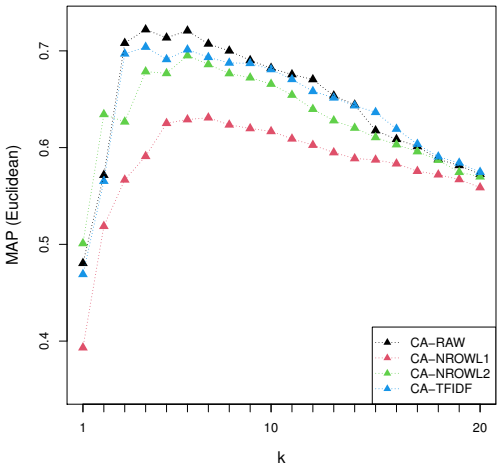
In this section, we introduce CA with weighting exponent α. Similar to Figure 3.4, Figure 3.6 shows MAP as a function of α in CA-RAW for the number of dimensions $k = 4, 6, 9, 12,$ and 24. Table 3.8 shows the optimal α and the corresponding MAP, which is a condensed version of Figure 3.6. We conclude the following from Figure 3.6 and Table 3.8:

- For CA, the overall MAP first increases and then decreases as a function of α. This means that varying α can potentially improve the performance of CA.

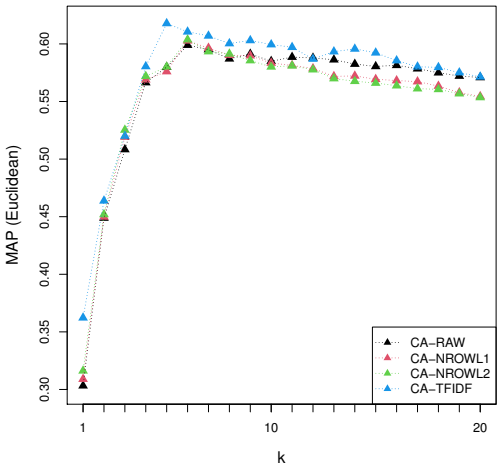


(a) $\alpha = 1$ (BBCNews)

(b) $\alpha = 1$ (BBCSport)



(c) $\alpha = 1$ (20 Newsgroups)



(d) $\alpha = 1$ (Wilhelmus)

Figure 3.5: MAP as a function of the number of dimensions k for the four versions of CA under standard coordinates.

- The increase in MAP by adjusting α is data and dimension dependent.
- If we compare the maxima in Table 3.6 with those in Table 3.8 , there is hardly a noticeable increase.

Now, we check the optimal α like Bullinaria and Levy (2012) did. Comparing Table 3.8 with part LSA-RAW of Table 3.6, the optimal α for CA-RAW is almost always larger than LSA-RAW and is almost always larger than 1. That is, CA-RAW needs a larger α than LSA-RAW to obtain its maximum MAP. Thus, compared to LSA, CA improves by placing more emphasis on its initial dimensions. The important difference between LSA and CA is that LSA involves margins, and CA does not. Therefore, we infer that margins in LSA considerably contribute to the initial dimensions; however, they are irrelevant ("noise") for information retrieval. On the other hand, CA effectively eliminates this irrelevant information.

We study MAP as a function of α under the optimal number of dimensions. The details including tables and figures are in the supplementary materials. Again, CA performs better than LSA. Adjusting α can potentially improve the performance of LSA and CA. Although the optimal α under the optimal number of dimensions is data dependent, the optimal α of CA is usually considerably larger than that of LSA.

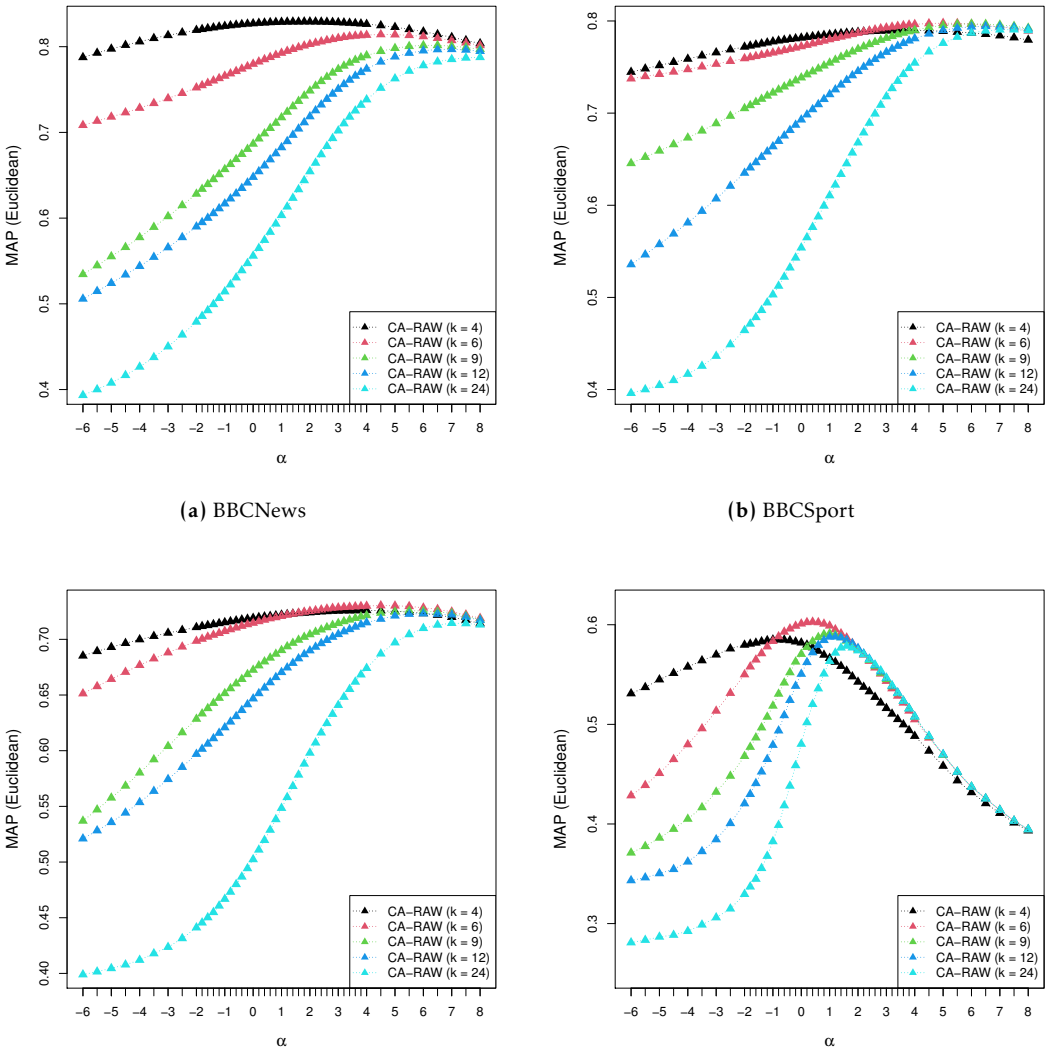
Table 3.8: MAP with the optimal α for CA-RAW under $k = 4, 6, 9, 12,$ and 24. Bold values are best.

	BBCNews		BBCSport		20 Newsgroups		Wilhelmus	
	α	MAP	α	MAP	α	MAP	α	MAP
CA-RAW ($k = 4$)	2	**0.829**	3.6	0.790	4	0.726	-1	0.585
CA-RAW ($k = 6$)	4.5	0.814	5	**0.798**	4.5	**0.730**	0.4	**0.603**
CA-RAW ($k = 9$)	6.5	0.802	6	0.797	5.5	0.726	1	0.591
CA-RAW ($k = 12$)	7	0.797	6.5	0.794	6	0.723	1.2	0.588
CA-RAW ($k = 24$)	8	0.788	7.5	0.791	7	0.715	1.6	0.579

3.5 Results for dot similarity and cosine similarity

In Section 3.4, we presented the results where Euclidean distance was used as a measure of similarity. Here, for comparison, we provide results for dot similarity and cosine similarity. Tables and figures for dot similarity and cosine similarity are presented in the supplementary materials.

The results for both dot similarity and cosine similarity lead to conclusions that match those for Euclidean distance. However, cosine similarity leads to a better performance in terms of MAP than Euclidean distance and dot similarity. We displayed the results for Euclidean distance in Section 3.4 because (1) it is more easily interpretable in the context of adjusting weighting exponent α: as α increases, Euclidean distances between row points (column points) on initial dimensions increase relative



(a) BBCNews

(b) BBCSport

(c) 20 Newsgroups

(d) Wilhelmus

Figure 3.6: MAP as a function of α for CA-RAW under various values of k.

to the later dimensions; and (2) in the literature, the Euclidean distance is the preferred way to interpret CA (in fact, we have never seen an interpretation of CA in terms of cosine or dot similarity).

3.6 Conclusion and discussion

Both LSA and CA make use of SVD. The main difference between LSA and CA is the matrix that is decomposed by SVD. In LSA, the decomposed matrix is the weighted matrix $\boldsymbol{A}$. In CA the decomposed matrix is the matrix $\boldsymbol{S}$ of standardized residuals, where in the part $(\boldsymbol{P} - \boldsymbol{E})$ the marginal effects are eliminated (Qi et al., 2023), and whose rank is one less the rank of $\boldsymbol{A}$. That is why the CA solution only displays the dependence between documents and terms. In LSA, on the other hand, the decomposed matrix also includes marginal effects, which are usually not relevant for information retrieval.

CA is related to the statistical independence model (Greenacre, 1984). The elements of $\boldsymbol{S}$ display the departure from marginal products, i.e., the departure form the statistical independence model. The sum of squared elements of $\boldsymbol{S}$ equals the Pearson chi-square statistic divided by the sum of elements of $\boldsymbol{F}$. CA decomposes the departure from statistical independence into a number of dimensions using SVD. LSA, on the other hand, has no connection with the statistical independence model.

In this paper, we compared four versions of LSA: LSA-RAW, LSA-NROWL1, LSA-NROWL2, and LSA-TFIDF with CA and found that CA always performs better than LSA in terms of MAP. Then, we compared LSA-RAW as a function of weighting exponent α with CA under a range of the numbers of dimensions. Even though LSA is improved by choosing an appropriate value for α, CA always performed better than LSA.

Next, we applied different weighting elements of the raw document-term matrix to CA. We found that weighting elements of the raw matrix sometimes improves the performance of CA, but improvements over CA-RAW are small and data dependent. The performance of CA-NROWL1 is better than that of LSA-NROWL1, the performance of CA-NROWL2 is better than that of LSA-NROWL2, and the performance of CA-TFIDF is better than that of LSA-TFIDF. Then, we adjusted the weighting exponents α in CA. For CA, as a function of α, MAP first increases and then decreases. Adjusting the weighting exponent α can potentially improve the performance of CA. However, the increased performance obtained by adjusting α is data and dimension dependent.

Using the standard coordinates of $\alpha = 1$, for LSA, the Euclidean distances between the rows of coordinates approximate the Euclidean distances between the rows of the decomposed matrix. For CA, the Euclidean distances between the rows of coordinates approximate the χ^2-distances between the rows of the decomposed matrix. $\alpha < 1$ gives less emphasis to the initial dimensions relative to the standard coordinates. Conversely, $\alpha > 1$ gives more emphasis to the initial dimensions relative to the standard coordinates. The optimal α for CA is almost always larger than that for LSA

and is almost always larger than 1.

Bullinaria and Levy (2012) argued that the initial dimensions in LSA tend not to contribute the most useful information about semantics and tend to be contaminated by "noise". The above mentioned results indicate that CA places more emphasis on the initial dimensions than LSA. The major difference between LSA and CA is that LSA involves margins but CA does not (Qi et al., 2023). Thus, we infer that margins considerably contribute to the initial dimensions in LSA. These margins are irrelevant for information retrieval. The CA effectively eliminates this irrelevant information.

In this paper, we focused on the performances of CA and LSA using Euclidean distances. We also performed identical experiments for dot similarity and cosine similarity. Both have nearly identical results with the Euclidean distance. Cosine similarity performs better than the Euclidean distance and dot similarity. We focus on Euclidean distance in the paper because (1) it is more easily interpretable in the context of adjusting α: as α increases, the Euclidean distances between row points (column points) on the initial dimensions increase relative to the later dimensions; (2) for CA, dot similarity and cosine similarity have never been used before, and therefore, by focusing on Euclidean distances, the results fit better into the existing literature.

Based on theoretical considerations and experimental results, we have the following three suggestions for practical guidance:

1. Use CA instead of LSA under the four kinds of feature extraction: RAW, NROWL1, NROWL2, and TF-IDF; use CA for visualizing data.

2. If information retrieval is the key issue, use cosine similarity instead of Euclidean distance and dot similarity for calculating MAP.

3. If optimal performance in terms of MAP is not of key importance, there is no need to weight the elements of raw document-term matrix for CA and optimize the performance over α for CA to save time. Otherwise, these two weightings may be considered potential approaches for improving the performance of CA.

Our finding that CA performs better than LSA for information retrieval is very important for creating next generation intelligent information systems. Among many other tasks, LSA has been widely used for information retrieval. We expect that the performance of these tasks can be improved by replacing LSA with CA.

Concluding, CA and LSA are both tools for information retrieval but the performance of CA is better. In our paper we tried to further improve CA by weighting the input matrix and by weighting dimensions. This did not lead to large or consistent improvements of the performance of CA.

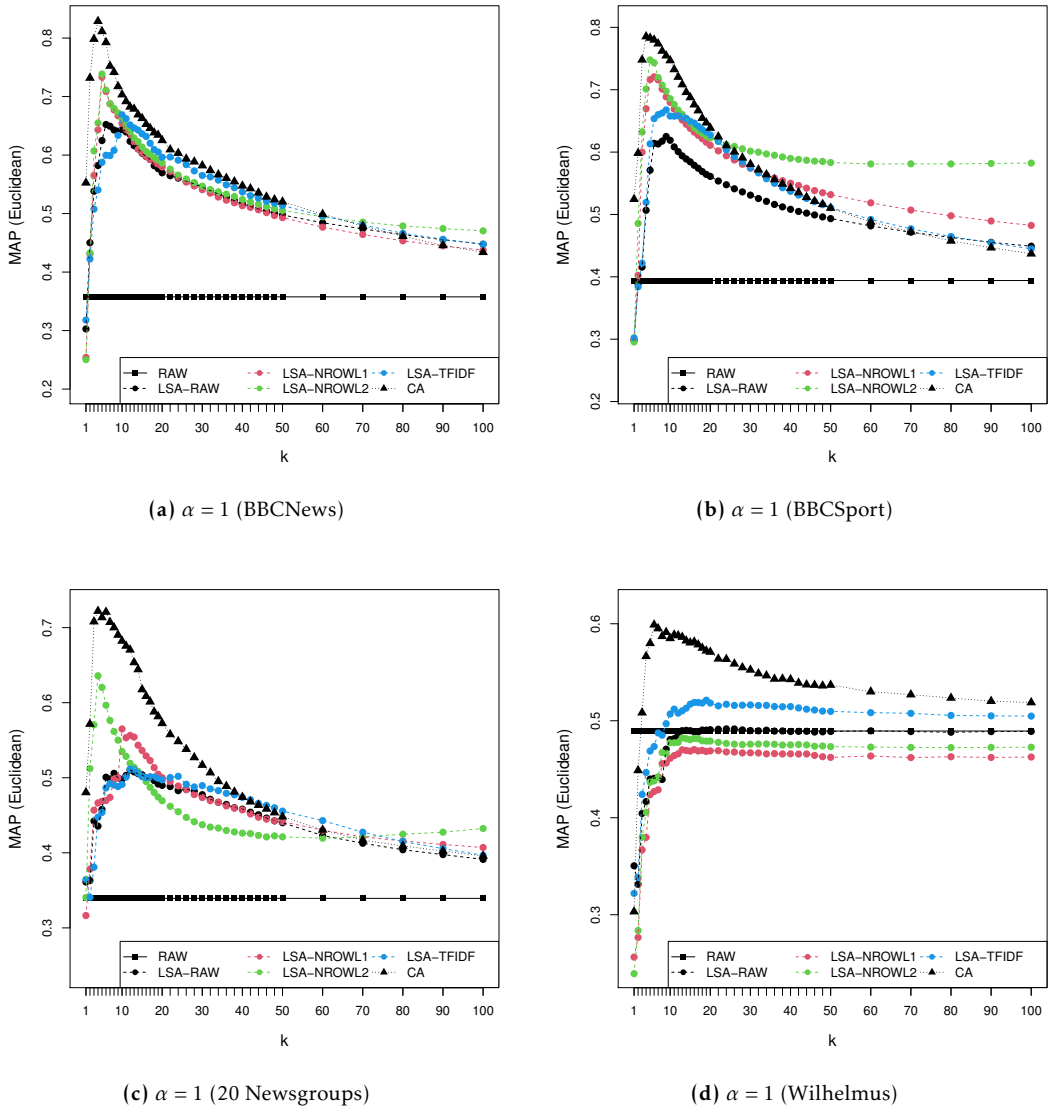
Further studies on the combination of LSA and CA will also be interesting. For example, creating an ensemble voting system using the coordinates from LSA and CA in the process of returning documents of a query. This paper, however, focuses on the comparison of LSA and CA for information retrieval and other explorations are left for future studies.

Appendix 3.A Euclidean distance

3.A.1 Comparing LSA and CA for information retrieval

3.A.1.1 MAP as a function of the number of dimensions for the four versions of LSA with the standard weighting exponent $\alpha = 1$ and for CA

Figure 3.7 shows MAP as a function of the number of dimensions k with dimensionality up to 100 for different weighting schemes of LSA, and for CA.



(a) $\alpha = 1$ (BBCNews)

(b) $\alpha = 1$ (BBCSport)

(c) $\alpha = 1$ (20 Newsgroups)

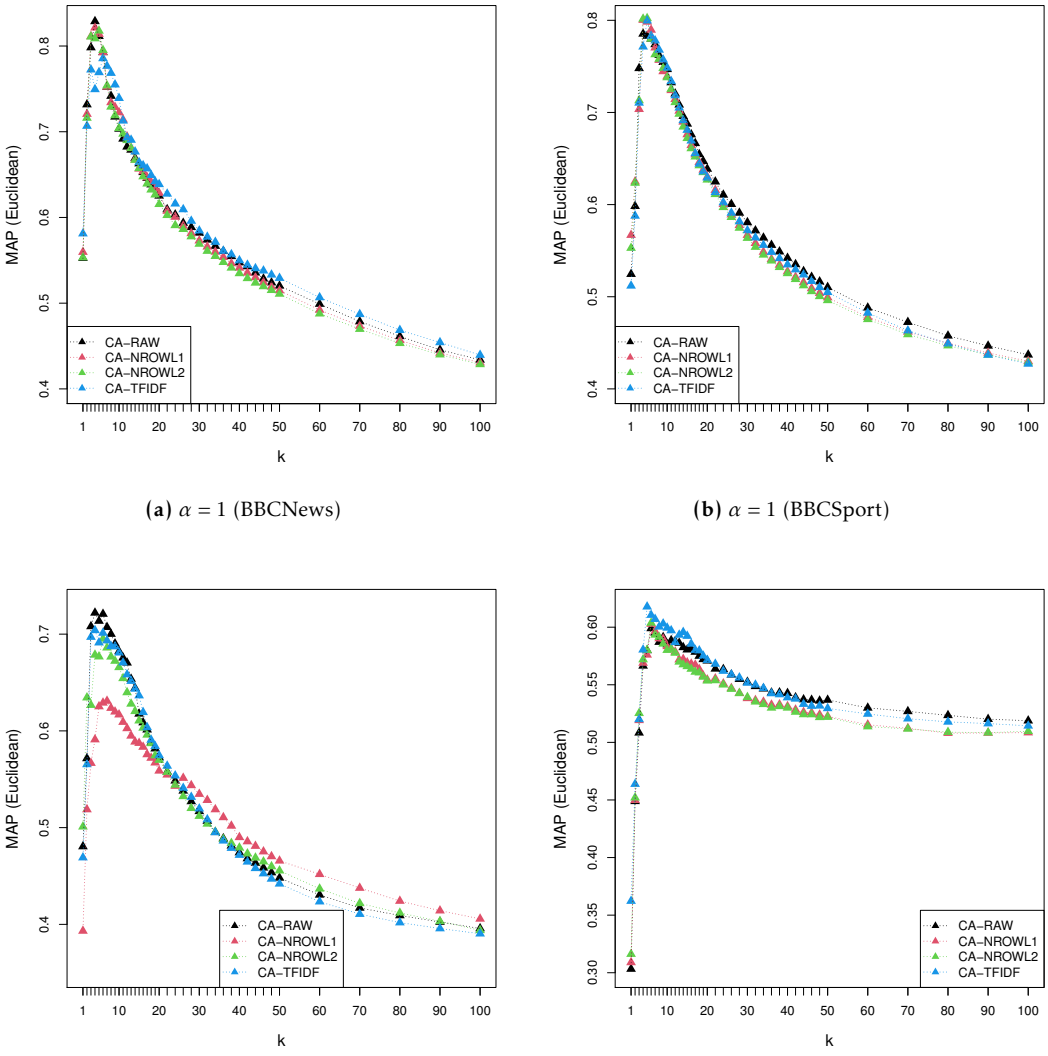
(d) $\alpha = 1$ (Wilhelmus)

Figure 3.7: MAP as a function of the number of dimensions k under standard coordinates.

3.A.2 Adjusting CA using weighting

3.A.2.1 Weighting the elements of the raw document-term matrix for CA

Similar to Figure 3.7, Figure 3.8 shows MAP as a function of k with dimensionality up to 100 for different weighting schemes of CA.



(a) $\alpha = 1$ (BBCNews)

(b) $\alpha = 1$ (BBCSport)

(c) $\alpha = 1$ (20 Newsgroups)

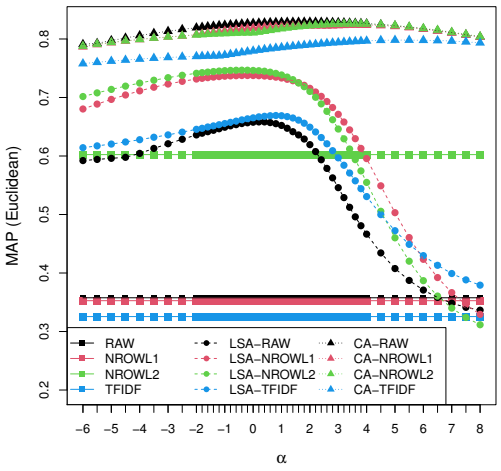
(d) $\alpha = 1$ (Wilhelmus)

Figure 3.8: MAP as a function of the number of dimensions k for the four versions of CA under standard coordinates.

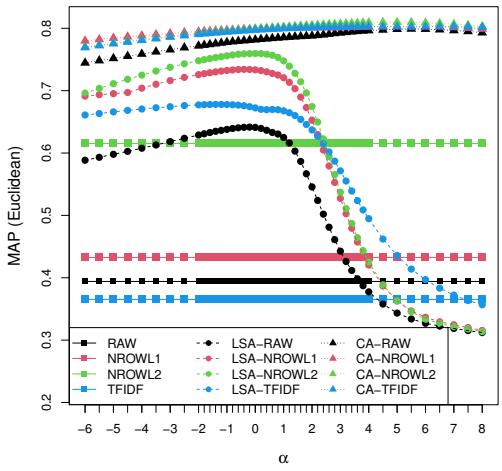
3.A.3 Exploring MAP as a function of α under the optimal number of dimensions for LSA and CA

Figure 3.9 shows the MAP as a function of α under the optimal number of dimensions. Similar to Section 3.4.1.1 in the context of $\alpha = 1$, we can obtain the corresponding optimal k (not shown in the figure) and the corresponding MAP (shown in the figure) for each α. Table 3.9 shows the optimal α, optimal k, and corresponding MAP, which is a condensed version of Figure 3.9. Based on Figure 3.9 and Table 3.9, we can see that

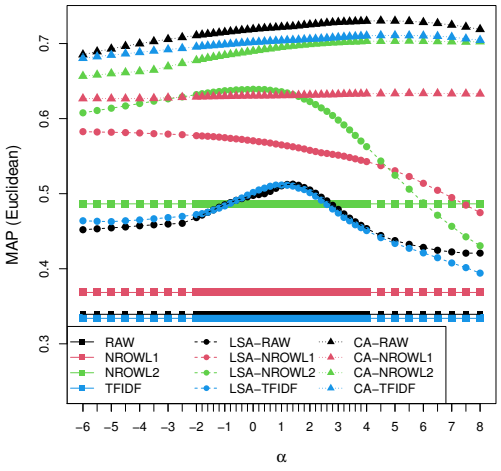
- CA methods are always better than the LSA methods and term matching methods under the optimal α and optimal number of dimensions k.
- Weighting the elements of the raw document-term matrix under the optimal number of dimensions k can improve the performance of CA; however, the improvements are small and data dependent.
- Similar to dimension $k = 4, 6, 9, 12,$ and 24, MAP as a function of α under the optimal number of dimensions k also first increases and then decreases. Thus, adjusting α can potentially improve the performance of LSA and CA.
- For different datasets, the optimal α under the optimal number of dimensions k is very different. In constrast to LSA, CA needs a greater α to reach the optimal performance under the optimal number of dimensions k. This illustrates that CA places more emphasis on initial dimensions than LSA.



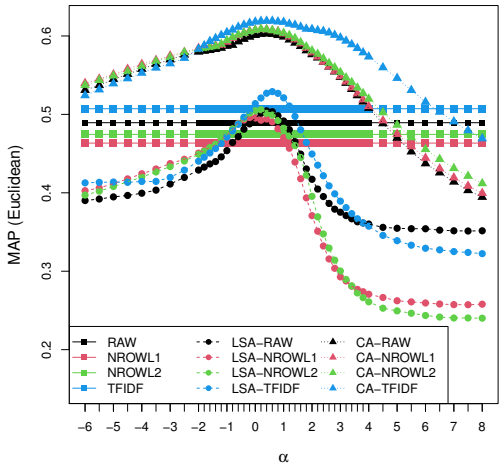
(a) BBCNews



(b) BBCSport



(c) 20 Newsgroups



(d) Wilhelmus

Figure 3.9: MAP as a function of α under the optimal number of dimensions.

Table 3.9: MAP under the optimal α and optimal dimension k. Bold values are best within group; underlined values are best overall.

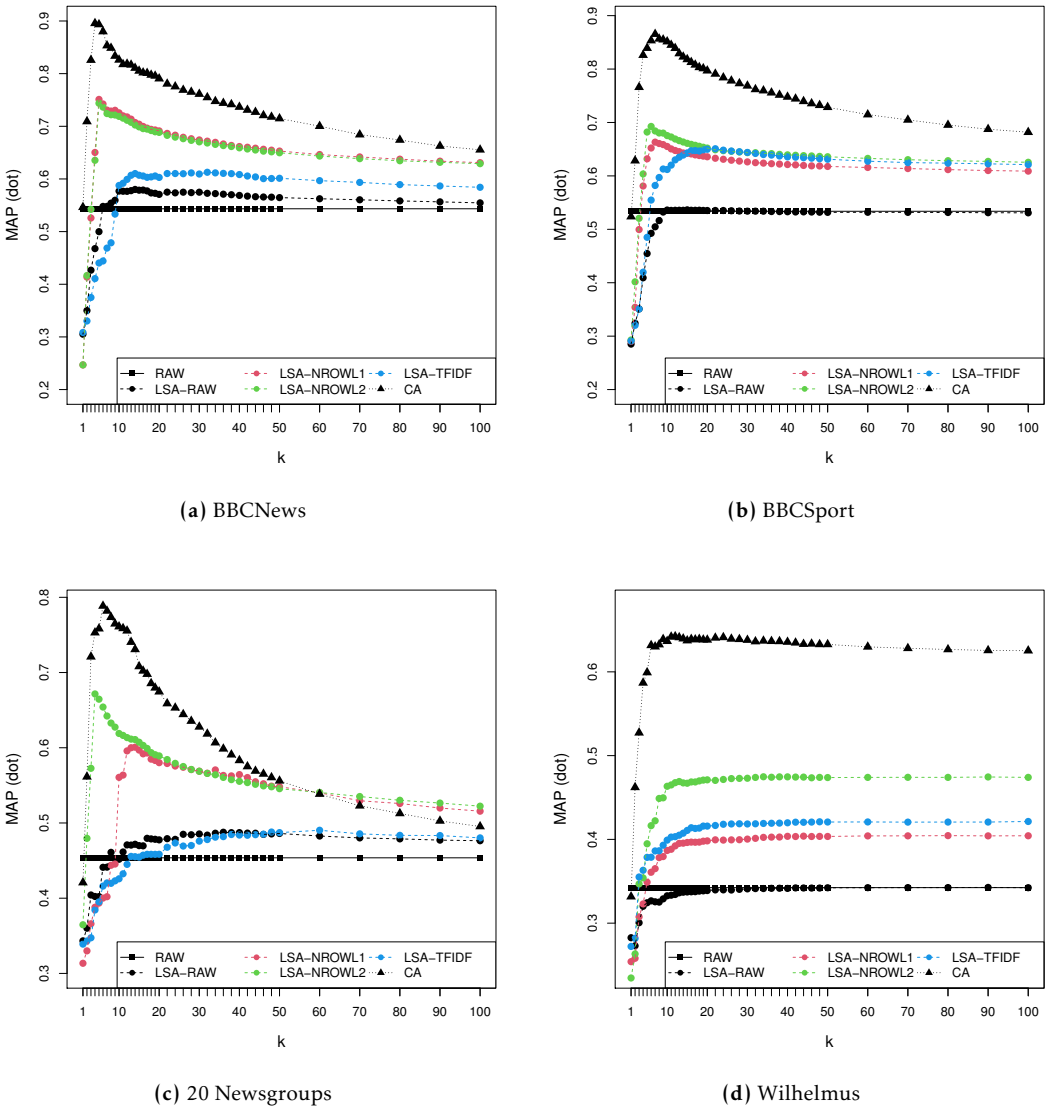
	BBCNews			BBCSport			20 Newsgroups			Wilhelmus		
	α	k	MAP	α	k	MAP	α	k	MAP	α	k	MAP
RAW			0.358			0.394			0.339			0.489
LSA-RAW	0.2	6	0.658	-0.2	6	0.642	1.4	12	0.513	0.4	13	0.505
CA-RAW	2	4	**0.829**	5.5	7	**0.799**	4.5	6	**0.730**	0.4	6	**0.603**
NROWL1			0.353			0.433			0.368			0.463
LSA-NROWL1	-0.4	5	0.738	-0.4	5	0.734	-6	10	0.583	-0.2	8	0.496
CA-NROWL1	3.6	5	**0.824**	5.5	5	**0.803**	5.5	7	**0.634**	0.2	6	**0.609**
NROWL2			0.602			0.615			0.486			0.474
LSA-NROWL2	-0.4	5	0.747	0	5	0.760	0	4	0.639	0.2	10	0.506
CA-NROWL2	3.6	5	**0.826**	5	5	**0.810**	5.5	6	**0.703**	0.2	6	**0.609**
TFIDF			0.326			0.365			0.334			0.507
LSA-TFIDF	0.8	10	0.669	-1.4	6	0.678	1	12	0.512	0.6	16	0.529
CA-TFIDF	5	6	**0.798**	6.5	7	**0.803**	5	6	**0.711**	0.6	5	**0.619**

Appendix 3.B Dot similarity

We performed identical experiments to the main paper as well as Section 3.A.3, but using dot similarity, instead of Euclidean distance, as similarity measurement method. The results lead to matching conclusions as those for Euclidean distance used in the main paper.

3.B.1 Comparing LSA and CA for information retrieval

3.B.1.1 MAP as a function of the number of dimensions for four versions of LSA with standard weighting exponent $\alpha = 1$ and CA



(a) BBCNews

(b) BBCSport

(c) 20 Newsgroups

(d) Wilhelmus

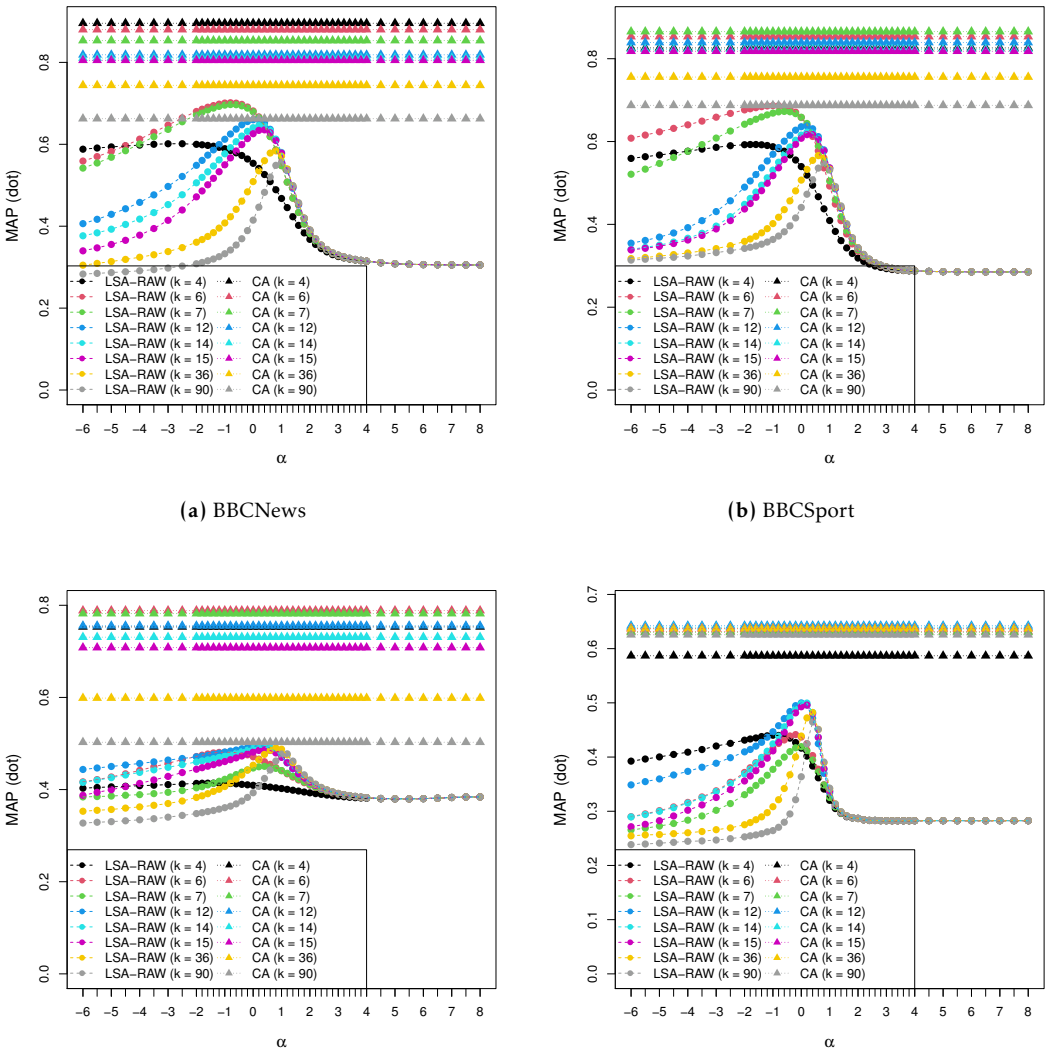
Figure 3.10: MAP as a function of the number of dimensions under standard coordinates.

Table 3.10: MAP with the optimal number of dimensions k about dot similarity. Bold values are best.

	BBCNews		BBCSport		20 Newsgroups		Wilhelmus	
	k	MAP	k	MAP	k	MAP	k	MAP
RAW		0.543		0.534		0.454		0.342
LSA-RAW	14	0.580	15	0.536	36	0.488	90	0.343
LSA-NROWL1	5	0.751	7	0.663	14	0.601	80	0.404
LSA-NROWL2	5	0.744	6	0.693	4	0.672	34	0.475
LSA-TFIDF	32	0.613	22	0.651	60	0.490	100	0.421
CA	4	**0.896**	7	**0.865**	6	**0.788**	12	**0.642**

3.B.1.2 MAP as a function of the weighting exponent α about LSA and MAP about CA for various values of the number of dimensions

Figure 3.11 shows MAP as a function of α about LSA-RAW and MAP about CA for the number of dimensions: $k = 4, 6, 7, 12, 14, 15, 36,$ and 90.



(a) BBCNews

(b) BBCSport

(c) 20 Newsgroups

(d) Wilhelmus

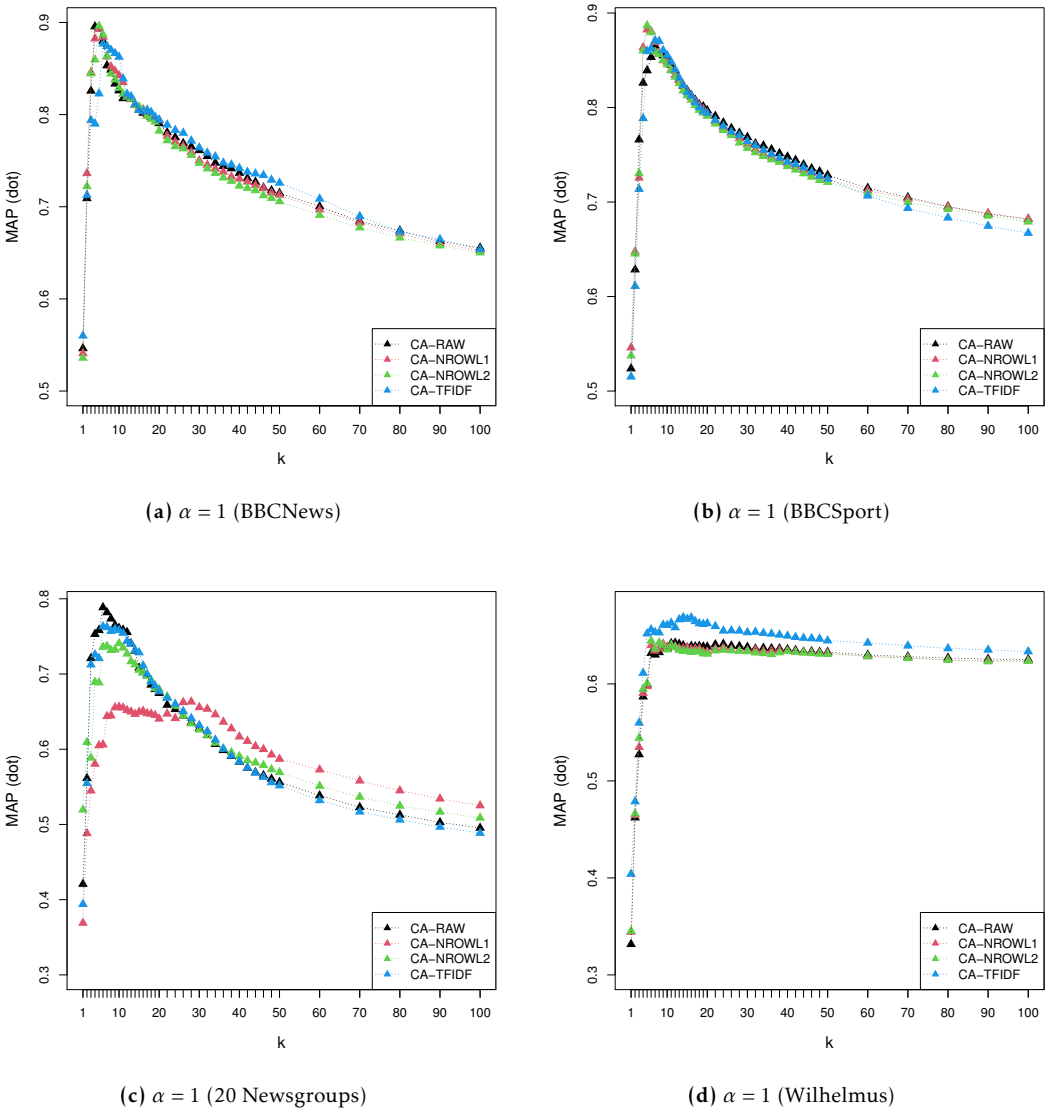
Figure 3.11: MAP as a function of α for LSA-RAW and MAP for CA under various values of k.

Table 3.11: MAP with the optimal weighting exponent α for LSA-RAW and MAP for CA under $k = 4, 6, 7, 12, 14, 15, 36,$ and 90 about dot similarity. Bold values are best.

	BBCNews		BBCSport		20 Newsgroups		Wilhelmus	
	α	MAP	α	MAP	α	MAP	α	MAP
LSA-RAW ($k = 4$)	-2.5	0.601	-1.8	0.593	-1.4	0.414	-1	0.440
LSA-RAW ($k = 6$)	-0.8	0.701	-1	0.687	-1	0.481	-0.2	0.441
LSA-RAW ($k = 7$)	-0.8	0.697	-0.6	0.673	0.4	0.450	0	0.419
LSA-RAW ($k = 12$)	0	0.660	0.2	0.638	0.2	0.495	0	0.500
LSA-RAW ($k = 14$)	0.2	0.646	0.2	0.623	0.4	0.492	0.2	0.500
LSA-RAW ($k = 15$)	0.4	0.635	0.4	0.617	0.4	0.486	0.2	0.496
LSA-RAW ($k = 36$)	0.8	0.584	0.6	0.566	0.8	0.490	0.4	0.482
LSA-RAW ($k = 90$)	1	0.556	0.8	0.547	1.2	0.478	0.4	0.465
CA ($k = 4$)		**0.896**		0.826		0.753		0.587
CA ($k = 6$)		0.880		0.853		**0.788**		0.632
CA ($k = 7$)		0.853		**0.865**		0.782		0.630
CA ($k = 12$)		0.818		0.839		0.756		**0.642**
CA ($k = 14$)		0.811		0.823		0.731		0.640
CA ($k = 15$)		0.805		0.818		0.708		0.637
CA ($k = 36$)		0.744		0.756		0.599		0.637
CA ($k = 90$)		0.663		0.687		0.503		0.626

3.B.2 Improving performance of CA for information retrieval

3.B.2.1 Weighting scheme of raw document-term matrix for CA



(a) $\alpha = 1$ (BBCNews)

(b) $\alpha = 1$ (BBCSport)

(c) $\alpha = 1$ (20 Newsgroups)

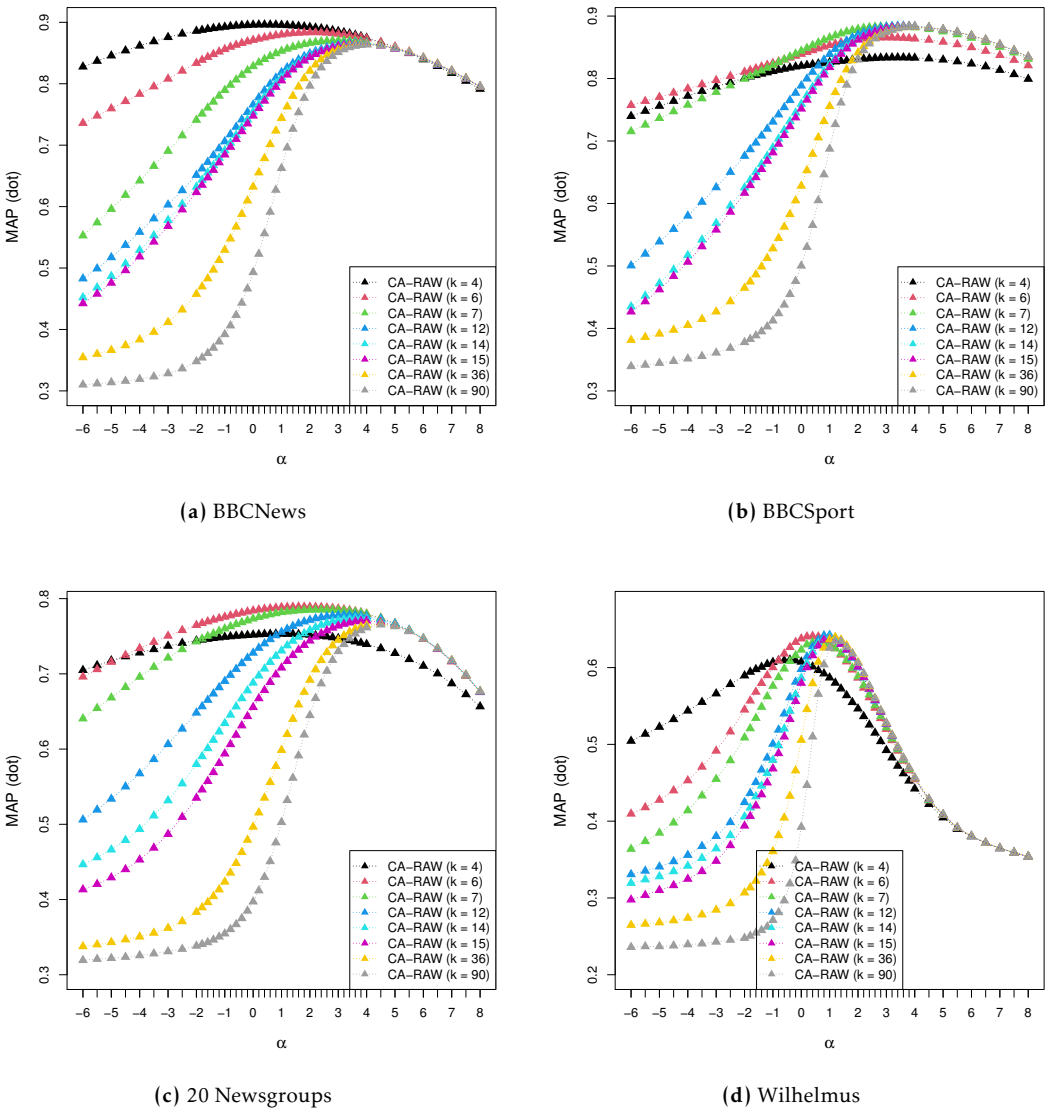
(d) $\alpha = 1$ (Wilhelmus)

Figure 3.12: MAP as a function of the number of dimensions k for the four versions of CA under standard coordinates.

Table 3.12: MAP with the optimal number of dimensions k for the four versions of CA about dot similarity. Bold values are best.

	BBCNews		BBCSport		20 Newsgroups		Wilhelmus	
	k	MAP	k	MAP	k	MAP	k	MAP
CA-RAW	4	**0.896**	7	0.865	6	**0.788**	12	0.642
CA-NROWL1	5	0.893	5	0.882	28	0.663	9	0.641
CA-NROWL2	5	**0.896**	5	**0.887**	10	0.741	6	0.644
CA-TFIDF	6	0.877	7	0.871	6	0.763	14	**0.669**

3.B.2.2 Weighting exponent α in CA



(a) BBCNews

(b) BBCSport

(c) 20 Newsgroups

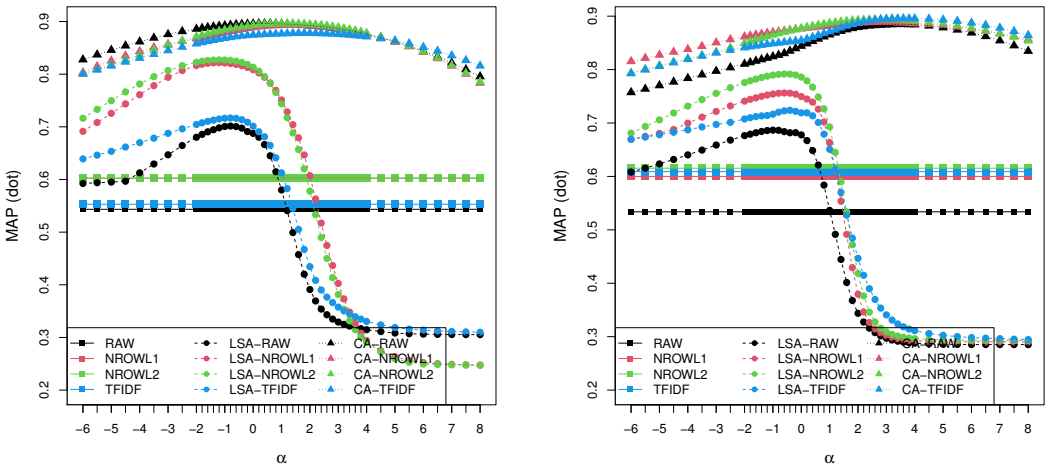
(d) Wilhelmus

Figure 3.13: MAP as a function of α for CA-RAW under various values of k.

Table 3.13: MAP with the optimal α for CA-RAW under $k = 4, 6, 7, 12, 14, 15, 36,$ and 90 about dot similarity. Bold values are best.

	BBCNews		BBCSport		20 Newsgroups		Wilhelmus	
	α	MAP	α	MAP	α	MAP	α	MAP
CA ($k = 4$)	0.4	**0.896**	3.4	0.834	0.8	0.753	-0.4	0.610
CA ($k = 6$)	2.2	0.884	3	0.866	1.6	**0.789**	0.4	0.641
CA ($k = 7$)	3.2	0.870	3	0.883	2.4	0.785	0.6	0.635
CA ($k = 12$)	3.8	0.866	3.6	**0.884**	3.4	0.779	1	**0.642**
CA ($k = 14$)	3.8	0.866	3.8	0.883	3.8	0.774	1	0.640
CA ($k = 15$)	3.8	0.865	3.8	0.883	4	0.771	1	0.637
CA ($k = 36$)	4	0.864	4	0.883	4.5	0.767	1.2	0.640
CA ($k = 90$)	4	0.863	4	0.883	4.5	0.765	1.2	0.635

3.B.3 Exploring MAP as a function of α under the optimal number of dimensions for LSA and CA



(a) BBCNews

(b) BBCSport

(c) 20 Newsgroups

(d) Wilhelmus

Figure 3.14: MAP as a function of α under optimal dimension.

Table 3.14: MAP under the optimal α and optimal dimension k about dot similarity. Bold values are best within group; underlined values are best overall.

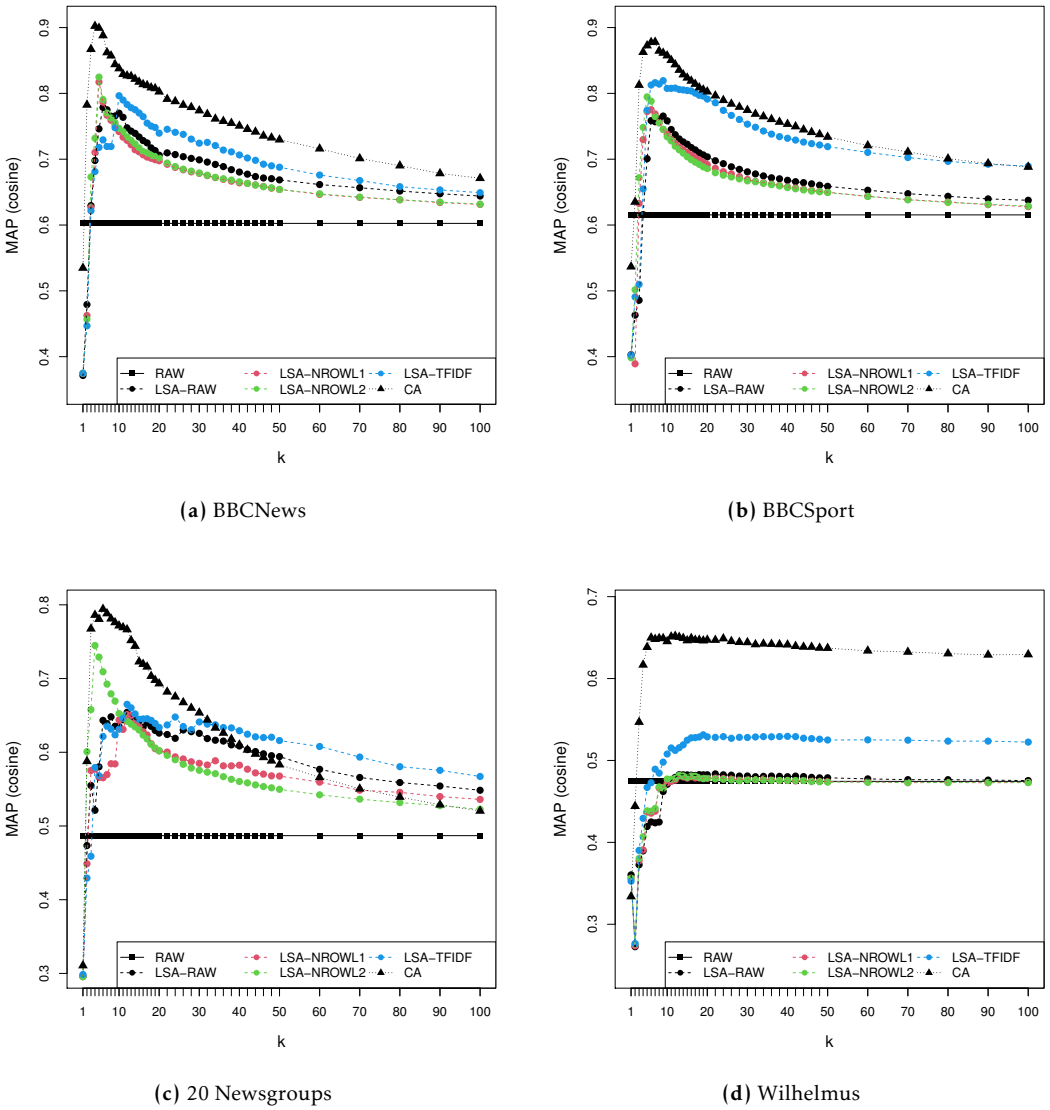
	BBCNews			BBCSport			20 Newsgroups			Wilhelmus		
	α	k	MAP	α	k	MAP	α	k	MAP	α	k	MAP
RAW			0.543			0.534			0.454			0.342
LSA-RAW	-0.8	6	0.701	-1	6	0.687	0.4	17	0.496	0	13	0.505
CA-RAW	0.4	4	**0.896**	3.6	10	**0.885**	1.6	6	**0.789**	1	12	**0.642**
NROWL1			0.603			0.600			0.472			0.405
LSA-NROWL1	-1.2	5	0.822	-0.6	5	0.756	-2	12	0.634	0.2	11	0.525
CA-NROWL1	1.4	5	**0.893**	2.8	6	**0.889**	5.5	32	**0.690**	0.2	6	**0.651**
NROWL2			0.602			0.615			0.487			0.474
LSA-NROWL2	-1.2	5	0.827	-0.6	5	0.792	-1.6	4	0.742	0.2	10	0.534
CA-NROWL2	1.2	5	**0.896**	3	6	**0.893**	3.8	10	**0.753**	0.4	6	**0.654**
TFIDF			0.553			0.609			0.453			0.422
LSA-TFIDF	-0.8	10	0.717	-0.4	9	0.724	0.2	24	0.498	0.2	16	0.555
CA-TFIDF	2	6	**0.878**	4	10	**0.896**	2.6	9	**0.767**	1.4	16	**0.674**

Appendix 3.C Cosine similarity

We performed identical experiments to the main paper as well as Section 3.A.3, but using cosine similarity, instead of Euclidean distance, as similarity measurement method. The results follow the same trend of the main paper, leading similarity conclusions.

3.C.1 Comparing LSA and CA for information retrieval

3.C.1.1 MAP as a function of the number of dimensions for four versions of LSA with standard weighting exponent $\alpha = 1$ and CA



(a) BBCNews

(b) BBCSport

(c) 20 Newsgroups

(d) Wilhelmus

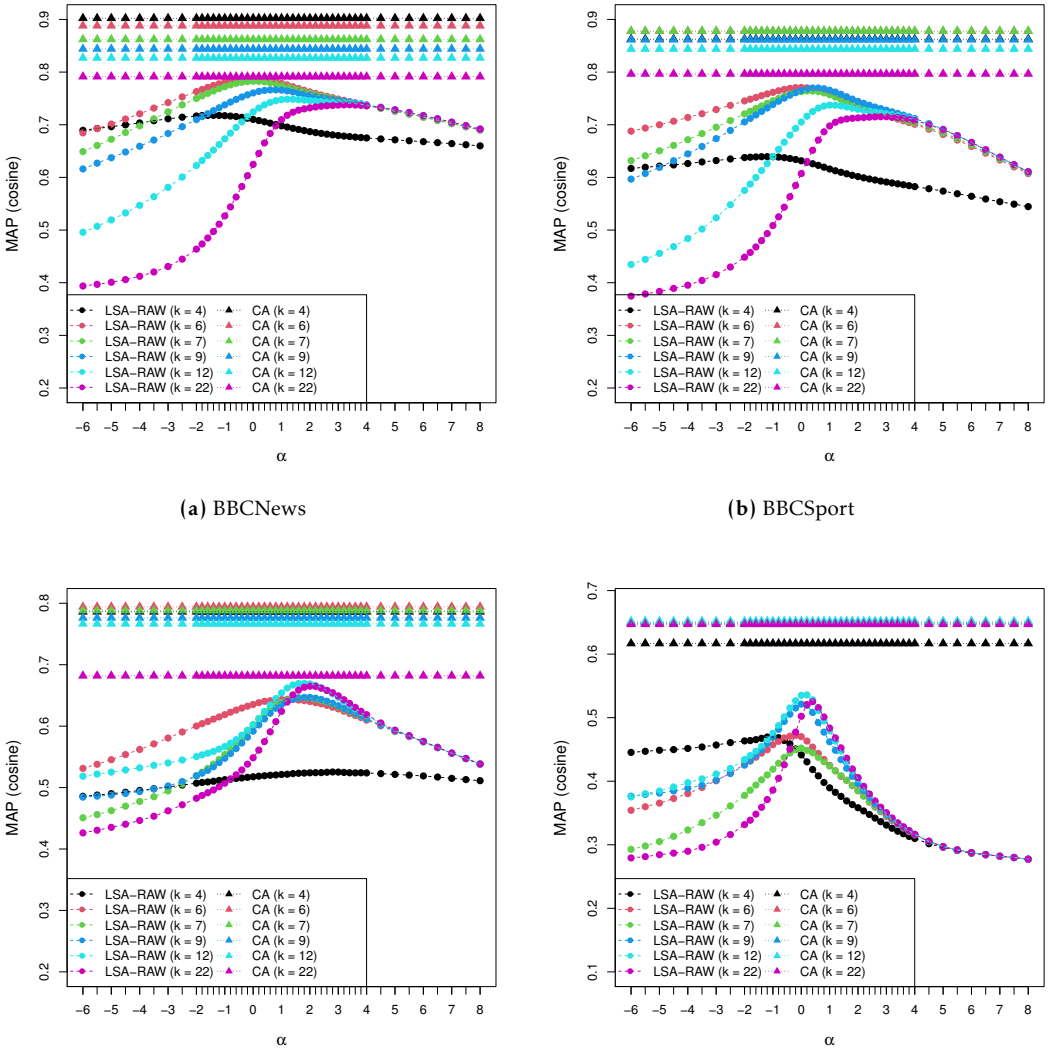
Figure 3.15: MAP as a function of the number of dimensions under standard coordinates.

Table 3.15: MAP with the optimal number of dimensions k about cosine similarity. Bold values are best.

	BBCNews		BBCSport		20 Newsgroups		Wilhelmus	
	k	MAP	k	MAP	k	MAP	k	MAP
RAW		0.602		0.615		0.487		0.474
LSA-RAW	6	0.779	9	0.766	12	0.654	22	0.484
LSA-NROWL1	5	0.817	6	0.775	12	0.651	13	0.481
LSA-NROWL2	5	0.825	5	0.795	4	0.745	13	0.482
LSA-TFIDF	10	0.796	9	0.819	12	0.665	19	0.531
CA	4	**0.902**	7	**0.878**	6	**0.794**	12	**0.652**

3.C.1.2 MAP as a function of the weighting exponent α about LSA and MAP about CA for various values of the number of dimensions

Figure 3.16 shows MAP as a function of α about LSA-RAW and MAP about CA for the number of dimensions: $k = 4, 6, 7, 9, 12,$ and 22.



(a) BBCNews

(b) BBCSport

(c) 20 Newsgroups

(d) Wilhelmus

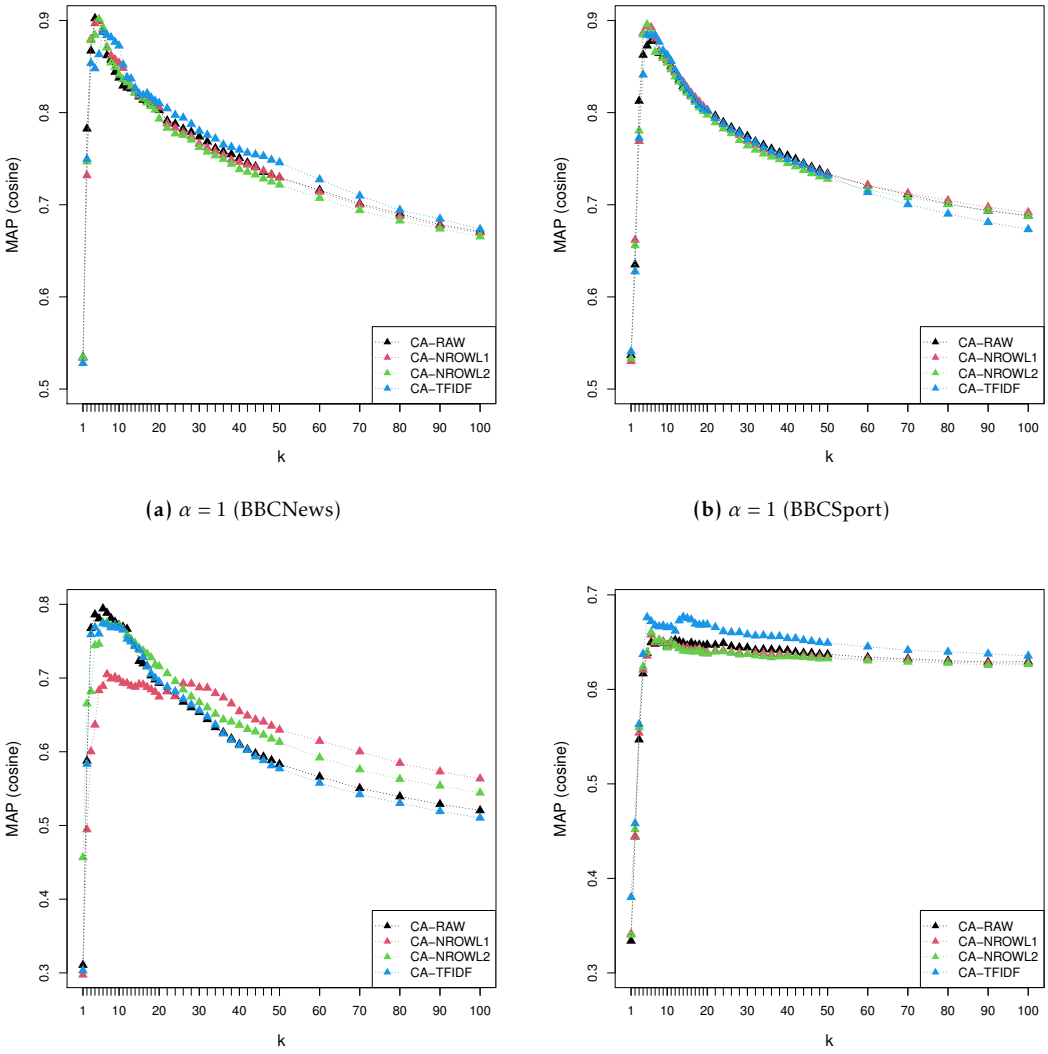
Figure 3.16: MAP as a function of α for LSA-RAW and MAP for CA under various values of k.

Table 3.16: MAP with the optimal weighting exponent α for LSA-RAW and MAP for CA under $k = 4, 6, 7, 9, 12,$ and 22 about cosine similarity. Bold values are best.

	BBCNews		BBCSport		20 Newsgroups		Wilhelmus	
	α	MAP	α	MAP	α	MAP	α	MAP
LSA-RAW ($k = 4$)	-1.4	0.718	-1.2	0.639	2.8	0.525	-1	0.470
LSA-RAW ($k = 6$)	0	0.788	0	0.771	1.2	0.644	-0.2	0.472
LSA-RAW ($k = 7$)	0	0.783	0.2	0.764	1.6	0.646	0	0.452
LSA-RAW ($k = 9$)	0.8	0.766	0.6	0.770	2	0.647	0	0.521
LSA-RAW ($k = 12$)	1.2	0.748	1	0.737	1.8	0.670	0.2	0.536
LSA-RAW ($k = 22$)	3.4	0.738	2.8	0.715	2	0.665	0.4	0.525
CA ($k = 4$)		**0.902**		0.863		0.786		0.617
CA ($k = 6$)		0.888		**0.878**		**0.794**		0.650
CA ($k = 7$)		0.862		**0.878**		0.788		0.648
CA ($k = 9$)		0.844		0.861		0.776		0.649
CA ($k = 12$)		0.827		0.844		0.767		**0.652**
CA ($k = 22$)		0.791		0.796		0.682		0.647

3.C.2 Improving performance of CA for information retrieval

3.C.2.1 Weighting scheme of raw document-term matrix for CA



(a) $\alpha = 1$ (BBCNews)

(b) $\alpha = 1$ (BBCSport)

(c) $\alpha = 1$ (20 Newsgroups)

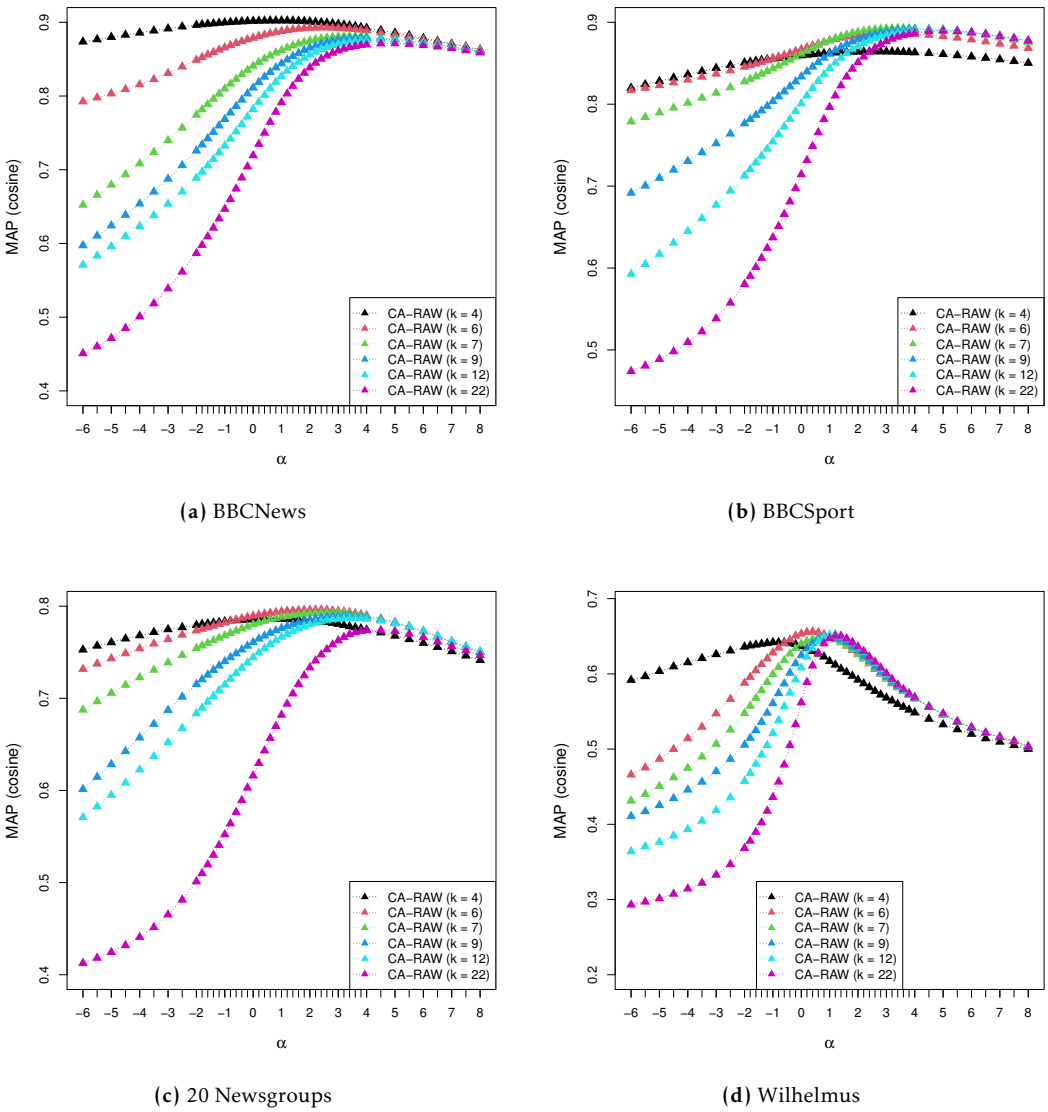
(d) $\alpha = 1$ (Wilhelmus)

Figure 3.17: MAP as a function of the number of dimensions k for the four versions of CA under standard coordinates.

Table 3.17: MAP with the optimal number of dimensions k for the four versions of CA about cosine similarity. Bold values are best.

	BBCNews		BBCSport		20 Newsgroups		Wilhelmus	
	k	MAP	k	MAP	k	MAP	k	MAP
CA-RAW	4	**0.902**	7	0.878	6	**0.794**	12	0.652
CA-NROWL1	5	0.900	5	0.894	7	0.705	6	0.658
CA-NROWL2	5	**0.902**	5	**0.896**	6	0.775	6	0.660
CA-TFIDF	6	0.887	5	0.884	6	0.774	14	**0.677**

3.C.2.2 Weighting exponent α in CA



(a) BBCNews

(b) BBCSport

(c) 20 Newsgroups

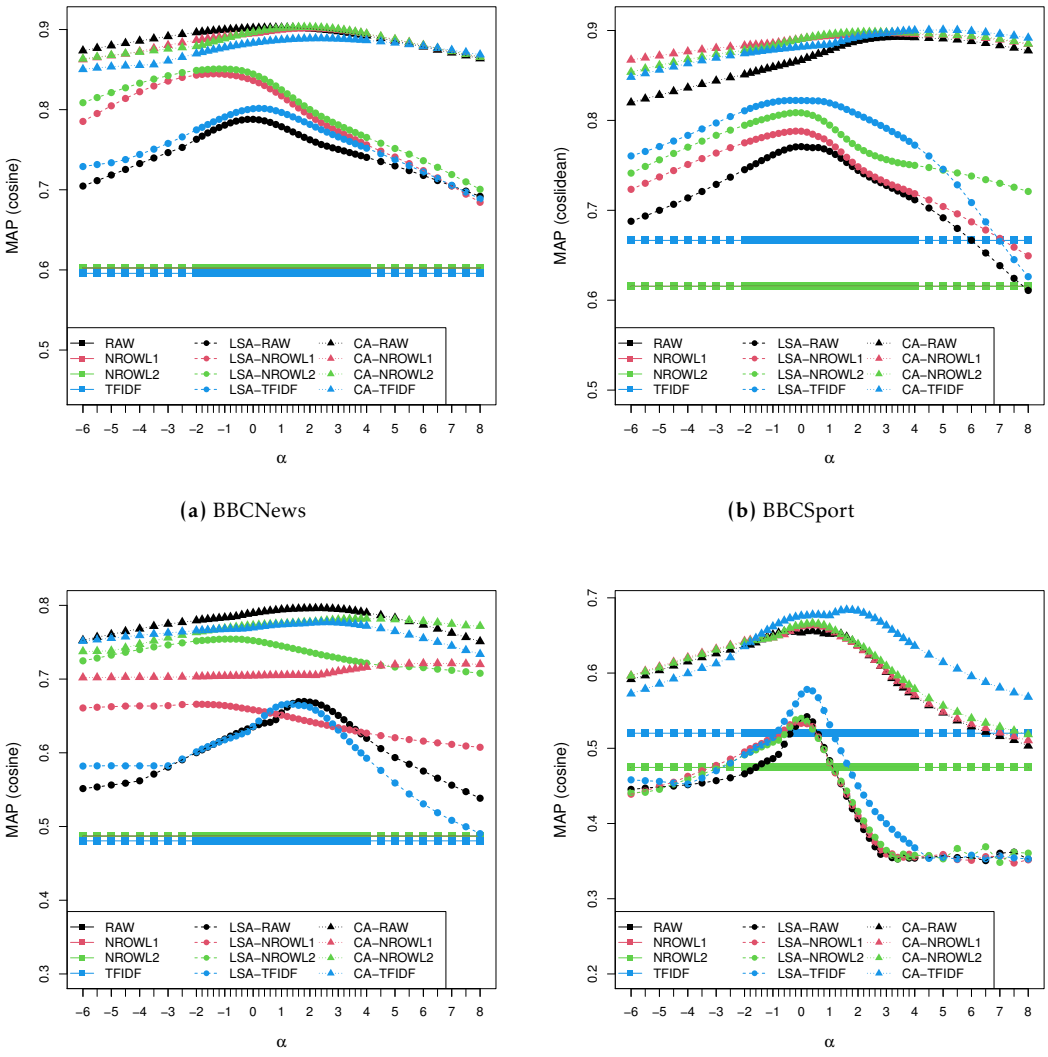
(d) Wilhelmus

Figure 3.18: MAP as a function of α for CA-RAW under various values of k.

Table 3.18: MAP with the optimal α for CA-RAW under $k = 4, 6, 7, 9, 12,$ and 22 about cosine similarity. Bold values are best.

	BBCNews		BBCSport		20 Newsgroups		Wilhelmus	
	α	MAP	α	MAP	α	MAP	α	MAP
CA ($k = 4$)	0.8	**0.902**	2.6	0.864	0.8	0.786	-0.8	0.642
CA ($k = 6$)	2.4	0.893	3.2	0.887	2.4	**0.796**	0.4	**0.656**
CA ($k = 7$)	3.6	0.881	3.4	**0.893**	2.8	0.793	0.8	0.649
CA ($k = 9$)	3.8	0.879	4	0.892	3.4	0.789	1	0.649
CA ($k = 12$)	4.5	0.876	4.5	0.890	3.6	0.787	1.2	0.652
CA ($k = 22$)	5	0.871	5	0.890	4.5	0.774	1.2	0.651

3.C.3 Exploring MAP as a function of α under the optimal number of dimensions for LSA and CA



(a) BBCNews

(b) BBCSport

(c) 20 Newsgroups

(d) Wilhelmus

Figure 3.19: MAP as a function of α under optimal dimension.

Table 3.19: MAP under the optimal α and optimal dimension k about cosine similarity. Bold values are best within group; underlined values are best overall.

	BBCNews			BBCSport			20 Newsgroups			Wilhelmus		
	α	k	MAP	α	k	MAP	α	k	MAP	α	k	MAP
RAW			0.602			0.615			0.487			0.474
LSA-RAW	0	6	0.788	0	6	0.771	1.8	12	0.670	0.2	13	0.542
CA-RAW	0.8	4	**0.902**	3.4	7	**0.893**	2.4	6	**<u>0.796</u>**	0.4	6	**0.656**
NROWL1			0.602			0.615			0.487			0.474
LSA-NROWL1	-1.4	5	0.845	-0.2	5	0.788	-2	12	0.666	-0.2	8	0.535
CA-NROWL1	2	5	**0.901**	2.8	6	**0.897**	6.5	32	**0.721**	0.4	6	**0.664**
NROWL2			0.602			0.615			0.487			0.474
LSA-NROWL2	-1	5	0.851	-0.2	5	0.809	-1	4	0.754	0.0	10	0.540
CA-NROWL2	1.8	5	**<u>0.903</u>**	2.6	5	**0.898**	4	10	**0.782**	0.4	6	**0.666**
TFIDF			0.596			0.667			0.481			0.520
LSA-TFIDF	0.2	10	0.802	-0.4	6	0.822	1.2	12	0.666	0.2	16	0.578
CA-TFIDF	2.4	6	**0.889**	5	10	**<u>0.901</u>**	2.6	9	**0.777**	1.6	14	**<u>0.684</u>**

Chapter 4

A comparison of correspondence analysis with PMI-based word embedding methods

Abstract

Popular word embedding methods such as GloVe and Word2Vec are related to the factorization of the pointwise mutual information (PMI) matrix. In this paper, we link correspondence analysis (CA) to the factorization of the PMI matrix. CA is a dimensionality reduction method that uses singular value decomposition (SVD), and we show that CA is mathematically close to the weighted factorization of the PMI matrix. In addition, we present variants of CA that turn out to be successful in the factorization of the word-context matrix, i.e. CA applied to a matrix where the entries undergo a square-root transformation (ROOT-CA) and a root-root transformation (ROOTROOT-CA). An empirical comparison among CA- and PMI-based methods shows that overall results of ROOT-CA and ROOTROOT-CA are slightly better than those of the PMI-based methods.

This chapter is under review as: Qi, Q., Hessen, D. J., & Van der Heijden, P. G. M.. A comparison of correspondence analysis with PMI-based word embedding methods. Author contributions: QQ, DH, and PvdH posed the problem. QQ worked out the idea, set up the experiments, and carried them out. QQ, DH, and PvdH discussed and edited the text. The code used in this study can be found at `https://github.com/qianqianqi28/ca-pmi`.

4.1 Introduction

Word embeddings, i.e., dense and low dimensional word representations, are useful in various natural language processing (NLP) tasks (Jurafsky & Martin, 2023; Sasaki, Heinzerling, Suzuki, & Inui, 2023). Three successful methods to derive such word representations are related to the factorization of the pointwise mutual information (PMI) matrix, an important matrix to be analyzed in NLP (Egleston et al., 2021; Bae et al., 2021; Alqahtani, Al-Twairesh, & Alsanad, 2023). The PMI matrix is a weighted version of the word-context co-occurrence matrix and measures how often two words, a target word and a context word, co-occur, compared with what we would expect if the two words were independent. The analysis of a positive PMI (PPMI) matrix, where all negative values in a PMI matrix are replaced with zero (Turney & Pantel, 2010; M. Zhang, Palade, Wang, & Ji, 2022; Alqahtani et al., 2023), generally leads to a better performance in semantic tasks (Bullinaria & Levy, 2007), and in most applications the PMI matrix is replaced with the PPMI matrix (Salle et al., 2016).

The first method, PPMI-SVD, decomposes the PPMI matrix with a singular value decomposition (SVD) (Levy & Goldberg, 2014; Levy et al., 2015; Stratos, Collins, & Hsu, 2015; M. Zhang et al., 2022). The second one is GloVe (Pennington et al., 2014). GloVe factorizes the logarithm of the word-context matrix with an adaptive gradient algorithm (AdaGrad) (Duchi, Hazan, & Singer, 2011). According to Shi and Liu (2014); Shazeer et al. (2016), GloVe is almost equivalent to factorizing a PMI matrix shifted by the logarithm of the sum of the elements of a word-context matrix. The third method is Word2Vec's skip-gram with negative sampling (SGNS) (Mikolov, Chen, et al., 2013; Mikolov, Sutskever, et al., 2013). SGNS uses a neural network model to generate word embeddings. Levy and Goldberg (2014) proved that SGNS implicitly factorizes a PMI matrix shifted by the logarithm of the number of negative samples in SGNS.

In this paper we study what correspondence analysis (CA) (Greenacre, 2017; Beh & Lombardo, 2021) has to offer for the analysis of word-context co-occurrence matrices. CA is an exploratory statistical method that is often used for visualization of a low dimensional approximation of a matrix. It is close to the T-test weighting scheme (Curran & Moens, 2002; Curran, 2004), where standardized residuals are studied, as CA is based on the SVD of the matrix of standardized residuals. In the context of document-term matrices, CA has been compared earlier with latent semantic analysis (LSA), where the document-term matrix is also decomposed with an SVD (Dumais et al., 1988; Deerwester et al., 1990). Although CA is similar to LSA, there is theoretical and empirical research showing that CA is to be preferred over LSA for text categorization and information retrieval (Qi et al., 2023; Qi, Hessen, & Van der Heijden, 2024).

CA of a two-way contingency table is equivalent to canonical correlation analysis (CCA) of the data in the form of indicator matrices for the row variable and the column variable of the two-way contingency table (Greenacre, 1984). Stratos et al. (2015) proposed to combine CCA with a square-root transformation of the elements of the

contingency table. In this paper we refer to this procedure as ROOT-CCA, to distinguish it from ROOT-CA introduced later. Stratos et al. (2015) found that, on word similarity tasks, (1) the performance of CCA is quite bad, but the performance of ROOT-CCA is a marked improvement, and (2) ROOT-CCA outperforms PPMI-SVD, GloVe, and SGNS. However, CA has not yet been linked to PMI-based methods.

A document-term matrix has some similarity to a word-context matrix, as they both use counts. In this paper, mathematically, we show that CA is close to a weighted factorization of the PMI matrix. We also propose a direct weighted factorization of the PMI matrix (PMI-GSVD). Furthermore, we empirically compare the performance of CA with the performance of PMI-based methods on a word similarity task.

In the context of CA, Nishisato, Beh, Lombardo, and Clavel (2021) point out, generally speaking, a two-way contingency table is prone to overdispersion. Overdispersion may negatively affect the performance of CA (Beh, Lombardo, & Alberti, 2018; Nishisato et al., 2021). To deal with this overdispersion, a fourth-root transformation can be used (Field, Clarke, & Warwick, 1982; Greenacre, 2009, 2010). The fourth root transformation has been widely discussed and applied (Downing, 1981; Kostensalo et al., 2023; France & Heung, 2023). Therefore, in addition to the word-context matrix, CA is also applied to the fourth-root transformation of the word-context matrix (ROOTROOT-CA). Inspired by ROOT-CCA, CA is also applied to the square-root transformation of the word-context matrix (ROOT-CA). Recently, ROOT-CA has been explored in biology (Hsu & Culhane, 2023). The difference between ROOT-CCA and ROOT-CA is discussed in Section 4.3.3.

In the following section, research objectives are presented. In Section 4.3 CA, the three variants of CA, and the T-Test weighting scheme are introduced. The three PMI-based methods are described in Section 4.4. Theoretical relationships between CA and the PMI-based methods are shown in Section 4.5. In Section 4.6 we present two corpora to build word vectors and five word similarity datasets to evaluate word vectors. Section 4.7 illustrates the setup of the empirical study using these two corpora where CA, PMI-SVD, PPMI-SVD, PMI-GSVD, ROOT-CA, ROOTROOT-CA, ROOT-CCA, SGNS, and GloVe are compared. Section 4.8 presents the results for these methods on word similarity tasks. Section 4.9 concludes and discusses this paper.

4.2 Research objectives

Considering the foregoing, this study focuses on word embeddings in NLP. The objective is to explore the relationship between CA and PMI-based methods and compare the performance in word similarity tasks. In addition, we explore the performance of variants of CA, namely ROOT-CA and ROOTROOT-CA.

4.3 Correspondence analysis

In this section, first we describe correspondence analysis (CA) using a distance interpretation (Benzécri, 1973; Greenacre & Hastie, 1987), which is a popular way to present CA. Then we present CA making use of an objective function, thus making the later comparison with PMI-based methods straightforward. Third, we present three variants of CA in word embedding. Finally, the T-Test weighting scheme (Curran & Moens, 2002; Curran, 2004) is described, as it turns out to be remarkably similar to CA.

A word-context matrix is a matrix with counts, in which the rows and columns are labeled by terms. In each cell a count represents the number of times the row (target) word and the column (context) word co-occur in a text (Jurafsky & Martin, 2023). Consider a word-context matrix denoted as $\boldsymbol{X}$ having I rows ($i = 1, 2, \cdots, I$) and J columns ($j = 1, 2, \cdots, J$), where the element for row i and column j is x_{ij}. The joint observed proportion is $p_{ij} = x_{ij}/x_{++}$, where "+" represents the sum over the corresponding elements. The marginal proportions of target word i and context word j are $p_{i+} = \sum_j p_{ij}$ and $p_{+j} = \sum_i p_{ij}$, respectively.

4.3.1 Introduction to CA

CA is an exploratory method for the analysis of two-way contingency tables. It allows to study how the counts in the contingency table depart from statistical independence. Here we introduce CA in the context of the word-context matrix $\boldsymbol{X}$. In CA of the matrix $\boldsymbol{X}$, first the elements x_{ij} are converted to joint observed proportions p_{ij}, and these are transformed into standardized residuals (Greenacre, 2017)

$$\frac{p_{ij} - p_{i+}p_{+j}}{\sqrt{p_{i+}p_{+j}}}. \tag{4.1}$$

Then an SVD is applied to this matrix of standardized residuals, yielding

$$\frac{p_{ij} - p_{i+}p_{+j}}{\sqrt{p_{i+}p_{+j}}} = \sum_{k=1}^{\min(I-1,J-1)} \sigma_k u_{ik} v_{jk}, \tag{4.2}$$

where σ_k is the kth singular value, with singular values in the decreasing order, and $u_{ik}, i = 1, 2, \cdots, I$ and $v_{jk}, j = 1, 2, \cdots, J$ are the kth left and right singular vectors, respectively. When $\boldsymbol{X}$ has full rank, the maximum dimensionality is $\min(I-1, J-1)$, where the "-1" is due to the subtraction of elements $p_{i+}p_{+j}$. Multiplying the singular vectors consisting of elements u_{ik} and v_{jk} by $p_{i+}^{-\frac{1}{2}}$ and $p_{+j}^{-\frac{1}{2}}$, respectively, leads to

$$\frac{p_{ij}}{p_{i+}p_{+j}} - 1 = \sum_{k=1}^{\min(I-1,J-1)} \sigma_k \phi_{ik} \gamma_{jk}, \tag{4.3}$$

where $\phi_{ik} = p_{i+}^{-\frac{1}{2}} u_{ik}$ and $\gamma_{jk} = p_{+j}^{-\frac{1}{2}} v_{jk}$. Scores $\phi_{ik}, k = 1, 2, \cdots, K$ and $\gamma_{jk}, k = 1, 2, \cdots, K$ provide the standard coordinates of row point i and column point j in K-dimensional space, respectively, because of $\sum_i p_{i+}\phi_{ik} = \sum_j p_{+j}\gamma_{jk} = 0$ and $\sum_i p_{i+}\phi_{ik}^2 = \sum_j p_{+j}\gamma_{jk}^2 = 1$. Scores $\phi_{ik}\sigma_k, k = 1, 2, \cdots, K$ and $\gamma_{jk}\sigma_k, k = 1, 2, \cdots, K$ provide the principle coordinates of row point i and column point j in K-dimensional space, respectively. When $K < \min(I-1, J-1)$, the Euclidean distances between these row (column) points approximate so-called χ^2-distances between rows (columns) of $\boldsymbol{X}$. The squared χ^2-distance between rows i and i' of $\boldsymbol{X}$ is

$$\delta_{ii'}^2 = \sum_j \frac{\left(\frac{p_{ij}}{p_{i+}} - \frac{p_{i'j}}{p_{i'+}}\right)^2}{p_{+j}}, \tag{4.4}$$

and similarly for the chi-squared distance between columns j and j'. Equation (4.4) shows that the χ^2–distance $\delta_{ii'}$ measures the difference between the ith vector of conditional proportions p_{ij}/p_{i+} and the i'th vector of conditional proportions $p_{i'j}/p_{i'+}$, where more weight is given to the differences in these columns if p_{+j} is relatively smaller compared to other columns.

Although the use of Euclidean distance is standard in CA, Qi et al. (2024) show that for information retrieval cosine similarity leads to the best performance among Euclidean distance, dot similarity, and cosine similarity. The superiority of cosine similarity also holds in the context of word embedding studies (Bullinaria & Levy, 2007). Therefore, in this paper we use cosine similarity to calculate the similarity of row points and of column points. It is worth noting that $p_{i+}^{-\frac{1}{2}}$ in $\phi_{ik} = p_{i+}^{-\frac{1}{2}} u_{ik}$ and $p_{+j}^{-\frac{1}{2}}$ in $\gamma_{jk} = p_{+j}^{-\frac{1}{2}} v_{jk}$ have no effects on the cosine similarity. Details are in Supplementary materials A. We coin scores $u_{ik}\sigma_k, k = 1, 2, \cdots, K$ and $v_{jk}\sigma_k, k = 1, 2, \cdots, K$ an alternative coordinates system for CA directly suited for cosine similarity.

The so-called total inertia is

$$\sum_i \sum_j \frac{\left(p_{ij} - p_{i+}p_{+j}\right)^2}{p_{i+}p_{+j}} = \sum_{k=1}^{\min(I-1,J-1)} \sigma_k^2. \tag{4.5}$$

This illustrates that CA decomposes the total inertia over $\min(I-1, J-1)$ dimensions. The total inertia equals the well-known Pearson χ^2 statistic divided by x_{++}. The relative contribution of cell (i, j) to the total inertia is calculated as $\frac{\left(p_{ij}-p_{i+}p_{+j}\right)^2}{p_{i+}p_{+j}} / \sum_i \sum_j \frac{\left(p_{ij}-p_{i+}p_{+j}\right)^2}{p_{i+}p_{+j}}$. The relative contribution of the ith row (jth column) to the kth dimension is calculated as u_{ik}^2 (v_{jk}^2).

4.3.2 The objective function of CA

To simplify the later comparison of CA with the other models, we present the objective function that is minimized in CA. The objective function is (Greenacre, 1984, pp. 345-

349):

$$\sum_{i,j} p_{i+}p_{+j}\left(\frac{p_{ij}}{p_{i+}p_{+j}} - 1 - \boldsymbol{e}_i^T \boldsymbol{o}_j\right)^2, \tag{4.6}$$

where $\boldsymbol{e}_i$ and $\boldsymbol{o}_j$ are parameter vectors for target word i and context word j, with respect to which the objective function is minimized. The vectors have length $K \leq \min(I-1, J-1)$. We call the part of the formula to be approximated, i.e. $\left(p_{ij}/p_{i+}p_{+j} - 1\right)$, the fitting function and the weighting part $p_{i+}p_{+j}$ the weighting function. Thus, according to (4.6), CA can be viewed as a weighted matrix factorization of $\left(p_{ij}/p_{i+}p_{+j} - 1\right)$ with weighting function $p_{i+}p_{+j}$.

The solution is found using the SVD as in Equation (4.2). The K-dimensional approximation of $\left(p_{ij}/p_{i+}p_{+j} - 1\right)$ is

$$\frac{p_{ij}}{p_{i+}p_{+j}} - 1 \approx \sum_{k=1}^{K} \sigma_k \phi_{ik} \gamma_{jk} = \boldsymbol{e}_i^T \boldsymbol{o}_j. \tag{4.7}$$

The matrix $[\boldsymbol{e}_i^T \boldsymbol{o}_j]$ minimizes (4.6) amongst all matrices of rank K in a weighted least-squares sense. The parameter vectors $\boldsymbol{e}_i$ and $\boldsymbol{o}_j$ can be represented, for example, as

$$\boldsymbol{e}_i = [\phi_{i1}\sigma_1, \phi_{i2}\sigma_2, \cdots, \phi_{iK}\sigma_K]^T \tag{4.8}$$

and

$$\boldsymbol{o}_j = [\gamma_{j1}, \gamma_{j2}, \cdots, \gamma_{jK}]^T \tag{4.9}$$

As described above, this representation $\boldsymbol{e}_i$ of target word i has the advantage that the χ^2-distance between target words i and i' in the original matrix is approximated by the Euclidean distance between $\boldsymbol{e}_i$ and $\boldsymbol{e}_{i'}$.

The parameter $\boldsymbol{e}_i$ can be adjusted by a singular value weighting exponent p, i.e., $\boldsymbol{e}_i = [\phi_{i1}\sigma_1^p, \phi_{i2}\sigma_2^p, \cdots, \phi_{iK}\sigma_K^p]^T$. Correspondingly, the alternative coordinate for the adjusted row i by a singular value weighting exponent is $[u_{i1}\sigma_1^p, u_{i2}\sigma_2^p, \cdots, u_{iK}\sigma_K^p]^T$.

4.3.3 Three variants of CA for word embeddings

We present three variants of CA. According to Stratos et al. (2015), word counts can be naturally modeled as Poisson variables. The square-root transformation of a Poisson variable leads to stabilization of the variance (Bartlett, 1936; Stratos et al., 2015). Stratos et al. (2015) proposed to combine CCA with the square-root transformation of the word-context matrix. Even though CA of a contingency table is equivalent to CCA of the data in the form of an indicator matrix, we call the proposal by Stratos et al. (2015) ROOT-CCA, to distinguish it from the alternative ROOT-CA, discussed later.

ROOT-CCA In ROOT-CCA, an SVD is performed on the matrix whose typical element is the square root of $x_{ij}/\sqrt{x_{i+}x_{+j}} = p_{ij}/\sqrt{p_{i+}p_{+j}}$, that is

$$\sqrt{\frac{p_{ij}}{\sqrt{p_{i+}p_{+j}}}} = \sum_{k=1}^{\min(I,J)} \sigma_k u_{ik} v_{jk}. \tag{4.10}$$

The reason that Stratos et al. (2015) ignore $p_{i+}p_{+j}$ in $p_{ij} - p_{i+}p_{+j}$ (compare Equation (4.2)) is that they believe that, when the sample size x_{++} is large, the first part $p_{ij}/\sqrt{(p_{i+}p_{+j})}$ in $(p_{ij} - p_{i+}p_{+j})/\sqrt{(p_{i+}p_{+j})}$ dominates the expression.

ROOT-CA Inspired by Stratos et al. (2015), we present CA of the square-root transformation of the word-context matrix (ROOT-CA) (Bartlett, 1936; Hsu & Culhane, 2023). ROOT-CA differs from ROOT-CCA in the following way. In the ROOT-CA, first we create a square-root transformation of the word-context matrix with elements $\sqrt{x_{ij}}$, and then we perform CA on this matrix. Let $p^*_{ij} = \frac{\sqrt{x_{ij}}}{\sum_{ij}\sqrt{x_{ij}}} = \frac{\sqrt{p_{ij}}}{\sum_{ij}\sqrt{p_{ij}}}$. Then ROOT-CA provides the decomposition

$$\frac{p^*_{ij} - p^*_{i+}p^*_{+j}}{\sqrt{p^*_{i+}p^*_{+j}}} = \sum_{k=1}^{\min(I-1,J-1)} \sigma_k u_{ik} v_{jk}. \tag{4.11}$$

ROOTROOT-CA According to Stratos et al. (2015), word counts can be naturally modeled as Poisson variables. In the Poisson distribution the mean and variance are identical. The phenomenon of the data having greater variability than expected based on a statistical model is called overdispersion (Agresti, 2007). In the context of CA, Nishisato et al. (2021) point out, generally speaking, a two-way contingency table is prone to overdispersion. Overdispersion may negatively affect the performance of CA (Beh et al., 2018; Nishisato et al., 2021).

Greenacre (2009, 2010), referring to Field et al. (1982), points out that in ecology abundance data is almost always highly over-dispersed and a particular school of ecologists routinely applies a fourth-root transformation before proceeding with the statistical analysis. Therefore we also study the effect of a root-root transformation before performing CA. We call it ROOTROOT-CA. That is, ROOTROOT-CA is a CA on the matrix with typical element $\sqrt{\sqrt{x_{ij}}}$ (Field et al., 1982). Suppose $p^{**}_{ij} = \frac{\sqrt{\sqrt{x_{ij}}}}{\sum_{ij}\sqrt{\sqrt{x_{ij}}}} = \frac{\sqrt{\sqrt{p_{ij}}}}{\sum_{ij}\sqrt{\sqrt{p_{ij}}}}$. Then, we have

$$\frac{p^{**}_{ij} - p^{**}_{i+}p^{**}_{+j}}{\sqrt{p^{**}_{i+}p^{**}_{+j}}} = \sum_{k=1}^{\min(I-1,J-1)} \sigma_k u_{ik} v_{jk}, \tag{4.12}$$

4.3.4 T-Test

The T-Test (TTEST) weighting scheme, described by Curran and Moens (2002) and Curran (2004), focuses on the matrix of standardized residuals, see Equation (4.1). Thus it is remarkably similar to CA, where the matrix of standardized residuals is decomposed. For a comparison between CA and TTEST weighting in word similarity tasks, as we will carry out below, the question is whether the performance is better on the matrix of standardized residuals, or on a low dimensional representation of this matrix provided by CA.

Inspired by Section 4.3.3, we also explore the performances of the matrix STRATOS-TTEST with typical element $\sqrt{p_{ij}/\sqrt{p_{i+}p_{+j}}}$ (compare Equation (4.10)), the matrix ROOT-TTEST with typical element $\left(p^*_{ij} - p^*_{i+}p^*_{+j}\right)/\sqrt{p^*_{i+}p^*_{+j}}$ (compare Equation (4.11)), and the matrix ROOTROOT-TTEST with typical element $\left(p^{**}_{ij} - p^{**}_{i+}p^{**}_{+j}\right)/\sqrt{p^{**}_{i+}p^{**}_{+j}}$ (compare Equation (4.12)).

4.4 PMI-based word embedding methods

4.4.1 PMI-SVD and PPMI-SVD

Pointwise mutual information (PMI) is an important concept in NLP. The PMI between a target word i and a context word j is defined as (Bullinaria & Levy, 2007; Levy & Goldberg, 2014; Levy et al., 2015; Jurafsky & Martin, 2023):

$$\text{PMI}(i,j) = \log \frac{p_{ij}}{p_{i+}p_{+j}} \tag{4.13}$$

i.e. the log of the contingency ratio $p_{ij}/\left(p_{i+}p_{+j}\right)$. If $p_{ij} = 0$, then $\text{PMI}(i,j) = \log 0 = -\infty$, and it is usual to set $\text{PMI}(i,j) = 0$ in this situation.

A common approach is to factorize the PMI matrix using SVD, which we call PMI-SVD. Thus the objective function is

$$\sum_{i,j}\left(\log \frac{p_{ij}}{p_{i+}p_{+j}} - \boldsymbol{e}_i^T \boldsymbol{o}_j\right)^2. \tag{4.14}$$

In terms of a weighted matrix factorization, PMI-SVD is the matrix factorization of the PMI matrix with the weighting function 1. The solution is provided directly via SVD. An SVD applied to the PMI matrix with elements $\log\left(p_{ij}/\left(p_{i+}p_{+j}\right)\right)$ yields

$$\log \frac{p_{ij}}{p_{i+}p_{+j}} = \sum_{k=1}^{\min(I,J)} \sigma_k u_{ik} v_{jk}, \tag{4.15}$$

where $\min(I,J)$ is the rank of the PMI matrix. The K-dimensional approximation of $\log\left(p_{ij}/\left(p_{i+}p_{+j}\right)\right)$ is

$$\log \frac{p_{ij}}{p_{i+}p_{+j}} \approx \sum_{k=1}^{K} \sigma_k u_{ik} v_{jk} = \boldsymbol{e}_i^T \boldsymbol{o}_j \tag{4.16}$$

where the matrix with elements $\boldsymbol{e}_i^T \boldsymbol{o}_j$ minimizes (4.14) amongst all matrices of rank K in a least squares sense, where $K \leq \min(I,J)$.

The parameters $\boldsymbol{e}_i$ and $\boldsymbol{o}_j$ can be represented as

$$\boldsymbol{e}_i = [u_{i1}\sigma_1, u_{i2}\sigma_2, \cdots, u_{iK}\sigma_K]^T \tag{4.17}$$

and

$$\boldsymbol{o}_j = \left[v_{j1}, v_{j2}, \cdots, v_{jK}\right]^T. \tag{4.18}$$

Thus the Euclidean distance between target words i and i' in the original matrix is approximated by the Euclidean distance between $\boldsymbol{e}_i$ and $\boldsymbol{e}_{i'}$. In practice, one regularly sees that the parameters $\boldsymbol{e}_i$ are adjusted by an exponent p used for weighting the singular values, i.e., $\boldsymbol{e}_i = \left[u_{i1}\sigma_1^p, u_{i2}\sigma_2^p, \cdots, u_{iK}\sigma_K^p\right]^T$, where p is usually set to 0 or 0.5 (Levy & Goldberg, 2014; Levy et al., 2015; Stratos et al., 2015).

It is worth noting that the elements in the PMI matrix, where word-context pairs that co-occur rarely are negative, but word-context pairs that never co-occur are set to 0 (Levy & Goldberg, 2014), are not monotonic transformations of observed counts divided by counts under independence. An alternative is the so-called positive PMI matrix, abbreviated as PPMI matrix. In the PPMI matrix all negative values are set to 0:

$$\text{PPMI}(i,j) = \max\left(\text{PMI}(i,j), 0\right) \tag{4.19}$$

In most applications, one makes use of the PPMI matrix instead of the PMI matrix (Salle et al., 2016). We call the factorization of the PPMI matrix using SVD PPMI-SVD (M. Zhang et al., 2022).

4.4.2 GloVe

The GloVe objective function to be minimized is (Pennington et al., 2014):

$$\sum_{i,j} f(x_{ij}) \left(\log x_{ij} - b_i - s_j - \boldsymbol{e}_i^T \boldsymbol{o}_j\right)^2 \tag{4.20}$$

where

$$f(x_{ij}) = \begin{cases} \left(x_{ij}/x_{\max}\right)^{\alpha} & \text{if } x_{ij} < x_{\max} \\ 1 & \text{otherwise} \end{cases}$$

In addition to parameter vectors $\boldsymbol{e}_i$ and $\boldsymbol{o}_j$, the scalar parameter terms b_i and s_j are referred to as *bias* of target word i and context word j, respectively. Pennington et al.

(2014) train the GloVe model using an adaptive gradient algorithm (AdaGrad) (Duchi et al., 2011). This algorithm trains only on the non-zero elements of a word-context matrix, as $f(0) = 0$, which avoids the appearance of the undefined $\log 0$ in Equation (4.20).

In the original proposal of GloVe (Pennington et al., 2014), $b_i = \log x_{i+}$ and then, due to the symmetric role of target word and context word, $s_j = \log x_{+j}$. Shi and Liu (2014) and Shazeer et al. (2016) show that the bias terms b_i and s_j are highly correlated with $\log x_{i+}$ and $\log x_{+j}$, respectively, in GloVe model training. This means that the GloVe model minimizes a weighted least squares loss function with the weighting function $f(x_{ij})$ and approximate fitting function $\log x_{ij} - \log x_{i+} - \log x_{+j} = \log\left(x_{ij}x_{++}/\left(x_{i+}x_{+j}\right)\right) - \log x_{++} = \log\left(p_{ij}/\left(p_{i+}p_{+j}\right)\right) - \log x_{++}$:

$$\sum_{i,j} f(x_{ij})\left(\log \frac{p_{ij}}{p_{i+}p_{+j}} - \log x_{++} - \boldsymbol{e}_i^T \boldsymbol{o}_j\right)^2 \tag{4.21}$$

4.4.3 Skip-gram with negative sampling

SGNS stands for skip-gram with negative sampling of word2vec embeddings (Mikolov, Chen, et al., 2013; Mikolov, Sutskever, et al., 2013). The algorithms used in SGNS are stochastic gradient descent and backpropagation (Rumelhart et al., 1986; Rong, 2014). SGNS trains word embeddings on every word of the corpus one by one.

Levy and Goldberg (2014) showed that SGNS implicitly factorizes a PMI matrix shifted by $\log n$:

$$\log \frac{p_{ij}}{p_{i+}p_{+j}} - \log n \approx \boldsymbol{e}_i^T \boldsymbol{o}_j \tag{4.22}$$

where n is the number of negative samples. According to Levy and Goldberg (2014) and Shazeer et al. (2016), the objective function of SGNS is approximately a minimization of the difference between $\boldsymbol{e}_i^T \boldsymbol{o}_j$ and $\log\left(p_{ij}/\left(p_{i+}p_{+j}\right) - \log n\right)$, tempered by a monotonically increasing weighting function of the observed co-occurrence count x_{ij}, $g(x_{ij})$:

$$\sum_{i,j} g(x_{ij})\left(\log \frac{p_{ij}}{p_{i+}p_{+j}} - \log n - \boldsymbol{e}_i^T \boldsymbol{o}_j\right)^2 \tag{4.23}$$

This shows that SGNS differs from GloVe in the use of n instead of x_{++}, and $g(x_{ij})$ instead of $f(x_{ij})$.

4.5 Relationships of CA to PMI-based models

4.5.1 CA and PMI-SVD / PPMI-SVD

In this section, we discuss PMI-SVD and PPMI-SVD together, as PMI and PPMI are the same except that in PPMI all negative values of PMI are set to 0.

CA is closely related to PMI-SVD. This becomes clear by comparing $\left(p_{ij}/\left(p_{i+}p_{+j}\right)-1\right)$ in (4.6) with $\text{PMI}(i,j) = \log\left(p_{ij}/\left(p_{i+}p_{+j}\right)\right)$ in (4.14). The relation lies in a Taylor expansion of $\log\left(p_{ij}/\left(p_{i+}p_{+j}\right)\right)$, namely that, if x is small, $\log(1+x) \approx x$ (Van der Heijden et al., 1989). Substituting x with $p_{ij}/(p_{i+}p_{+j}) - 1$ leads to:

$$\log \frac{p_{ij}}{p_{i+}p_{+j}} \approx \frac{p_{ij}}{p_{i+}p_{+j}} - 1 \tag{4.24}$$

This illustrates that if $\left(p_{ij}/p_{i+}p_{+j}-1\right)$ is small, the objective function of CA approximates

$$\sum_{i,j} p_{i+}p_{+j}\left(\log \frac{p_{ij}}{p_{i+}p_{+j}} - \boldsymbol{e}_i^T \boldsymbol{o}_j\right)^2. \tag{4.25}$$

From Equation (4.25) it follows that CA is approximately a weighted matrix factorization of $\log\left(p_{ij}/\left(p_{i+}p_{+j}\right)\right)$ with weighting function $p_{i+}p_{+j}$.

Comparing Equation (4.25) with Equation (4.14), both CA and PMI-SVD can be taken as weighted least squares methods having approximately the same fitting functions, namely $\left(p_{ij}/p_{i+}p_{+j}-1\right)$ for CA and $\log\left(p_{ij}/\left(p_{i+}p_{+j}\right)\right)$ for PMI-SVD. Both make use of an SVD.

However, they use different weighting functions, namely $p_{i+}p_{+j}$ in CA and 1 in PMI-SVD. It has been argued that equally weighting errors in the objective function, as is the case in PMI-SVD, is not a good approach (Salle et al., 2016; Salle & Villavicencio, 2023). For example, Salle and Villavicencio (2023) presented the reliability principle, that the objective function should have a weight on the reconstruction error that is a monotonically increasing function of the marginal frequencies of word and of context. On the other hand, CA, unlike PMI-SVD, weights errors in the objective function with a weighting function equal to the product of the marginal proportions of word and context (Greenacre, 1984, 2017; Beh & Lombardo, 2021).

4.5.1.1 PMI-GSVD

The weighting function of PMI-SVD is 1 while in the approximate version of CA it is $p_{i+}p_{+j}$. Therefore, we also investigate the performance of a weighted factorization of the PMI matrix, where $p_{i+}p_{+j}$ is the weighting function:

$$\sum_{i,j} p_{i+}p_{+j}\left(\log \frac{p_{ij}}{p_{i+}p_{+j}} - \boldsymbol{e}_i^T \boldsymbol{o}_j\right)^2. \tag{4.26}$$

Similar with CA, we use generalized SVD (GSVD) to find the optimum of the objective function (PMI-GSVD). That is, an SVD is applied as follows:

$$\sqrt{p_{i+}p_{+j}}\log\frac{p_{ij}}{p_{i+}p_{+j}} = \sum_{k=1}^{\min(I,J)}\sigma_k u_{ik}v_{jk}, \tag{4.27}$$

We call the matrix with typical element $\sqrt{p_{i+}p_{+j}}\log\frac{p_{ij}}{p_{i+}p_{+j}}$ the WPMI matrix.

4.5.2 CA and GloVe

Both CA and GloVe are weighted least squares methods. The weighting function in GloVe is $f(x_{ij})$, which is defined uniquely for each element of the word-context matrix, while the weighting function $p_{i+}p_{+j}$ in CA is defined by the row and column margins.

In the approximate fitting function of GloVe, $\log\left(p_{ij}/\left(p_{i+}p_{+j}\right)\right) - \log x_{++}$, the term $\log x_{++}$ can be considered as a shift of $\log\left(p_{ij}/\left(p_{i+}p_{+j}\right)\right)$. And as we showed in Section 4.5.1, the fitting function of CA is approximately $\log\left(p_{ij}/\left(p_{i+}p_{+j}\right)\right)$ when p_{ij} is close to $p_{i+}p_{+j}$. Thus, from a comparison of the objective functions of CA and GloVe, it is natural to expect that these two methods will yield similar results if $\left(p_{ij}/p_{i+}p_{+j} - 1\right)$ is small.

In comparing the algorithms of these two methods, we find that CA uses SVD while GloVe uses AdaGrad. These two algorithms have their own advantages and disadvantages. On the one hand, the AdaGrad algorithm trains GloVe only on the nonzero elements of word-context matrix, one by one, while in CA the SVD decomposes the entire word-context matrix in full in one step. On the other hand, the SVD always finds the global minimum while the AdaGrad algorithm cannot guarantee the global minimum.

4.5.3 CA and SGNS

By comparing Equations (4.23) and (4.25), both the approximation of CA and of SGNS are found by weighted least squares methods. The weighting function in SGNS is $g(x_{ij})$, which is defined for each element of word-context matrix where frequent word-context pairs pay more for deviations than infrequent ones (Levy & Goldberg, 2014), while the weighting function in CA is defined by the row and column margins, i.e. $p_{i+}p_{+j}$.

In the fitting function of the approximation of SGNS, $\log\left(p_{ij}/\left(p_{i+}p_{+j}\right)\right) - \log n$, the term $\log n$ can be considered as a shift of $\log\left(p_{ij}/\left(p_{i+}p_{+j}\right)\right)$. As shown in Section 4.5.1, the approximate fitting function in CA is $\log\left(p_{ij}/\left(p_{i+}p_{+j}\right)\right)$. Thus, considering the objective function view, both the approximation of CA and of SGNS make use of the PMI matrix.

Although the approximate objective function of SGNS is similar to that of CA, the training processing for SGNS is different from that of CA. SGNS trains word embeddings on the words of a corpus, one by one, to maximize the probabilities of target words and context words co-occurrence, and to minimize the probabilities between target words and randomly sampled words, by updating the vectors of target words and context words. In contrast, CA first counts all co-occurrences in the corpus and then performs SVD on the matrix of standardized residuals to obtain the vectors of target words and context words at once.

4.6 Two corpora and five word similarity datasets

All methods are trained on two corpora: Text8 (*Text8 dataset*, 2006) and British National Corpus (BNC) (BNC Consortium, 2007), respectively. Text8 is a widely used corpus in NLP (Xin, Yuan, He, & Jose, 2018; Roesler, Aly, Taniguchi, & Hayashi, 2019; Podkorytov, Biś, Cai, Amirizirtol, & Liu, 2020; S. Guo & Yao, 2021). It includes more than 17 million words from Wikipedia (Peng & Feldman, 2017) and only consists of lowercase English characters and spaces. Words that appeared less than 100 times in the corpus are ignored, resulting in a vocabulary of 11,815 terms.

BNC is from a representative variety of sources and is widely used (Raphael, 2023; Samuel, Kutuzov, Øvrelid, & Velldal, 2023). Data cited herein have been extracted from the British National Corpus, distributed by the University of Oxford on behalf of the BNC Consortium. We remove English punctuation and numbers and set words in lowercase form. Words that appeared less than 500 times in the corpus are ignored, resulting in a vocabulary of 11,332 terms.

Following previous studies (Levy et al., 2015; Pakzad & Analoui, 2021), we evaluate each word embeddings method on word similarity tasks using the Spearman's correlation coefficient ρ. We use five popular word similarity datasets: WordSim353 (Finkelstein et al., 2002), MEN (Bruni, Boleda, Baroni, & Tran, 2012), Mechanical Turk (Radinsky, Agichtein, Gabrilovich, & Markovitch, 2011), Rare (Luong, Socher, & Manning, 2013), and SimLex-999 (F. Hill, Reichart, & Korhonen, 2015). All these datasets consist of word pairs together with human-assigned similarity scores. For example, in WordSim353, one word pair is (tiger, cat) with human assigned similarity score 7.35. Out-of-vocabulary words are removed from all test sets. I.e., if either tiger or cat doesn't occur in the vocabularies of the 11,815 terms created by Text8 corpus, we delete (tiger, cat). Thus for evaluating the different word embedding methods in Text8 277 word pairs with scores are kept in WordSim353 instead of the original 353 word pairs. Table 4.1 provides the number of word pairs used by the datasets in Text8 and BNC.

After calculating the solutions for CA, PMI-SVD, PPMI-SVD, PMI-GSVD, ROOT-CA, ROOTROOT-CA, ROOT-CCA, GloVe, and SGNS, we obtain the word embeddings. We calculate the cosine similarity for each word pair in each word similarity dataset. For example, for WordSim353 using Text8, we obtain 277 cosine similarities.

Table 4.1: Datasets for word similarity evaluation.

Dataset	Word pairs	Word pairs in Text8	Word pairs in BNC
WordSim353	353	277	276
MEN	3000	1544	1925
Turk	287	221	197
Rare	2034	205	204
SimLex-999	999	726	847

The Spearman's correlation coefficient ρ between these similarities and the human similarity scores is calculated to evaluate these word embedding methods. Larger values are better.

4.7 Study setup

4.7.1 SVD-based methods

CA, PMI-SVD, PPMI-SVD, PMI-GSVD, ROOT-CA, ROOTROOT-CA, and ROOT-CCA are SVD-based dimensionality reduction methods. First, we create a word-context matrix of size 11,815×11,815 and 11,332×11,332 based on Text8 and BNC, respectively. We use a window of size 2, i.e., two words to each side of the target word. A context word one token and two tokens away will be counted as 1/1 and 1/2 of an occurrence, respectively. Then we perform SVD on the related matrices. We use the svd function from scipy.linalg in Python to calculate the SVD of a matrix, and obtain singular values σ_k, left singular vectors u_{ik}, and right singular vectors v_{jk}. We obtain the word embeddings as $\boldsymbol{e}_i = \left[u_{i1}\sigma_1^p, u_{i2}\sigma_2^p, \cdots, u_{ik}\sigma_k^p\right]^T$.

The choices of the exponent weighting p and number of dimensions k are important for SVD-based methods. In the context of PPMI-SVD and ROOT-CCA p is regularly set to $p = 0$ or $p = 0.5$ (Levy & Goldberg, 2014; Levy et al., 2015; Stratos et al., 2015). For $p = 0$, we have the standard coordinates with $\boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{V}^T\boldsymbol{V} = \boldsymbol{I}$. For $p = 0.5$, we have $\boldsymbol{A}_k = \boldsymbol{U}_k\Sigma_k\boldsymbol{V}_k^T = (\boldsymbol{U}_k\Sigma_k^{1/2})(\boldsymbol{V}_k\Sigma_k^{1/2})^T$. That is, the target words $\boldsymbol{U}_k\Sigma_k^{1/2}$ and context words $\boldsymbol{V}_k\Sigma_k^{1/2}$ reconstruct the decomposed matrix $\boldsymbol{A}_k$. The two created word-context matrices based on Text8 and BNC are symmetric, so the matrices to be decomposed are also symmetric. For the SVD of a symmetric matrix, using the target words $\boldsymbol{U}_k\Sigma_k^{1/2}$ for word embeddings is equivalent to using the context words $\boldsymbol{V}_k\Sigma_k^{1/2}$ for word embeddings. We vary the number of dimensions k from 2, 50, 100, 200, $\cdots$, 1,000, 2,000, $\cdots$, 10,000.

4.7.2 GloVe and SGNS

We use the public implementation by Pennington et al. (2014) to perform GloVe and choose the default hyperparameters. Pennington et al. (2014) proposed to use the

context vectors $\boldsymbol{o}_j$ in addition to target word vectors $\boldsymbol{e}_i$. Here, we only use target word vectors $\boldsymbol{e}_i$, set window size to 2 and set vocab minimum count to 100 for Text8 and 500 for BNC, in the same way as for the SVD-based methods to keep the settings consistent. We vary the dimension k of word embeddings from 200 to 600 with intervals of 100.

We use the public implementation by Mikolov, Sutskever, et al. (2013) to perform SGNS, and use the vocabulary created by GloVe as the input of SGNS. We choose the default values except for the dimensions k of word embeddings and window size, which are chosen in the same way as in GloVe, to keep the settings consistent.

4.8 Results

We make a distinction between conditions where no dimensionality reduction takes place, and conditions where dimensionality reduction is used. For no dimensionality reduction we compare TTEST, PMI, PPMI, WPMI, ROOT-TTEST, ROOTROOT-TTEST, STRATOS-TTEST. For dimensionality reduction we first compare CA with the more standard methods PMI-SVD, PPMI-SVD, PMI-GSVD, GloVe, SGNS, and then compare variants of CA.

4.8.1 TTEST, PMI, PPMI, WPMI, ROOT-TTEST, ROOTROOT-TTEST, and STRATOS-TTEST

First, we compare methods where no dimensionality reduction takes place. We show the Spearman's correlation coefficient ρ for the TTEST, PMI, PPMI, WPMI, ROOT-TTEST, ROOTROOT-TTEST, and STRATOS-TTEST matrices in Table 4.2. The results for the five word similarity datasets and the two corpora show that (1) either ROOT-TTEST or ROOTROOT-TTEST is best, and (2) ROOT-TTEST is consistently better than PPMI, PMI, STRATOS-TTEST, and WPMI. In the Total column of the block at the bottom of the table we provide the sum of ρ-values for all five datasets and two corpora. Overall, ROOT-TTEST and ROOTROOT-TTEST perform best, closely followed by PPMI and TTEST. PMI follows at some distance, and last, we find STRATOS-TTEST and WPMI.

4.8.2 CA, PMI-SVD, PPMI-SVD, PMI-GSVD, GloVe, and SGNS

Next, we compare CA (RAW-CA in Table 4.3) with the PMI-based methods PMI-SVD, PPMI-SVD, PMI-GSVD, GloVe, and SGNS. Table 4.3 has a left part, where $p = 0$, and a right part, where $p = 0.5$. As p does not exist in GloVe and SGNS, these methods have identical values for $p = 0$ and $p = 0.5$. Plots for ρ as a function of k for SVD-based methods are in Supplementary materials B.

Comparing the last block of Table 4.3 with the last block of Table 4.2 reveals that, overall, dimensionality reduction is beneficial for the size of ρ, as CA, PMI-SVD,

Table 4.2: Correlation coefficient ρ for TTEST, PMI, PPMI, WPMI, ROOT-TTEST, ROOTROOT-TTEST, and STRATOS-TTEST matrices.

		Text8	BNC	Total
WordSim353	TTEST	0.588	0.427	1.015
	PMI	0.587	0.292	0.879
	PPMI	0.609	0.505	1.115
	WPMI	0.233	0.221	0.454
	ROOT-TTEST	**0.658**	**0.539**	**1.197**
	ROOTROOT-TTEST	0.646	0.495	1.141
	STRATOS-TTEST	0.438	0.314	0.752
MEN	TTEST	0.248	0.260	0.509
	PMI	0.269	0.224	0.494
	PPMI	0.253	0.284	0.537
	WPMI	0.132	0.171	0.303
	ROOT-TTEST	0.305	**0.293**	**0.598**
	ROOTROOT-TTEST	**0.317**	0.263	0.580
	STRATOS-TTEST	0.156	0.130	0.286
Turk	TTEST	0.619	0.649	1.268
	PMI	0.629	0.514	1.143
	PPMI	0.651	0.625	1.276
	WPMI	0.343	0.417	0.760
	ROOT-TTEST	0.666	**0.659**	**1.325**
	ROOTROOT-TTEST	**0.667**	0.616	1.283
	STRATOS-TTEST	0.561	0.525	1.086
Rare	TTEST	0.392	0.428	0.820
	PMI	0.335	0.289	0.624
	PPMI	0.328	0.363	0.691
	WPMI	0.252	0.255	0.506
	ROOT-TTEST	0.389	**0.477**	0.866
	ROOTROOT-TTEST	**0.418**	0.454	**0.872**
	STRATOS-TTEST	0.243	0.196	0.439
SimLex-999	TTEST	0.220	0.230	0.450
	PMI	0.257	0.168	0.425
	PPMI	0.251	0.277	0.528
	WPMI	0.139	0.118	0.257
	ROOT-TTEST	**0.276**	**0.280**	**0.556**
	ROOTROOT-TTEST	0.271	0.239	0.509
	STRATOS-TTEST	0.181	0.125	0.306
Total	TTEST	2.067	1.994	4.061
	PMI	2.078	1.487	3.565
	PPMI	2.092	2.054	4.146
	WPMI	1.098	1.182	2.280
	ROOT-TTEST	2.293	**2.249**	**4.542**
	ROOTROOT-TTEST	**2.319**	2.067	4.386
	STRATOS-TTEST	1.579	1.289	2.869

PPMI-SVD, PMI-GSVD, ROOT-CA, ROOTROOT-CA, and ROOT-CCA do better than their respective counterparts TTEST, PMI, PPMI, WPMI, ROOT-TTEST, ROOTROOT-TTEST, and STRATOS-TTEST. For TTEST the improvement due to using SVD is less than for PMI, PPMI, WPMI, ROOT-TTEST, STRATOS-TTEST; for WPMI and STRATOS-TTEST the improvement due to using SVD is more than for TTEST, PPMI, ROOT-TTEST, and ROOTROOT-TTEST, which is a result consistent for each corpus and each word similarity dataset.

For an overall comparison of the dimensionality reduction methods, we study the block at the bottom of Table 4.3, which provides the sum of the ρ-values over the five word similarity datasets. For both $p = 0$ and $p = 0.5$, among RAW-CA, PMI-SVD, PPMI-SVD, PMI-GSVD, GloVe, and SGNS, overall PMI-SVD and PPMI-SVD perform best, closely followed by SGNS. RAW-CA and PMI-GSVD follow at some distance, and last, we find GloVe. The popular method GloVe does not perform well. Possibly the conditions of the study are not optimal for GloVe, as the Text8 and BNC corpora are, with 11,815 and 11,332 terms respectively, possibly too small to obtain reliable results (Jiang, Yu, Hsieh, & Chang, 2018).

As the focus in this paper is on the performance of CA, we give some extra attention to RAW-CA and the similar PMI-GSVD. Even though CA and PMI-GSVD have the same weighting function $p_{i+}p_{+j}$, and should be close when $p_{ij}/\left(p_{i+}p_{+j}\right)-1$ is small (compare the discussion around Equations (4.24, 4.25)) their performances are rather different. This may be because there are extremely large values (larger than 35,000) in the fitting function $\left(p_{ij}/\left(p_{i+}p_{+j}\right)-1\right)$ of CA, which makes the fitting function of CA not close to the fitting function $\log\left(p_{ij}/\left(p_{i+}p_{+j}\right)\right)$ of PMI-GSVD.

When we compare PMI-GSVD with PMI-SVD, we are surprised to find that weighting rows and columns appears to decrease the values of ρ. This is in contrast with the reliability principle of Salle and Villavicencio (2023) discussed above.

We now discuss why PMI-SVD and PPMI-SVD do better than PMI-GSVD. It turns out that the number and sizes of extreme values in the matrix WPMI decomposed by PMI-GSVD are much larger than in PMI and PPMI, and this results in PMI-GSVD dimensions being dominated by single words. We only include non-zero elements in the PMI matrix as the PMI matrix is sparse: 94.2% of the entries are zero for Text8; for a fair comparison, the corresponding 94.2% of entries in the PPMI and WPMI matrices are also ignored. Following box plot methodology (Tukey, 1977; Schwertman, Owens, & Adnan, 2004; Dodge, 2008), extreme values are determined as follows: let q_1 and q_3 be the first and third sample quartiles, and let $f_1 = q_1 - 1.5(q_3 - q_1)$, $f_3 = q_3 + 1.5(q_3 - q_1)$. Then extreme values are defined as values less than f_1 (LTf_1) or greater than f_3 (GTf_3). The first three rows in Table 4.4 show the number of extreme elements in the PMI, PPMI, WPMI matrices. The number of extreme values of the WPMI matrix (1,384,231) is much larger than that of PMI and PPMI (32,319 and 27,984). Furthermore, in WPMI the extremeness of values is much larger than in PMI and PPMI. Let the averaged contribution of each cell, expressed as a proportion, be $1/(11,815 \times 11,815)$. However, in WPMI, the most extreme entry, found for (the, the),

Table 4.3: Correlation coefficient ρ for SVD-based methods with $p = 0, 0.5$ and for GloVe and SGNS.

		$p = 0$					$p = 0.5$				
		Text8		BNC			Text8		BNC		
		k	ρ	k	ρ	total	k	ρ	k	ρ	total
WordSim353	RAW-CA	600	0.578	400	0.465	1.043	9000	0.609	10000	0.498	1.107
	PMI-SVD	400	0.675	600	0.628	1.303	400	0.683	500	0.579	1.262
	PPMI-SVD	400	0.681	700	0.628	1.309	200	0.694	2000	0.623	1.317
	GloVe	200	0.422	600	0.522	0.943	200	0.422	600	0.522	0.943
	SGNS	300	0.668	600	0.551	1.219	300	0.668	600	0.551	1.219
	PMI-GSVD	700	0.512	600	0.468	0.980	6000	0.548	3000	0.449	0.997
	ROOT-CA	300	0.668	400	0.623	1.291	500	0.688	900	**0.657**	**1.345**
	ROOTROOT-CA	200	**0.692**	200	**0.635**	**1.327**	300	**0.697**	400	0.630	1.327
	ROOT-CCA	100	0.682	700	0.627	1.310	300	0.684	600	0.620	1.304
MEN	RAW-CA	300	0.223	600	0.293	0.516	7000	0.256	9000	0.299	0.556
	PMI-SVD	800	0.328	700	0.393	0.721	600	0.317	2000	0.357	0.674
	PPMI-SVD	800	0.336	500	0.394	0.730	800	0.324	1000	0.358	0.681
	GloVe	300	0.175	600	0.310	0.485	300	0.175	600	0.310	0.485
	SGNS	400	0.295	400	0.333	0.627	400	0.295	400	0.333	0.627
	PMI-GSVD	800	0.267	600	0.318	0.585	5000	0.256	3000	0.308	0.564
	ROOT-CA	800	0.325	500	**0.400**	0.725	9000	0.324	800	**0.374**	**0.698**
	ROOTROOT-CA	600	**0.340**	400	0.396	**0.735**	1000	**0.332**	4000	0.359	0.690
	ROOT-CCA	600	0.315	400	0.392	0.706	900	0.298	800	0.355	0.653
Turk	RAW-CA	400	0.549	100	0.562	1.111	400	0.592	10000	0.588	1.181
	PMI-SVD	100	0.656	50	0.652	1.308	300	**0.677**	500	0.661	1.338
	PPMI-SVD	50	0.668	50	0.671	1.339	50	**0.677**	50	0.683	**1.361**
	GloVe	600	0.502	200	0.540	1.042	600	0.502	200	0.540	1.042
	SGNS	200	0.651	300	0.650	1.302	200	0.651	300	0.650	1.302
	PMI-GSVD	900	0.495	200	0.506	1.000	5000	0.563	10000	0.584	1.147
	ROOT-CA	50	0.649	50	**0.695**	**1.344**	100	0.661	50	**0.684**	1.345
	ROOTROOT-CA	50	**0.669**	50	0.666	1.334	50	0.664	300	0.673	1.337
	ROOT-CCA	50	0.633	50	0.672	1.305	100	0.665	100	0.678	1.343
Rare	RAW-CA	600	0.396	500	0.450	0.846	900	0.411	3000	0.465	0.875
	PMI-SVD	100	0.476	700	0.480	0.957	300	0.471	5000	0.464	0.936
	PPMI-SVD	100	0.483	400	0.470	0.952	100	0.475	6000	0.469	0.944
	GloVe	400	0.181	600	0.379	0.560	400	0.181	600	0.379	0.560
	SGNS	600	0.456	200	**0.532**	**0.988**	600	0.456	200	**0.532**	0.988
	PMI-GSVD	400	0.451	500	0.418	0.869	900	0.431	600	0.429	0.860
	ROOT-CA	400	0.468	400	0.501	0.970	600	**0.479**	7000	0.526	**1.006**
	ROOTROOT-CA	100	**0.503**	500	0.476	0.978	100	0.475	4000	0.478	0.953
	ROOT-CCA	200	0.469	200	0.505	0.974	600	0.469	900	0.511	0.979
SimLex-999	RAW-CA	4000	0.219	2000	0.322	0.541	8000	0.243	7000	0.327	0.571
	PMI-SVD	700	0.310	900	0.409	0.719	3000	0.315	900	0.372	0.687
	PPMI-SVD	700	0.309	500	0.393	0.702	3000	0.308	500	0.368	0.676
	GloVe	500	0.148	500	0.255	0.403	500	0.148	500	0.255	0.403
	SGNS	600	0.306	400	0.376	0.682	600	0.306	400	0.376	0.682
	PMI-GSVD	900	0.272	4000	0.365	0.637	5000	0.271	3000	0.312	0.583
	ROOT-CA	2000	0.295	900	0.415	0.710	5000	0.309	2000	**0.395**	**0.704**
	ROOTROOT-CA	700	**0.321**	900	0.410	**0.731**	700	**0.317**	900	0.376	0.693
	ROOT-CCA	1000	0.294	1000	**0.421**	0.715	7000	0.303	2000	0.391	0.693
total	RAW-CA		1.965		2.092	4.057		2.111		2.178	4.290
	PMI-SVD		2.445		2.562	5.007		2.465		2.433	4.897
	PPMI-SVD		2.476		2.556	5.033		2.478		2.501	4.979
	GloVe		1.427		2.006	3.433		1.427		2.006	3.433
	SGNS		2.376		2.442	4.819		2.376		2.442	4.819
	PMI-GSVD		1.997		2.075	4.072		2.069		2.082	4.151
	ROOT-CA		2.405		**2.635**	5.039		2.462		**2.637**	**5.098**
	ROOTROOT-CA		**2.525**		2.582	**5.107**		**2.484**		2.515	4.999
	ROOT-CCA		2.393		2.617	5.011		2.417		2.555	4.972

Table 4.4: Text8: the number of extreme values

	LT f_1	GT f_3	total
PMI	4,335	27,984	32,319
PPMI	0	27,984	27,984
WPMI	1,038,236	345,995	1,384,231
TTEST	50,560	627,046	677,606
ROOT-TTEST	5,985	448,860	454,845
ROOTROOT-TTEST	4,942	396,437	401,379
STRATOS-TTEST	0	400,703	400,703

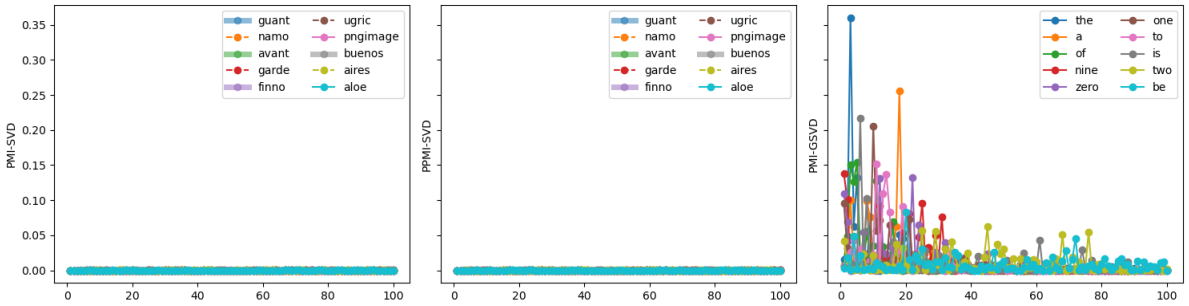


Figure 4.1: Text8: the contribution of the rows, corresponding to the top 10 extreme values, to the first 100 dimensions of PMI-SVD, PPMI-SVD, PMI-GSVD.

contributes around 0.01126 to the total inertia. In PMI (PPMI) the most extreme entry is (guant, namo) or (namo, guant) and contributes around 3.1×10^{-6} (3.2×10^{-6}) to the total inertia. Figure 4.1 shows the contribution of the rows for the corresponding to top 10 extreme values, to the first 100 dimensions of PMI-SVD, PPMI-SVD, PMI-GSVD. The rows, corresponding to the top extreme values in the WPMI matrix, take up a much bigger contribution to the first dimensions of PMI-GSVD. For example, in PMI-GSVD, the "the" row contributes more than 0.3 to the third dimension, while in PMI-SVD and PPMI-SVD, the contributions are much more even. Thus the PMI-GSVD solution is hampered by extreme cells in the WPMI matrix that is decomposed. Similar results can be found for BNC in Supplementary materials C.

4.8.3 The results for three variants of CA

Now we compare the three variants of CA (ROOT-CA, ROOTROOT-CA, ROOT-CCA) with CA-RAW and the winner of the PMI-based methods, PPMI-SVD.

First, in Table 4.3, the three variants of CA perform much better than RAW-CA in each word similarity dataset and each corpus, both for $p = 0$ and $p = 0.5$. In the block at the bottom of Table 4.3, overall, the performance of the three variants is similar, where ROOT-CA outperforms ROOT-CCA slightly.

The lower part of Table 4.4 shows the number of extreme values of TTEST, ROOT-TTEST, ROOTROOT-TTEST and STRATOS-TTEST matrices for Text8. Similar with

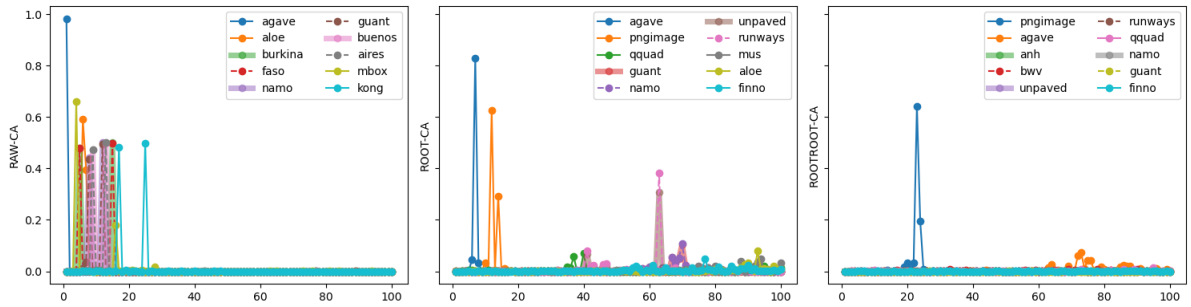


Figure 4.2: Text8: the contribution of the rows, corresponding to top 10 extreme values, to first 100 dimensions of RAW-CA, ROOT-CA, ROOTROOT-CA.

PMI, PPMI, WPMI, 94.2% of entries are ignored. The number of extreme values of the TTEST matrix (677,606) is larger than that of ROOT-TTEST, ROOTROOT-TTEST and STRATOS-TTEST (454,845, 401,379, and 400,703). Furthermore, in TTEST the extremeness of the extreme values is larger than those in ROOT-TTEST, ROOTROOT-TTEST, and STRATOS-TTEST. For example, in TTEST the most extreme entry (agave, agave) contributes around 0.02117 to the total inertia, while in ROOT-TTEST, ROOTROOT-TTEST, and STRATOS-TTEST, the most extreme entries (agave, agave), (pngimage, pngimage), and (agave, agave) contribute around 0.00325, 0.00119, and 0.00017, respectively. Figure 4.2 shows the contribution of the rows for the top 10 extreme values, to the first 100 dimensions of RAW-CA, ROOT-CA, and ROOTROOT-CA (The corresponding plot about ROOT-CCA is in Supplementary materials D). In RAW-CA, the rows, corresponding to top extreme values of TTEST, take up a big contribution to the first dimensions of RAW-CA. For example, in RAW-CA, the "agave" row contributes around 0.983 to the first dimension, while in ROOT-CA and ROOTROOT-CA, the contributions are much smaller which also holds for ROOT-CCA. Similar results can be found for BNC in Supplementary materials E. Thus, we infer that the extreme values in TTEST are the important reason that RAW-CA performs badly.

Second, in the rows of the block at the bottom of Table 4.3, the overall performances of ROOT-CA, ROOTROOT-CA, ROOT-CCA are comparable to or sometimes slightly better than PPMI-SVD. Specifically, ROOTROOT-CA and ROOT-CA achieve the highest ρ for Text8 and BNC corpora, respectively. Based on these results, no matter what we know about the corpus, ROOTROOT-CA and ROOT-CA appear to have potential to improve the performance in NLP tasks.

4.9 Conclusion and discussion

PMI is an important concept in natural language processing. In this paper, we theoretically compare CA with three PMI-based methods with respect to their objective functions. CA is a weighted factorization of a matrix where the fitting function is

$\left(p_{ij}/\left(p_{i+}p_{+j}\right)-1\right)$ and the weighting function is the product of row margins and column margins $p_{i+}p_{+j}$. When the elements in the fitting function $\left(p_{ij}/\left(p_{i+}p_{+j}\right)-1\right)$ of CA are small, CA is close to a weighted factorization of the PMI matrix where the weighting function is the product $p_{i+}p_{+j}$. This is because $\left(p_{ij}/\left(p_{i+}p_{+j}\right)-1\right)$ is close to $\log\left(p_{ij}/\left(p_{i+}p_{+j}\right)\right)$ when $\left(p_{ij}/\left(p_{i+}p_{+j}\right)-1\right)$ is small.

The extracted word-context matrices are prone to overdispersion. To remedy the overdispersion, we presented ROOTROOT-CA. That is, we perform CA on the root-root transformation of the word-context matrix. We also apply CA to the square-root transformation of the word-context matrix (ROOT-CA). In addition, we present ROOT-CCA, described in Stratos et al. (2015), which is similar with ROOT-CA. The empirical comparison on word similarity tasks shows that ROOTROOT-CA achieves the best overall results in the Text8 corpus, and ROOT-CA achieves the best overall results in the BNC corpus. Overall, the performance of ROOT-CA and ROOTROOT-CA is slightly better than the performance of PMI-based methods.

Concluding, our theoretical and empirical comparisons about CA and PMI-based methods shed new light on SVD-based and PMI-based methods. Our results show that, regularly, in NLP tasks the performance can be improved by making use of ROOT-CA and ROOTROOT-CA.

Appendix 4.A An alternative coordinates system for CA

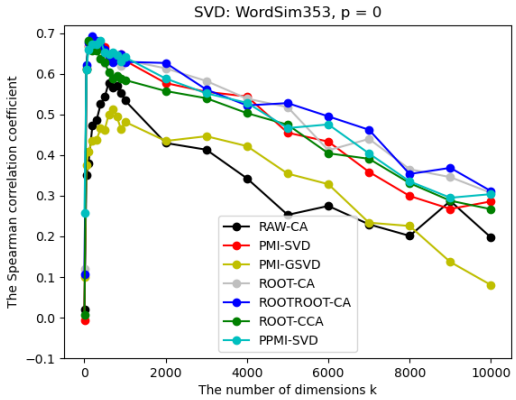
For row points i and i', with coordinates $\sigma_k \phi_{ik}$ and $\sigma_k \phi_{i'k}$ on dimension k in $K-$dimensional space we have

$$\begin{aligned}
\text{cosine}(\text{row}_i, \text{row}_{i'}) &= \frac{\sum_{k=1}^{K} (\phi_{ik}\sigma_k)(\phi_{i'k}\sigma_k)}{\sqrt{\sum_{k=1}^{K} (\phi_{ik}\sigma_k)^2 \cdot \sum_{k=1}^{K} (\phi_{i'k}\sigma_k)^2}} \\
&= \frac{\sum_{k=1}^{K} \left(p_{i+}^{-\frac{1}{2}} u_{ik}\sigma_k\right)\left(p_{i'+}^{-\frac{1}{2}} u_{i'k}\sigma_k\right)}{\sqrt{\sum_{k=1}^{K} \left(p_{i+}^{-\frac{1}{2}} u_{ik}\sigma_k\right)^2 \cdot \sum_{k=1}^{K} \left(p_{i'+}^{-\frac{1}{2}} u_{i'k}\sigma_k\right)^2}} \\
&= \frac{\sum_{k=1}^{K} (u_{ik}\sigma_k)(u_{i'k}\sigma_k)}{\sqrt{\sum_{k=1}^{K} (u_{ik}\sigma_k)^2 \cdot \sum_{k=1}^{K} (u_{i'k}\sigma_k)^2}},
\end{aligned} \tag{4.28}$$

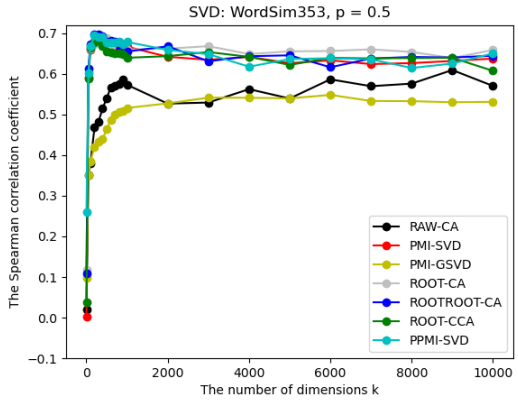
so the terms $p_{i+}^{-\frac{1}{2}}$ and $p_{i'+}^{-\frac{1}{2}}$ drop out of the equation. A similar result is found for column points.

Appendix 4.B Plots for ρ as a function of k for SVD-based methods

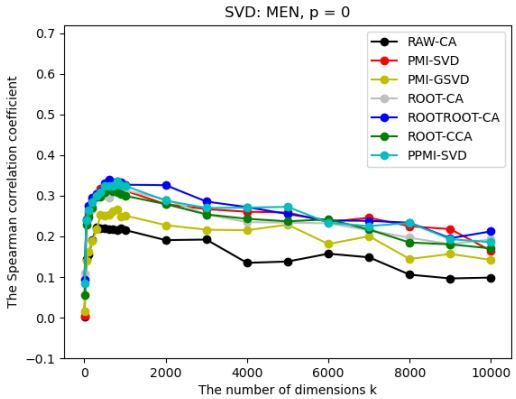
Plots are for ρ as a function of k for SVD-based methods.



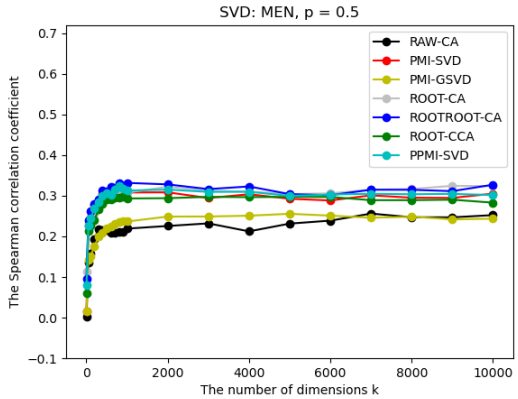
(a) WordSim353: $p = 0$



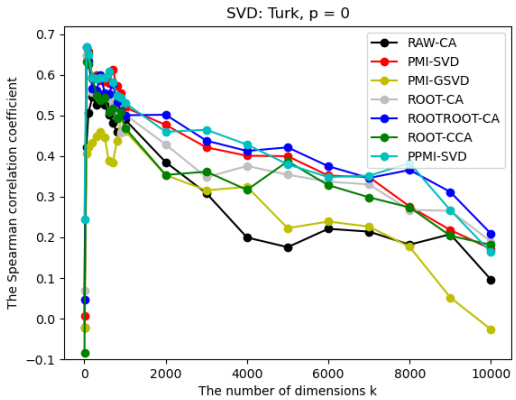
(b) WordSim353:$p = 0.5$



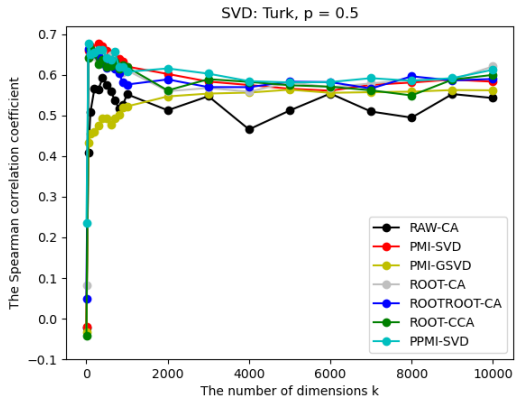
(c) MEN: $p = 0$



(d) MEN: $p = 0.5$

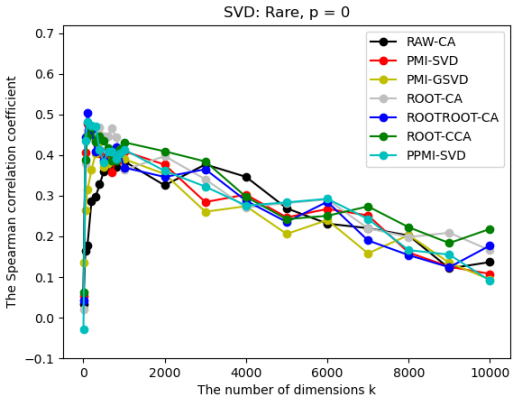


(e) Turk: $p = 0$

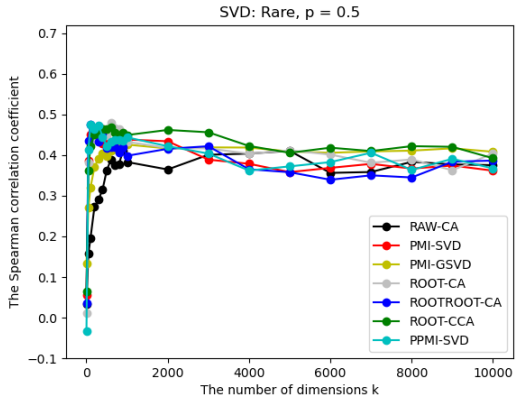


(f) Turk: $p = 0.5$

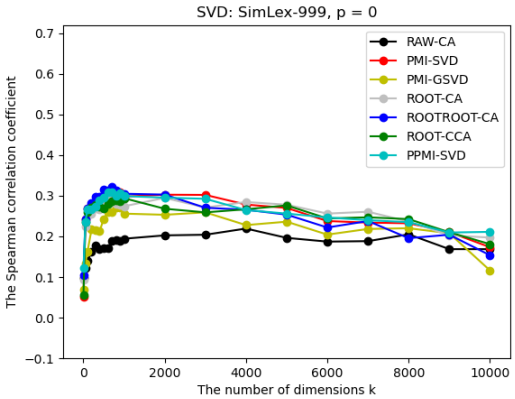
Figure 4.3: Text8



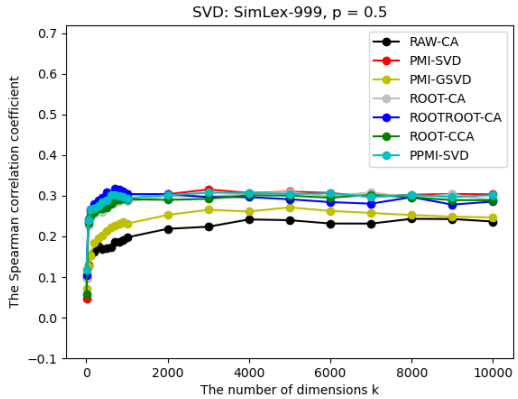
(g) Rare: $p = 0$



(h) Rare: $p = 0.5$

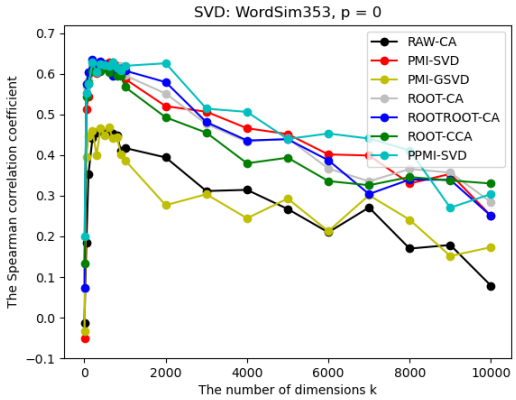


(i) SimLex-999: $p = 0$

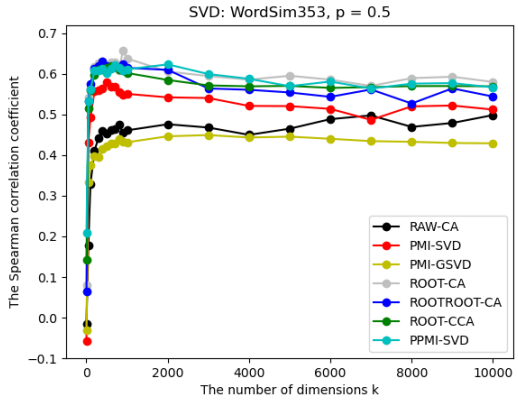


(j) SimLex-999: $p = 0.5$

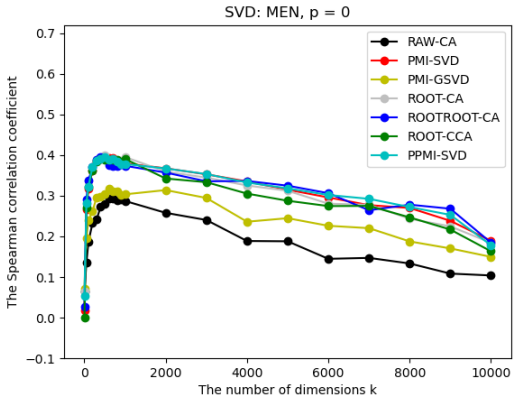
Figure 4.3: Text8



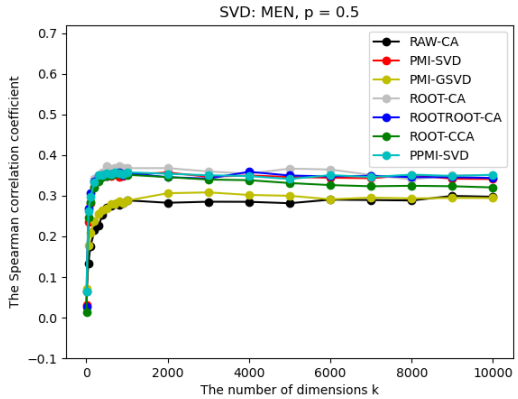
(a) WordSim353: $p = 0$



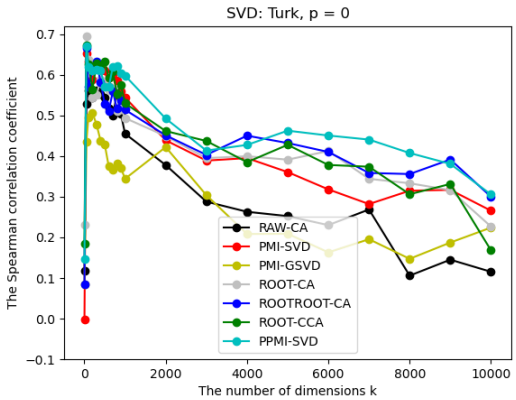
(b) WordSim353: $p = 0.5$



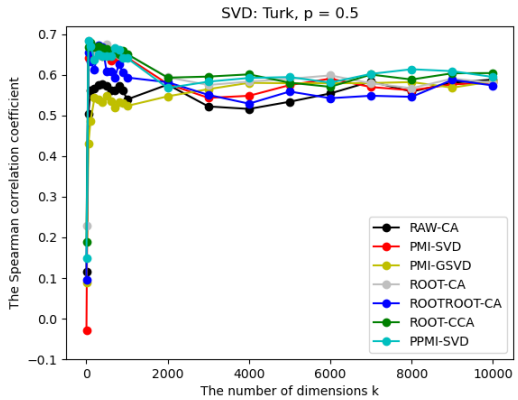
(c) MEN: $p = 0$



(d) MEN: $p = 0.5$

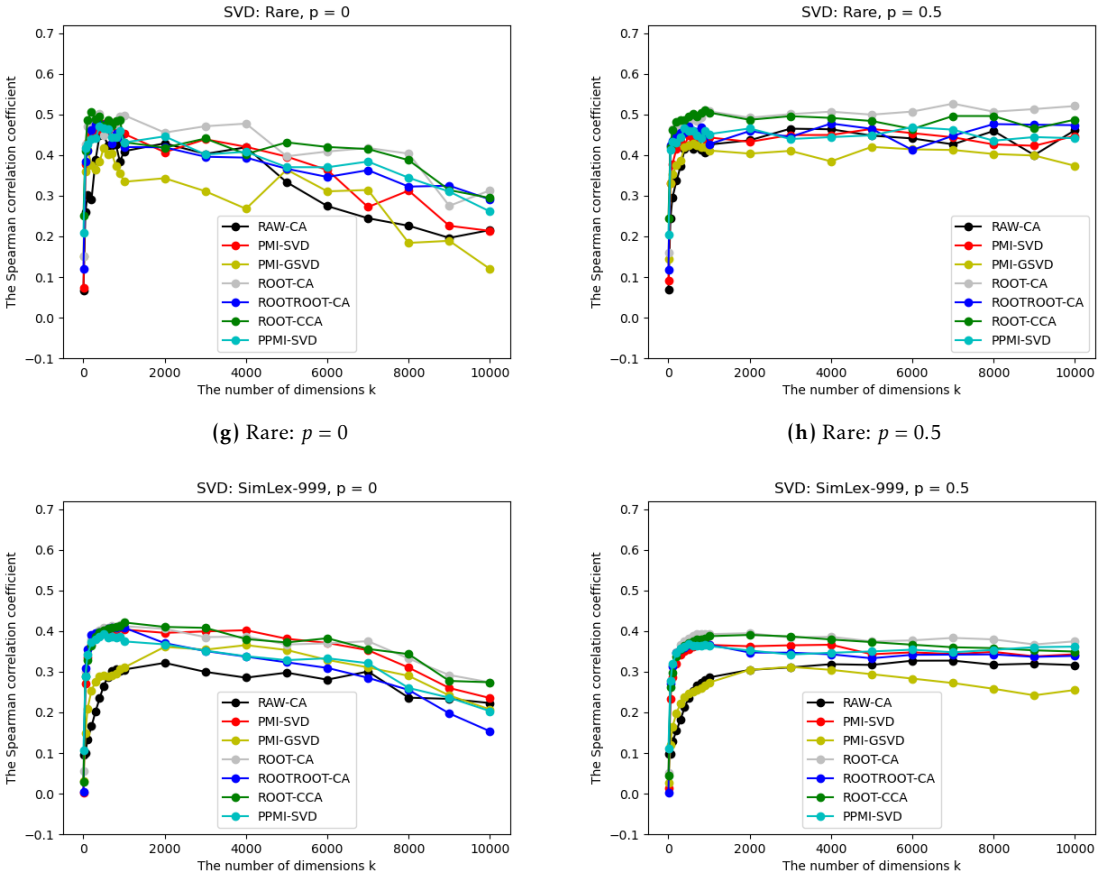


(e) Turk: $p = 0$



(f) Turk: $p = 0.5$

Figure 4.4: BNC



(g) Rare: $p = 0$

(h) Rare: $p = 0.5$

(i) SimLex-999: $p = 0$

(j) SimLex-999: $p = 0.5$

Figure 4.4: BNC

Appendix 4.C BNC: the number and sizes of extreme values of PMI, PPMI, and WPMI, and plots showing the contribution of the rows about PMI-SVD, PPMI-SVD, and PMI-GSVD

Table 4.5, part PMI, PPMI, WPMI, shows the number of extreme values of PMI, PPMI, WPMI matrices. We only include non-zero pairs of PMI matrix because the PMI matrix is sparse: 84.1% of the entries are zero. The corresponding 84.1% of entries in PPMI and WPMI are also ignored. The number of extreme values of WPMI matrix (2,525,345) is much larger than that of PMI and PPMI (141,366 and 405,830). Furthermore, in WPMI the extremeness of the extreme values is much larger than those in PMI and PPMI. For example, where the average contribution of each cell is

$1/(11,332 \times 11,332)$, in WPMI the most extreme entry (the, the) contributes around 0.01150 to the total inertia, while in PMI (PPMI), the most extreme entry (ee, ee) contributes around 2.2×10^{-6} (2.7×10^{-6}) to the total inertia. Figure 4.5 shows the contribution of the rows, corresponding to top 10 extreme values, to the first 100 dimensions of PMI-SVD, PPMI-SVD, PMI-GSVD. The rows, corresponding to the top extreme values of WPMI, take up a bigger contribution to the first dimensions of PMI-GSVD. For example, in PMI-GSVD, the "the" row contributes more than 0.3 to the third dimension, while in PMI-SVD and PPMI-SVD, the contributions are much smaller.

Table 4.5: BNC: the number of extreme values

	LT f_1	GT f_3	total
PMI	13,982	127,384	141,366
PPMI	0	405,830	405,830
WPMI	2,037,800	487,545	2,525,345
TTEST	334,512	1,480,336	1,814,848
ROOT-TTEST	35,418	927,470	962,888
ROOTROOT-TTEST	31,234	750,433	781,667
STRATOS-TTEST	0	1,173,717	1,173,717

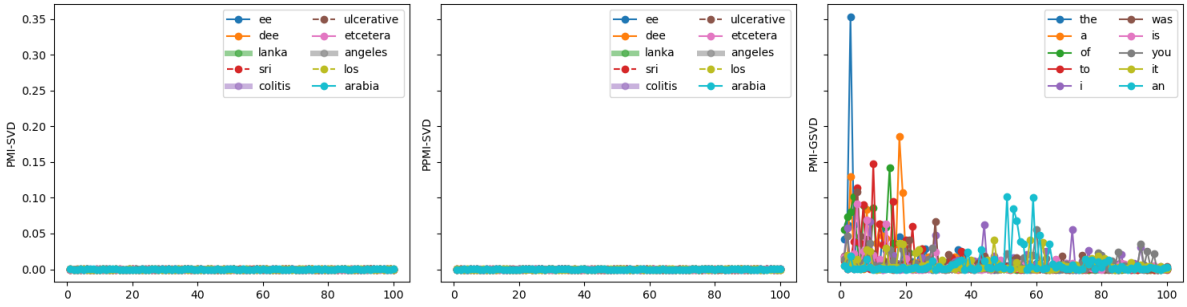


Figure 4.5: BNC: the contribution of the rows, corresponding to top 10 extreme values, to first 100 dimensions of PMI-SVD, PPMI-SVD, PMI-GSVD.

Appendix 4.D Text8: plots showing the contribution of the rows about ROOT-CCA

Figure 4.6 shows the contribution of the rows, corresponding to top 10 extreme values, to first 100 dimensions of ROOT-CCA.

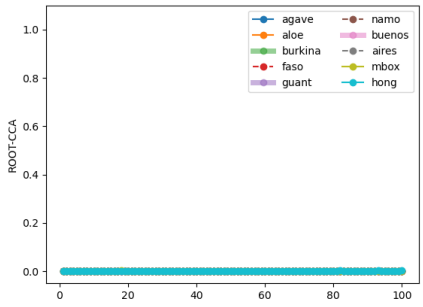


Figure 4.6: Text8: the contribution of the rows, corresponding to top 10 extreme values, to first 100 dimensions of ROOT-CCA.

Appendix 4.E BNC: the number and sizes of extreme values of TTEST, ROOT-TTEST, ROOTROOT-TTEST, and STRATOS-TTEST, and plots showing the contribution of the rows about RAW-CA, ROOT-CA, ROOTROOT-CA, and ROOT-CCA

The bottom part of Table 4.5 shows the number of extreme values of TTEST, ROOT-TTEST, ROOTROOT-TTEST and STRATOS-TTEST matrices. Similar with PMI, PPMI, WPMI, 84.1% of entries are ignored. The number of extreme values of TTEST matrix (1,814,848) is much larger than that of ROOT-TTEST, ROOTROOT-TTEST and STRATOS-TTEST (962,888, 781,667, and 1,173,717). Furthermore, in TTEST the extremeness of the extreme values is much larger than in ROOT-TTEST, ROOTROOT-TTEST and STRATOS-TTEST. For example, in TTEST the most extreme entry (kong, hong) or (hong, kong) contributes around 0.00965 to the total inertia, while in ROOT-TTEST, ROOTROOT-TTEST and STRATOS-TTEST, the most extreme entries (colitis, ulcerative) or (ulcerative, colitis), (colitis, ulcerative) or (colitis, ulcerative), (hong, kong) or (kong, hong) contribute around 0.00047, 0.00003, and 0.00008 respectively. Figure 4.7 shows the contribution of the rows, corresponding to top 10 extreme values, to first 100 dimensions of RAW-CA, ROOT-CA, and ROOTROOT-CA. The corresponding plot about ROOT-CCA is in Figure 4.8. In RAW-CA, the rows, corresponding to top extreme values of TTEST, have a big contribution to the first dimensions of RAW-CA, while in ROOT-CA and ROOTROOT-CA, the contributions are much smaller, which also holds for ROOT-CCA.

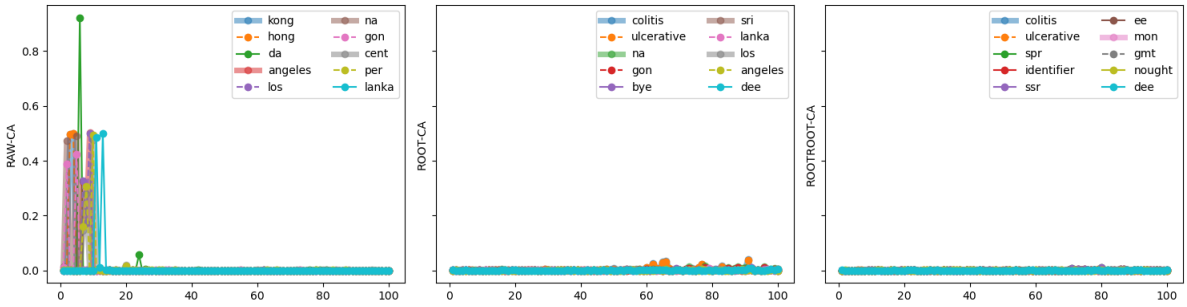


Figure 4.7: BNC: the contribution of the rows, corresponding to top 10 extreme values, to first 100 dimensions of RAW-CA, ROOT-CA, ROOTROOT-CA.

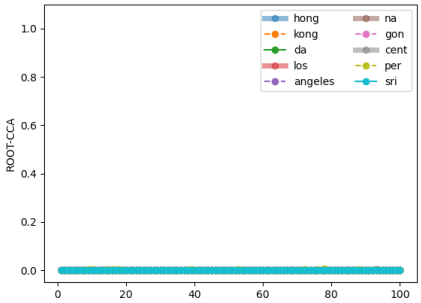


Figure 4.8: BNC: the contribution of the rows, corresponding to top 10 extreme values, to first 100 dimensions of ROOT-CCA.

Chapter 5

Correspondence analysis: handling cell-wise outliers via reconstitution algorithm

Abstract

Correspondence analysis (CA) is a popular technique to visualize the relationship between two categorical variables. CA uses the data from a two-way contingency table and is affected by the presence of outliers. The supplementary points method is a popular method to handle outliers. Its disadvantage is that the information from entire rows or columns is removed. However, outliers can be caused by cells only. In this paper, a reconstitution algorithm is introduced to cope with such cells. This algorithm can reduce the contribution of cells in CA instead of deleting entire rows or columns. Thus the remaining information in the row and column involved can be used in the analysis. The reconstitution algorithm is compared with two alternative methods for handling outliers, the supplementary points method and MacroPCA. It is shown that the proposed strategy works well.

This chapter is under review as: Qi, Q., Hessen, D. J., Vonk, A. N., & Van der Heijden, P. G. M.. Author contributions: PvdH provided the idea. QQ worked out the idea, set up the experiments, and carried them out. AV provided and analyzed the ocean plastic data. QQ, DH, AV, and PvdH discussed and edited the text. The code used in this study can be found at `https://github.com/qianqianqi28/ca-outlier`.

5.1 Introduction

Correspondence analysis (CA) is an exploratory data analysis method that visualizes the dependence of the two categorical variables in a two-way contingency table using a two-dimensional plot (Greenacre, 1984; Greenacre & Hastie, 1987; Greenacre, 2017). CA has received considerable attention in a variety of areas such as marketing (Pitt et al., 2020), psychology (Kim et al., 2021), and text categorization and authorship attribution (Qi et al., 2023). However, relatively little attention has been given to CA in the presence of outliers (Riani, Atkinson, Torti, & Corbellini, 2022).

Outliers may be errors or unexpected observations which could shed new light on the researched phenomenon (Sripriya & Srinivasan, 2018). In general, the data are arranged in a matrix where rows correspond to the individual observations and columns are variables (Grubbs, 1969; Rousseeuw & Van Den Bossche, 2018; Hubert, Rousseeuw, & Van den Bossche, 2019; Raymaekers & Rousseeuw, 2024). The term outlier typically refers to an individual observation that deviates markedly from other members of the sample in which it occurs.

In a contingency table, the definition of an outlier is different from the general case (Kuhnt, Rapallo, & Rehage, 2014; Sripriya & Srinivasan, 2018). An entry in the table represents the number of individuals that occurs jointly in a category of one variable and a category of the other. Thus, in the contingency table, a row does not correspond to a single observation but to a number of joint sample frequencies of individual observations. Here, extreme counts that do not follow the general pattern in the table are viewed as outliers.

In the context of CA, an outlier can be defined in different ways and the procedure to detect outliers depends on the definition of an outlier. Two detection procedures stand out. On the one hand, Greenacre (2013, 2017) uses visual inspecting of CA plots to detect outliers. Greenacre (2013, 2017) considers a row or column point as an outlier when it clearly lies far from other points in the CA plot. In addition to large absolute coordinates, Hoffman and Franke (1986) and Bendixen (1996) define a row or column point as an outlier if the row or column point has a high contribution to an axis. The contribution of a point to an axis is determined not only by the position of the point in the CA plot but also by the marginal proportion of the point. According to Hoffman and Franke (1986) and Bendixen (1996), if the marginal proportion of a point is very small, it may not be an outlier, even though, following Greenacre's definition, it is an outlier in the sense that it lies far from other points in the CA plot.

On the other hand, Riani et al. (2022) and Raymaekers and Rousseeuw (2024) detect outliers making use of distributional assumptions. Riani et al. (2022) state (p. 8) "... an outlier is a row which does not agree with the multiplicative model assuming independence fitted to the data." This outlier detection procedure is less attractive, because, in interesting applications, the independence model assumption would be rejected almost always (De Leeuw, Van der Heijden, & Verboon, 1990), and thus, in this situation, this procedure tends to detect too many rows as outlying points. Raymaekers and Rousseeuw (2024) use MacroPCA to detect outliers. MacroPCA is

originally proposed by Rousseeuw and Van Den Bossche (2018) for principal component analysis (PCA) and subsequently used in CA by Raymaekers and Rousseeuw (2024). MacroPCA assumes that the data are generated from a multivariate Gaussian distribution. However, the two variables in the contingency table are categorical variables, and therefore the normality assumption for the input matrix of MacroPCA may be not appropriate for CA.

Hoffman and Franke (1986), Bendixen (1996), Greenacre (2017) and Riani et al. (2022) detect outlying rows or columns, and, after detecting the outliers, they cope with the outliers by the supplementary points method. That is, CA is performed on the contingency table without the outlying rows and columns. Afterwards, the outliers are projected into the CA solution of the reduced table. Therefore, the outliers cannot determine the CA solution.

In contrast, Raymaekers and Rousseeuw (2024) detect outlying cells and outlying rows and handle the outliers in the same step. The basic idea is to impute the outlying cells by an iterative PCA algorithm while excluding outlying rows. Their method does not have a good fit with the theory of CA, and important properties of CA, such as that Euclidean distances in a CA display can be interpreted in terms of chi-squared distances, are lost. Moreover, this method seems to flag a lot more rows as outliers than necessary.

The supplementary points method and MacroPCA delete outlying rows or columns completely, and therefore, also remove information from these rows or columns that is not related to this outlying problem. So, the removal of an entire row or column causes a unnecessary loss of information.

According to Bendixen (1996), a cell frequency that causes its row to be identified as an outlier might also cause its column to be identified as an outlier, and vice versa. Thus, an outlying row or column may be caused by a specific joint frequency. This suggests that we only need to deal with the specific cell and do not need to delete the entire row or column.

In this paper, the focus is on cell-wise outliers. To detect outlying cells, we follow Greenacre's definition using visual inspection of the CA plot, as a main aim of CA is to summarize the structure of data via a two-dimensional plot and such outliers cause the other points to be tightly clustered and thus reduce the readability of a CA plot. A cell is an outlying cell if the corresponding row and column points of this cell lie far from other points. Here, once a cell is identified as an outlier, the cell is not removed but its contribution is reduced. For reducing the contribution of an outlying cell the reconstitution algorithm is proposed. The reconstitution algorithm has been proposed originally by Nora-Chouteau (1974) and has later been used by Greenacre (1984) and De Leeuw and Van der Heijden (1988) to handle missing values in cells.

The paper is built up as follows. We start with a description of CA in Section 5.2. Section 5.3 presents how outliers originate. Section 5.4 presents the reconstitution algorithm to handle cell-wise outliers and describes MacroPCA and the supplementary points method. Section 5.5 compares these three methods on a contingency table, the brands of cars dataset, and compares the reconstitution algorithm and the supple-

mentary points method on an incidence table, the ocean plastic dataset. Section 5.6 discusses and concludes this paper. Finally, Section 5.7 introduces the implementation of code.

5.2 Correspondence analysis background

Let $\boldsymbol{X}$ be a contingency table having I rows and J columns with non-negative entries x_{ij}, and suppose that $\boldsymbol{X}$ has full rank. An index is replaced by '+' when summed over the corresponding elements, such as $x_{i+} = \sum_j x_{ij}$. It is customary to rescale $\boldsymbol{X}$ to the correspondence matrix $\boldsymbol{P} = \boldsymbol{X}/x_{++}$, so that $\sum_i \sum_j p_{ij} = 1$. The row profile for row i is the vector having elements $p_{ij}/p_{i+}, j = 1,\ldots,J$ and, similarly, the column profile for column j is the vector with elements $p_{ij}/p_{+j}, i = 1,\ldots,I$. The average row profile is the vector with elements $p_{+j}, j = 1,\ldots,J$, i.e., the column margins, and the average column profile is the vector with elements $p_{i+}, i = 1,\ldots,I$, i.e. row margins. Let $\boldsymbol{E} = [p_{i+}p_{+j}]$ be the matrix with elements under statistical independence. Let $\boldsymbol{D}_r$ and $\boldsymbol{D}_c$ be diagonal matrices with the row margins p_{i+} and column margins p_{+j} in the diagonal, respectively.

CA can be introduced in many ways. We introduce CA here using the concept of total inertia (Greenacre, 2017), i.e., the well-known Pearson χ^2 statistic divided by x_{++}

$$\text{Total inertia} = \sum_i \sum_j \frac{(p_{ij} - p_{i+}p_{+j})^2}{p_{i+}p_{+j}}. \tag{5.1}$$

The aim of CA is to provide a multidimensional representation of the matrix $\boldsymbol{X}$ where the total inertia is projected as much as possible onto a low-dimensional space. The computational procedure to obtain the solution makes use of the singular value decomposition (SVD). In the first step the matrix $\boldsymbol{X}$ is transformed into the matrix of standardized residuals $\boldsymbol{D}_r^{-\frac{1}{2}}(\boldsymbol{P} - \boldsymbol{E})\boldsymbol{D}_c^{-\frac{1}{2}}$ with elements $(p_{ij} - p_{i+}p_{+j})/\sqrt{p_{i+}p_{+j}}$, and then SVD is applied to this matrix, yielding

$$\boldsymbol{D}_r^{-\frac{1}{2}}(\boldsymbol{P} - \boldsymbol{E})\boldsymbol{D}_c^{-\frac{1}{2}} = \boldsymbol{U}\Sigma\boldsymbol{V}^T, \tag{5.2}$$

where $\boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{V}^T\boldsymbol{V} = \boldsymbol{I}$ and Σ is a diagonal matrix with singular values $\sigma_k, k = 1,\cdots,\min(I-1, J-1)$ in descending order on the diagonal. Subtracting the matrix $\boldsymbol{E}$ of rank 1 leads to a reduction of 1 for the rank of the resulting matrix $\boldsymbol{D}_r^{-\frac{1}{2}}(\boldsymbol{P} - \boldsymbol{E})\boldsymbol{D}_c^{-\frac{1}{2}}$.

If we pre-multiply and post-multiply both sides of Equation (5.2) by $\boldsymbol{D}_r^{-\frac{1}{2}}$ and $\boldsymbol{D}_c^{-\frac{1}{2}}$, respectively, on the left hand side we get $\boldsymbol{D}_r^{-1}(\boldsymbol{P} - \boldsymbol{E})\boldsymbol{D}_c^{-1}$ with elements $(p_{ij} - p_{i+}p_{+j})/(p_{i+}p_{+j})$, and this yields

$$\boldsymbol{D}_r^{-1}(\boldsymbol{P} - \boldsymbol{E})\boldsymbol{D}_c^{-1} = \boldsymbol{D}_r^{-\frac{1}{2}}\boldsymbol{U}\Sigma(\boldsymbol{D}_c^{-\frac{1}{2}}\boldsymbol{V})^T = \boldsymbol{\Phi}\Sigma\boldsymbol{\Gamma}^T = \boldsymbol{F}\Sigma^{-1}\boldsymbol{G}^T \tag{5.3}$$

where $\boldsymbol{\Phi} = \boldsymbol{D}_r^{-\frac{1}{2}}\boldsymbol{U}$, $\boldsymbol{\Gamma} = \boldsymbol{D}_c^{-\frac{1}{2}}\boldsymbol{V}$, $\boldsymbol{F} = \boldsymbol{\Phi}\Sigma$, and $\boldsymbol{G} = \boldsymbol{\Gamma}\Sigma$. $\boldsymbol{\Phi}$ and $\boldsymbol{\Gamma}$ are called the standard coordinates for the row profiles and column profiles, respectively. They have the property that, for each k, their weighted sum is 0 and their weighted sum of squares is 1, i.e. $\mathbf{1}^T\boldsymbol{D}_r\boldsymbol{\Phi} = \mathbf{1}^T\boldsymbol{D}_c\boldsymbol{\Gamma} = \mathbf{0}^T$ and $\boldsymbol{\Phi}^T\boldsymbol{D}_r\boldsymbol{\Phi} = \boldsymbol{\Gamma}^T\boldsymbol{D}_c\boldsymbol{\Gamma} = \boldsymbol{I}$. $\boldsymbol{F}$ and $\boldsymbol{G}$ are called principal coordinates for the row profiles and column profiles, respectively.

Euclidean distances between rows of $\boldsymbol{F}$ ($\boldsymbol{G}$) are equal to the so-called χ^2- distances between rows (columns) of $\boldsymbol{X}$. The squared χ^2-distance between the row profiles i and i' is

$$\delta_{i,i'}^2 = \sum_j \frac{(\frac{p_{ij}}{p_{i+}} - \frac{p_{i'j}}{p_{i'+}})^2}{p_{+j}}. \tag{5.4}$$

The χ^2-distance $\delta_{i,i'}$ between row profiles i and i' gives more weight to differences in a column j when this column has a lower margin p_{+j}. The χ^2-distance $\delta_{j,j'}$ between column profiles j and j' is defined in a similar way.

Joint graphic displays of row points and column points are usually made to study the relationship between the rows and the columns in the matrix $\boldsymbol{P}$. For this asymmetric and symmetric maps are used. In an asymmetric map rows of $\boldsymbol{P}$ can be displayed as points in a multidimensional space using principle coordinates, and columns as points using standard coordinates. Thus, in full-dimensional space the dot products of row points $\boldsymbol{F}$ and column points $\boldsymbol{\Gamma}$ are equal to the elements of $\boldsymbol{D}_r^{-1}(\boldsymbol{P}-\boldsymbol{E})\boldsymbol{D}_c^{-1}$. Usually low-dimensional representations are made of the first few columns of $\boldsymbol{F}$ and $\boldsymbol{\Gamma}$, as the SVD ensures that the first few dimensions provide an optimal approximation of $\boldsymbol{D}_r^{-1/2}(\boldsymbol{P}-\boldsymbol{E})\boldsymbol{D}_c^{-1/2}$ in a least-squares sense. Together, the configurations of row points and column points form a biplot (Gabriel, 1971) of the matrix $\boldsymbol{D}_r^{-1}(\boldsymbol{P}-\boldsymbol{E})\boldsymbol{D}_c^{-1}$. Asymmetric maps have the interesting property that the row points are in the weighted average of the column points and the other way around. This is evident from the so-called transition equations

$$\boldsymbol{F} = \boldsymbol{D}_r^{-1}\boldsymbol{P}\boldsymbol{\Gamma} \quad \text{and} \quad \boldsymbol{G} = \boldsymbol{D}_c^{-1}\boldsymbol{P}^T\boldsymbol{\Phi} \tag{5.5}$$

The points for the average row profile and the average column profile fall in the origin. Thus, for the combination of $\boldsymbol{F}$ and $\boldsymbol{\Gamma}$, the transition formulas pull individual row points towards the column points for which $p_{ij}/p_{i+} > p_{+j}$.

Asymmetric maps have the drawback that, when for example the pair $\boldsymbol{F}$ and $\boldsymbol{\Gamma}$ is used, the Euclidean distances between the columns are not chi-squared distances. Also, there is the practical disadvantage that the cloud of column points may be huge in comparison to the cloud of row points, and thus row points tend to huddle together and reduce the readability of the plot. For this reason one often sees the use of the so-called symmetric map. That is, both rows and columns are displayed in principle coordinates. Therefore, the Euclidean distances between row points, i.e., rows of $\boldsymbol{F}$ (column points, i.e., rows of $\boldsymbol{G}$) are equal to the χ^2-distances between rows (columns) of $\boldsymbol{X}$, and in low-dimensional representations the Euclidean distances between row

points and between column points provide approximations of these distances. The Euclidean distance between row points and column points is not meaningful. However, the direction between row points and column points is still meaningful, because the only difference between principal and standard coordinates is a dimensionwise scalar (compare Equation (5.3)).

The total inertia can be expressed as a weighted sum of squared χ^2–distances of row profiles and of column profiles to the average profile:

$$\text{Total inertia} = \sum_i p_{i+} \sum_j \frac{(\frac{p_{ij}}{p_{i+}} - p_{+j})^2}{p_{+j}} = \sum_j p_{+j} \sum_i \frac{(\frac{p_{ij}}{p_{+j}} - p_{i+})^2}{p_{i+}}. \tag{5.6}$$

This shows that the total inertia can be split up over the rows and over the columns. The inertia of the row point i and the column point j in dimension k are $p_{i+}f_{ik}^2 = u_{ik}^2\sigma_k^2$ and $p_{+j}g_{jk}^2 = v_{jk}^2\sigma_k^2$, respectively. The contributions of row i and column j to dimension k are $p_{i+}f_{ik}^2/\sigma_k^2 = u_{ik}^2\sigma_k^2/\sigma_k^2 = u_{ik}^2$ and $p_{+j}g_{jk}^2/\sigma_k^2 = (v_{jk}\sigma_k)^2/\sigma_k^2 = v_{jk}^2$, respectively. The contributions quantify to what extent individual rows and columns, both by their positions (f_{ik} or g_{jk}) and margins (p_{i+} or p_{+j}), affect the solution (Greenacre, 2013). This means that, for rows that have equal margins p_{i+} for dimension k, the further this point is from the origin, the larger its contribution is to dimension k. In a so-called *contribution biplot*, elements f_{ik} (u_{ik}) are as row coordinates and v_{jk} (g_{jk}) as column coordinates.

The total inertia can also be split up over cells. The inertia of each cell in the matrix $\boldsymbol{D}_r^{-\frac{1}{2}}(\boldsymbol{P}-\boldsymbol{E})\boldsymbol{D}_c^{-\frac{1}{2}}$ of standardized residuals is $(p_{ij}-p_{i+}p_{+j})^2/(p_{i+}p_{+j})$.

By rewriting Equation (5.3), the correspondence matrix $\boldsymbol{P}$ can be decomposed as follows

$$\boldsymbol{P} = \boldsymbol{D}_r(\mathbf{1}\mathbf{1}^T + \boldsymbol{\Phi}\boldsymbol{\Sigma}\boldsymbol{\Gamma}^T)\boldsymbol{D}_c \approx \boldsymbol{D}_r(\mathbf{1}\mathbf{1}^T + \boldsymbol{\Phi}_K\boldsymbol{\Sigma}_K\boldsymbol{\Gamma}_K^T)\boldsymbol{D}_c \tag{5.7}$$

Equation (5.7) is called the reconstitution formula and is the foundation of the *reconstitution algorithm*, discussed in Section 5.4.

Similar to Equation (5.5), an additional row can be projected as a supplementary point in an existing CA plot. Let the extra row (supplementary) point be the row vector $\boldsymbol{a} = [a_1, a_2, \cdots, a_J]$ and an extra column (supplementary) point $\boldsymbol{b} = [b_1, b_2, \cdots, b_I]$, as a row vector. The projections for the row point $\boldsymbol{a}$ and the column point $\boldsymbol{b}$ are found by

$$\frac{\boldsymbol{a}}{\sum_j a_j}\boldsymbol{\Gamma} \quad \text{and} \quad \frac{\boldsymbol{b}}{\sum_i b_i}\boldsymbol{\Phi} \tag{5.8}$$

respectively. These supplementary points do not determine the CA solution, but from these projections we can see the relationships between the configurations of row and column points in the existing CA solution to these supplementary points.

5.3 How outliers originate

CA is sensitive to outliers (Choulakian, 2020). Here, we enumerate three potential causes for the presence of outliers: an (approximate) block diagonal matrix, rows or columns with relatively small margins, and cells with relatively high values. We do not claim that these causes give an exhaustive view, and also note that these three causes may overlap.

5.3.1 Block diagonal matrix

As we discussed in Equation (5.6), the total inertia can be expressed as a weighted sum of squared χ^2–distances of rows points to the origin (Greenacre, 2017). If all profiles are the same and thus equal to the average profile, then all χ^2-distances of the points to the origin would be 0 and thus the total inertia would be 0. On the other hand, maximum inertia can be obtained when all profiles are totally different. For example, when a matrix is an identity matrix and $m = n$, the inertia is equal to $m - 1$.

If, after reordering the rows and columns of a matrix in an appropriate way, $\boldsymbol{X}$ is a block diagonal matrix with t blocks, the first $t-1$ dimensions of the CA solution have singular values equal to 1 (Choulakian, 2020; Handan-Nader, 2023). For example, let $t = 2$. Table 5.1a is an illustration. One block is cell (2, b) and the other block consists of rows 1, 3 and 4 together with columns a, c, d. The CA solution is as follows

$$\begin{aligned}\boldsymbol{D}_r^{-1}(\boldsymbol{P}-\boldsymbol{E})\boldsymbol{D}_c^{-1} &= \boldsymbol{\Phi}\boldsymbol{\Sigma}(\boldsymbol{\Gamma})^T \\ &= \begin{bmatrix} 0.27 & 1.59 & 0.18 \\ -3.67 & 0.00 & 0.00 \\ 0.27 & -0.50 & -1.52 \\ 0.27 & -0.79 & 0.97 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0.44 & 0 \\ 0 & 0 & 0.10 \end{bmatrix} \begin{bmatrix} 0.27 & -0.83 & 1.54 \\ -3.67 & 0.00 & 0.00 \\ 0.27 & -0.40 & -0.91 \\ 0.27 & 1.91 & 0.34 \end{bmatrix}^T \end{aligned} \tag{5.9}$$

The first singular value equals 1. The reordered matrix based on the first coordinates of the rows and columns, shown in Table 5.1b, is a block diagonal matrix. On dimension 1 row 2 and column b are outliers with scores -3.67.

Table (5.1c) is a less extreme case, where the elements approximate a block diagonal matrix. The CA solution is as follows

$$\begin{aligned}\boldsymbol{D}_r^{-1}(\boldsymbol{P}-\boldsymbol{E})\boldsymbol{D}_c^{-1} &= \boldsymbol{\Phi}\boldsymbol{\Sigma}(\boldsymbol{\Gamma})^T \\ &= \begin{bmatrix} 0.19 & 1.56 & 0.18 \\ -5.03 & -0.03 & 0.02 \\ 0.20 & -0.49 & -1.49 \\ 0.21 & -0.78 & 0.95 \end{bmatrix} \begin{bmatrix} 0.93 & 0 & 0 \\ 0 & 0.44 & 0 \\ 0 & 0 & 0.09 \end{bmatrix} \begin{bmatrix} 0.20 & -0.82 & 1.52 \\ -5.03 & -0.07 & -0.03 \\ 0.21 & -0.39 & -0.90 \\ 0.17 & 1.88 & 0.33 \end{bmatrix}^T \end{aligned} \tag{5.10}$$

Now the first singular value is 0.93, close to 1. The reordered matrix, where the row and column scores for dimension 1 are used, is in Table 5.1d. It approximates a block diagonal matrix. On dimension 1 row 2 and column b are outliers with scores -5.03.

By these examples we want to illustrate that, if the rows and columns of the table can be reordered so that a block diagonal matrix or an approximate block diagonal matrix arises, this may lead to outlying points for those rows and columns that form the smaller (approximate) block diagonal matrix.

Table 5.1: Document-term matrix $\boldsymbol{X}$: size 4×4

	a	b	c	d
1	1	0	3	4
2	0	2	0	0
3	2	0	5	1
4	4	0	6	1

(a) Block diagonal matrix

	b	d	c	a
2	2	0	0	0
1	0	4	3	1
3	0	1	5	2
4	0	1	6	4

(b) Block diagonal matrix; reordered table

	a	b	c	d
1	100	2	300	400
2	2	100	1	4
3	200	3	500	100
4	400	2	600	100

(c) Approximate block diagonal matrix

	b	d	a	c
2	100	4	2	1
1	2	400	100	300
3	3	100	200	500
4	2	100	400	600

(d) Approximate block diagonal matrix; reordered table

	a	b	c	d
1	1	2	3	4
2	2	100	1	4
3	2	3	5	1
4	4	2	6	1

(e) Large value 100

	c	a	d	b
4	6	4	1	2
3	5	2	1	3
1	3	1	4	2
2	1	2	4	100

(f) Large value 100; reordered table

5.3.2 Rows or columns with relatively small margins

For the rows of the matrix $\boldsymbol{X}$, Equation (5.6), the squared chi-squared distance of row i to the origin O, δ_{iO}^2, is

$$\delta_{iO}^2 = \sum_j \frac{\left(\frac{p_{ij}}{p_{i+}} - p_{+j}\right)^2}{p_{+j}}. \tag{5.11}$$

Following Greenacre (2013), we will argue that, in principle, rows (and columns) with smaller margins p_{i+} have relatively more potential than rows (and columns) with larger margins to be in the periphery of a cloud of points, as these rows with smaller margins p_{i+} have a higher potential to have a larger chi-squared distance δ_{iO}.

- The marginal profile with elements p_{+j} falls in the origin. Outliers fall relatively far away from the origin. Thus we are interested in what makes the distance of row i from the origin, δ^2_{iO}, larger.

- In Equation (5.11) $\left(p_{ij}/p_{i+} - p_{+j}\right)^2$ stands for the squared difference between the row profile element j and the marginal profile element j.

- The marginal profile is the weighted average of the row profiles with weights p_{i+}, as for each element j, $\sum_i p_{i+}(p_{ij}/p_{i+}) = p_{+j}$.

- Therefore, if row i has a larger size p_{i+}, we expect that, in principle, row i will be closer to the marginal profile, as it makes up a larger part of the marginal profile. In other words, row profiles with larger p_{i+} have a larger expected correlation with the marginal profile. *So in principle rows with smaller p_{i+} have a higher potential to be relatively further away from the origin.*

- Now consider the denominator p_{+j} in the squared chi-squared distance. For a fixed difference $\left(p_{ij}/p_{i+} - p_{+j}\right)^2$ columns with smaller p_{+j} add more to the chi-squared distance of row i to the origin.

- Now consider the above reasoning for the squared chi-squared distance between column j and the origin, δ^2_{jO}. The same argument holds, *columns with smaller p_{+j} have in principle a higher potential to be relatively further away from the origin.*

- At the same time we noticed that, for a fixed difference $\left(p_{ij}/p_{+j} - p_{i+}\right)^2$, rows with smaller p_{i+} add more to the chi-squared distance of column j to the origin.

We conclude that smaller margins have the potential to lead to larger chi-squared distances of individual rows and columns to the origin because of their potential to deviate more from the marginal profile falling in the origin, and due to the role of the marginal probability in the denominator. If row i has a large difference $\left(p_{ij}/p_{i+} - p_{+j}\right)^2$ in element j, and in particular in element j where p_{+j} is small, then it is more likely that an outlier arises.

We also note that, if row i is an outlier due to profile element j having a low p_{+j}, then if profile element i has a low p_{i+}, column j will also be an outlier. The reason is that we can formulate independence in three ways, namely as $\left(p_{ij} = p_{i+}p_{+j}\right)$, as $\left(p_{ij}/p_{i+} = p_{+j}\right)$ and as $\left(p_{ij}/p_{+j} = p_{i+}\right)$. If there is positive dependence, then $\left(p_{ij} > p_{i+}p_{+j}\right)$, then also $\left(p_{ij}/p_{i+} > p_{+j}\right)$ and $\left(p_{ij}/p_{+j} > p_{i+}\right)$. The latter two conditional formulations of positive dependence link directly to the squared chi-squared distances δ^2_{iO} to δ^2_{jO}. Thus a single cell (i, j) with a strong positive relation can cause a row i as well as column j to be an outlier.

5.3.3 Cells with relatively high values

Outliers may occur due to relatively large frequencies (Langovaya, Kuhnt, & Chouikha, 2013; Choulakian, 2020). Table (5.1e) is an illustration where row 2 and column b have a relative large frequency of 100. The CA solution is

$$\begin{aligned}\boldsymbol{D}_r^{-1}(\boldsymbol{P}-\boldsymbol{E})\boldsymbol{D}_c^{-1} &= \boldsymbol{\Phi}\boldsymbol{\Sigma}(\boldsymbol{\Gamma})^T \\ &= \begin{bmatrix} -1.49 & -3.28 & 0.36 \\ 0.56 & 0.05 & 0.04 \\ -1.67 & 0.74 & -2.91 \\ -2.05 & 1.44 & 1.89 \end{bmatrix} \begin{bmatrix} 0.75 & 0 & 0 \\ 0 & 0.31 & 0 \\ 0 & 0 & 0.08 \end{bmatrix} \begin{bmatrix} -1.76 & 1.44 & 3.08 \\ 0.55 & 0.12 & -0.07 \\ -2.18 & 0.55 & -1.83 \\ -0.99 & -3.41 & 0.71 \end{bmatrix}^T \end{aligned} \tag{5.12}$$

The reordered matrix based on the first coordinates of the rows and columns is shown in Table 5.1f. On dimension 1 row 2 and column b are outliers with scores 0.56 and 0.55, respectively.

5.4 Methods to handle outliers

We discuss three methods to handle outliers. Two methods are cell-wise outlier methods: reconstitution of order h and MacroPCA. The third is the supplementary points method. It is worth noting that reconstitution of order h has been used to handle missing data, but has not been proposed to handle outliers.

5.4.1 Reconstitution of order h

In this paper we propose to deal with an outlier or outliers by changing the data. Specifically, we assume that specific cells in a matrix are outlying cells if they cause row and column points to be outliers. We propose to make such cells in the data matrix missing. We use visual inspection of the CA plot to define outlying cells. In a second step, we apply an algorithm that imputes a new value for each missing value. For this, we use the reconstitution algorithm, originally proposed by Nora-Chouteau (1974) and revisited by Greenacre (1984), De Leeuw and Van der Heijden (1988), and Josse, Chavent, Liquet, and Husson (2012).

We assume for the moment that there is only a single cell causing a row and a column to be outliers, but the procedure that we describe can be applied to multiple outlying cells simultaneously. The idea is to adjust the value in this single cell in such a way that it is perfectly reconstituted in a h-dimensional CA solution. This reconstitution is obtained iteratively.

As by iteratively imputing the missing cell the margins also change, it is easier to describe the method using the raw data x_{ij} instead of the proportions p_{ij}. For x_{ij} we have

$$x_{ij} = \frac{x_{i+}x_{+j}}{x_{++}}\left(1 + \sum_{k=1}^{\min\{I-1,J-1\}} \phi_{ik}\sigma_k\gamma_{jk}\right), \tag{5.13}$$

i.e. x_{ij} is reconstituted if the maximum dimensionality min $(I-1, J-1)$ is used. Let $\hat{x}_{ij}$ be the reconstituted value using $h < \min(I-1, J-1)$ dimensions. Then

$$\hat{x}_{ij} = \frac{x_{i+}x_{+j}}{x_{++}}\left(1 + \sum_{k=1}^{h} \phi_{ik}\sigma_k\gamma_{jk}\right). \tag{5.14}$$

We first explain reconstitution of order 0, meaning that no CA dimensions are used in the reconstitution. Assume that cell (m,n) is an outlier made missing, and assume that at iteration $t = 0$ we impute a non-negative value. Then we iteratively find updates for this missing value as follows:

$$x_{mn}^{t+1} = \frac{x_{m+}^t x_{+n}^t}{x_{++}^t}. \tag{5.15}$$

After convergence, we have the converged value x_{mn}^*. Then CA is applied to the original data where the outlier value in cell (m,n) is replaced by x_{mn}^*. As $x_{mn}^* = x_{m+}^* x_{+n}^* / x_{++}^*$, in (5.13) the residual for cell (m,n) $x_{mn}^* - x_{m+}^* x_{+n}^* / x_{++}^* = 0$. In this sense, the influence of the original outlying cell is eliminated. De Leeuw and Van der Heijden (1988) use reconstitution of order zero in the context of the statistical quasi-independence model. They adjust CA so that it can decompose the departure from this model, a model that assumes independence for some but not all cells in a contingency table. Reconstitution of order zero is also available in the R Package *anacor* (De Leeuw & Mair, 2009).

However, as the residual for cell (m,n) is 0, the inner-product $\sum_{k=1}^{\min\{I-1,J-1\}} \phi_{mk}^*\sigma_k^*\gamma_{nk}^* = 0$ as well, meaning that in the full-dimensional space the vectors m and n are orthogonal. This may be an undesirable bi-product of reconstitution of order 0. An alternative, reconstitution of order h, does not have this problem. In reconstitution h, the value in cell (m,n) is reconstituted by

$$x_{mn}^{t+1} = \frac{x_{m+}^t x_{+n}^t}{x_{++}^t}\left(1 + \sum_{k=1}^{h} \phi_{mk}^t\sigma_k^t\gamma_{nk}^t\right). \tag{5.16}$$

Thus in the h-dimensional solution the value in cell (m,n) is reconstituted perfectly by $x_{m,n}^* = (x_{m+}^* x_{+n}^* / x_{++}^*)(1 + \sum_{k=1}^{h} \phi_{mk}^*\sigma_k^*\gamma_{nk}^*)$, and only for higher dimensions than h the residual as well as inner-product is zero. This means that the parameters ϕ_{ik}^*, σ_k^*, and γ_{jk}^*, $k = 1, 2, \cdots, h$ provide the CA solution based on the non-outlying cells in the matrix only. So, when interest goes out to a CA solution of two dimensions, theoretically it makes sense to eliminate the influence of an outlier by applying reconstitution of order 2. However, in practice this may lead to a negative value for $x_{m,n}^*$, as is the case in the second example of Section 5.5. In such instances reconstitution of order zero is the preferred option.

As far as we know, there is no R package in which reconstitution of order h is implemented, where $h \geq 1$. We present the R function *reconca*, that we created by rewriting the function *imputeCA* taken from the R package *missMDA* which implements a regularized reconstitution algorithm (Josse et al., 2012; Josse & Husson, 2016) that is meant for the missing value problem where the number of missing values in the data is relatively large. This is a situation different from our idea to make outlying values missing and therefore we further ignore this regularized version in this paper.

5.4.2 MacroPCA

MacroPCA was originally proposed for PCA (Hubert et al., 2019) and subsequently adjusted for CA (Raymaekers & Rousseeuw, 2024). MacroPCA is quite involved and detects outliers and handles outliers at the same time. It includes two parts. The first part of MacroPCA is a multivariate method called DetectDeviatingCells (DDC) (Rousseeuw & Van Den Bossche, 2018; Hubert et al., 2019) that assumes that data are generated from a multivariate Gaussian distribution but some cells were corrupted. DDC detects cellwise outliers, and provides these cellwise outliers with initial values. It also detects initial row-wise outliers. In the second part, the set of outlying rows will be improved. Low-dimensional representations are obtained in a way that is similar but not identical to the reconstitution algorithm. The low-dimensional representations of MacroPCA are not nested. That is, for example, the two-dimensional representation is not a subset of three-dimensional representations. We refer to Rousseeuw and Van Den Bossche (2018); Hubert et al. (2019) for details.

MacroPCA is modified to handle missing data and outlier problems in the context of CA (Raymaekers & Rousseeuw, 2024). For CA the original matrix is replaced with the matrix of standardized residuals. As in CA the standardized residuals are only a starting point in finding the CA solution, the modification is close to but different from CA. Also, in the DCC step of MacroPCA where outlying cells are detected, the algorithm makes the assumption of a Gaussian distribution, for which there is no clear rationale in the context of CA.

5.4.3 Supplementary points method

The supplementary points method is a well-known method to deal with row-wise outliers or column-wise outliers. That is, after noticing outlying points, for which we use visual inspection, a new CA is performed on the data matrix where these row-wise or column-wise outliers are removed. Then, as a second step, these outliers are projected as supplementary points into the existing CA solution. Using Equation (5.8) in Section 5.2, if an outlier $\boldsymbol{a}$ is a row point, its coordinates in the K-dimensional CA solution are given by $(\boldsymbol{a}/\sum_j a_j)\boldsymbol{\Gamma}_K$ and if an outlier $\boldsymbol{b}$ is a column point, its coordinates in the K-dimensional CA solution are given by $(\boldsymbol{b}/\sum_i b_i)\boldsymbol{\Phi}_K$.

The supplementary points method is a standard method to deal with outliers in CA, see, for example, Hoffman and Franke (1986), Bendixen (1996), Greenacre (2017),

and Riani et al. (2022). However, as we argued above, outliers may be caused by a single cell in the data matrix, and deleting an entire row or column where cell-wise outliers occur from the contingency table leads to a loss of the entire category, including outlying and non-outlying cells. In contrast, reconstitution of order h eliminates the effect of only the outlying cells, thus keeping as much information as possible in the analysis.

5.5 Empirical studies/Results

We consider two datasets, the attributes of brands of cars and ocean plastic datasets. The attributes of brands of cars dataset is a classic dataset to study CA with the problem of outliers, see, for example, Riani et al. (2022); Raymaekers and Rousseeuw (2024). Therefore, we compare reconstitution of order h, MacroPCA, and the supplementary points method on this dataset.

The ocean plastic dataset is an incidence dataset created by Vonk, Bos, Smeets, and Van Sebille (2024). We use this dataset to show that the reconstitution algorithm is appropriate for incidence data as well. However, we do not discuss MacroPCA for this example, as MacroPCA applied to this dataset yielded a degenerate solution (See Supplementary materials). The reason for this is not clear to us, but we notice that assumptions underlying MacroPCA are severely violated by the matrix of standardized residuals. Therefore, for this dataset, we only compare reconstitution of order h and the supplementary points method.

5.5.1 The attributes of brands of cars data

As a first dataset, we use the attributes of brands of cars dataset to illustrate our method. The dataset has been analysed before in Riani et al. (2022); Raymaekers and Rousseeuw (2024). This dataset is a part of the R package *cellWise* (Raymaekers, Rousseeuw, Van den Bossche, & Hubert, 2023). See Table 5.2 for the data. The contingency table consists of 39 rows and 7 columns. The rows represent 39 brands of cars, such as *Jeep*, *Porsche*, and *Volvo*. The seven columns represent the attributes: *Fuel Economy*, *Innovation*, *Performance*, *Quality*, *Safety*, *Style*, and *Value*. In total 1,578 participants were asked what they considered attributes for the 39 different vehicle brands. They selected all attributes in the list which they felt applied to a brand. An entry in the table represents the number of respondents that chose the attribute for a car. In total this led to 11,713 scorings. We note that this is not a typical contingency table as in a typical table the total count is identical to the number of respondents.

Figure 5.1a shows the symmetric plot of CA. The first four singular values, with percentage of inertia displayed between brackets, are 0.335 (41.3%), 0.281 (28.9%), 0.171 (10.7%), and 0.157 (9.0%). Using the elbow criterion, we decide to interpret two dimensions.

The first dimension contrasts cars that score high on *Fuel Economy* versus cars that

score high on *Style* and *Performance*. On the second dimension the car brand *Volvo* is far from other brands of cars, and the attribute *Safety* is close by. Where the marginal proportion of *Volvo* is 0.024, its contribution to the second dimension is 65.7%. For *Safety* the marginal proportion is 0.132, but the contribution to the second dimension is 75.2%. In addition, the contribution of cell (*Volvo*, *Safety*) to the total inertia is 17.7%. Hence the cell (*Volvo*, *Safety*) is cell-wise outlier, leading to outlying points for *Volvo* and *Safety* on dimension 2.

5.5.1.1 Reconstitution algorithm

Here we use reconstitution algorithm of order 2 to handle the cell-wise outlier (*Volvo*, *Safety*). Using the reconstitution algorithm, the value 180 in (*Volvo*, *Safety*) becomes 27.0. (Reconstitution of order 0 leads an imputed value of 13.1, but the graphic results are similar.) The contribution of cell (*Volvo*, *Safety*) to the total inertia went down from 17.7% to 0.4%. The first four singular values become 0.334 (51.0%), 0.186 (15.8%), 0.170 (13.2%), and 0.156 (11.1%). It is clear that the second dimension now is less important, the proportion of inertia went down from 28.9% to 15.8%. The singular values of dimensions 2, 3 and 4 do not differ much, and using the elbow criterion, we decide only to study the first dimension. Also, since in a contingency table the singular value can be interpreted as the canonical correlation between the row variable and the column variable, with 0.186 the second singular value is quite small.

Figure 5.1b is a symmetric CA plot of the reconstituted table. On the first dimension the configuration of row and column points is similar to the configuration of the CA of the original table, except for the change of location of *Volvo*. *Safety* is still in a similar position, and the reason for this difference between *Volvo* and *Safety* is that bringing down the value of 180 to 27 has a much larger impact on the profile of *Volvo*, that originally had a marginal total of 276, than the marginal total of *Safety*, that originally was 1,551. Note that by eliminating the impact of a single cell the new figure is much better readable than Figure 5.1a.

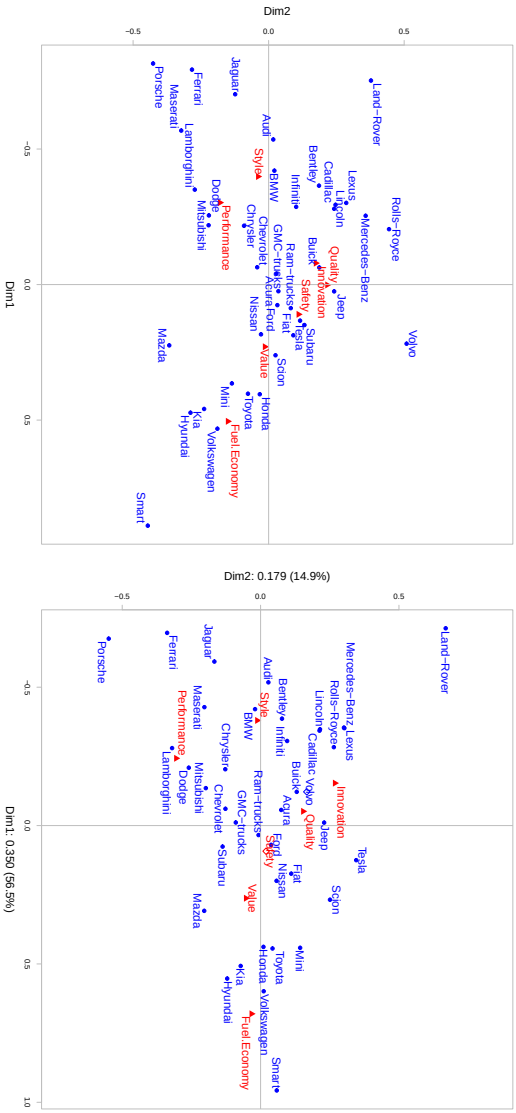
By eliminating the influence of a single cell the reconstitution method allows us to arrive at the simple conclusion that (i) there is a single outlying cell for *Volvo* and *Safety*, as *Safety* is chosen as the outstanding characteristic of *Volvo* (180 out of 276 scores for *Volvo* come from *Safety*), and (ii) there is largely a one-dimensional structure for the cars and features going from *Land-Rover*, *Ferrari* and *Porsche* on the left, scoring higher than average on *Style* and *Performance*, to *Smart*, *Volkswagen*, *Hyundai* and *Kia*, on the right, scoring higher on *Fuel Economy*, with the other car types and features ordered in between.

5.5.1.2 MacroPCA

We obtain the results of MacroPCA by applying the *MacroPCA* function in the R package *cellWise* (Raymaekers et al., 2023). We use the same parameter setting as in Raymaekers and Rousseeuw (2024) and Raymaekers et al. (2023), except for $\alpha = 0.97$ and $k = 2$. By setting $\alpha = 0.97$ we make the number of non-outlying rows as large as

Table 5.2: Car data matrix

	Fuel Econo.	Innov.	Perform.	Quality	Safety	Style	Value	Total	Proport.
Acura	24	38	28	20	28	33	25	196	0.017
Audi	9	54	54	30	19	67	8	241	0.021
Bentley	0	16	18	25	9	27	17	112	0.010
BMW	14	83	94	55	38	93	35	412	0.035
Buick	25	48	39	58	52	52	43	317	0.027
Cadillac	14	73	50	76	40	83	36	372	0.032
Chevrolet	114	103	202	174	140	160	145	1,038	0.089
Chrysler	38	65	96	54	54	103	72	482	0.041
Dodge	60	61	141	61	63	133	69	588	0.050
Ferrari	0	20	45	10	8	46	5	134	0.011
Fiat	19	21	17	20	15	7	16	115	0.010
Ford	167	180	169	179	161	157	188	1,201	0.103
GMC-trucks	40	40	64	57	80	50	58	389	0.033
Honda	163	68	73	118	104	50	135	711	0.061
Hyundai	97	25	31	27	35	42	82	339	0.029
Infiniti	5	39	31	15	10	17	16	133	0.011
Jaguar	0	3	18	19	3	47	12	102	0.009
Jeep	18	33	14	51	19	41	52	228	0.019
Kia	68	30	17	13	24	42	109	303	0.026
Lamborghini	5	19	37	8	6	23	24	122	0.010
Land-Rover	0	43	0	5	0	47	2	97	0.008
Lexus	10	62	29	50	27	64	26	268	0.023
Lincoln	6	37	23	31	24	40	19	180	0.015
Maserati	0	6	9	0	0	41	25	81	0.007
Mazda	46	23	34	10	12	26	38	189	0.016
Mercedes-Benz	8	83	44	87	58	82	42	404	0.034
Mini	23	12	4	4	13	12	4	72	0.006
Mitsubishi	20	13	33	23	7	32	13	141	0.012
Nissan	80	68	51	53	52	55	70	429	0.037
Porsche	0	17	66	14	6	42	5	150	0.013
Ram-trucks	9	22	21	10	18	1	16	97	0.008
Rolls-Royce	0	4	4	35	11	25	17	96	0.008
Scion	20	24	11	6	11	4	4	80	0.007
Smart	38	9	3	7	0	5	10	72	0.006
Subaru	19	14	32	33	75	20	40	233	0.020
Tesla	23	35	10	12	9	15	12	116	0.010
Toyota	238	116	95	134	113	74	150	920	0.079
Volkswagen	90	30	25	37	27	22	46	277	0.024
Volvo	9	15	16	31	180	14	11	276	0.024
Total	1,519	1,652	1,748	1,652	1,551	1,894	1,697	11,713	
Proport.	0.130	0.141	0.149	0.141	0.132	0.162	0.145		1.000



(c) CA MacroPCA of order 2

(d) CA supplementary points Volvo and Safety

(a) CA original table

(b) CA reconstitution of order 2 (Volvo, safety)

Figure 5.1: Car dataset CA plot

possible. We choose $k = 2$ because this simplifies the comparison with the reconstitution of order $h = 2$ in CA.

The results from the first step in MacroPCA, DCC, provides a cellmap. See Figure 5.2. The red or blue cells indicate cellwise outliers. Specifically, red cells indicate that the observed values are much larger than the predicted values, and for blue cells the opposite holds. Thus DDC finds 19 cellwise outliers, including the cellwise outlier (*Volvo*, *Safety*) found using the visual inspection employed in Section 5.5.1.1. DDC shows there is no row-wise outlier. However, in the second part of MacroPCA, there are 2 row-wise outliers, which are *Land-Rover* and *Volvo*.

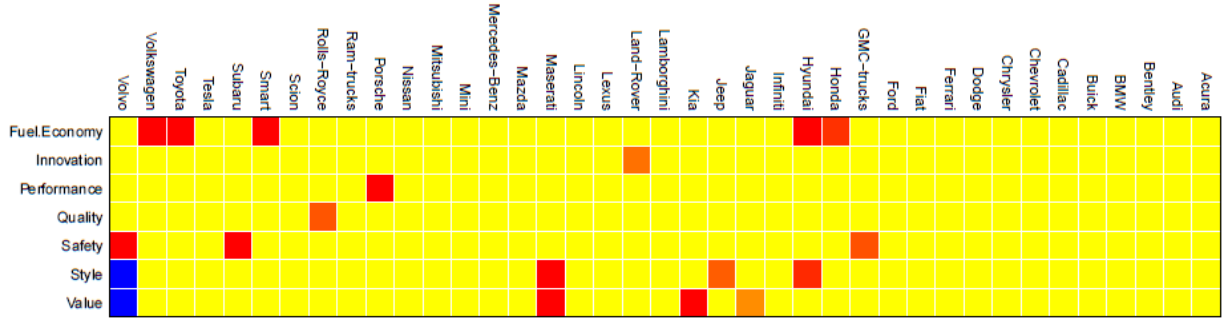


Figure 5.2: Car dataset

Figure 5.1c is the corresponding symmetric CA-type plot. On the first dimension, the configuration of row and column points is similar to the original Figure 5.1a.

5.5.1.3 Supplementary points method

Here we treat *Volvo* and *Safety* as supplementary points. Thus the table analysed has size 38×6, and now, row *Volvo* and column *Safety* have no effect on the solution of CA but are projected into it afterwards. The first four singular values are 0.350 (56.5%), 0.179 (14.9%), 0.166 (12.7%), and 0.150 (10.3%). As in the reconstitution approach, the second dimension is now less important, the proportion of inertia went down from 28.9% to 14.9%, and using the elbow criterion, only the first dimension is to be studied.

Figure 5.1d shows a symmetric CA plot, where *Volvo* and *Safety* are added as supplementary points. On the first dimension the configuration of row and column points is similar to the original Figure 5.1a, except for *Volvo*. Again, *Safety* is still in a similar position. For this dataset the interpretation using the supplementary points method is very similar to the interpretation using the reconstitution approach.

5.5.2 The ocean plastic data

The ocean plastic dataset is created by Vonk et al. (2024) to analyze how scientific studies on ocean plastic are communicated in press releases. The study analyzed press releases published on EurekAlert! between January 2017 and December 2021. In the

Table 5.3: Frame variables

CCC	Cause: Ocean climate change
Resp.C.P	Actor responsible for cause: Politics
Resp.C.I	Actor responsible for cause: Industry
Resp.C.C	Actor responsible for cause: Regions/Countries
Resp.C.S	Actor responsible for cause: Society
Resp.C.O	Actor responsible for cause: Other

(a) Causal interpretation

PH	Health
PE	Economic
PB	Biological
PnB	Non-Biological
PT	Treatment
PC	Conflict
OC	Opportunity

(b) Problem definition

Tr	Treatment recommendation

(c) Treatment recommendation

OT	Opportunity due to treatment
Resp.T.P	Actor responsible for treatment: Politics
Resp.T.I	Actor responsible for treatment: Industry
Resp.T.C	Actor responsible for treatment: Regions/Countries
Resp.T.S	Actor responsible for treatment: Society
Resp.T.O	Actor responsible for treatment: Other
Ur	Urgency to take action

(d) Moral evaluation

analysis, variables defining the four frame elements of Entman (1993), namely causal interpretation, problem definition, moral evaluation, and treatment recommendation were noted, resulting in 21 frame variables. Table 5.3 summarizes these framing variables, while a more detailed description can be found in Appendix 1 of Vonk et al. (2024).

The causal interpretation (a) was coded, when the text referred to climate change (*CCC*) as a cause of problems. It was coded whether an entity was held responsible for causing climate change, ocean plastics or related problems (*Resp.C.P*, *Resp.C.I*, *Resp.C.C*, *Resp.C.S*, and *Resp.C.O*). The problem definition (b) describes different problems (*PH*, *PE*, *PB*, *PnB*, *PT*, *PC*) or opportunities (*OC*) stated in the text. The moral evaluation (d) was coded when an entity was held responsible for solving problems (*Resp.T.P*, *Resp.T.I*, *Resp.T.C*, *Resp.T.S*, and *Resp.T.O*); when opportunities would be named if problems were mitigated (*OT*); or when the text stated that mitigation of problems was urgently needed (*Ur*). The treatment recommendation (c) described a solution that reduced or remedied problems or their cause (*Tr*).

The ocean plastic dataset has 81 press releases in the rows and 21 framing variables in the columns with 0 or 1 in each cell where 1 means the framing variable is present in the text and 0 otherwise. See Table 5.4 in supplementary materials A. The table has $81 \times 21 = 1{,}701$ cells of which 1,389 have a value 0. Note that Documents *10*, *34*, *50*, and *81* are identical, and so are Documents *13*, *19*, *26*, *27*, *46*, *56*, *65*, *69*, and *84*, Documents *15*, *71*, and *75*, Documents *17* and *59*, Documents *28* and *31*, Documents *30* and *86*, Documents *41*, *44*, *63*, and *67*, Documents *48* and *77*, and Documents *64*, *72*, and *85*. As the profiles are identical in each group, the points have an identical position in the graphic configurations and we only provide the label *10*, *13*, *15*, *17*, *28*, *30*, *41*, *48* and *64*.

Figure 5.3a is a symmetric plot of the dataset. The first four singular values, with percentages of inertia displayed between brackets, are 0.671 (13.2%), 0.588 (10.2%), 0.570 (9.6%), and 0.544 (8.7%). The closeness of the singular values shows that the

dataset cannot be summarized in a small number of dimensions.

The first dimension contrasts Opportunity due to treatment (*OT*), Treatment related problems (*PT*) and Treatment recommendation (*Tr*), Responsibility for treatment framings *T.O*, *T.P*, *T.C* and *T.I* on the left versus responsibility for causes framings *C.P*, *C.S* and *C.I*, and Problem definitions such as Opportunity (*OC*), Health (*PH*), Economic (*PE*), Non-Biological (*PnB*), and Biological (*PB*) on the right. On the second dimension, *Resp.C.I*, i.e. industry is responsible for cause, is far from the origin. The marginal proportion of *Resp.C.I* is 0.013, and its contribution to the second dimension is 76.9%. *Resp.C.I* masks the visualisation of the structure in the dataset and reduces the readability of this map. Documents *17*, *59*, which have identical scores, are far from the origin and are closest to *Resp.C.I*. The marginal proportion of documents *17/59* jointly is 0.013, yet its contribution to the second dimension is 61.0%. Also, the contribution of the two cells (*17/59*, *Resp.C.I*) to the total inertia is 7.0%, which is large (note that there are 81×21 cells). Hence the cells (*17/59*, *Resp.C.I*) are cell-wise outliers, leading to outlying points for *17*, *59* and *Resp.C.I* on dimension 2.

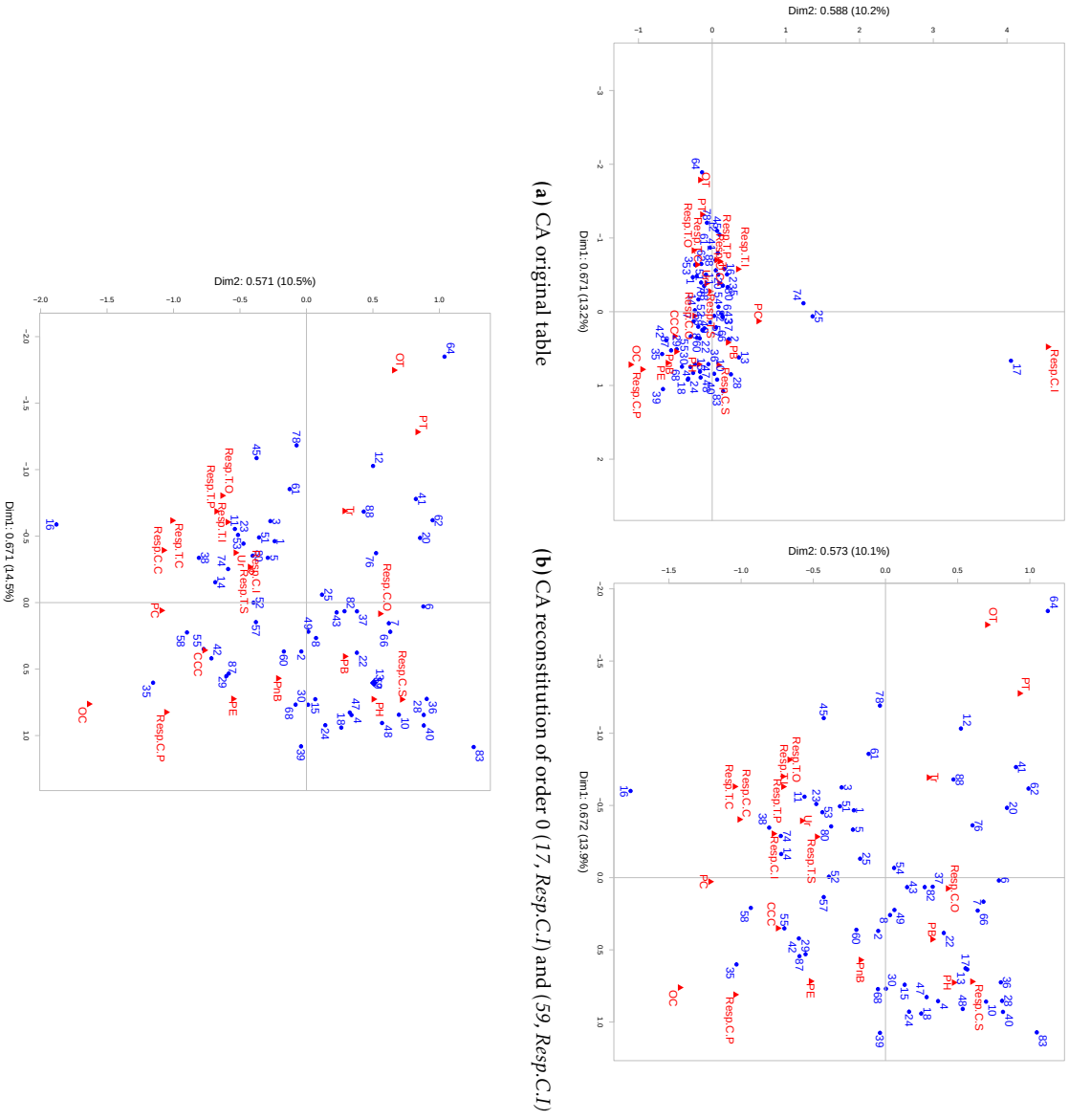
5.5.2.1 Reconstitution algorithm

Again, we used the reconstitution algorithm of order 2 to handle the cell-wise outliers. However, this created a negative imputed value -0.0006 for outlying cells (*17/59*, *Resp.C.I*). Negative values are not easy to interpret in an incidence matrix. Therefore, we applied reconstitution of order 0. This yields value 0.0065 for the cells (*17/59*, *Resp.C.I*). Now the documents *17/59*, having a 1 in the framing variable *PB*, 0.0065 in Resp.C.I and otherwise 0, are similar to documents *13*, *19*, *26*, *27*, *46*, *56*, *65*, *69*, *84* which have 1 in *PB* and 0 otherwise. The first four singular values are 0.672 (13.9%), 0.573 (10.1%), 0.548 (9.3%), and 0.519 (8.3%).

Figure 5.3b is a symmetric CA plot of the reconstituted table. On the first dimension the configuration of column points is similar to the configuration in Figure 5.3a, except for *Resp.C.I*. *Resp.C.I* is not close to Documents *17/59*, and *Resp.C.I*, *17/59* are not far from the origin. Now, the contributions to the second dimension of *Resp.C.I* is only 1.2% and of *17/59* jointly 0.6%.

By reducing the influence of cells (*17/59*, *Resp.C.I*), the new figure is much better readable than Figure 5.3a. A full interpretation of the table makes use of the outliers found in standard CA, and the CA solution found with the reconstitution method. The standard CA reveals a strong positive relation between *17/59* and *Resp.C.I*. We interpret the CA solution found with the reconstitution method by interpreting the four quadrants of Figure 5.3b.

- Press releases in the first quadrant focus on problems related to biology (*PB*) and human health (*PH*), and place the responsibility for causes at society (*Resp.C.S*);
- The second quadrant represents problems related to treatment (*PT*) and solutions to these problems in the form of treatment (*Tr*), and opportunity if treatment is carried out (*OT*);



(a) CA original table

(b) CA reconstitution of order 0 (*17*, *Resp.C.I*) and (*59*, *Resp.C.I*)

(c) CA supplementary points *17*, *59*, and *Resp.C.I*

Figure 5.3: Ocean plastic dataset CA plot

- Press releases in the third quadrant focus on the urgency to treat ocean plastic (*Ur*) and hold entity responsible for carrying out that treatment (*Resp.T.C*, *Resp.T.P*, *Resp.T.I*, *Resp.T.O*, *Resp.T.S*). In some cases they also state the responsibility for cause at industry (*Resp.C.I*) and specific regions/countries (*Resp.C.C*);
- Press releases in the fourth quadrant focus on the interconnections between ocean plastic and climate change (*CCC*) and they state non-biological (*PnB*) and economic consequences (*PE*). The fourth quadrant also represents the responsibility for cause at politics (*Resp.C.P*) and opportunity due to problems (*OC*). We note that the marginal frequencies of *Resp.C.P* and *OC* are low, namely 1 and 2 respectively.

5.5.2.2 Supplementary points method

Here we treat *17/59* and *Resp.C.I* as supplementary points. Thus the size of the table analysed is 79×20. Now, due to deleting *Resp.C.I*, documents *25* and *54* are also identical. Rows 17/59 and column *Resp.C.I* have no effect on the solution of CA but are projected into it afterwards. The first four singular values are 0.671 (14.5%), 0.571 (10.5%), 0.544 (9.6%), and 0.511 (8.4%).

Figure 5.3c shows the symmetric CA plot for the supplementary points method. Figure 5.3c is similar to Figure 5.3b.

5.6 Discussion and conclusion

In this paper, we propose to use the reconstitution algorithm of order h to deal with outlying cells in CA. The reconstitution algorithm of order h can reduce the effects of single outlying cells on the CA solution. We compare it with MacroPCA and the supplementary points method.

In comparison to the reconstitution approach, MacroPCA imputes outlying cells in the matrix of standardized residuals instead of in the original matrix. Apart from imputing cell-wise outliers, it can also eliminate complete rows. Yet, MacroPCA is not as transparent and straightforward as the reconstitution approach. One of the reasons is that it is originally proposed for the analysis continuous data and makes distributional assumptions, which does not hold for the reconstitution approach. Due to these distributional assumptions, that do not always fit with how the data originate, in our view it appears to flag too many cells as outlying cells.

The supplementary points method deletes complete rows or columns. In contrast, the reconstitution algorithm only reduces the influence of outlying cells. Thus, the reconstitution algorithm uses more information in the data and is, from this perspective, preferable.

We analysed two real data sets to illustrate the use of the reconstitution algorithm and compared the algorithm with the supplementary points method and MacroPCA. For the contingency table car dataset, the three methods yielded similar results. For

the ocean plastic dataset, the reconstitution algorithm and the supplementary points method had similar results, but MacroPCA failed.

We are not able to show empirically that the reconstitution method is preferable over the supplementary points method and MacroPCA. However, on theoretical grounds the reconstitution method is preferable: it eliminates only single cells to handle outlier problems, thus it is not necessarily deleting more information than is necessary.

5.7 Software

The reconstitution algorithm of order h is implemented by a function *reconca* both for $h = 0$ and $h > 0$. The function is written by adjusting the function *imputeCA* in the R Package *missMDA*. Josse and Husson (2016) proposed the R package *missMDA* for handling missing values in multivariate data analysis, where the function *imputeCA* is meant for missing values in CA. Another R Package, which can perform a reconstitution algorithm of order zero, is *anacor*, proposed for simple and canonical CA by De Leeuw and Mair (2009), to deal with missing data in CA.

The MacroPCA method is performed by the *MacroPCA* function in R package *cellWise*. The MacroPCA method is proposed for PCA (Hubert et al., 2019) and adjusted for CA (Raymaekers & Rousseeuw, 2024). To fit CA, the original matrix is replaced with the matrix of standardized residuals.

Appendix 5.A Ocean plastic dataset of size 81×21

Table 5.4: Ocean plastic data matrix

	PH	PE	PB	PnB	PT	PC	OC	OT	CCC	Resp.C.P	Resp.C.I	Resp.C.C	Resp.C.S	Resp.C.O	Resp.T.P	Resp.T.I	Resp.T.C	Resp.T.S	Resp.T.O	Ur	Tr	Total	Proport.
1	0	0	1	1	0	0	0	1	1	0	0	0	0	0	1	0	0	0	0	1	1	7	0.022
2	0	0	1	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	3	0.010
3	1	0	0	1	1	0	0	1	1	0	0	0	0	1	1	1	1	1	1	1	1	13	0.042
4	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0.010
5	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	3	0.010
6	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	2	0.006
7	1	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0.013
8	1	1	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	1	1	7	0.022
10	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0.006
11	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	0	1	5	0.016
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0.003
13	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0.003
14	1	0	1	1	0	0	0	0	1	0	0	0	0	0	1	1	1	1	1	0	0	9	0.029
15	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0.006
16	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0.003
17	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	2	0.006
18	1	1	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	5	0.016
19	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0.003
20	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	1	4	0.013
22	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	4	0.013
23	0	0	1	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	1	4	0.013
24	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0.010
25	0	0	1	0	0	0	0	0	0	0	1	0	1	0	0	0	0	1	0	1	1	6	0.019
26	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0.003
27	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0.003
28	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	2	0.006
29	0	1	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	5	0.016
30	0	0	1	1	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	4	0.013
31	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	2	0.006
34	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0.006
35	0	1	1	1	0	0	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0	6	0.019
36	1	0	1	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	4	0.013
37	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	1	4	0.013
38	0	0	1	0	0	0	0	0	1	0	0	1	0	0	0	0	1	1	0	1	1	7	0.022
39	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0.006
40	1	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	3	0.010
41	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	3	0.010
42	0	0	1	1	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	1	5	0.016
43	1	0	1	0	0	0	0	0	0	0	0	0	1	0	0	1	0	1	0	0	1	6	0.019
44	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	3	0.010
45	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1	0	1	0	1	1	6	0.019
46	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0.003
47	1	0	1	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	4	0.013
48	1	0	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	4	0.013
49	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	3	0.010
50	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0.006
51	0	1	1	0	1	0	0	0	0	0	0	1	0	0	1	0	0	0	0	1	1	7	0.022
52	0	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	1	1	6	0.019
53	0	0	0	1	1	0	0	0	1	0	0	1	0	0	0	0	0	1	0	1	1	7	0.022
54	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	1	1	5	0.016
55	0	0	1	1	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	4	0.013
56	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0.003
57	0	0	1	1	0	1	0	0	1	0	0	0	1	0	0	0	0	1	0	1	1	8	0.026
58	0	1	1	0	0	1	0	0	1	0	0	0	0	0	0	0	0	1	0	1	0	6	0.019
59	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	2	0.006
60	1	0	1	1	0	0	0	0	1	0	0	0	1	0	1	0	0	0	0	1	0	7	0.022
61	0	0	1	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1	1	6	0.019
62	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	3	0.010
63	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	3	0.010
64	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	3	0.010
65	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0.003
66	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	3	0.010
67	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	3	0.010
68	1	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	4	0.013
69	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0.003
71	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0.006
72	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	3	0.010
74	1	0	1	0	0	1	0	0	0	0	1	1	0	0	1	1	0	0	0	0	1	8	0.026
75	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0.006
76	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	4	0.013
77	1	0	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	4	0.013
78	0	0	0	0	1	0	0	1	0	0	0	1	0	0	0	0	0	1	0	1	1	6	0.019
80	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	1	4	0.013
81	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0.006
82	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	4	0.013
83	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0.003
84	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0.003
85	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	3	0.010
86	0	0	1	1	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	4	0.013
87	1	1	1	1	0	0	0	0	1	1	0	0	0	0	0	0	0	1	0	1	0	8	0.026
88	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	4	0.013
Total	26	10	68	25	16	3	2	8	18	1	4	12	21	2	10	5	4	20	3	16	38	312	
Proport.	0.083	0.032	0.218	0.080	0.051	0.010	0.006	0.026	0.058	0.003	0.013	0.038	0.067	0.006	0.032	0.016	0.013	0.064	0.010	0.051	0.122		1.000

Appendix 5.B MacroPCA for ocean plastic dataset

We use the same parameter as in the car dataset to deal with the ocean plastic dataset. Figure 5.4 is a cellmap based on the results from the first part DDC of MacroPCA. MacroPCA flags 225 cellwise outliers. The 225 cellwise outliers include (*17*, *Resp.C.I*) and (*59*, *Resp.C.I*), which we took as cellwise outliers in reconstitution of order h.

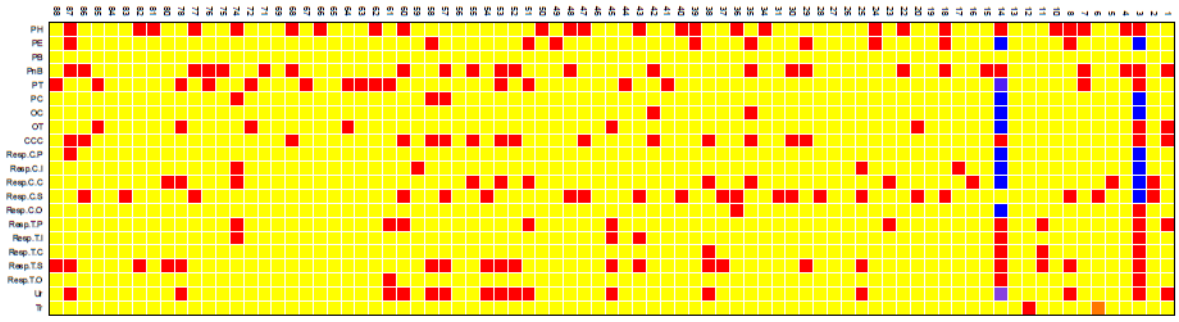


Figure 5.4: Ocean plastic dataset

Figure 5.5 is the corresponding symmetric CA-type plot based on MacroPCA. MacroPCA does not work well in the ocean plastic dataset. The reason may be that the analyzed matrix severely violates the assumption of multivariate Gaussian distribution.

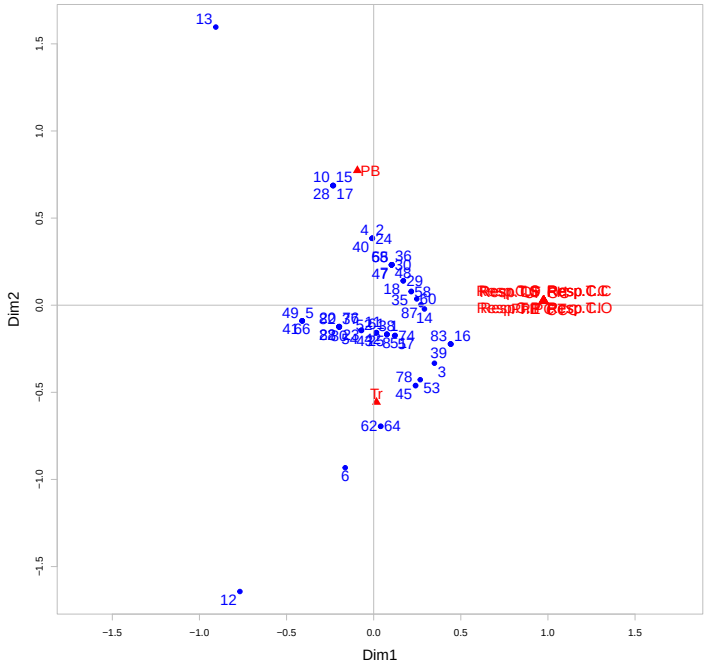


Figure 5.5: CA MacroPCA of order 2

References

Ab Samat, N., Murad, M. A. A., Abdullah, M. T., & Atan, R. (2008). Term weighting schemes experiment based on SVD for Malay text retrieval. *International Journal of Computer Science and Network Security (IJCSNS)*, *8*(10), 357-361.

Aggarwal, C. C. (2018). *Machine learning for text*. Springer.

Agresti, A. (2007). *An introduction to categorical data analysis*. Wiley.

Albright, R. (2004). Taming Text with the SVD. *SAS Institute Inc*.

Al-Qahtani, M., Amira, A., & Ramzan, N. (2015). An efficient information retrieval technique for e-health systems. In *2015 International Conference on Systems, Signals and Image Processing (IWSSIP)* (pp. 257–260).

Alqahtani, Y., Al-Twairesh, N., & Alsanad, A. (2023). Improving sentiment domain adaptation for arabic using an unsupervised self-labeling framework. *Information Processing & Management*, *60*(3), 103338.

Altszyler, E., Sigman, M., Ribeiro, S., & Slezak, D. F. (2016). Comparative study of LSA vs Word2vec embeddings in small corpora: a case study in dreams database. *arXiv preprint arXiv:1610.01520*.

Arenas-Márquez, F. J., Martinez-Torres, R., & Toral, S. (2021). Convolutional neural encoding of online reviews for the identification of travel group type topics on tripadvisor. *Information Processing & Management*, *58*(5), 102645.

Azmi, A. M., Al-Jouie, M. F., & Hussain, M. (2019). AAEE–Automated evaluation of students' essays in Arabic language. *Information Processing & Management*, *56*(5), 1736–1752.

Bacciu, A., Morgia, M. L., Mei, A., Nemmi, E. N., Neri, V., & Stefa, J. (2019). Bot and Gender Detection of Twitter Accounts Using Distortion and LSA. In *CLEF*.

Bae, Y. S., Kim, K. H., Kim, H. K., Choi, S. W., Ko, T., Seo, H. H., ... Jeon, H. (2021). Keyword extraction algorithm for classifying smoking status from unstructured bilingual electronic health records based on natural language processing. *Applied Sciences*, *11*(19), 8812.

Bagheri, A. (2021). *Text mining in healthcare: bringing structure to electronic health records* (Doctoral dissertation, Utrecht University).

Barman, D., & Chowdhury, N. (2020). A novel semi supervised approach for text classification. *International Journal of Information Technology*, *12*(4), 1147–1157.

Baroni, M., Bernardini, S., Ferraresi, A., & Zanchetta, E. (2009). The WaCky wide web: a collection of very large linguistically processed web-crawled corpora. *Language Resources and Evaluation*, *43*(3), 209–226.

Bartlett, M. S. (1936). The square root transformation in analysis of variance. *Supplement to the Journal of the Royal Statistical Society*, *3*(1), 68–78.

Beh, E. J., & Lombardo, R. (2021). *An introduction to correspondence analysis*. John Wiley & Sons.

Beh, E. J., Lombardo, R., & Alberti, G. (2018). Correspondence analysis and the freeman–tukey statistic: A study of archaeological data. *Computational Statistics & Data Analysis*, *128*, 73–86.

Bendixen, M. (1996). A practical guide to the use of correspondence analysis in marketing research. *Marketing Research On-Line*.

Benzécri, J.-P. (1973). *L'analyse des données* (Vol. 1 and 2). Paris: Dunod.

Berry, M. W., Dumais, S. T., & O'Brien, G. W. (1995). Using linear algebra for intelligent information retrieval. *SIAM Review*, *37*(4), 573–595.

Bianco, G. D., Duarte, D., & Gonçalves, M. A. (2023). Reducing the user labeling effort in effective high recall tasks by fine-tuning active learning. *Journal of Intelligent Information Systems*.

Billhardt, H., Borrajo, D., & Maojo, V. (2002). A context vector model for information retrieval. *Journal of the American Society for Information Science and Technology*, *53*(3), 236–249.

BNC Consortium. (2007). *British National Corpus, XML edition.* Retrieved from `http://hdl.handle.net/20.500.14106/2554` (Literary and Linguistic Data Service)

Bounabi, M., Moutaouakil, K. E., & Satori, K. (2019). A comparison of text classification methods using different stemming techniques. *International Journal of Computer Applications in Technology*, *60*(4), 298–306.

Bozkurt, I. N., Baghoglu, O., & Uyar, E. (2007). Authorship attribution. In *2007 22nd international symposium on computer and information sciences* (pp. 1–5).

Bruni, E., Boleda, G., Baroni, M., & Tran, N.-K. (2012). Distributional semantics in technicolor. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 136–145).

Bullinaria, J. A., & Levy, J. P. (2007). Extracting semantic representations from word co-occurrence statistics: A computational study. *Behavior Research Methods*, *39*, 510–526.

Bullinaria, J. A., & Levy, J. P. (2012). Extracting semantic representations from word co-occurrence statistics: stop-lists, stemming, and SVD. *Behavior Research Methods*, *44*(3), 890–907.

Caron, J. (2001). Experiments with LSA scoring: Optimal rank and basis. In *Proceedings of the SIAM Computational Information Retrieval Workshop* (pp. 157–169).

Cauchy, A. (1847). Méthode générale pour la résolution des systemes d'équations simultanées. *Comp. Rend. Sci. Paris*, *25*(1847), 536–538.

Chang, C.-Y., Lee, S.-J., Wu, C.-H., Liu, C.-F., & Liu, C.-K. (2021). Using word semantic concepts for plagiarism detection in text documents. *Information Retrieval Journal*, *24*, 298–321.

Choulakian, V. (2020). Taxicab correspondence analysis of sparse two-way contingency tables. *Statistica Applicata - Italian Journal of Applied Statistics*, *29*(2-3), 153–179.

Church, K. W., & Hanks, P. (1990). Word association norms, mutual information, and lexicography. *Computational Linguistics*, *16*(1), 22–29.

Curran, J. R. (2004). *From distributional to semantic similarity* (Unpublished doctoral dissertation). University of Edinburgh.

Curran, J. R., & Moens, M. (2002). Improvements in automatic thesaurus extraction. In *Proceedings of the ACL-02 workshop on Unsupervised lexical acquisition* (pp. 59–66).

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, *41*(6), 391–407.

Deisenroth, M. P., Faisal, A. A., & Ong, C. S. (2020). *Mathematics for machine learning*. Cambridge University Press.

De Leeuw, J., & Mair, P. (2009). Simple and canonical correspondence analysis using the R package anacor. *Journal of Statistical Software*, *31*(5), 1–18.

De Leeuw, J., & Van der Heijden, P. G. M. (1988). Correspondence analysis of incomplete contingency tables. *Psychometrika*, *53*(2), 223–233.

De Leeuw, J., Van der Heijden, P. G. M., & Verboon, P. (1990). A latent time–budget model. *Statistica Neerlandica*, *44*(1), 1–22.

Di Gangi, M. A., Bosco, G. L., & Pilato, G. (2019). Effectiveness of data-driven induction of semantic spaces and traditional classifiers for sarcasm detection. *Natural Language Engineering*, *25*(2), 257–285.

Dodge, Y. (2008). *The concise encyclopedia of statistics*. Springer.

Downing, J. (1981). How well does the 4th-root transformation work-reply. *Canadian Journal of Fisheries and Aquatic Sciences*, *38*(1), 127–129.

Drozd, A., Gladkova, A., & Matsuoka, S. (2016). Word embeddings, analogies, and machine learning: Beyond king - man + woman = queen. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers* (pp. 3519–3530).

Duan, L., Gao, T., Ni, W., & Wang, W. (2021). A hybrid intelligent service recommendation by latent semantics and explicit ratings. *International Journal of Intelligent Systems*, *36*(12), 7867–7894.

Duchi, J., Hazan, E., & Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, *12*, 2121–2159.

Dumais, S. T. (1991). Improving the retrieval of information from external sources. *Behavior Research Methods, Instruments, & Computers*, *23*(2), 229–236.

Dumais, S. T., Furnas, G. W., Landauer, T. K., Deerwester, S., & Harshman, R. (1988). Using latent semantic analysis to improve access to textual information. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 281–285).

Dzisevič, R., & Šešok, D. (2019). Text classification using different feature extraction approaches. In *2019 Open Conference of Electrical, Electronic and Information Sciences (eStream)* (pp. 1–4).

Egleston, B. L., Bai, T., Bleicher, R. J., Taylor, S. J., Lutz, M. H., & Vucetic, S. (2021). Statistical inference for natural language processing algorithms with a demonstration using type 2 diabetes prediction from electronic health record notes. *Biometrics*, *77*(3), 1089–1100.

Elghazel, H., Aussem, A., Gharroudi, O., & Saadaoui, W. (2016). Ensemble multilabel text categorization based on rotation forest and latent semantic indexing. *Expert Systems with Applications*, *57*, 1–11.

Entman, R. M. (1993). Framing: Toward clarification of a fractured paradigm. *Journal of Communication*, *43*(4), 51–58.

Field, J., Clarke, K., & Warwick, R. M. (1982). A practical strategy for analysing multispecies distribution patterns. *Marine ecology progress series*, *8*(1), 37–52.

Finkelstein, L., Gabrilovich, E., Matias, Y., Rivlin, E., Solan, Z., Wolfman, G., & Ruppin, E. (2002). Placing search in context: The concept revisited. *ACM Transactions on Information Systems*, *20*(1), 116–131.

France, R. L., & Heung, B. (2023). Density variability of COVID-19 face mask litter: A cautionary tale for pandemic PPE waste monitoring. *Journal of Hazardous Materials Advances*, *9*, 100220.

Frobenius, G. (1912). *Über matrizen aus nicht negativen elementen.*

Gabriel, K. R. (1971). The biplot graphic display of matrices with application to principal component analysis. *Biometrika*, *58*(3), 453–467.

Gareth, J., Daniela, W., Trevor, H., & Robert, T. (2021). *An introduction to statistical learning: with applications in R*. Springer.

Gifi, A. (1990). *Nonlinear multivariate analysis*. Wiley.

Greenacre, M. (1984). *Theory and applications of correspondence analysis*. Academic Press.

Greenacre, M. (2009). Power transformations in correspondence analysis. *Computational Statistics & Data Analysis*, *53*(8), 3107–3116.

Greenacre, M. (2010). Log-ratio analysis is a limiting case of correspondence analysis. *Mathematical Geosciences*, *42*, 129–134.

Greenacre, M. (2013). The contributions of rare objects in correspondence analysis. *Ecology*, *94*(1), 241–249.

Greenacre, M. (2017). *Correspondence analysis in practice*. CRC press.

Greenacre, M., & Hastie, T. (1987). The geometric interpretation of correspondence analysis. *Journal of the American Statistical Association*, *82*(398), 437–447.

Greene, D., & Cunningham, P. (2006). Practical solutions to the problem of diagonal dominance in kernel document clustering. In *Proceedings of the 23rd International Conference on Machine Learning* (pp. 377–384).

Grubbs, F. E. (1969). Procedures for detecting outlying observations in samples. *Technometrics*, *11*(1), 1–21.

Guo, J., Cai, Y., Fan, Y., Sun, F., Zhang, R., & Cheng, X. (2022). Semantic models for the first-stage retrieval: A comprehensive review. *ACM Transactions on Information Systems (TOIS)*, *40*(4), 1–42.

Guo, S., & Yao, N. (2021). Document vector extension for documents classification. *IEEE Transactions on Knowledge & Data Engineering*, *33*(8), 3062–3074.

Gupta, H., & Patel, M. (2021). Method Of Text Summarization Using Lsa And Sentence Based Topic Modelling With Bert. In *2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS)* (pp. 511–517).

Guthrie, D. (2008). *Unsupervised detection of anomalous text* (Unpublished doctoral dissertation). University of Sheffield.

Handan-Nader, C. (2023). *Graph embeddings with influential outliers using correspondence analysis.* Retrieved October 2, 2023, from https://www.dropbox.com/scl/fi/7rc8jg5g6lwd1u9q2z71b/CA_Algorithms_Paper.pdf?rlkey=mg5jw71q17861nbbocahafn2g&dl=0.

Harris, Z. S. (1954). Distributional structure. *WORD*, *10*(2-3), 146–162.

Hassani, A., Iranmanesh, A., & Mansouri, N. (2021). Text mining using nonnegative matrix factorization and latent semantic analysis. *Neural Computing and Applications*, *33*, 13745-13766.

Hayashi, C. (1956). Theory and example of quantification (II). *Proceedings of the Institute of Statistical Mathematics*, *4*, 19–30.

Hayashi, C. (1992). Quantification method III or correspondence analysis in medical science. *Annals of Cancer Research and Therapy*, *1*(1), 17–21.

Hearst, M. A. (1999). Untangling text data mining. In *Proceedings of the 37th Annual meeting of the Association for Computational Linguistics* (pp. 3–10).

Hill, F., Reichart, R., & Korhonen, A. (2015). SimLex-999: Evaluating Semantic Models With (Genuine) Similarity Estimation. *Computational Linguistics*, *41*(4), 665–695.

Hill, M. O. (1973). Reciprocal averaging: an eigenvector method of ordination. *Journal of Ecology*, *61*(1), 237–249.

Hill, M. O. (1974). Correspondence analysis: a neglected multivariate method. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, *23*(3), 340–354.

Hoffman, D. L., & Franke, G. R. (1986). Correspondence Analysis: Graphical Representation of Categorical Data in Marketing Research. *Journal of Marketing Research*, *23*(3), 213–227.

Horasan, F. (2022). Latent Semantic Indexing-Based Hybrid Collaborative Filtering for Recommender Systems. *Arabian Journal for Science and Engineering*, *47*, 10639-10653.

Horasan, F., Erbay, H., Varçın, F., & Deniz, E. (2019). Alternate Low-Rank Matrix Approximation in Latent Semantic Analysis. *Scientific Programming*, *2019*, 1-12.

Hossain, T., Zahin Mauni, H., & Rab, R. (2022). Reducing the Effect of Imbalance in Text Classification Using SVD and GloVe with Ensemble and Deep Learning. *Computing and Informatics*, *41*(1), 98–115.

Hou, R., & Huang, C.-R. (2020). Classification of regional and genre varieties of chinese: A correspondence analysis approach based on comparable balanced corpora. *Natural Language Engineering*, *26*(6), 613–640.

Hsu, L. L., & Culhane, A. C. (2023). Correspondence analysis for dimension reduction, batch integration, and visualization of single-cell RNA-seq data. *Scientific Reports*, *13*, 1197.

Hu, X., Cai, Z., Franceschetti, D., Penumatsa, P., Graesser, A., Louwerse, M., ... Tutoring Research Group (2003). LSA: First dimension and dimensional weighting. In *Proceedings of the Annual Meeting of the Cognitive Science Society* (Vol. 25).

Hubert, M., Rousseeuw, P. J., & Van den Bossche, W. (2019). MacroPCA: An All-in-One PCA Method Allowing for Missing Values as Well as Cellwise and Rowwise Outliers. *Technometrics*, *61*(4), 459–473.

Jarman, A. M. (2020). Hierarchical Cluster Analysis: Comparison of Single linkage, Complete linkage, Average linkage and Centroid Linkage Method. *Georgia Southern University*.

Jiang, C., Yu, H.-F., Hsieh, C.-J., & Chang, K.-W. (2018). Learning word embeddings for low-resource languages by PU learning. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)* (pp. 1024–1034).

Jiao, Q., & Zhang, S. (2021). A brief survey of word embedding and its recent development. In *2021 IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)* (pp. 1697–1701).

Josse, J., Chavent, M., Liquet, B., & Husson, F. (2012). Handling Missing Values with Regularized Iterative Multiple Correspondence Analysis. *Journal of Classification*, *29*, 91–116.

Josse, J., & Husson, F. (2016). missMDA: A Package for Handling Missing Values in Multivariate Data Analysis. *Journal of Statistical Software*, *70*(1), 1–31.

Jurafsky, D., & Martin, J. H. (2023). Vector semantics and embeddings. *Speech and language processing*, 1–34. Retrieved December 21, 2023, from `https://web.stanford.edu/~jurafsky/slp3/6.pdf`.

Kalmukov, Y. (2022). Comparison of Latent Semantic Analysis and Vector Space Model for Automatic Identification of Competent Reviewers to Evaluate Papers. *International Journal of Advanced Computer Science and Applications*, *13*(2), 77-85.

Kestemont, M. (2017). *Who Wrote the Wilhelmus?* Retrieved July 17, 2021, from `https://github.com/mikekestemont/anthem`.

Kestemont, M., Stover, J., Koppel, M., Karsdorp, F., & Daelemans, W. (2016). Authenticating the writings of Julius Caesar. *Expert Systems with Applications*, *63*, 86-96.

Kestemont, M., Stronks, E., De Bruin, M., & Winkel, T. d. (2017a). Did a poet with donkey ears write the oldest anthem in the world? Ideological implications of the computational attribution of the Dutch national anthem to Petrus Dathenus. In *Digital humanities 2017, conference abstracts.*

Kestemont, M., Stronks, E., De Bruin, M., & Winkel, T. d. (2017b). *Van wie is het Wilhelmus? De auteur van het Nederlandse volkslied met de computer onderzocht*. Amsterdam University Press.

Kim, S.-K., McKay, D., Murphy, T. K., Bussing, R., McNamara, J. P., Goodman, W. K., & Storch, E. A. (2021). Age moderated–anxiety mediation for multimodal treatment outcome among children with obsessive-compulsive disorder: An evaluation with correspondence analysis. *Journal of Affective Disorders*, *282*, 766–775.

Kolda, T. G., & O'leary, D. P. (1998). A semidiscrete matrix decomposition for latent semantic indexing information retrieval. *ACM Transactions on Information Systems (TOIS)*, *16*(4), 322–346.

Koppel, M., & Seidman, S. (2013). Automatically identifying pseudepigraphic texts. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing* (pp. 1449–1454).

Kostensalo, J., Lidauer, M., Aernouts, B., Mäntysaari, P., Kokkonen, T., Lidauer, P., & Mehtiö, T. (2023). Short communication: Predicting blood plasma non-esterified fatty acid and beta-hydroxybutyrate concentrations from cow milk—addressing systematic issues in modelling. *animal*, *17*(9), 100912.

Kuhnt, S., Rapallo, F., & Rehage, A. (2014). Outlier detection in contingency tables based on minimal patterns. *Statistics and Computing*, *24*, 481–491.

Landauer, T. K., & Dumais, S. T. (1997). A solution to plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review*, *104*(2), 211-240.

Langovaya, A., Kuhnt, S., & Chouikha, H. (2013). Correspondence Analysis in the Case of Outliers. In A. Giusti, G. Ritter, & M. Vichi (Eds.), *Classification and Data Mining* (pp. 63–70).

Le, Q., & Mikolov, T. (2014). Distributed representations of sentences and documents. In E. P. Xing & T. Jebara (Eds.), *Proceedings of the 31st international conference on machine learning* (pp. 1188–1196).

Levy, O., & Goldberg, Y. (2014). Neural word embedding as implicit matrix factorization. In *Advances in Neural Information Processing Systems* (Vol. 27).

Levy, O., Goldberg, Y., & Dagan, I. (2015). Improving distributional similarity with lessons learned from word embeddings. *Transactions of the Association for Computational Linguistics*, *3*, 211–225.

Liu, T., Ungar, L., & Sedoc, J. (2019). Unsupervised post-processing of word vectors via conceptor negation. In *Proceedings of the AAAI Conference on Artificial Intelligence* (pp. 6778–6785).

Luong, T., Socher, R., & Manning, C. (2013). Better word representations with recursive neural networks for morphology. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning* (pp. 104–113).

Mannion, D., & Dixon, P. (2004). Sentence-length and authorship attribution: the case of Oliver Goldsmith. *Literary and Linguistic Computing*, *19*(4), 497–508.

McCarthy, P. M., Lewis, G. A., Dufty, D. F., & McNamara, D. S. (2006). Analyzing writing styles with Coh-Metrix. In *Proceedings of the Nineteenth International Florida Artificial Intelligence Research Society Conference* (pp. 764–769).

Mealand, D. L. (1995). Correspondence analysis of Luke. *Literary and Linguistic Computing*, *10*(3), 171–182.

Mealand, D. L. (1997). Measuring genre differences in Mark with correspondence analysis. *Literary and Linguistic Computing*, *12*(4), 227–245.

Michailidis, G., & De Leeuw, J. (1998). The Gifi system of descriptive multivariate analysis. *Statistical Science*, *13*(4), 307–336.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. In *Proceedings of the International Conference on Learning Representations ICLR'13.*

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems* (Vol. 26).

Morin, A. (1999). *Knowledge extraction in texts: a comparison of two methods.*

Morin, A. (2004). Intensive use of correspondence analysis for information retrieval. In *26th International Conference on Information Technology Interfaces, 2004* (pp. 255–258).

Mu, J., & Viswanath, P. (2018). All-but-the-top: Simple and effective post-processing for word representations. *6th International Conference on Learning Representations, ICLR 2018*.

Nakov, P., Popova, A., & Mateev, P. (2001). Weight functions impact on LSA performance. In *EuroConference Recent Advances in Natural Language Processing* (pp. 187–193).

Nishisato, S., Beh, E. J., Lombardo, R., & Clavel, J. G. (2021). On the analysis of overdispersed categorical data. In *Modern quantification theory: Joint graphical display, biplots, and alternatives* (pp. 215–231).

Nora-Chouteau, C. (1974). *Une méthode de reconstitution et d'analyse de données incomplètes* (Unpublished doctoral dissertation). Université Pierre et Marie Curie.

Österlund, A., Ödling, D., & Sahlgren, M. (2015). Factorization of latent variables in distributional semantic models. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing* (pp. 227–231).

Pakzad, A., & Analoui, M. (2021). A word selection method for producing interpretable distributional semantic word vectors. *Journal of Artificial Intelligence Research*, *72*, 1281–1305.

Parali, U., Zontul, M., & Ertuğrul, D. C. (2019). Information retrieval using the reduced row echelon form of a term-document matrix. *Journal of Internet Technology*.

Patil, A. (2022). Word Significance Analysis in Documents for Information Retrieval by LSA and TF-IDF using Kubeflow. In *Expert Clouds and Applications* (pp. 335–348).

Peng, J., & Feldman, A. (2017). Automatic idiom recognition with word embeddings. In *Information Management and Big Data* (pp. 17–29).

Pennington, J., Socher, R., & Manning, C. (2014). GloVe: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 1532–1543).

Perron, O. (1907). Zur Theorie der Matrices. *Mathematische Annalen*, *64*(2), 248-263.

Phillips, T., Saleh, A., Glazewski, K. D., Hmelosilver, C. E., Lee, S., Mott, B., & Lester, J. C. (2021). Comparing Natural Language Processing Methods for Text Classification of Small Educational Data. In *Companion proceedings 11th international conference on learning analytics & knowledge.*

Pitt, C. S., Bal, A. S., & Plangger, K. (2020). New approaches to psychographic consumer segmentation: Exploring fine art collectors using artificial intelligence, automated text analysis and correspondence analysis. *European Journal of Marketing*, *54*(2), 305–326.

Podkorytov, M., Biś, D., Cai, J., Amirizirtol, K., & Liu, X. (2020). Effects of Architecture and Training on Embedding Geometry and Feature Discriminability in BERT. In *2020 International Joint Conference on Neural Networks (IJCNN)* (pp. 1–8).

Qi, Q., Hessen, D. J., Deoskar, T., & Van der Heijden, P. G. M. (2023). A comparison of latent semantic analysis and correspondence analysis of document-term matrices. *Natural Language Engineering*, 1–31.

Qi, Q., Hessen, D. J., & Van der Heijden, P. G. M. (2024). Improving information retrieval through correspondence analysis instead of latent semantic analysis. *Journal of Intelligent Information Systems*, *62*, 209–230.

Radinsky, K., Agichtein, E., Gabrilovich, E., & Markovitch, S. (2011). A word at a time: Computing word relatedness using temporal semantic analysis. In *Proceedings of the 20th International Conference on World Wide Web* (pp. 337–346).

Raphael, E. B. (2023). Gendered representations in language: A corpus-based comparative study of adjective-noun collocations for marital relationships. *Theory and Practice in Language Studies*, *13*(5), 1191–1196.

Raymaekers, J., & Rousseeuw, P. J. (2024). Challenges of cellwise outliers. *Econometrics and Statistics*.

Raymaekers, J., Rousseeuw, P. J., Van den Bossche, W., & Hubert, M. (2023). *cellWise: Analyzing Data with Cellwise Outliers.* R package version 2.5.3.

Ren, X., & Coutanche, M. N. (2021). Sleep reduces the semantic coherence of memory recall: An application of latent semantic analysis to investigate memory reconstruction. *Psychonomic Bulletin & Review*, *28*(4), 1336–1343.

Rennie, J. (2005). *20 newsgroups data set.* Retrieved April 21, 2022, from `http://qwone.com/~jason/20Newsgroups/`.

Riani, M., Atkinson, A. C., Torti, F., & Corbellini, A. (2022). Robust correspondence analysis. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, *71*(5), 1381-1401.

Roesler, O., Aly, A., Taniguchi, T., & Hayashi, Y. (2019). Evaluation of Word Representations in Grounding Natural Language Instructions Through Computational Human-Robot Interaction. In *2019 14th ACM/IEEE International Conference on Human-Robot Interaction (HRI)* (pp. 307–316).

Rong, X. (2014). word2vec parameter learning explained. *arXiv preprint arXiv:1411.2738*.

Rousseeuw, P. J., & Van Den Bossche, W. (2018). Detecting deviating data cells. *Technometrics*, *60*(2), 135–145.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *nature*, *323*(6088), 533–536.

Salle, A., & Villavicencio, A. (2023). Understanding the effects of negative (and positive) pointwise mutual information on word vectors. *Journal of Experimental & Theoretical Artificial Intelligence*, *35*(8), 1161-1199.

Salle, A., Villavicencio, A., & Idiart, M. (2016). Matrix factorization using window sampling and negative sampling for improved word representations. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)* (pp. 419–424).

Salton, G., & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information processing & management*, *24*(5), 513–523.

Samuel, D., Kutuzov, A., Øvrelid, L., & Velldal, E. (2023). Trained on 100 million words and still in shape: BERT meets British National Corpus. In *Findings of the Association for Computational Linguistics: EACL 2023* (pp. 1954–1974).

Sasaki, S., Heinzerling, B., Suzuki, J., & Inui, K. (2023). Examining the effect of whitening on static and contextualized word embeddings. *Information Processing & Management*, *60*(3), 103272.

Satyam, A., Dawn, A. K., & Saha, S. K. (2014). A statistical analysis approach to author identification using latent semantic analysis: Notebook for PAN at CLEF 2014. In *2014 Working Notes for CLEF Conference*.

Schwertman, N. C., Owens, M. A., & Adnan, R. (2004). A simple more general boxplot method for identifying outliers. *Computational Statistics & Data Analysis*, *47*(1), 165-174.

Séguéla, J., & Saporta, G. (2011). A comparison between latent semantic analysis and correspondence analysis. In *CARME 2011 International conference on Correspondence Analysis and Related Methods*.

Séguéla, J., & Saporta, G. (2013). A hybrid recommender system to predict online job offer performance. *Revue des Nouvelles Technologies de l'Information*, *RNTI-E-25*, 177-197.

Shazeer, N., Doherty, R., Evans, C., & Waterson, C. (2016). Swivel: Improving embeddings by noticing what's missing. *arXiv preprint arXiv:1602.02215*.

Shi, T., & Liu, Z. (2014). Linking glove with word2vec. *arXiv preprint arXiv:1411.5595*.

Silge, J., & Robinson, D. (2017). *Text mining with R: A tidy approach*. O'Reilly Media.

Soboroff, I. M., Nicholas, C. K., Kukla, J. M., & Ebert, D. S. (1997). Visualizing document authorship using n-grams and latent semantic indexing. In *Proceedings of the 1997 workshop on New paradigms in information visualization and manipulation* (pp. 43–48).

Sripriya, T. P., & Srinivasan, M. R. (2018). Detection of Outlying Cells in Two-Way Contingency Tables. *Statistics and Applications*, *16*(2), 103–113.

Stamatatos, E. (2009). A survey of modern authorship attribution methods. *Journal of the American Society for information Science and Technology*, *60*(3), 538–556.

Stratos, K., Collins, M., & Hsu, D. (2015). Model-based word embeddings from decompositions of count matrices. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)* (pp. 1282–1291).

Suleman, R. M., & Korkontzelos, I. (2021). Extending latent semantic analysis to manage its syntactic blindness. *Expert Systems with Applications*, *165*, 114130.

Text8 dataset. (2006). Retrieved October 3, 2023, from `http://mattmahoney.net/dc/text8.zip`.

Tseng, H.-C., Chen, B., Chang, T.-H., & Sung, Y.-T. (2019). Integrating LSA-based hierarchical conceptual space and machine learning methods for leveling the readability of domain-specific texts. *Natural Language Engineering*, *25*(3), 331–361.

Tukey, J. W. (1977). *Exploratory data analysis*. Addison-Wesley, Reading, MA.

Turney, P. D., & Pantel, P. (2010). From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research*, *37*, 141–188.

Van der Heijden, P. G. M., De Falguerolles, A., & De Leeuw, J. (1989). A combined approach to contingency table analysis using correspondence analysis and loglinear analysis. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, *38*(2), 249–292.

Van Dam, A., Dekker, M., Morales-Castilla, I., Rodríguez, M. Á., Wichmann, D., & Baudena, M. (2021). Correspondence analysis, spectral clustering and graph embedding: applications to ecology and economic complexity. *Scientific Reports*, *11*(1), 1–14.

Van Leeuwen, M., Vreeken, J., & Siebes, A. (2006). Compression picks item sets that matter. In *Proceedings of the 10th European Conference on Principles and Practice of Knowledge Discovery in Databases* (pp. 585–592).

Vargas Quiros, J. (2017). *Information-theoretic anomaly detection and authorship attribution in literature* (Unpublished master's thesis). Utrecht University.

Vonk, A. N., Bos, M., Smeets, I., & Van Sebille, E. (2024). A comparative study of frames and narratives identified within scientific press releases on ocean climate change and ocean plastic. *Journal of Science Communication*, *23*(1), A01.

Winkel, T. d. (2015). *Of Deutsches blood* (Unpublished master's thesis). Utrecht University.

Xin, X., Yuan, F., He, X., & Jose, J. M. (2018). Batch IS NOT heavy: Learning word representations from all samples. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 1853–1862).

Yim, W.-w., Yetisgen, M., Harris, W. P., & Kwan, S. W. (2016). Natural language processing in oncology: a review. *JAMA oncology*, *2*(6), 797–804.

Yin, Z., & Shen, Y. (2018). On the dimensionality of word embedding. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems* (pp. 895–906).

Zhang, M., Palade, V., Wang, Y., & Ji, Z. (2022). Word representation using refined contexts. *Applied Intelligence*, *52*, 12347–12368.

Zhang, W., Yoshida, T., & Tang, X. (2011). A comparative study of TF*IDF, LSI and multi-words for text classification. *Expert Systems with Applications*, *38*(3), 2758–2765.

English Summary

In text mining and natural language processing (NLP) applications, a vector representation of text data is the key in designing an effective machine learning algorithm. Document-term and word-context matrices are two important matrices to represent texts as vectors. These two matrices are usually sparse and high-dimensional. The process of creating low-dimensional representations of texts is referred to as dimensionality reduction. Dimensionality reduction is associated with the representation of text data and thus very important. In the machine learning literature, little to no attention has been paid to a popular statistical technique, correspondence analysis (CA). Other popular dimensionality reduction methods receive more attention, like latent semantic analysis (LSA). This project is to study whether CA is a good dimensionality reduction technique in text mining and NLP.

Chapter 2 theoretically compares CA and LSA of a document-term matrix. In addition, the performance of CA is compared to the performance of different versions of LSA in the context of text categorization and authorship attribution. The criterion used to make comparisons is mainly a measure for accuracy. From a theoretical point of view it appears that CA has more attractive properties than LSA. For example, in LSA, the effect of the margins as well as the dependence between documents and terms is part of the matrix that is analyzed, while CA eliminates the effect of the margins and thus the solution only displays the dependence. The results for four empirical datasets show that CA can obtain higher accuracies on text categorization and authorship attribution than the different versions of LSA.

Chapter 3 also studies the performance of CA and LSA in the context of document-term matrices. CA and LSA are empirically compared in information retrieval by calculating the mean average precision. An attempt is made to improve CA by applying the two kinds of weighting, that are also used in LSA. These are weighting schemes for the elements of the document-term matrix and the adjustment of the singular value weighting exponent. The results for four empirical datasets show that CA always performs better than LSA. Weighting the elements of the raw data matrix can improve CA; however, it is data dependent and the improvement is small. Adjusting the singular value weighting exponent often improves the performance of CA; however, the extent of the improvement depends on the dataset and the number of dimensions.

Chapter 4 compares CA with PPMI-SVD, GloVe, and SGNS. Theoretically, like PPMI-SVD, GloVe, and SGNS, we are able to link CA to the factorization of the PMI matrix. An attempt is made to improve CA by making use of weighting schemes for the elements of the word-context matrix. An empirical comparison on word similarity tasks shows that the overall results for CA with the two weighting schemes are slightly

better than those of PPMI-SVD, GloVe, and SGNS.

CA is susceptible to outliers. In Chapter 5, the so-called reconstitution algorithm is introduced to cope with outlying cells. This algorithm can reduce the contribution of the outlying cells in CA. The reconstitution algorithm is compared with two alternative methods for handling outliers, the supplementary points method and MacroPCA. It is shown that the proposed strategy works well.

Summarizing, we have shown that CA is a technique that matches or outperforms techniques that are now commonly used in computing science. We think that the performance of CA in the studies of this dissertation shows that CA deserves more attention in this field.

Nederlandse Samenvatting

Bij tekst mining en natuurlijke taalverwerking (NLP) is een vectorrepresentatie van tekstgegevens de sleutel tot het ontwerpen van een effectief algoritme voor machinaal leren. Document-term en woord-tekst matrices zijn twee belangrijke matrices om teksten als vectoren weer te geven. Deze twee matrices zijn meestal spaarzaam gevuld en hoogdimensionaal. Het proces om laagdimensionale representaties van teksten te maken, wordt dimensionaliteitsreductie genoemd. Dimensionaliteitsreductie wordt geassocieerd met de representatie van tekstgegevens en is dus erg belangrijk. In de literatuur over machinaal leren is weinig tot geen aandacht besteed aan een populaire statistische techniek, correspondentieanalyse (CA). Andere populaire dimensionaliteitsreductiemethoden krijgen meer aandacht, zoals latente semantische analyse (LSA). In dit project wordt onderzocht of CA een goede dimensionaliteitsreductietechniek is in tekst mining en NLP.

In hoofdstuk 2 worden CA en LSA van een document-term-matrix theoretisch vergeleken. Daarnaast worden de prestaties van CA vergeleken met de prestaties van verschillende versies van LSA in de context van tekstcategorisatie en auteurschapstoewijzing. Het criterium dat gebruikt wordt om vergelijkingen te maken is voornamelijk een maat voor nauwkeurigheid, de zgn. *accuracy*. Vanuit theoretisch oogpunt blijkt dat CA aantrekkelijkere eigenschappen heeft dan LSA. In LSA maakt bijvoorbeeld zowel het effect van de marges als de afhankelijkheid tussen documenten en termen deel uit van de matrix die wordt geanalyseerd, terwijl CA het effect van de marges elimineert en de oplossing dus alleen de afhankelijkheid weergeeft. De resultaten voor vier empirische datasets laten zien dat CA hogere *accuracy* kan bereiken voor tekstcategorisatie en auteurschapstoewijzing dan de verschillende versies van LSA.

Hoofdstuk 3 bestudeert ook de prestaties van CA en LSA in de context van document-term-matrices. CA en LSA worden empirisch vergeleken in *information retrieval* door de gemiddelde precisie te berekenen. Er wordt geprobeerd om CA te verbeteren door twee soorten wegingen toe te passen die ook in LSA worden gebruikt. Dit zijn wegingsschema's voor de elementen van de document-term matrix en de aanpassing van de exponent voor weging van de singuliere waarde. De resultaten voor vier empirische datasets laten zien dat CA altijd beter presteert dan LSA. Het wegen van de elementen van de ruwe gegevensmatrix kan CA verbeteren; de mate van verbetering is echter afhankelijk van de specifieke dataset en de verbetering is klein. Het aanpassen van de exponent voor het wegen van de singuliere waarde verbetert vaak de prestaties van CA; de mate van verbetering hangt echter af van de dataset en het aantal dimensies.

Hoofdstuk 4 vergelijkt CA met PPMI-SVD, GloVe en SGNS. Theoretisch zijn we, net als PPMI-SVD, GloVe en SGNS, in staat om CA te koppelen aan de factorisatie van de PMI-matrix. Er wordt geprobeerd om CA te verbeteren door gebruik te maken van wegingsschema's voor de elementen van de woord-tekstmatrix. Een empirische vergelijking op woordovereenkomsttaken laat zien dat de algemene resultaten voor CA met de twee wegingsschema's iets beter zijn dan die van PPMI-SVD, GloVe en SGNS.

CA is gevoelig voor uitbijters. In hoofdstuk 5 wordt het zogenaamde reconstitutiealgoritme geïntroduceerd om met uitbijtende waarden in cellen om te gaan. Dit algoritme kan de bijdrage van de uitbijtende cellen in CA verminderen. Het reconstitutiealgoritme wordt vergeleken met twee alternatieve methoden voor het omgaan met uitschieters, de *supplementary points*-methode en MacroPCA. Er wordt aangetoond dat de voorgestelde strategie goed werkt.

Samenvattend hebben we laten zien dat CA een techniek is die overeenkomt met of beter presteert dan technieken die nu veel gebruikt worden in de computerwetenschap. We denken dat de prestaties van CA in de studies van dit proefschrift laten zien dat CA meer aandacht verdient op dit gebied.

About the Author

Qianqian Qi was born on March 1st, 1993 in Shandong, China. In 2016, she received her BSc in Mathematics and Applied Mathematics from Northwest Normal University, China. In 2019, she received her MSc in Applied Mathematics from Northwestern Polytechnical University, China.

In November 2019, she started as a PhD candidate at the Department of Methodology and Statistics, Utrecht University, the Netherlands. During her PhD, she focused on some studies in correspondence analysis of texts. She published papers in Natural Language Engineering and in Journal of Intelligent Information Systems.

Publications

Qi, Q., Hessen, D. J., Deoskar, T., & Van der Heijden, P. G. M. (2023). A comparison of latent semantic analysis and correspondence analysis of document-term matrices. Natural Language Engineering, 1–31.

Qi, Q., Hessen, D. J., & Van der Heijden, P. G. M. (2024). Improving information retrieval through correspondence analysis instead of latent semantic analysis. Journal of Intelligent Information Systems, 62, 209–230.

Qi, Q., Hessen, D. J. & Van der Heijden, P. G. M. (2024). A comparison of correspondence analysis with PMI-based word embedding methods. arXiv preprint arXiv: 2405.20895.

Qi, Q., Hessen, D.J., Vonk, A. N. & Van der Heijden, P. G. M. (2024). Correspondence analysis: handling cell-wise outliers via the reconstitution algorithm. arXiv preprint arXiv: 2404.17380.

Acknowledgement

This dissertation would not have been possible without the support of many people. I want to take this opportunity to thank them.

I want to thank my supervisor, Prof. dr. P.G.M. van der Heijden, for his patience and guidance for the projects. You always carefully read the document I sent you and provided comments on it. You always have such passion and unique perspective on projects, which inspires me a lot. You introduced me to the world of correspondence analysis and text mining. Without you, I cannot even think about starting and finishing the PhD journey. This dissertation has condensed your hard work from selecting topics to writing to finalization.

I want to thank my co-supervisor Dr. D.J. Hessen. I am grateful for all your patience, constant support, excellent advice, and encouragement throughout my whole PhD journey. Your comments on the projects play an essential role in finishing these projects. Your profound knowledge and fascinating insight have deeply infected and inspired me. This dissertation has condensed your hard work in every stage. Without you, this dissertation cannot be finished.

I want to thank my supervisor, Prof. dr. D.L. Oberski, for his patience and guidance on my projects and on my classes. I appreciate your valuable and insightful feedback very much.

I want to thank the members of the reading committee and the defence committee for their time and effort spent on evaluating this dissertation.

I want to thank my co-authors for my projects, Tejaswini and Aike. Your valuable insights give the projects a ladder. Collaborating with you has made me a better researcher.

I want to thank Ayoub, Qixiang, Anastasia, Pablo, and Erik-Jan for their help when I have questions. I want to thank all the other members of the NLTP lab: Huyen, Paul, Jelle, Niek, Hadi, Daniel, Arjan, Özgür, Nikolaj, Tina, and Aron. I enjoy the weekly meeting with you very much. I want to thank Ellen, Herbert, Gerko, Mirjam, Javier, Mahdi, Beth, Camilla, and all the teachers who taught and helped me. I want to thank every member of the Department of Methodology and Statistics. I know that whenever I have some questions, I can always find someone to ask.

I want to thank my office-mates Pia, Jeroen, Dan, Danielle, Ria, Mingyang, and others for their guidance in the research and for their companionship in life. I had such a good time with you at the office. Thanks to my friends Shanshan, Dandan, Shiya, Chuenjai, Daniëlle, Melanie, and others for their companionship.

I want to thank Qianrao for introducing Utrecht University to me and for the help with the application for scholarship. You helped me a lot both in life and in research.

Particular acknowledgment also should go to the China Scholarship Council (CSC) for financial support.

Finally, I want to thank my family for their understanding and support, which is the biggest motivation for my continuous progress and future work and life.