UNIVERSITY OF NOVI SAD
FACULTY OF SCIENCE
DEPARTMENT OF
MATHEMATICS AND INFORMATICS

Miloš Radovanović

High-Dimensional Data Representations and Metrics for Machine Learning and Data Mining

– Doctoral Dissertation –

Novi Sad, 2010

Contents

List of Figures

List of Tables

Acknowledgments

I would like to thank the members of the dissertation defense board, Dr. Zoran Budimac (University of Novi Sad, Serbia), Dr. Branimir Todorović (University of Niš, Serbia), and Dr. Alexandros Nanopoulos (University of Hildesheim, Germany) for helpful comments on early versions of this dissertation. My special thanks goes to Alexandros Nanopoulos, who helped me become a better researcher, and especially a better research collaborator. I am indebted to Dr. Zagorka Lozanov-Crvenković (University of Novi Sad) for helpful discussions and suggestions regarding the mathematical aspects of the dissertation. My colleagues and Computer Science students from the Department of Mathematics and Informatics at the University of Novi Sad are responsible for providing a pleasant working environment, for which I am grateful. My gratitude also goes out to Dr. Dunja Mladenić and Marko Grobelnik (Jožef Stefan Institute, Ljubljana, Slovenia), for outstanding hospitality during our past and ongoing research cooperation. Last but not least, I thank my advisor, Dr. Mirjana Ivanović (University of Novi Sad), for the good will, patience, encouragement, and iterative nudges over the years towards the fields of machine learning and data mining, starting from the work on my master's thesis and continuing to this day.

I also thank the authors of all software packages, libraries, and resources used in this research: Matlab with the Statistics Toolbox, Dimensionality Reduction Toolbox [210], the FastICA package [70], GML AdaBoost Toolbox [217], code for semi-supervised learning with the basic harmonic function [238], SpectraLIB package for symmetric spectral clustering [216, 135], MatlabBGL graph library [72]; the Weka machine-learning toolkit [224], and Wolfram Mathematica with invaluable online resources MathWorld (`mathworld.wolfram.com/`) and the Wolfram Functions Site (`functions.wolfram.com/`).

The presented research was supported by the Serbian Ministry of Science and Technological Development, the Provincial Secretariat for Science and Technological Development, and the Provincial Department of Education and Culture of Vojvodina.

Novi Sad,
July 2010

Miloš Radovanović

Part I

Preliminaries

Chapter 1

Introduction

The information age we are living in brings numerous benefits to many aspects of human endeavor. Automation and computerization of tasks which were performed manually in the past is only one example of how the use of computers is changing peoples' lives, both on professional and personal levels. With the benefits, however, come numerous challenges. This dissertation studies problems which stem from the increasing volume of information being generated, stored, and used on today's computer systems. The rate at which information is being generated typically outpaces the rate at which it can be processed, structured, and effectively used as *knowledge*, giving rise to the term *information overload*, whose impact can be observed, for example, on the World Wide Web [112, 11, 161]. Besides large volumes and weak structure, information gathered as data often contains noise, in the sense of being erroneous, irrelevant, or simply superfluous with respect to a particular task.

The above-mentioned properties of data – large volumes, weak or inappropriate structure, and noisiness – make it amenable to application of *machine learning* (ML), *data mining* (DM), and *information retrieval* (IR) techniques. The fields of machine learning and data mining provide many useful methods for discovering patterns and inferring knowledge from raw or shallowly processed data, such as (hyper)text commonly found on the Web. Machine learning and data mining possess the means to perform such tasks *automatically* (albeit after careful human analysis of the concrete problem and preparation of data). Information retrieval, on the other hand, provides the user with techniques for locating larger units of information (for example, documents) that may satisfy a user's information need expressed by a query.

The basic representation in which information is gathered and stored is a data table (also referred to as a data set), where rows correspond to objects (or instances, examples, points) that are described by one or more features (or attributes, variables) that form the columns of the table. Increasing volumes of available information can be manifested in one or both of the following properties: (1) having a large number of data instances in a table, and (2) having a large number of features. This dissertation

focuses on the second property, typically called *high dimensionality*, which is known to be able to cause problems in tasks related to many fields, including machine learning, data mining, and information retrieval. These problems are commonly referred to as *the curse of dimensionality*.

The curse of dimensionality, a term originally introduced by Bellman [10], can be manifested in many different forms and contexts. Within the fields of machine learning and data mining, the curse can affect Bayesian modeling [18] by making the estimation of mutual dependencies between features (as random variables) infeasible. Nearest-neighbor prediction is also affected [82], for example, by the exponential raise of the number of required samples of data points to achieve a required sampling density. The search for nearest neighbors may suffer from high dimensionality [111], since indexing methods tend to lose their effectiveness in high dimensions [105]. High dimensionality is also known to reduce the performance of learning methods such as neural networks [17].

An aspect of the dimensionality curse that has attracted recent attention refers to the behavior of distance measures, which are used to express the proximity of data points. The behavior in question is known as the phenomenon of *distance concentration*, which is manifested by the tendency of all pairwise distances between points in high-dimensional data to become approximately equal. Distance concentration and the meaningfulness of nearest-neighbor relations in high dimensions has been studied for distance measures in general [16, 51], and specifically for Minkowski and fractional distances [44, 85, 2, 61, 59, 87].

A common approach to tackling the curse of dimensionality is by applying some form of *dimensionality reduction* to data, transforming it into a lower-dimensional representation that aims to preserve important information contained in the original. There exist numerous methods to achieve this goal, a selection of which is reviewed in Section 2.8. One approach is through *feature selection*, which encompasses techniques for choosing a subset of features appropriate for a given task, while *feature extraction* (often referred to by the term dimensionality reduction in its own right) includes methods which combine the values contained in different features, forming a completely new feature space.

The principal reason why dimensionality-reduction methods are expected to work in practice is the observation that for the majority of available data, its *intrinsic dimensionality* is lower that the embedding dimensionality, that is, the total number of features. The intrinsic dimensionality can be loosely defined as the minimal number of features needed to express all information contained in the data [59]. Depending on the exact notion of "information" that is considered relevant, different precise definitions of intrinsic dimensionality can be formulated.[1] A concept related to intrinsic dimensionality is the *manifold assumption* [33], which states that data instances usually lie on a low-dimensional manifold (or, loosely speaking, subspace) of the original data space.

This dissertation will study the implications of the curse of dimensionality in high-dimensional data representations from two angles:

[1] In Chapter 5 we will discuss the notion of intrinsic dimensionality relevant to this dissertation.

1. the behavior of distance (and similarity) measures with increasing dimensionality of data, and

2. feature-selection methods, primarily through their interaction with high-dimensional document representation schemes for text data.

These two angles are manifested in the structure of this dissertation, outlined in the following section.

1.1 Dissertation Outline

This dissertation is organized into three major parts. Part I: Preliminaries, which includes the first two chapters of the dissertation, introduces the motivation and problems studied in the subsequently presented research, and provides an overview of techniques for machine learning, data mining, and information retrieval to assist the reader in understanding the material which follows. The overview, presented in Chapter 2, includes descriptions of data representations, distance and similarity metrics, classification, clustering, semi-supervised learning, outlier detection, and dimensionality reduction.

Part II: Metrics, which encompasses Chapters 3–7, presents the angle of research focused on the behavior of distance and similarity measures in data spaces of increasing dimensionality. Chapter 3, the results of which are published in [141, 171], studies the concentration phenomenon with respect to the cosine similarity measure. The remaining chapters of Part II explore a novel phenomenon of *hubness*, which refers to the tendency of neighborhood graphs of high-dimensional data points to contain nodes (called hubs) that are more frequently included in nearest-neighbor lists of other points. Chapter 4 studies the phenomenon in isolation on synthetic data distributions, from a theoretical and empirical perspective, with the results included in [169, 168]. Chapter 5 places the hubness phenomenon in the context of real data from various application domains, generalizing the conclusions from Chapter 4, and exploring the effects on machine learning and data-mining techniques for classification, semi-supervised learning, clustering, and outlier detection, with the results published in [169, 168]. Chapter 6, the results of which are presented in [172], studies the hubness phenomenon within the time-series domain, and discusses its implications on time-series classification. Finally, Chapter 7 examines hubness within the setting of information retrieval, where hubs represent documents which persistently appear in search result lists of many different queries, with the results published in [171].

Part III: Document Representation and Feature Selection, consisting of Chapters 8 and 9, turns its attention to the second angle of research into the implications of the curse of dimensionality: feature-selection methods, and their interaction with high-dimensional representation schemes for text data. Chapter 8, whose results are presented in [164, 166], describes an experimental study on the impact of high-dimensional document representations on the performance of five major classifiers. Different transformations of input data: stemming, normalization, logtf, and idf, together with dimen-

sionality reduction, are found to have a statistically significant improving or degrading effect on classification performance. Besides determining the best document representation which corresponds to each classifier, the study describes the effects of every individual transformation on classification, together with their mutual relationships. Chapter 9, the results of which are published in [165, 167], examines the relationship between text data transformations and several widely used feature-selection methods, in the context of classification, showing that the idf transformation considerably effects the distribution of classification performance over feature-selection reduction rates, and offering an evaluation method which permits the discovery of relationships between different document representations and feature-selection methods which is independent of absolute differences in classification performance. The chapter also briefly discusses hubness in the context of feature selection and generation [170].

Finally, Chapter 10 concludes the dissertation, summarizing the main results, and discussing the possibilities for future work.

1.2 Contributions of the Dissertation

The scientific contributions of this dissertation can be viewed separately within the research angles presented in Parts II and III.

In Part II, original contributions begin with the theoretical results concerning the concentration behavior of the cosine similarity measure (Chapter 3). In the remaining chapters of Part II, the attention of the ML, DM, and IR communities is drawn to the phenomenon of hubness, which is a fundamental property of data distributions in high-dimensional spaces that has received surprisingly little interest to date. Hubness and its origins are explained from a theoretical and empirical perspective in artificial data distributions (Chapter 4), and put into the context of real data from various domains (Chapter 5), describing the connection between hubness, intrinsic dimensionality, and the cluster structure of the data. Chapter 5 also studies the interaction of hubness with the information provided by class labels, and shows that the phenomenon is relevant to various distance-based classification schemes. Furthermore, the same chapter provides evidence which describes the effect of hubness on graph-based methods for semi-supervised learning, as well as distance-based clustering and outlier-detection methods. The importance of the phenomenon is demonstrated on time-series data as well (Chapter 6) in the setting for time-series classification currently considered as the state-of-the-art, providing a framework for categorizing time-series data sets based on hubness properties and the distribution of class labels which facilitates time-series classification. Finally, the importance of the phenomenon is demonstrated in the classical IR setting involving vector space models, explaining its origins with respect to the behavior of cosine similarity in high dimensions, and generalizing the conclusions to more advanced document representation and matching schemes.

In Part III, the principal contributions are contained in the experimental results and methodologies for classification of high-dimensional text data, showing the impacts

and relationships between various transformations of the bag-of-words document representation in the context of classifiers commonly used on text data (Chapter 8), and quantifying the interaction between transformations of the bag-of-words document representation and feature selection (Chapter 9).

Chapter 2

Machine Learning, Data Mining, and Information Retrieval

This chapter will review some of the tasks and data representations relevant to the fields of machine learning, data mining, and information retrieval. The aim of the overview is not to provide comprehensive coverage of the fields (which would be infeasible due to the sheer amount of relevant material), but rather to aid the reader in understanding the subsequent chapters by describing the main ideas and principles behind various techniques, and offering references to work that provides additional details.

The field of machine learning (ML) is concerned with the question of how to construct computer programs that automatically improve with experience [136]. On the other hand, data mining (DM), also referred to as knowledge discovery from data (KDD), deals with concepts and techniques for uncovering interesting data patterns hidden in large data sets [81]. Despite the apparent difference in the central motivation, the two fields share many tasks, techniques, and data representations. Another related field is information retrieval (IR) which is concerned with finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within large collections (usually stored on computers) [38]. Many text representations and distance measures commonly employed for ML and DM tasks were initially developed and explored within the IR domain. Generally, high-dimensional data domains will be given somewhat greater emphasis in this survey due to their prominence in the research presented in subsequent chapters of this dissertation, with particular attention being given to text and time-series.

Regarding machine-learning techniques, this chapter will take a broader view of the learning process, and consider supervised, unsupervised, and semi-supervised approaches. In supervised learning, computer programs capture structural information and derive conclusions from *examples* (also called *instances*; in the textual domain documents, or parts of text), previously annotated by labels denoting *classes*. This enables supervised learning algorithms to process new examples and apply the con-

clusions to them. Unsupervised learning deals more with the *analysis* of data, in the sense of capturing relationships between examples without relying on outside information. Semi-supervised learning in concerned with ways to combine the two paradigms, predominantly by using unlabeled data to assist supervised learning.

ML tasks can roughly be divided into four distinct areas: classification, clustering, association learning, and numeric prediction [224]. In the machine-learning approach, classification algorithms (classifiers) are trained beforehand on previously sorted (labeled) data, before being applied to sorting unseen examples. Classification applied to text is the subject of *text categorization* (TC – also known as *text classification* or *topic spotting*), which is the task of automatically sorting a set of documents into *categories* (or *classes*, or *topics*) from a predefined set [191]. Straightforward classification of documents is employed in document indexing for information-retrieval systems, text filtering (including protection from e-mail spam), categorization of Web pages, routing news articles, and many other applications. Classification can also be used on smaller parts of text (paragraphs, sentences, words) depending on the concrete application, like document segmentation, topic tracking, or word sense disambiguation. Classification is also a useful task in the time-series domain, applied to labeling trajectories of vehicles monitored by video surveillance systems, indexing ECG diagrams for medical diagnosis, segmenting signals from robotic sensors, etc.

Clustering is a basic unsupervised-learning task concerned with finding groups of examples based on some inherent notion of similarity between them. The use of clustering techniques with text can be achieved on two levels. Analyzing collections of documents by identifying clusters of similar ones can be achieved by straightforward application of known clustering algorithms coupled with document similarity measures (see Section 2.2). Within-document clustering can be somewhat more challenging since it requires preprocessing text and isolating objects to cluster – sentences, words, or some construct which requires derivation. In time-series analysis, clustering can be applied to group similar stocks, analyze weather imagery, recognize motion, etc.

Association learning can be viewed as a generalization of classification which aims to capture relationships between arbitrary *features* (also called *attributes*) of examples in a data set. In this sense, classification captures only the relationships of all features to the one feature specifying the class. Straightforward application of association learning to text is not feasible because of the high dimensionality of document representations, that is, the large number of features. Applying association learning to information extracted from text (for example, using classification, clustering, or dimensionality reduction) is more feasible, and can yield useful insights, as can association learning from information extracted from time-series data.

Numeric prediction (also called *regression*, in a wider sense of the word), may be viewed as another generalization of classification, where the class feature is not discrete, but continuous. This small shift in definition results in large differences in the internal workings of classification and regression algorithms. However, by dividing the predicted numeric feature into a finite number of intervals, regression algorithms can generally also be used for classification, while the opposite is not generally possible.

Another important task associated with data mining is *outlier detection*, which is concerned with locating examples which are in some sense different from the majority of others. Applications of outlier detection include detection of credit card fraud, intrusion into computer systems, medical diagnosis, and many others.

It is widely acknowledged that a large number of features, that is, high dimensionality of a data set, can cause severe problems in many machine learning, data mining, and information-retrieval applications. These problems are commonly referred to as the curse of dimensionality (see Chapter 1). A solution that is often employed is to reduce the dimensionality of the data space by selecting only a subset of the feature set, or applying some transformation on data points, projecting them to a feature space of lower dimensionality. Techniques for achieving this task are collectively known as *dimensionality-reduction* methods.

The rest of this chapter will review in more detail the data representations, techniques, and tasks from the fields of machine learning, data mining, and information retrieval, relevant to the results of the research presented in this dissertation. Section 2.1 introduces the most common tabular approach to data representation, and details various methods for representation of textual and time-series data. Section 2.2 describes many of the widely-used distance and similarity measures for data with numeric features. The following three sections are devoted to the basic machine-learning tasks, starting with Section 2.3 which discusses classification, including descriptions of several illustrative algorithms, possible difficulties, and means of evaluation. Approaches to semi-supervised learning are reviewed in Section 2.4, and clustering techniques are discussed in Section 2.5. Section 2.6 is devoted to outlier detection, while Section 2.7 summarizes the main principles and techniques of textual information retrieval. Finally, Section 2.8 presents a taxonomy of dimensionality-reduction techniques, and Section 2.9 concludes the survey.

2.1 Data Representation

General-purpose techniques, like classification and clustering, are usually designed for examples which have a fixed set of nominal (symbolic), discrete (ordinal), or numeric (integer or continuous) features. A data set is then represented by a table where columns correspond to features, and rows are individual examples. One such data set is shown in Table 2.1 (from [224]). The data set consists of 14 examples characterized by 5 features, three of which are nominal (outlook, windy and play), and the other two discrete. Play can be considered the *class attribute*, since it indicates whether one should engage in his/her favorite sport depending on the weather conditions expressed by other attributes.

2.1.1 Text Data

When examining the described tabular form of representing data, it is evident that free-flowing or semi-structured text (for example, HTML) needs to be transformed in order

outlook	temperature	humidity	windy	play
sunny	85	85	FALSE	no
sunny	80	90	TRUE	no
overcast	83	86	FALSE	yes
rainy	70	96	FALSE	yes
rainy	68	80	FALSE	yes
rainy	65	70	TRUE	no
overcast	64	65	TRUE	yes
sunny	72	95	FALSE	no
sunny	69	70	FALSE	yes
rainy	75	80	FALSE	yes
sunny	75	70	TRUE	yes
overcast	72	90	TRUE	yes
overcast	81	75	FALSE	yes
rainy	71	91	TRUE	no

Table 2.1: The weather data set
Tabela 2.1: Skup podataka o vremenskim uslovima

to apply methods for machine learning, data mining, or information retrieval. The most widely used approach is the bag-of-words representation.

The Bag-of-Words Representation

In the bag-of-words (BOW) representation, word order is discarded from a document and single words are treated as features. Actually, other entities can be used as features (for example, phrases), hence textual features are referred to as *terms* instead of *words*. Let W be the *dictionary* – the set of all words (terms) that occur at least once in the set of documents D. The BOW representation of document d_i is a vector of weights $\mathbf{w}_i = (w_{i1}, \ldots, w_{i|W|})$. There are many variations of the BOW representation, depending on the weight values. For the simplest *binary representation*, $w_{ij} \in \{0, 1\}$; the weight $w_{ij} = 1$ if the jth word is present in document d_i, otherwise $w_{ij} = 0$. In the *term-frequency representation* (denoted tf), $w_{ij} = tf_{ij}$, the frequency of appearance of the jth term in the ith document. Figure 2.1 (from [137]) shows a short document together with its tf representation.

Many transformations of term frequencies are used in practice. *Normalization* (denoted norm) can be employed to scale the term frequencies, accounting for differences in the lengths of documents. The logtf transformation may also be applied to term frequencies, resulting in the representation:

$$\mathrm{logtf}(d_i) = (\log(1 + w_{i1}), \ldots, \log(1 + w_{i|W|})).$$

The *inverse document frequency* (idf) transformation yields the representation:

$$\mathrm{idf}(d_i) = (w_{i1} \log(|D|/\mathrm{docfreq}(D, 1)), \ldots, w_{i|W|} \log(|D|/\mathrm{docfreq}(D, |W|))),$$

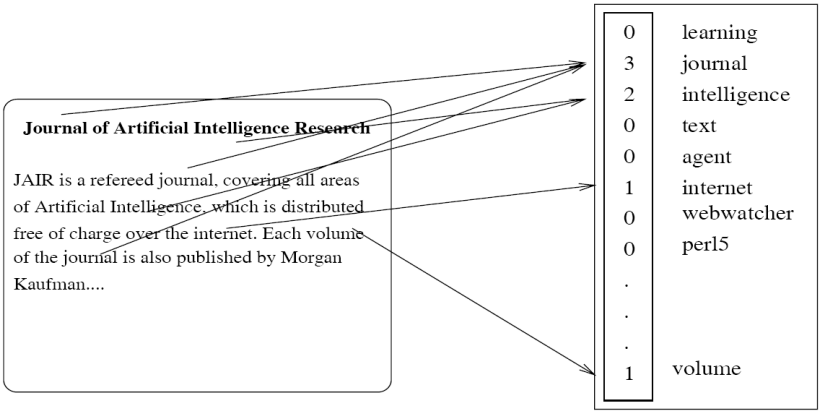


Figure 2.1: The bag-of-words representation of a document, with term frequencies
Slika 2.1: *Bag-of-words* reprezentacija dokumenta, sa frekvencijama termova

where $\mathrm{docfreq}(D, j)$ is the number of documents from D the ith term occurs in. It can be used by itself (with binary weights w_{ij}), or with term frequencies to form the popular tfidf representation. We refer to Appendix A for a comprehensive illustrative example of term-frequency transformations.

There are many rationales behind different transformations of the bag-of-words representation. Term frequencies supposedly stress the importance of the more frequent terms for determining relationships between documents. Normalization stops the term frequencies in longer documents from overriding the frequencies in shorter texts. The logtf transformation scales down all frequencies, making differences less influential in high ranges, but without bounding values from above. The issue of a term occurring in many documents is addressed by the idf transformation – the more documents a term appears in, the less it is considered important, and the weight is scaled down. It can be said that the tfidf representation attempts to strike a balance between intra- and inter-document frequencies of terms.

N-grams

Two very different notions have been referred to as "n-grams" in the literature. The first are *phrases*, as sequences of n words; this meaning was adopted by the Statistical natural language processing community [132]. The other notion are n-grams as sequences of *characters*.

N-grams as phrases can be viewed as a generalization of words, since 1-grams *are* words, and therefore 2-grams up to 5-grams are usually used to *enrich* the BOW representation, rather than on their own. The main problem is sheer magnitude – the number of n-grams grows exponentially with n – therefore many strategies for efficient

generation of a useful set of n-grams have been developed. One such algorithm, presented by Mladenić [137], iterates over n, generating all possible n-grams from known $(n-1)$-grams, immediately discarding all n-grams which appear too infrequently in the document set.

N-grams as sequences of characters, at first glance, are not very intuitive. For example, the string "not very intuitive" could be represented by the following 3-grams: `not ot_ t_v _ve ver ery ry_ y_i _in int ntu tui uit iti tiv ive.`[1] In this case, n-grams are commonly used *instead* of words in the BOW representation, so now not only is the word order lost, but words themselves are not preserved. Nevertheless, character n-grams proved useful in situations with grammatical and typographical errors in documents, and are also an effective way to achieve language independence [25, 130]. Besides the classical text-categorization task of sorting documents into different topics or routing news articles, character n-grams are an effective document representation for the task of language identification. This is because different languages tend to exhibit different distributions of n-grams in documents, that is, some n-grams are expected to appear more frequently in documents written in language A than in language B (for example, `th` is much more frequent in English than in Serbian).

Hypertext Features

Hypertext documents in HTML or XML format offer many other features to be exploited by machine learning. *Hyperlink* information is arguably an obvious choice, and there are many ways it can be employed: adding to a document all the words from documents *it links to*, or representing a document only with names (or identifiers) of documents it links to [228]; and, vice versa, using features from the contexts of links *referring to the document* [8]. For classification, class labels of neighboring documents can be added to the feature space of a document [29]. All these techniques were employed with variable degrees of reported success.

The tree structure of HTML/XML is another possible source of features. Terms can be labeled with their paths in the tag hierarchy containing them, which was shown to be an effective method. Even more successful was prefix labeling – a new feature is constructed by labeling a term with every possible prefix of its path in the tag tree [28]. For example, consider the following XML marked-up text (from [28]):

```
<resume>
  <publication>
    <title>Statistical Models for Web-surfing</title>
  </publication>
  <hobbies>
    <item>Wind-surfing</item>
  </hobbies>
</resume>
```

[1] The underscore represents space, and is treated as a character.

The term *surfing* may be labeled resume.publication.title.*surfing* and resume.hobbies.item.*surfing*, and, in addition, with all prefixes: resume.*surfing*, resume.publication.*surfing*, resume.hobbies.*surfing*.

Features can also be derived from the text found in TITLE and META tags of HTML pages [228]. This can be achieved, for example, by including words from these tags into the BOW representation of a document, possibly using a higher weight to signify the increased significance of such words with respect to the semantics of the document.

The ILP Approach

Relations provide a completely different means for representing documents, compared to the BOW model. They offer more expressive power, at the cost of a limited range of learning algorithms that can be applied. These algorithms fall into the scope of *inductive logic programming* (ILP), and deal with generating rules based on examples. Although it is possible to use these algorithms for association learning, classification is more commonly attempted on text. Typical representatives include Ross Quinlan's FOIL [160] and William Cohen's RIPPER [39].

Extracting relations from documents of the form contains(document, term) or contains(document, term, weight) is a straightforward way to represent the same information that is captured by the BOW model. Furthermore, word order can be expressed by a relation like after(term1, term2), or word context by near(term1, term2, k), where k denotes term distance in text [39].

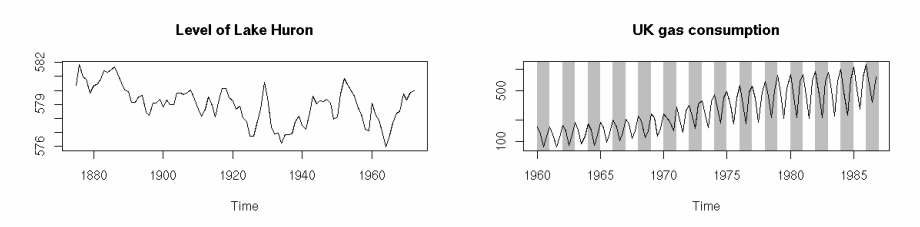
Relations are particularly suited for representing hypertext information. Consider several examples (adapted from [28]):

```
contains-text(treeNode, term).
part-of(treeNode1, treeNode2).
tagged(treeNode, tagName).
links-to(srcTreeNode, dstTreeNode).
contains-anchor-text(srcTreeNode, dstTreeNode, term).
classified(treeNode, label).
```

Based on information represented in this form, a rule learner could generate output of the following form:

```
classified(A, facultyPage) :-
    contains-text(A, professor), contains-text(A, phd),
    links-to(B, A), contains-text(B, faculty).
```

Even an enriched BOW representation like the one including hypertext features, described in the previous section, would never have made such expressiveness (and clarity) of learned classifiers possible. However, a crucial downside of ILP methods is the inherent slowness in the face of high-dimensional, large-volume textual data. Therefore, hybrid approaches may be considered, like a combination of a BOW representation for text and a relational representation for hyperlinks [27].



(a) (b)

Figure 2.2: Example time series
Slika 2.2: Primeri vremenskih serija

2.1.2 Time-Series Data

A time-series database consists of a series of values or events obtained over repeated measurements of time [81]. The values are usually measured at equal time intervals. Using the terminology from previous sections, every attribute corresponds to one *point* in time when measurement is taken, with the dimensionality of data referred to as time-series *length*.[2]

Time-series data is generated in many application areas, including the stock market, study of natural astronomical, geological or biological phenomena, medicine, etc. As examples, Figure 2.2(a) shows the water levels of Lake Huron measured from the late 1800s to the late 1900s, while Figure 2.2(b) plots the amount of gas consumed in the United Kingdom between 1960 and 1986 [239]. It is important to distinguish time series from *sequence* data, which refers to measurements taken without regard to time, with the only considered relationship between measurements being *before/after*. Examples of sequence data include Web-page access logs, customer shopping transactions, command sequences issued by a user to a remote computer system, etc.

A wide range of tasks were identified to be applicable to time-series data, including prediction, trend analysis, classification, clustering, outlier detection, indexing, etc. Many of these tasks require the computation of similarity between two time series. In Section 2.2 we give an overview of distance and similarity measures, many of which are directly applicable to time-series data. Before application of a similarity measure it may be necessary to normalize the time series in a particular data set, which is typically done for each time-series point by subtracting from its value the sample mean of the whole time series, and dividing by the sample standard deviation. More precisely, given time series $\mathbf{x} = (x_1, x_2, \ldots, x_d)$, its sample mean $\mu_{\mathbf{x}}$ and standard deviation $\sigma_{\mathbf{x}}$, the normalized time series is $\mathbf{x}' = (x'_1, x'_2, \ldots, x'_d)$, where $x'_i = (x_i - \mu_{\mathbf{x}})/\sigma_{\mathbf{x}}$, for all $i \in \{1, 2, \ldots, d\}$.

[2]Strictly speaking, it is not necessary for all time-series in a data set to be of the same length. In such cases some form of scaling like uniform scaling [65] can be performed to reduce or extend all time series to the same length.

An often used approach to preprocessing time series involves transforming the data set to a completely new representation where individual attributes do not necessarily correspond to measurements in time. The best-known methods include discrete Fourier transform (DFT) [56], discrete wavelet transform (DWT) [31], singular value decomposition (SVD) [56], piecewise aggregate approximation (PAA) [102], piecewise constant approximation (APCA) [26], and symbolic aggregate approximation (SAX) [127]. The first three of the aforementioned methods will be reviewed in Section 2.8, which discusses dimensionality reduction.

2.2 Distance and Similarity Measures

Computing (dis)similarity of data instances represents a key operation in many machine-learning, data-mining, and information-retrieval applications. A number of distance and similarity measures have been developed for this task; in this section we will review the measures that will be analyzed or mentioned later in this dissertation, applicable to numerical data.

Distance measures express the degree of dissimilarity between two data instances, with higher values signifying less similarity. With *similarity* measures, on the other hand, higher values indicate *more* similarity. Depending on the type of measure, the following equations can be used to perform conversion between distance and similarity. Given data points $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$,

$$dist(\mathbf{x}, \mathbf{y}) = 1 - sim(\mathbf{x}, \mathbf{y}) \,,$$

$$sim(\mathbf{x}, \mathbf{y}) = \frac{1}{1 + dist(\mathbf{x}, \mathbf{y})} \,.$$

Distance measures that are used in practice often, but not necessarily, satisfy all of the following properties for any given $\mathbf{x}, \mathbf{y}, \mathbf{z} \in \mathbb{R}^d$:

1. $dist(\mathbf{x}, \mathbf{y}) \geq 0$ (*non-negativity*),
2. $dist(\mathbf{x}, \mathbf{y}) = 0$ if and only if $\mathbf{x} = \mathbf{y}$ (*identity of indiscernibles*),
3. $dist(\mathbf{x}, \mathbf{y}) = dist(\mathbf{y}, \mathbf{x})$ (*symmetry*),
4. $dist(\mathbf{x}, \mathbf{z}) \leq dist(\mathbf{x}, \mathbf{y}) + dist(\mathbf{y}, \mathbf{z})$ (*triangle inequality*).

If a distance measure satisfies all of the above properties it is said to be *metric* in the strict mathematical sense. In this dissertation, unless explicitly stated, we will not distinguish between measures that are metric from those that are not, and will use the two terms interchangeably.

In the remainder of this section we will review the commonly used Minkowski, fractional, Bray-Curtis, normalized Euclidean, and Canberra distance measures, cosine and Jaccard similarity measures, and the dynamic time warping (DTW) distance measure

applicable to time-series data. For the interested reader, comprehensive overviews of measures that express distance or similarity are given by Chapman [34], Kohonen [108], Teknomo [206], and Gan et al. [69]; Egghe [53] analyzes the relationships between several well-known similarity measures, while Cohen et al. [40] provide a thorough experimental comparison of string metrics for name-matching tasks.

2.2.1 Minkowski Distances

For two vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$, and a given $p \in \{1, 2, \ldots\}$, the Minkowski (l_p) distance is defined as the p-norm of the difference between the two vectors:

$$l_p(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|_p = \left(\sum_{i=1}^{d} |x_i - y_i|^p \right)^{\frac{1}{p}}. \tag{2.1}$$

When p = 1, the metric is also referred to as Manhattan distance, while l_2 is the well-known and universally employed Euclidean distance.

2.2.2 Fractional Distances

By taking p from Equation 2.1 to be a positive *rational* number from the $(0, 1)$ range, *fractional* distance measures are obtained. A typically used value of p is $1/2$. Fractonal distances were advocated for handling high-dimensional data by Aggarwal et al. [2], and shown to be robust to certain types of noise [2, 60].

2.2.3 Bray-Curtis and Normalized Euclidean Distance

To achieve meaningful distance measurements in some applications (like numerical ecology [118]), it may be beneficial to normalize the distance values obtained by Minkowski metrics. Distance measures that perform normalization include Bray-Curtis distance (which normalizes Manhattan distance) and the normalized Euclidean metric. For $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$, when $\mathbf{x} \neq \mathbf{0}$ or $\mathbf{y} \neq \mathbf{0}$:

$$\text{bray-curtis}(\mathbf{x}, \mathbf{y}) = \frac{\|\mathbf{x} - \mathbf{y}\|_1}{\|\mathbf{x}\|_1 + \|\mathbf{y}\|_1} = \frac{\sum_{i=1}^{d} |x_i - y_i|}{\sum_{i=1}^{d} |x_i| + \sum_{i=1}^{d} |y_i|},$$

$$\text{norm-euclidean}(\mathbf{x}, \mathbf{y}) = \frac{\|\mathbf{x} - \mathbf{y}\|_2}{\|\mathbf{x}\|_2 + \|\mathbf{y}\|_2} = \frac{\sqrt{\sum_{i=1}^{d} (x_i - y_i)^2}}{\sqrt{\sum_{i=1}^{d} x_i^2} + \sqrt{\sum_{i=1}^{d} y_i^2}}.$$

In case $\mathbf{x} = \mathbf{y} = \mathbf{0}$, the values of both measures are 0. Properties of these measures are discussed in [118, 231].

2.2.4 Canberra Distance

Canberra distance focuses on the relative difference between coordinate values, as opposed to absolute differences expressed by Minkowski and fractional distances. For $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$, when $\mathbf{x} \neq \mathbf{0}$ or $\mathbf{y} \neq \mathbf{0}$, Canberra distance given by [206]:

$$\mathrm{canberra}(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{d} \frac{|x_i - y_i|}{|x_i| + |y_i|} .$$

In case $\mathbf{x} = \mathbf{y} = \mathbf{0}$, the value of Canberra distance is 0.

2.2.5 Cosine Similarity

For two vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$, the cosine similarity is defined as

$$\cos(\mathbf{x}, \mathbf{y}) = \frac{\langle \mathbf{x}, \mathbf{y} \rangle}{\|\mathbf{x}\| \cdot \|\mathbf{y}\|} = \frac{\sum\limits_{i=1}^{d} x_i y_i}{\sqrt{\sum\limits_{i=1}^{d} x_i^2} \cdot \sqrt{\sum\limits_{i=1}^{d} y_i^2}} .$$

The measure expresses the cosine of the angle between the two vectors in d-dimensional space. A smaller angle signifies greater similarity between two document vectors and, presumably, greater similarity between the semantics of their contents.

2.2.6 Jaccard Similarity

Jaccard similarity (also referred to as Tanimoto similarity) between vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$ may be defined as

$$\mathrm{jaccard}(\mathbf{x}, \mathbf{x}) = \frac{\langle \mathbf{x}, \mathbf{y} \rangle}{\|\mathbf{x}\|^2 + \|\mathbf{y}\|^2 - \langle \mathbf{x}, \mathbf{y} \rangle} = \frac{\sum\limits_{i=1}^{d} x_i y_i}{\sum\limits_{i=1}^{d} x_i^2 + \sum\limits_{i=1}^{d} y_i^2 - \sum\limits_{i=1}^{d} x_i y_i} ,$$

which is reminiscent of the cosine measure. However, its origins in the comparison of *sets* allow a different definition. Let N and M be sets containing designators of coordinates with nonzero values of vectors $\mathbf{x}$ and $\mathbf{y}$ (for example, if $\mathbf{x}$ and $\mathbf{y}$ correspond to document vectors in the BOW representation, N and M are the respective sets of terms). Jaccard similarity then represents the ratio between the number of nonzero attributes shared by $\mathbf{x}$ and $\mathbf{y}$, and the number of all nonzero attributes appearing in both vectors:[3]

$$\mathrm{jaccard}(N, M) = \frac{|N \cap M|}{|N \cup M|} = \frac{|N \cap M|}{|N| + |M| - |N \cap M|} .$$

[3]This definition is applicable to *multisets* as well.

Jaccard similarity is often used in situations where a common dictionary is not available, or not necessary for solving the problem at hand. In the BOW setting, this would mean that the dictionary is formed online only from the two documents being compared.

2.2.7 Dynamic Time Warping Distance

The dynamic time warping (DTW) distance can be viewed as a modification of Euclidean distance which can meaningfully be applied to speech signals [183], time series [14], and similar types of data. It differs from Euclidean distance by allowing the vector components that are compared to "drift" from exactly corresponding positions, in order to minimize the distance and compensate for possible "stretching" and "shrinking" of parts of the series along the temporal axis. This is achieved by forming a "warping" matrix representing every possible combination of components of the two compared series of length d.[4] The distance is determined by the "warping" path from component tuple $(1, 1)$ to (d, d) of the matrix, which minimizes the sum of squared differences between the components on the path, where the allowed steps are $(+1, 0)$, $(0, +1)$, and $(+1, +1)$. The warping path can be computed in quadratic time using dynamic programming.

In order to disable excessive "drift" of the warping path in any of the two directions in the matrix, a constraint parameter may be introduced which limits the warping path to a narrow "strip" between matrix entries $(1, 1)$ and (d, d). The constraint parameter, c, usually expresses the percentage of dimensionality d. The resulting distance measure is referred to as the constrained dynamic time warping (CDTW) distance. Regarding the extreme values of the constraint, for $c = 0\%$ CDTW becomes equivalent to Euclidean distance, while $c = 100\%$ yields the unconstrained DTW. It has been observed in the time-series domain that low values of c below 10% usually work well [174].

2.3 Classification

The process of automated construction of a classifier can be viewed as algorithmically building a mathematical model for separating examples of different classes, from the evidence given in a data set. The process of model building is referred to as classifier *training*, with the employed data set called the *training set*. The computed model can then be used to classify a previously unseen data instance (that is, an instance not included in the training set).

There exist many different approaches to building a classifier model, which resulted in the development of many kinds of classifiers with different properties. Some classification algorithms can only discern between two different classes, making them *two-class* (or *binary*) classifiers, others are naturally *multi-class*. However, this restriction may not be a great mishap, since there exist ways to use binary classifiers for classification into more than two classes (see Section 2.3.2).

[4] The compared series do not necessarily have to be of the same length.

Binary classification can also be viewed as a *one-class* problem, where instances can be *positive* (belonging to the class) or *negative*. If a data set contains both positive and negative instances, the view shift is a mere formality, but not if negative evidence is missing from the data set – then the problem of *separating* the classes becomes a problem of *describing* the positive class (albeit it can be solved by modifications of standard classification techniques [225]).

There exist classifiers that are able to give a real-valued estimate of their conviction about an instance belonging to a particular class (for example, naïve Bayes), which may be valuable in particular applications. Some classifiers produce decisions which can be interpreted by a human (like decision-tree learners), while others output answers which may not be easy to trace (neural networks, support vector machines). The ability to learn *online* is also an important property a classifier can possess, meaning that the learned model may be incrementally updated with each new training instance.

This section attempts to present classification algorithms from the viewpoint of their application to high-dimensional data. First, several key classification algorithms are presented, illustrating the diversity of approaches to the task, followed by a description of the ways to use binary classifiers for multi-class classification. Then, evaluation of classifiers is discussed, introducing several ways to use data sets, and providing an overview of evaluation measures and data collections. *Overfitting*, a problem which is often encountered in high-dimensional domains, is explained next. The section is concluded with a brief discussion of the similarities and differences between various classification techniques.

2.3.1 Algorithms

Perceptrons

The *perceptron*, originally introduced by Rosenblatt [181], is a binary classifier which uses the value of the inner product of vectors $\mathbf{w} \cdot \mathbf{x}$ to classify instance $\mathbf{x}$ according to the previously learned vector of weights $\mathbf{w}$. If the inner product, summed with some value b (called the bias, or threshold), is greater than or equal to 0, the instance is assigned to the positive class, and vice versa. More precisely, for binary class $c \in \{-1, 1\}$, $c = \mathrm{sg}(\mathbf{w} \cdot \mathbf{x} + b)$, where $\mathrm{sg}(x) = 1$ if $x \geq 0$, and $\mathrm{sg}(x) = 0$ otherwise. This means that $\mathbf{w}$ and b define a hyperplane which linearly separates the vector space, as exemplified in Figure 2.8(a) for the two-dimensional case.

Figure 2.3 (adapted from [133]) shows a schematic representation of the perceptron. Every *input* x_i represents a component of vector $\mathbf{x}$, and has an associated weight w_i ($i \in \{1, 2, \ldots, d\}$). The bias can be viewed as constant input +1, with the associated weight b. To compute the *output* of the perceptron, every input is multiplied by its corresponding weight, the sum is taken, and the classification decision made based on its sign.[5]

[5]The sigmoid function $1/(1 + e^{-x})$ is often used instead of the sign.

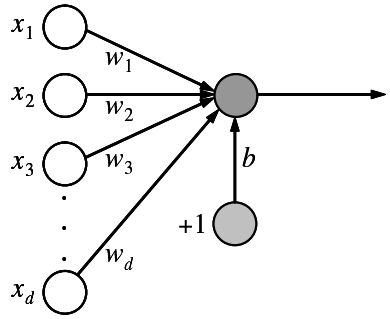


Figure 2.3: The perceptron
Slika 2.3: Perceptron

Learning the vector $\mathbf{w}$ starts by assigning it a $\mathbf{0}$ vector (or a vector of small positive weights) and continues by examining each training instance $\mathbf{x}$ one at a time, classifying it using the currently learned $\mathbf{w}$. If the classification is incorrect, the vector is updated: $\mathbf{w} \leftarrow \mathbf{w} \pm \eta\mathbf{x}$, where addition (subtraction) is used when $\mathbf{x}$ belongs to the positive (negative) class, and η is a small positive number – the *learning rate*. The effect of the update is to shift the weights of $\mathbf{w}$ towards the correct classification of $\mathbf{x}$, in proportion to their "importance" signified by the values of weights in $\mathbf{x}$. The algorithm iterates multiple times over instances in the training set, until all examples are classified correctly, or some other stopping criterion is met.

The perceptron was shown to exhibit solid performance on high-dimensional data such as text [190], despite its simplicity. There exist numerous extensions, one of them being the *voted-perceptron* by Freund and Schapire [63].

In the voted-perceptron algorithm, *all* vectors $\mathbf{w}$ calculated during training are retained, together with the number of training instances they "survive" without being modified. Then, for a list of such weighed perceptrons $(\mathbf{w}_1, c_1), \ldots, (\mathbf{w}_k, c_k)$, the classification is calculated as the sign of the weighed sum of the classifications given by each saved perceptron:

$$c = \mathrm{sg}\left(\sum_{i=1}^{k} c_i \mathrm{sg}(\mathbf{w}_i \cdot \mathbf{x})\right),$$

assuming all thresholds are zero. The voted-perceptron was shown to be effective on high-dimensional data, at the same time being simple to implement and having a low training time [63]. Since multiple "simple" classifiers are explicitly combined when making a classification decision using the voting mechanism, voted-perceptron can be regarded as a classifier *ensemble* method.

Perceptrons are also the building blocks of one type of neural networks. Neural networks have been applied to high-dimensional data such as text [221, 182], but their use is not particularly widespread, since more complex nonlinear models did not show significant performance improvement over the simpler linear ones [190], to compensate

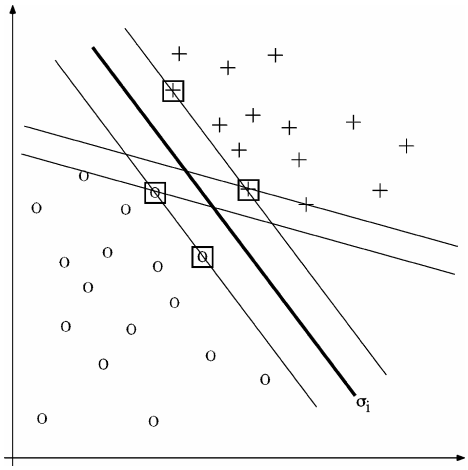


Figure 2.4: The maximum margin hyperplane determined by the SVM, which separates the two classes, with highlighted support vectors
Slika 2.4: Maksimalno razdvajajuća hiperravan određena pomoću SVM, koja razdvaja dve klase, sa naglašenim *support* vektorima

for the inherently long training times. Neural networks have also been used for time-series classification [156].

Support Vector Machines

One of the most sophisticated classifiers, suitable for application to high-dimensional data, is the support vector machine (SVM) classifier. It has been successfully used for classification of both text [191], and time-series data [52]. SVM is a binary classifier, and its main idea lies in using a predetermined *kernel function*, whose principal effect is the transformation of the feature vector space into another space, usually with a higher number of dimensions, where the data is linearly separable. Quadratic programming methods are then applied to find a *maximum margin hyperplane*, that is, the optimal linear separation in the new space, whose inverse transformation should yield a good classifier in the original vector space. Figure 2.4 (from [190]) shows a graphical representation of the separating hyperplane for a two-dimensional space (after the transformation), where the class feature is depicted with labels + and ∘. The hyperplane, in this case a line, lies in the middle of the widest strip separating the two classes, and is constructed using only the instances adjacent to the strip – the *support vectors* (outlined by squares in the figure).

Although the theoretical foundations for SVMs were laid out by Vapnik in the 1970s [214, 213], the computational complexity of various solutions to the quadratic programming problem restricted the use of SVMs in practice. Only relatively recently

were approximate solutions derived which enabled feasible and, compared with some other classifiers, superior training times. One solution was by Osuna et al. [151], improved by Joachims [97] and implemented in his SVM^{light} package. An alternative is Platt's *sequential minimal optimization* (SMO) algorithm [157, 101], available, for instance, as part of the Weka machine-learning workbench [224].

Support vector machines can handle very high dimensionality, and are not particularly sensitive to overfitting (see Section 2.3.4), making them highly suitable for application to text without dimensionality reduction [96]. Many practical studies have confirmed this argument [119], and there is a wide consensus that SVMs are one of the best performing text classifiers available today.

Bayesian Learners

The probabilistic approach to modeling data has resulted in several useful machine-learning techniques which can be used on high-dimensional data. One of them is the simple, but effective *naïve Bayes classifier*, and another, more expressive but also more complex and still actively researched – *Bayesian networks*.

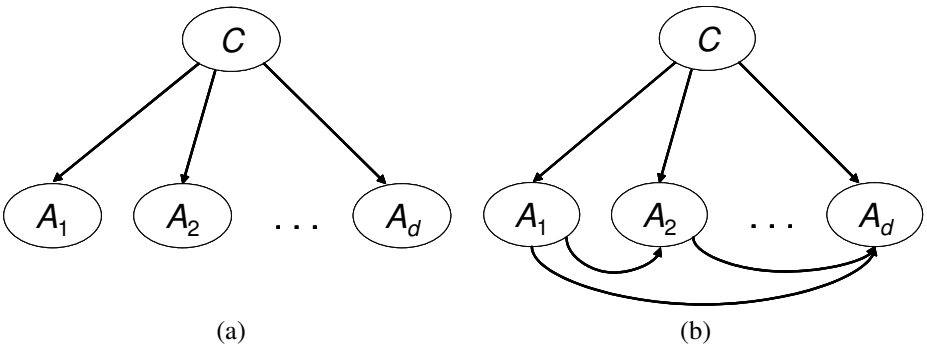
Naïve Bayes. The naïve Bayes classifier has been "around" for a long time, but was initially more in the focus of information retrieval, rather than the machine-learning community [122]. The main principles of its functioning are as follows. Let random variable C denote the class feature, and $A_1, A_2, \ldots, A_d$ the d components of the attribute vector. Then, the classification of a specific vector $(a_1, a_2, \ldots, a_d)$ is

$$c = \operatorname*{argmax}_{c_j \in C} \mathrm{P}(c_j | a_1, a_2, \ldots, a_d)\,,$$

providing that one class maximizes the expression. Application of the Bayes theorem transforms the expression to

$$\begin{aligned} c &= \operatorname*{argmax}_{c_j \in C} \frac{\mathrm{P}(a_1, a_2, \ldots, a_d | c_j)\,\mathrm{P}(c_j)}{\mathrm{P}(a_1, a_2, \ldots, a_d)} \\ &= \operatorname*{argmax}_{c_j \in C} \mathrm{P}(a_1, a_2, \ldots, a_d | c_j)\,\mathrm{P}(c_j) \\ &= \operatorname*{argmax}_{c_j \in C} \mathrm{P}(c_j) \prod_{i=1}^{d} \mathrm{P}(a_i | c_j)\,. \end{aligned}$$

The last derivation uses the assumption that attributes are mutually independent, which obviously does not hold in reality, hence the prefix "naïve." Nevertheless, the assumption has been shown to work in practice. Training involves approximating the values $\mathrm{P}(c_j)$ and $\mathrm{P}(a_i|c_j)$ from data. Several approaches exist, depending on the assumed data distribution. The approach most often used on text involves the multinomial model, and was recently subjected to several enhancements [176, 106]. In the classification phase, if multiple classes maximize the expression $\mathrm{P}(c_j) \prod_{i=1}^{d} \mathrm{P}(a_i|c_j)$ different strategies



(a) (b)

Figure 2.5: The naïve Bayes classifier (a), and the Bayesian network that captures inter-attribute dependencies (b)
Slika 2.5: „Naivni“ Bejesov klasifikator (a), i Bejesova mreža koja izražava međuzavisnost atributa

may be employed to resolve the ambiguity, for example, by selecting the class with the highest prior probability $\mathrm{P}(c_j)$, or simply by choosing one of the classes randomly.

Bayesian networks. Note that without the independence assumption in naïve Bayes, estimating the values of $\mathrm{P}(a_1, a_2, \ldots, a_d|c_j)$ would have been infeasible. However, data attributes usually *are* interrelated, and one way to capture such dependencies is by means of Bayesian networks.

Generally speaking, Bayesian networks consist of nodes which are random variables, and vertices representing conditional probabilities between them. Their aim is to offer a computationally feasible and graphically representable way to express and calculate dependencies between events. The graphic in Figure 2.5(a) shows the naïve Bayes classifier, with conditional probabilities $\mathrm{P}(A_i|C)$ depicted as arcs from C to A_i. The dependencies between attributes, which are missing in naïve Bayes, are added to the Bayesian network shown in Figure 2.5(b).

Again, it would be computationally infeasible (and not even allowed in a Bayesian network) to calculate dependencies between *all* attributes, especially with high-dimensional data, such as text. The trick with Bayesian networks is to express only the dependencies which are necessary (or strong enough to have an impact on the solution to a particular problem), under constraints which ensure the correctness and feasibility of computation. This can be done manually, by supplying the structure of the network – then training a Bayesian network resembles the training phase of the naïve Bayes classifier, with conditionals being estimated from the data set. If estimation of dependencies from data is not possible, training becomes more difficult, with several solutions being available [136]. Learning the *structure* of the network presents a bigger challenge, and is still an area of active research. Several books devoted to the subject of Bayesian networks have recently appeared [94, 142].

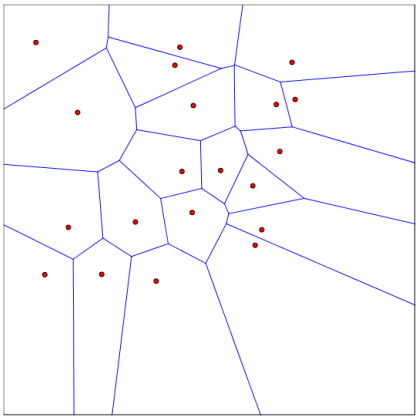


Figure 2.6: Voronoi tessellation of the data space, for the 1-NN classifier
Slika 2.6: Voronojeva teselacija prostora podataka, za 1-NN klasifikator

Nearest-Neighbor Classifiers

The training phase of *nearest-neighbor* (also known as *instance-based*, or *memory-based*) classifiers is practically trivial, and consists of storing all examples in a data structure suitable for their later retrieval. Unlike other classifiers, all computation concerning the classification of an unseen example is deferred until the classification phase. Then, k instances most similar to the example in question – its k *nearest neighbors* – are retrieved, and the class is computed from the classes of the neighbors. The computation of the class can consist of selecting the majority class of all the neighbors, or distance weighing may be used to reduce the influence of faraway neighbors on the classification decision. The choice of k depends on the concrete data and application – there is no universally best value. The similarity function is usually the cosine of the angle between vectors (see Section 2.2); with richer representations of instances and more complex similarity functions the issue moves into the field of case based reasoning [136, 114].

To illustrate the workings of the k-nearest neighbor (k-NN) classifier, for the simple case of $k = 1$ and data dimensionality 2, Figure 2.6 shows the Voronoi tessellation of the data space by training points [205], where the region surrounding every point represents the area in which the point is the nearest one of all points in the training set. At classification phase, the label of an unseen point is determined as the label of the training point in the corresponding region. Therefore, depending on the labeling of training points, the boundaries between classes for the 1-NN classifier model follow the boundaries between regions.

A major problem with applying k-NN to text is the sheer volume of practical textual data, which consumes memory and slows down retrieval. One of the first applications of the k-NN classifier to text was by Yang [226], who addressed this problem by organizing data into a three-layer network of weights, with one layer for words, one for

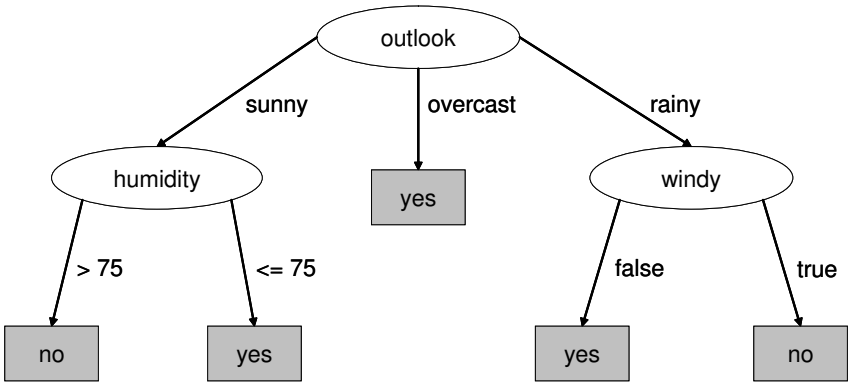


Figure 2.7: The decision tree generated from the weather data
Slika 2.7: Stablo odlučivanja generisano za podatke o vremenskim uslovima

documents and one for categories. The same problem can be tackled by storing only the instances for which there is evidence during training that they would contribute significantly to classification [4]. Other improvements to the k-NN algorithm include feature weight adjusting [80] and document clustering [91].

On the other hand, the simple method combining the 1-NN classifier and some form of dynamic time warping (DTW) distance (see Section 2.2) was shown to be one of the best-performing time-series classification techniques [47, 103].

Decision Trees

A decision tree (DT) is a tree whose internal nodes represent features, where arcs are labeled with outcomes of tests on the value of the feature from which they originate, and leaves denote categories. The decision tree constructed from the weather data set in Table 2.1 is shown in Figure 2.7. Classifying a new instance using a decision tree involves starting from the root node and following the branches labeled with the test outcomes which are true for the appropriate feature values of the instance, until a leaf with a class value is reached.

Two of the most widely used decision-tree learning algorithms are classification and regression trees (CART) by Breiman et al. [20], and Quinlan's C4.5 [159] (an improved commercialized version C5.0 exists, which focuses on better generation of rules). Learning a decision tree with C4.5 involves selecting the most informative feature using a combination of the information-gain and gain-ratio criteria described in Section 2.8, determining how best to split its values using tests, and repeating the process recursively for each branch/test, without considering features which were already assigned to nodes. Recursion stops when the tree perfectly fits the data, or when all features have been used up. The tree in Figure 2.7 was generated using the C4.5 algorithm.

To avoid overfitting (see Section 2.3.4), pruning can be performed on the learned tree, which reduces its fit to the training data, at the same time attempting to improve its accuracy in the general case. C4.5 performs this task by converting the tree to an equivalent rule form (one for each path from root to leaf), estimating the general accuracy of each, and improving it by removing some tests. Then, rules are sorted in decreasing order of estimated accuracy and used in this form for classification.

Decision trees (and rules) are especially useful when the workings of the classifier need to be interpreted by humans, offering insight into the structure of data. As for text data, DTs may be unsuitable for many applications since they are known not to be able to efficiently handle great numbers of features. Nevertheless, sometimes they do prove superior, for instance with data sets in which a few highly discriminative features stand out from the many [68].

AdaBoost

AdaBoost is one of the most well known ensemble methods for classification, using voting to combine the classification decision of multiple "weak" learners, where "weak" refers to simple, fast classifiers that do not tend to produce overly complex models. The rationale behind AdaBoost, as well as other "boosting" methods, is to iteratively build new weak classifier models that seek to perform well on examples that the models from previous iterations found problematic.

In its original formulation [62, algorithm "AdaBoost.M1"], AdaBoost assumes a "weak learner" that may be any classifier able to handle weighted instances in a data set, where the weight of an instance is a positive number. In the presence of weights, the training error of a weak learner is the sum of weights of misclassified instances divided by the total sum of weights in the training set, as opposed to the proportion of incorrectly classified instances. Through weights, the weak classifier can be directed to give special attention to a particular set of instances, that is, those instances with high associated weights. The decision-tree algorithms C4.5 [159] and CART [20] are examples of classifiers that can naturally handle instance weights.[6]

The training phase in the AdaBoost algorithm commences by assigning equal weight $1/n$ to each training instance, where n is the size of the training set. Then, the weak classifier is trained, and weights are updated based on its training error. The weight associated to correctly classified instance i is changed to:

$$w_i \leftarrow w_i \cdot error\,/(1 - error)\,,$$

where $error$ denotes training error, while the weights of incorrectly classified instances are left unchanged. After each update, instance weights are normalized to unit sum. The procedure is iterated a specified number of times, or until $error = 0$, or $error \geq 0.5$.

[6] If a classifier is unable to explicitly handle instance weights, the same effect can be achieved by resampling points from the training set [224].

In the classification phase of one data instance, a weighted vote is taken between all weak classifiers. More precisely, zero weights are first assigned to each class. Then, the class c predicted by each weak learner j has its weight incremented:

$$y_c \leftarrow y_c + \log \frac{1 - error_j}{error_j},$$

where $error_j$ is the training error of classifier j.[7] Finally, the class with the highest weight is returned as the predicted class.

The original AdaBoost algorithm has witnessed numerous extensions, including versions optimizing Hamming loss (AdaBoost.MH), and producing real-valued predictions ("Real AdaBoost") [188, 64].

Generally, AdaBoost is known to be able to generate excellent classifiers in a wide variety of domains [82]. Its weaknesses, however, include the tendency to overfit the training data [224], and sensitivity to outliers [175].

2.3.2 Using Binary Classifiers for Multi-Class Problems

There are several techniques to reduce multi-class classification problems to a set of binary problems. The oldest and most commonly used is referred to as *one-vs-all binarization*: the original data set consisting of m classes is split into m data sets, with each class once declared as positive, and the negative class composed of the leftover $m - 1$ original classes. During classification, m binary decisions are made, and in the case of more than one positive outcome the most confident one is taken as the final classification decision.

The one-vs-all scheme may be prone to errors emerging from inadequate confidence values returned by the binary classifiers. One possible alternative is *round robin binarization* [67], also known as all-vs-all. In this scheme, $\binom{m}{2}$ classifiers are trained, one for each pair of classes, and the "winning" class during classification is determined as the outcome of a round robin tournament held between classes via the binary classifiers.

Another alternative to the one-vs-all method relies on the use of *error-correcting codes* [224], applicable in cases when $m > 3$. Instead of training m binary classifiers, a much larger number of $2^{m-1} - 1$ is trained, with the difference that several original classes may be combined into the positive class, instead of using only one. This results in classification decisions that do not rely on one, but several positive binary outcomes, with the series of expected positive (1) and negative (0) decisions forming the error correcting code for each of the original m classes. This leads to a classification decision which is less prone to errors – in the case of one, or even several incorrect binary classifications, the true class may still be recognized from the remaining correct binary decisions which match the appropriate code.

The above approach, referred to as the *exhaustive error-correcting code method*, is infeasible for large values of m. Therefore, other schemes for constructing shorter

[7] This voting scheme explains the stopping criteria in the training phase: for $error = 0$ and $error \geq 0.5$ the logarithm is undefined.

Class	Class vector
a	1000
b	0100
c	0010
d	0001

(a)

Class	Class vector
a	1111111
b	0000111
c	0011001
d	0101010

(b)

Table 2.2: Using binary classifiers for multi-class problems: (a) standard method (one-vs-all); (b) error-correcting code method
Tabela 2.2: Korišćenje binarnih klasifikatora za probleme sa više klasa: (a) standardna metoda („sam protiv svih“), i (b) metoda kodova koji ispravljaju greške

error correcting codes are often used to reduce the number of trained classifiers without significant loss of performance, resulting, for instance, in the *randomly selected error-correcting code method*.

To illustrate the use of error-correcting codes, consider a multiclass problem with four classes a, b, c, and d. Four binary classifiers, each of which is trained to recognize one of the classes as positive (for example, using the one-vs-all method), can be viewed as producing a 0/1 value in a four-digit code vector representing each class. Table 2.2(a) shows the (row) vectors which represent each class, where columns constitute the expected output of every classifier for each class. In this scheme, if a classifier produces erroneous output (a 1 instead of a 0, or vice versa), there does not exist a way to recover from the error. However, each class can be represented by, for example, seven binary digits instead of four, and the corresponding classifiers constructed to potentially produce an output of 1 for more than one class, as shown in Table 2.2(b). At classification time of a particular instance, which belongs to class a, the classifier output vector 1011111 can still be correctly interpreted as signifying class a, despite of the erroneous output produced by the second classifier. This is because, of all class vectors in Table 2.2(b), vector 1011111 is most similar to vector 1111111, since they contain only one different binary digit [224].

2.3.3 Classifier Evaluation

There are three different aspects of classifier performance [191]:

- training efficiency,
- classification efficiency,
- correctness of classification.

Training and classification efficiency are measured in terms of execution speed and memory consumption, and present very important factors in practical applications of

classification. The end user will certainly be affected by low classification efficiency, and if online classifiers are used, by low training (that is, "updating") efficiency as well. Nevertheless, the attention of the research community is dominated by the correctness aspect, often giving the other two only a passing glance. This is also true for this text, as "classification performance" mentioned in previous sections was mostly referring to correctness, and will continue in the same manner.

The data set used for classification is usually divided into the *training set*, used to train the classifier, and the *test set* for evaluating classifier performance. One widely used measure for evaluating classifier performance in machine learning is *accuracy* – the percentage of correctly classified examples from the test set. Sometimes, a third set is extracted from the data set, called the *validation set*, which is used in the training phase to evaluate the classifier and help tune its parameters to yield optimal performance. It is important to separate the validation and test sets, because a classifier tuned on the test set would exhibit excellent performance when evaluated on it, which would in all probability be misleading.

The ratio between the sizes of the training and test set (the *split*) may depend on the amount of available data, the particular application and many other factors – there are no firm rules. The usual splits include 2/1, 3/1, 4/1, and 9/1, and there are cases when test sets larger than the training sets are used.

Several problems plague the singular split scheme. The first one concerns the distribution of classes in the original data set, which may not be preserved in the training and test sets if they are generated randomly. *Class distribution* is an important property of data sets, and practically all classifiers either implicitly or explicitly use it while learning the model. Therefore, a split which breaks the class distribution may also break classifier performance. A cure for this is *stratification*, the notion of preserving the class distribution in all data sets derived from the original.

The second, more serious problem lies in the arbitrariness of which examples end up in which set after the split (even with stratification). There are no guarantees that a particular split into a training and test set will yield a realistic evaluation of a classifier. The problem is even more emphasized when the amount of available data is small. Then, the exclusion of different test sets from training data may led to great variance in classifier performance measurements.

A common solution to this problem is in *cross-validation*, a technique borrowed from statistics: the data set is split into n subsets, one is declared the test set, the others are merged into the training set, and the classifier is evaluated. The procedure is repeated n times, every subset once being the test set, and the results are averaged. Each iteration is called a *fold*, and the whole process n-fold cross-validation. Stratification is also possible here, yielding *stratified n-fold cross-validation*. The whole procedure can be repeated k times, making sure that in each run the n subsets of the original data set are sufficiently different. This is known as k *runs of n-fold cross-validation* (with the adjective "stratified" also applicable).

As with the split, there are no firm rules for choosing the values of n and k. In machine learning in general, there is some agreement that 10 as the value of both n and k is

a satisfactory solution, but for applications to text these values may simply be too high for feasible training efficiency. For the same reason *leave-one-out cross-validation*, the extreme case of n-fold cross-validation where n equals the total number of examples, is avoided on text. Many experiments in text categorization were performed on single splits (see page 48), there are also examples of 5 runs of 4-fold cross-validation [57, 68]. In time-series classification, cross-validation splits with the number of folds dependent on the data set have been used, with the sizes of the training and test sets reversed (one of the n subsets of a data set is declared the training set, and the others merged into the test set), making the classification problem somewhat more difficult [47].

Having multiple measurements means that a statistical test, like the *t-test*, can be used to determine whether the performance of two classifiers is *significantly* different. However, caution needs top be exercised when multiple runs are performed, since in such cases the performance measures become interdependent [187]. In fact, increasing the number of runs can eventually result in any difference being judged as statistically significant, when in reality it is not [224]. Numerous methods have been proposed to address this issue, including averaging over folds and runs [19], approaches involving 5×2-fold cross-validation [6], and a modification of the t-test referred to as the *corrected resampled t-test* [139, 224].

Statistical tests may be used in a scenario where, for example, multiple classifiers are being compared over multiple data sets. Then, the number of statistically significant *wins* and *losses* can be counted for each classifier, and the subtracted value of *wins–losses* used to rank the classifiers relative to one another.

Evaluation Measures

Accuracy – the percentage of correctly classified examples – is a good measure for evaluating classifiers in a wide variety of applications. However, consider a binary classification problem with *imbalanced class distribution*, where examples from the negative class constitute 95% of the data set. Then, the *trivial rejector* (that is, the classifier which assigns all examples to the negative class) has an accuracy of 95%, but is totally unusable in practice if the positive class is of any importance. Class imbalance not only dismisses accuracy as an evaluation measure, but creates the need to fine-tune classifiers to assign an adequate importance to the minority class, in order to achieve desired performance.

Several evaluation measures which originated in information retrieval (IR) are commonly used to evaluate classifiers, especially in the context of text categorization. IR is concerned with the *relevance* of documents retrieved from a database as a response to a user query, where "relevant" may now be considered as "belonging to the positive class." Then, *precision* is defined as the ratio of the number of relevant documents that were retrieved (the number of documents *correctly* classified as positive), and the total number of retrieved documents (the number of documents classified as positive). In terms of outcomes of binary classification summarized in Table 2.3, it is calculated as

$$precision = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} .$$

		Predicted class: yes	Predicted class: no
Actual class	yes	True Positive (TP)	False Negative (FN)
	no	False Positive (FP)	True Negative (TN)

Table 2.3: Outcomes of binary classification
Tabela 2.3: Rezultati binarne klasifikacije

Similarly, *recall* is the ratio between the number of relevant documents retrieved, and the total number of relevant documents:

$$recall = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} .$$

For comparison, *accuracy* is

$$accuracy = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}} .$$

Although they differ by only one term in the formula, precision and recall are really on the opposite sides of the spectrum – while precision characterizes the mistakes made in making the positive decision, recall expresses the coverage of the real positives by the decision, regardless of mistakes. The *trivial acceptor* has 100% recall and very low precision, while a classifier which makes only one positive classification, and it happens to be correct, has 100% precision and very low recall. Therefore, these two measures are seldom used by themselves, and may be combined to form the *F-measure*:

$$\mathrm{F}_\beta = \frac{(\beta^2 + 1) \cdot precision \cdot recall}{\beta^2 \cdot precision + recall} .$$

When $\beta = 1$, F-measure represents the harmonic mean of precision and recall, taking both of them equally into account. For $\beta < 1$ precision is given more importance, ending with $\mathrm{F}_0 = precision$, while $\beta > 1$ means recall gets the upper hand, with the other extreme at $\mathrm{F}_\infty = recall$. Besides $\beta = 1$, the usual values of β that are used are $\beta = 0.5$ when precision is considered more important, and $\beta = 2$ when recall is preferred [137].

Classifiers can generally be tuned to favor precision or recall during training. The point where (averaged) precision and recall are equal for a particular test set is the *break-even point* (BEP), and is also used as a measure of classifier performance, although it has received some criticism [41].

In case of multi-class classification, all these measures can be considered for each class separately. If we denote the classification outcomes with regards to class $i \in \{1, 2, \ldots, n\}$ by TP_i, TN_i, FP_i, and FN_i, then $precision_i$ and $recall_i$ calculated using them refer to classification performance on the ith class. There are two ways to express "global" precision and recall: *microaveraging* and *macroaveraging*.

Microaveraged precision and recall are obtained by first summing up classification outcomes by class:

$$\begin{aligned} precision^m &= \frac{\sum_{i=1}^{n} \mathrm{TP}_i}{\sum_i (\mathrm{TP}_i + \mathrm{FP}_i)}, \\ recall^m &= \frac{\sum_{i=1}^{n} \mathrm{TP}_i}{\sum_i (\mathrm{TP}_i + \mathrm{FN}_i)}, \end{aligned}$$

while macroaveraging involves averaging of precision and recall calculated for each individual class:

$$\begin{aligned} precision^M &= \frac{\sum_{i=1}^{n} precision_i}{n}, \\ recall^M &= \frac{\sum_{i=1}^{n} recall_i}{n}. \end{aligned}$$

Data Collections

Over the course of research into techniques for machine learning, data mining, and information retrieval, a number of data repositories were set up to facilitate the usage of standard evaluation benchmarks and reproducibility of results.

Regarding machine-learning and data-mining research, one of the most influential data-set repositories is the University of California, Irvine (UCI) Machine Learning Repository [7], which currently hosts around 200 data sets originating from numerous application areas, suitable for evaluation of classification, clustering, regression, and other ML tasks. Also, numerous repositories exist, that specialize on data from particular application areas, for example the Kent Ridge Bio-Medical Data Set Repository [124], which predominantly hosts gene expression data, and the University of California, Riverside (UCR) Time Series Repository [104], which focuses on time-series data sets suitable for the tasks of classification and clustering. As for information retrieval, the series of Text Retrieval Conferences (TREC) continually provides corpora for evaluation of various IR tasks.

Several freely available textual data sets have been repeatedly used for evaluating the performance of classification, as well as information retrieval. They include:

- the *Reuters* corpus, assembled by Lewis [121] from Reuters news stories related to economics, and enduring several modifications afterwards. The Reuters-22173 data set (where the number refers to the number of examples) was experimented on using several different subsets/splits which became almost standard – ModLewis, ModApté, and ModWiener, while the most recent version is Reuters-21578 ModApté, with several different subsets used. A new version of the Reuters corpus called Reuters Corpus Volume 1 (RCV1) was released more recently [123], containing a much larger volume of data. It should eventually replace Reuters-21578 as the standard Reuters data set for experimentation.

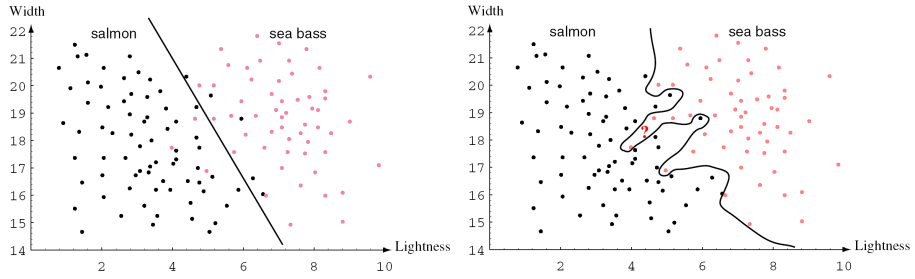
- the *20-newsgroups* data set, consisting of messages taken from 20 Usenet groups, where the groups themselves represent categories [95, 136],
- the *WebKB* corpus of university Web pages, classified into seven categories by type [42],
- the OHSUMED corpus (a subset of the Medline database), where documents are titles and abstracts published in medical journals, initially used for evaluation of information retrieval [83],
- the *Cora* data set, originating from a Web search engine focusing on the domain of computer science research papers [57],
- the *dmoz* Open Directory collection, available for download in RDF format, which contains tiles, hyperlinks, and short descriptions of a large number of Web pages organized into a multi-level topic hierarchy. It is constantly evolving, but nevertheless has been extensively experimented on [46, 30, 68],
- the collection made available by Han and Karypis [79], also used by Forman [57], includes documents from TREC collections, the OHSUMED collection, Reuters and Los Angeles Times news stories, etc.

Despite the commonality of data collections used to evaluate classifiers, exact comparison of classifiers is difficult due to different experimental setups – subsets of collections, splits, evaluation measures, and versions and parameters of algorithms. Nevertheless, some conclusions can be reached, and are discussed in Section 2.3.5.

2.3.4 Overfitting

Training a classifier "too much," in the sense of maximizing its performance on the training set, may in fact lead to suboptimal performance on a separate test set and real life data. This phenomenon is referred to as *overfitting*, and may come as a consequence of a large number of training instances, noisy data, and/or high dimensionality. Some classifiers are more prone to overfitting than others, and many of them employ complex strategies to avoid it. The philosophical equivalent of the problem lies in the *Occam's razor* principle, which in ML terms translates to preferring a simple model which reasonably fits the data, to a complex one which does so more accurately.

To illustrate, consider one binary classification problem, in a two-dimensional feature space, of separating salmon and sea bass on evidence of their width and lightness of scales, shown in Figure 2.8 (from [50]). The data is clearly not linearly separable, therefore the linear classifier in Figure 2.8(a) leaves several misclassified examples on both sides of the boundary. On the other hand, the complex model in Figure 2.8(b) perfectly fits all examples, making great efforts to "pick up" every fish which strayed deep into the waters occupied by instances of the other species. This leads to whole regions being marked for one class on the evidence of a single example, when, considering all surrounding examples, they have a greater probability of belonging to the other



(a) (b)

Figure 2.8: A simple and complex model for binary classification in a two-feature space
Slika 2.8: Prost i složen model binarne klasifikacije u prostoru sa dva atributa

class. In one such region a new example is marked by a question mark in the figure – it will be classified as sea bass although it is more likely to be a salmon, considering the surroundings. In all probability, the linear model will perform the same or better on real-world data than the complex one, at the same time being much simpler to derive, apply, and maintain.

2.3.5 Discussion

A natural question which arises when considering classifiers is: which one is best? There seems to exist no universal answer, however some domain-dependent conclusions have been reached in the literature.

Regarding text categorization, as far as accuracy is concerned, SVMs have consistently dominated other classification algorithms in experimental benchmarks. Nevertheless, much may depend on the properties of a data set (as was effectively demonstrated by Gabrilovich and Markovitch [68]), the evaluation measure that is considered important, and the final application of the classifier. Too many times were sophisticated techniques employed only to later discover that simple techniques performed as well, or even better, at the same time being easier to manage [224].

In the time-series domain, somewhat counterintuitively, the straightforward combination of the 1-NN classifier and some form of dynamic time warping (DTW) distance (Section 2.2) was shown to outperform almost all other classification techniques, in the sense that 1-NN with DTW is the single best-performing out-of-the-box algorithm [47, 103]. Chapter 6 will provide additional interpretation of this observation.

If the classifier needs to be trained online, perceptrons and nearest-neighbor methods are generally good choices, as is naïve Bayes which can be adjusted for that purpose. SVMs are difficult to use in this way as new examples may alter the configuration of support vectors and render the classification unstable. If interpretability of classifi-

cation is a big concern, C4.5 may be a good choice. Nearest-neighbor classifiers have very short training times, as does naïve Bayes; SVMs take a little longer, while C4.5 and Bayesian networks may be prohibitively slow in certain situations. Classification time is longest for nearest neighbor, but takes very little for naïve Bayes, SVMs, and the perceptron. All these properties of classifiers, together with the characteristics of data and the nature of the particular problem being solved, need to be taken into account when selecting the right tools for the task at hand.

2.4 Semi-Supervised Learning

In many situations, obtaining data sets with complete class information may be difficult or infeasible, especially when dealing with large numbers of data instances. Semi-supervised learning (SSL) is a family of techniques which makes use of both labeled and unlabeled examples during learning.

Generally, in order for SSL to work, that is, for unlabeled data to be helpful in the process of learning a model, certain conditions need to be met. One of the most widely adopted conditions is the *cluster assumption*, which can informally be formulated as follows [33].

Cluster assumption: If points are in the same cluster, they are likely to be of the same class.

An equivalent formulation is referred to as *low-density separation* [33].

Low-density separation: The decision boundary should lie in a low-density region.

Both formulations of the cluster assumption deal with the relationship between the cluster structure in the data and the information provided by class labels. If the cluster assumption holds then there exists enough positive correspondence between class and cluster structure in the data so that using unlabeled points, and the cluster information carried by them, can help the learning process.

Semi-supervised learning algorithms can be viewed as belonging to one of three major families: generative models, algorithms which directly implement the low-density separation assumption, and graph-based methods [33]. The following three paragraphs briefly summarize the main ideas behind these families of methods. For more comprehensive overviews of SSL, we refer the interested reader to the survey by Zhu [237], as well as books by Chapelle et al. [33], Liu [128], and Sebe et al. [192].

Generative models. This family of algorithms attempts to model the conditional probabilities $\mathrm{P}(X|C)$ for each class using some unsupervised learning procedure, where random variable C denotes the class attribute, and X corresponds to (multiple) data attributes. Then, the probabilities for prediction, $\mathrm{P}(C|X)$, are obtained by applying the Bayes theorem [33]. The term "generative" originates from the explicit modeling of the

structure of the problem by incorporating knowledge about $\mathrm{P}(X)$ and $\mathrm{P}(X|C)$ into prediction. A classic example of the generative model approach is the combination of the naïve Bayes classifier (Section 2.3.1) and the EM approach to probabilistic clustering (see Section 2.5.1) [128].

Low-density separation. The most common approach to direct implementation of the low-density separation assumption is through a maximum-margin algorithm like support vector machines [33]. One way to use unlabeled data in SVM training is to select the labels of the unlabeled data points in such a way that the margin of the trained classifier is maximized [128]. This procedure requires that the unlabeled instances which need to be classified are known beforehand, with the labels of all these instances determined collectively. In other words, data points (but not the class labels) from the test set are used in the process of building the maximal-margin model. This approach is commonly referred to as *transduction*, giving rise to the term *transductive support vector machines* [128, 33].

Graph-based methods. Algorithms that belong to this family of approaches to SSL represent data (both labeled and unlabeled) as nodes of a graph, the edges of which are weighted according to pairwise distances between incident nodes [33]. The graph, with similar instances connected by larger weights, is used to assign classes to unlabeled points in such a way that labels of nodes connected by edges with high weight tend to agree with each other [128]. A representative algorithm by Zhu et al. [238] involves computing a real-valued function f on graph nodes, and assigning labels to nodes based on its values. Function f, which exhibits harmonic properties, is obtained by optimizing a quadratic energy function that involves graph edge weights, with the probability distribution on the space of functions f formed using Gaussian fields. For data points $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$ the edge weights may be assigned by the radial basis function (RBF) of the following form:

$$W(\mathbf{x}, \mathbf{y}) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{y}\|^2}{\sigma^2}\right),$$

where σ is a data-dependent constant. Therefore, large edge weights are assigned between nodes that are close to one another with respect to Euclidean distance.

2.5 Clustering

While classification is concerned with finding models by *generalization* of evidence produced by a data set, clustering deals with the *discovery* of models which describe patterns in data, with little or no external guidance. This section will overview clustering techniques in a manner similar to Section 2.3 – the principles behind several key algorithms will be presented first, followed by a discussion of issues in clustering evaluation. The section will be concluded by a discussion of the similarities and differences between various approaches to clustering.

2.5.1 Algorithms

K-means Clustering

The basic K-means clustering algorithm is one of the oldest and simplest clustering algorithms suitable for application to high-dimensional data, which may still produce good results. It involves randomly choosing K points to be the centroids of clusters, and grouping instances around centroids based on proximity. Then, centroids are iteratively recomputed for each cluster, and instances regrouped until there is sufficiently little change in centroid positions. This algorithm depends heavily on the choice of K (which may not be obvious at all for a particular application), and the initial positioning of centroids. Having K-means generate empty clusters is not a rare occurrence.

Instead of explicitly assigning examples to clusters (*hard* assignment), each cluster can be represented by a vector of features and updated on witnessing an example (*soft* or *fuzzy* assignment), based on proximity. That way, representations of clusters are not limited to centroids and may fit some data distributions more naturally.

A close relative of the "soft" variant of the K-means algorithm are *self-organizing maps* (SOMs), a technique with strong origins in neural networks. While K-means is concerned with finding relations among examples in their own space, SOM projects the examples down to a two dimensional grid of interconnected points. Each example activates the point closest to its projection, and the activation is propagated through the grid in a neural network-like manner. Kohonen et al. [109] used a triangular grid SOM to organize a large collection of newsgroup documents.

Hierarchical Clustering

Hierarchical clustering techniques derive a nested hierarchy of clusters, with the extreme of a single cluster containing all instances on one end, and a collection of one-element clusters on the other. One such partition is shown by the *dendrogram* in Figure 2.9 (from [28]). The hierarchy can be constructed from the bottom up (the *agglomerative* approach), by starting with single-element clusters and merging two of the most similar in each step, and from the top down (the *divisive* approach), by repeatedly dividing the cluster with least internal similarity. In both approaches, a concrete set of clusters can be obtained simply by choosing a suitable degree of inter-cluster similarity and "reading off" the clusters from the dendrogram using a horizontal cut-off.

Compared to the divisive approach, the agglomerative approach is relatively straightforward, with the main issues concerning the choice of the inter-cluster distance metric and optimizing the search for most similar clusters. Methods for determining the distance between two clusters include: *single link*, where the distance between two clusters is taken to be the minimum distance (maximum similarity) between two points from the two different clusters, *complete link*, where cluster distance is defined as the maximum distance (minimum similarity) between two points from the two different clusters, and *group average*, where average distance between all pairs of points from the different clusters is used [203]. The divisive approach, on the other hand, offers a wide variety

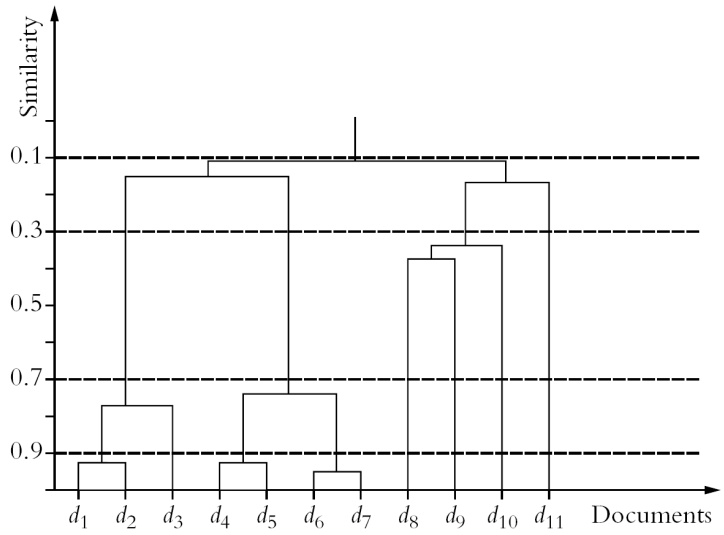


Figure 2.9: A dendrogram representing a hierarchical clustering of a set of examples
Slika 2.9: Dendrogram koji predstavlja hijerarhijski klastering skupa instanci

of ways to split the chosen cluster. One way is to use the basic K-means algorithm multiple times with different choices of starting points, and select the split with the highest overall similarity. This technique, referred to as *bisecting K-means*, exhibited surprisingly good performance at clustering documents [201].

Divisive hierarchical clustering can also be achieved using singular value decomposition, which produced the *principal direction divisive partitioning* (PDDP) algorithm for clustering documents [13]. SVD-based techniques can also be very effective as feature-extraction methods (see Section 2.8.2).

Probabilistic Clustering

In the probabilistic clustering approach, instances are considered to be generated from a *mixture model* of k probability distributions, by first choosing model j with probability p_j, and then drawing an example adhering to the distribution [13]. Each cluster corresponds to a distribution, with instances gathering around its mean at distances determined by variance. The *likelihood* that a particular data set is drawn from a particular mixture model of k distributions is given by

$$\mathrm{L}(X|R) = \prod_i \sum_j p_j \, \mathrm{P}(\mathbf{x}_i|r_j),$$

for instances $\mathbf{x}_i$ and clusters r_j. One probabilistic method, the EM algorithm [136], is based on alternatively estimating (the "E" step) and maximizing (the "M" step) the expected value of the *log-likelihood* function $\log \mathrm{L}(X|R)$.

To illustrate the functioning of the EM algorithm, Figure 2.10 (from [223]) shows several steps of its execution on the "Old Faithful" data set, whose attributes indicate the eruption and waiting times of the famous Yellowstone geyser. Figure 2.10(a) depicts the initial configuration which consists of two Gaussian distributions that do not fit the data, the plots in Figure 2.10(b, c) show two intermediate steps in adjusting the parameters of the Gaussians, and Figure 2.10(d) represents the final state where the distributions correspond with the data very well.

Benefits of probabilistic clustering include the ability to build clusters using different data sets (because clusters are represented independently from examples), iterative examination of instances (the approach is online), and output of results which are easy to interpret [13]. More details on the probabilistic approach to clustering can be found in [28].

Spectral Clustering

Spectral clustering refers to a family of algorithms designed to work well in situations where clusters in the data can have non-convex shapes [18]. Starting from a matrix of pairwise similarities of points in a data set, an adjacency matrix is formed using, for example, mutual k-nearest neighbors or ϵ-neighborhoods, resulting in an undirected graph whose edges are weighted by similarities between nodes. Some notion of a graph Laplacian matrix is then computed, and the eigenvectors corresponding to the smallest eigenvalues are found.[8] The final step usually involves using some standard clustering algorithm like K-means to find the groups of points.

The most well-known spectral clustering algorithms include those by Shi and Malik [194], Ng et al. [145], and Meilă and Shi [135], with the principal differences being in the type of graph Laplacian adopted [220].

The objective behind spectral clustering methods is to identify different groups of data points by finding local neighborhoods within a graph. There exist several viewpoints on how this objective should be achieved, resulting in different approaches to spectral clustering, and explanations as to why spectral clustering methods work: the graph cut point of view (partitioning the graph so that the edges between different groups have a low weights, and the edges within a group have high weights), the random walk point of view (through a stochastic process which jumps from node to node), and the perturbation theory point of view (examining the behavior of eigenvalues and eigenvectors with the introduction of small changes to the matrix, that is, perturbations) [220].

[8] The set of eigenvalues of a matrix is refereed to as "the spectrum," giving rise to the name of the clustering method family.

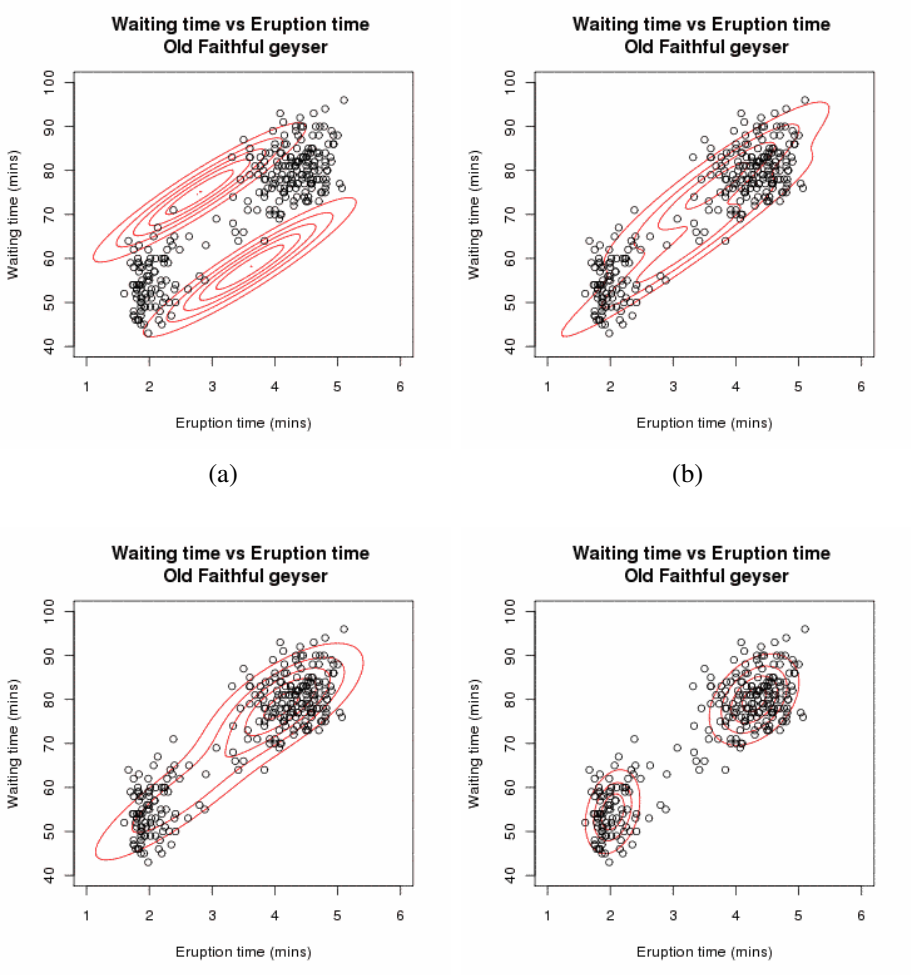


Figure 2.10: Steps of the EM algorithm on the Old Faithful data set: (a) the initial configuration; (b, c) two intermediate steps; (d) the final state
Slika 2.10: Koraci EM algoritma na podacima o "*Old Faithful*" gejziru: (a) početna konfiguracija; (b, c) dva međukoraka; (d) završno stanje

Co-Clustering

Simultaneous clustering of not only instances, but also attributes, has led to the idea of *co-clustering*, which is especially useful for data with large numbers of both instances and attributes. The principle of the approach is to iteratively improve the clustering of examples by examining clusters of features, and vice versa. An advanced method involving bi-partite graphs for text clustering was proposed in [45]. The K-means algorithm can also be used in this context – in an algorithm which gradually co-clusters examples and attributes which has industrial applications in Web analysis [13].

2.5.2 Clustering Evaluation

Evaluating clustering algorithms is a much vaguer notion than evaluating classifiers, as there is no clear-cut definition of what constitutes "good" clusters. Nevertheless, clustering quality can be evaluated from several viewpoints. If there already exists a partition (a classification) of the data set, *external quality measures* can be used, which express how well the clustering measures up to the prescribed labels. *Internal quality measures* require no such labeling, because they try to assess inter-cluster difference and intra-cluster similarity. Several other types of clustering evaluation measures exist, described in the study by Zhao and Karypis [236], and books by Gan et al. [69] and Tan et al. [203].

Comparing clustering algorithms is difficult not only because of the problems of performance measurement, but also because of the lack of consensus on standard evaluation corpora. Document collections presented in Section 2.3.3 have all been used in clustering experiments, together with multitudes of other data collections originating from specific application domains.

Evaluation Measures

Evaluation measures typically used to assess the performance of clustering include *Rand*, *Jaccard*, *entropy*, and the *F-measure* (external); as well as *overall similarity* and *silhouette coefficients* (internal) [173, 69, 203, 201].

To define the *Rand* and *Jaccard* measures, we introduce the following definitions. Let a be the number of pairs of points in a data set which are in different classes but in the same cluster, b the number of pairs of points which are in the same class and cluster, c the number of pairs of points which are in the same class but in different clusters, and d the number of pairs of points in different classes and clusters. Then, the Rand and Jaccard measures are expressed as

$$\begin{aligned} Rand &= \frac{b+d}{a+b+c+d}, \\ Jaccard &= \frac{b}{a+b+c}. \end{aligned}$$

In Section 2.8 we defined the *entropy* of a feature and used it to formulate the information-gain criterion. Entropy can also be used as an external quality measure of clustering. The probability that a document from cluster R_j belongs to class C_i, denoted by p_{ij}, can be calculated from the clustered data set for every i and j, and then the entropy of cluster j with regards to the class distribution expressed as

$$\mathrm{H}(R_j) = -\sum_i p_{ij} \log(p_{ij}).$$

The entropy of the complete set of clusters is now the average of the entropies of all clusters, normalized by cluster size:

$$\mathrm{H} = \sum_j \frac{|R_j|}{n} \mathrm{H}(R_j),$$

where $|R_j|$ is the size of the jth cluster, and n the total number of examples.

The *F-measure*, presented in Section 2.3.3 and typically used for evaluating classifiers, is another metric which can conveniently be utilized to express the external quality of clustering. In information-retrieval terms, if cluster R_j is viewed as an answer to a query with the correct answers constituting class C_i, precision and recall of cluster R_j, relative to class C_i, can be formulated as:

$$\mathit{precision}^{ij} = \frac{n_{ij}}{|R_j|} \qquad \mathit{recall}^{ij} = \frac{n_{ij}}{|C_i|},$$

where n_{ij} is the number of instances shared by the cluster and the class. Then, F_1^{ij} is calculated in the standard way, and the F-measure of the whole clustering is derived by weight-averaging over all classes the maximal F_1 values obtained for all clusters:

$$\mathrm{F} = \sum_i \frac{|C_i|}{n} \max_j(\mathrm{F}_1^{ij}).$$

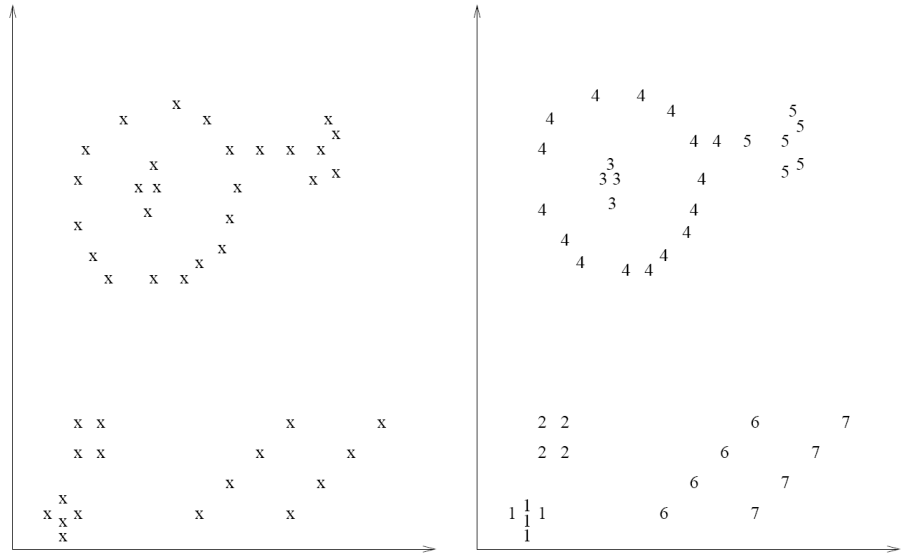
In case of hierarchical clustering (see page 53), the maximum is taken over all clusters at all levels.

As an internal quality measure, *overall similarity* first calculates intra-cluster similarity for cluster j as

$$\frac{1}{|R_j|^2} \sum_{\mathbf{x},\mathbf{y}} \mathit{sim}(\mathbf{x}, \mathbf{y}),$$

where $\mathbf{x}$ and $\mathbf{y}$ are examples from the cluster, and then a sum weighed by $|R_j|$ values is taken to form the overall measure.

Silhouette coefficients (SC) combine the notions of intra-cluster ("within") and inter-cluster ("between") distance, and are computed as follows. For the ith point, a_i is its average distance to all points in its cluster (a_i corresponds to intra-cluster distance), whereas b_i is the minimum average distance to points of other clusters (b_i corresponds to inter-cluster distance). The SC of the ith point is $(b_i - a_i)/\max(a_i, b_i)$, ranging between -1 and 1 (higher values are preferred). The SC of a set of points is obtained by averaging the silhouette coefficients of the individual points.



(a) (b)

Figure 2.11: A challenging example for many data clustering algorithms
Slika 2.11: Izazovan primer za mnoge algoritme klasteringa podataka

2.5.3 Discussion

Most of the points that were discussed in Section 2.3.5 about classification, also apply to clustering. Properties of the data set play an even more important role, since clustering techniques attempt to describe some of these properties, but often without external guidance in the form of class labels. There is no "universal" clustering algorithm which is able to discover the structure of every conceivable data set. Consider the simple example shown in Figure 2.11 (from [92]), where data is shown on chart (a), and an "ideal" clustering, with instances labeled by their corresponding cluster numbers, on chart (b). Not all clustering techniques can uncover all the clusters presented here with equal facility [92].

The problem with high-dimensional data is that it may not be easy to visualize the results of clustering, for an expert to evaluate. Even methods like SOM, which provide an explicit projection into a low-dimensional space, do not guarantee that the visualization will depict anything useful. As for automatic evaluation measures, even when they report good results there is a possibility that valuable structural information was missed, unlike the evaluation of classifiers which gives pretty good ideas about classifier performance. A further contrast to the classification community is the lack of consensus on standard evaluation corpora, which makes the comparison of clustering algorithms more difficult.

Despite the problems with evaluation, most of the presented clustering techniques are known to perform well on text. Bisecting K-means (a hierarchical method) outperformed basic K-means and agglomerative hierarchical clustering in one study [201]. Despite that, the basic K-means algorithm is attractive because of its runtime efficiency. If clustering is to be performed in an online fashion, probabilistic methods are most easily employed, and K-means is simple to adapt as well.

Regarding time-series clustering, different variants of K-means and agglomerative hierarchical clustering have been used with success, as well as various fuzzy approaches [126].

2.6 Outlier Detection

Outliers (also termed *anomalies*, *deviations*, *exceptions*) refer to data objects that are in some sense different from the majority of other objects. In the often-quoted definition by Hawkins [203]: "An outlier is an observation that differs so much from other observations as to arouse suspicion that it was generated by a different mechanism." This definition encompasses one possible cause for a data instance to become an outlier – being of a different class (that is, nature) than the rest of the data.[9] Other causes for the appearance of outliers include natural variability in data (in this case an outlier is of the same class as the rest of the data, but may nevertheless be interesting), and measurement errors (in which case outliers may be considered as noise, and removed prior to applying another data-analysis technique).

Detecting outliers can therefore be a useful task in many fields which employ data-mining techniques. These include [203]: *fraud detection* (purchasing patterns of a credit card thief are usually different from those of a regular user), *intrusion detection* (monitoring the behavior of computer system and network users to detect attacks and unsolicited gathering of information), *medicine* (an outlying test result may signify illness), etc.

To illustrate in an intuitive way the appearance of outliers in data, Figure 2.12 (originally from [21]) depicts two clusters of points, C_1 and C_2, and two outlying points, o_1 and o_2. Point o_1 is a clear outlier, since it is far away from both clusters of points. On the other hand, in terms of average distances between points in cluster C_1, it can be said that point o_2 is close to cluster C_2 and therefore not an outlier. However, relative to average distances between points in cluster C_2, point o_2 is an outlier, since it is much farther away from all points in C_2 than any other point from the cluster.

From the above discussion of causes and applications of outliers and their detection, it is clear that there exist many different standpoints for judging the degree of "outlierness" of a given data point. This, in turn, has resulted in a wide variety of approaches to outlier detection. The rest of this section will briefly review the techniques for outlier detection belonging to three major families: statistical, distance-based, and density-

[9] For example, a fraudulent credit card user is of a different class than a regular user.

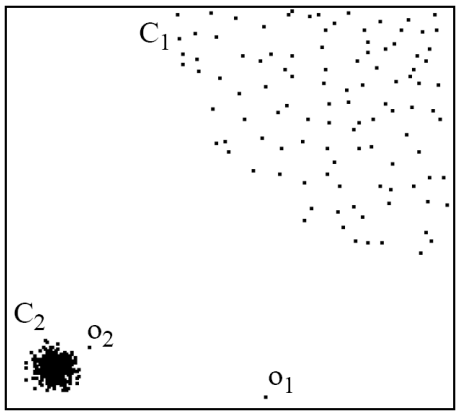


Figure 2.12: Outliers in two-dimensional data
Slika 2.12: *Outlier*-i u dvodimenzionalnim podacima

based. For a more detailed overview of outlier-detection techniques and applications, we refer to the recent survey by Tan et al. [32].

Statistical model-based outlier detection. The statistical approach to outlier detection assumes a particular distribution or probabilistic model for a data set, and determines outliers using a *discordancy test*. For each data point, the working hypothesis that the point is generated by the distribution model assumed for the entire data set, is tested against the alternative hypothesis that the point originates from another distribution model. Depending of the assumed distribution model, there exist numerous methods for performing discordancy testing [203, 81].

Although the statistical approach is theoretically sound, various issues may limit its applicability in practice. One issue pertains to the selection of a correct particular distribution for a data set, which in some situations can be difficult, or even impossible to achieve. Another issue is that discordancy tests for multivariate data are not as well developed as tests for the univariate case, impairing the use of statistical methods on high-dimensional data sets. Furthermore, real data usually originates from a mixture of distributions, making the statistical models more difficult to use and understand [203].

Distance-based outlier detection. The distance-based approach views an outlier as a point which is distant from most other points according to a given distance measure. Depending on distance measurements, every point in a data set is assigned an *outlier score*, and a certain number of points with the highest score considered as outliers.

There are several ways by which distances can be used to assign outlier scores to data points. One of the simplest is to take as the outlier score the distance of a point to its kth nearest neighbor [203]. Another method is to determine the outlier score of a

point as the percentage of other points from the data set which lie at a distance greater than some threshold distance $dmin$ from the point [81].

Generally, distance-based outlier-detection methods are simple, easy to implement, and can produce good results. On the other hand, they may be computationally expensive on large data sets, since they usually require the computation of all pairwise distances. Indexing methods can be used to mitigate this problem, however these tend to work well only on low or moderately-dimensional data spaces. Furthermore, distance-based methods cannot handle data sets with regions of highly differing densities, since global thresholds like k and $dmin$ cannot take into account density variations [203], which may result in whole (low-density) clusters being considered as outliers, or in failure to detect outliers in high-density regions.

Density-based outlier detection. In density-based outlier detection, the outlier score of a given point is determined as the inverse of the density around the point. The density-based approach is related to the distance-based approach, since densities are generally computed through use of distance measurements.

As with the distance-based approach, there exist various ways to express the density of a point (that is, density surrounding the point). One way is to compute density as the inverse of the average distance from a point to its k nearest neighbors. Another way is to take density to be the number of data points within a specified radius $dmax$ from the point.

Besides being sensitive to the choices of parameters k or $dmax$, both notions of density given above suffer from the same drawbacks as distance-based approaches, particularly with regards to the inability to handle regions of varying density. For this reason, the notion of *relative density* is useful, which may be computed by dividing the density of a point with the average density of its surrounding points (for example, its k nearest neighbors). By using relative instead of absolute densities, outliers from regions of differing can be correctly identified. A well-known algorithm that uses a more elaborate notion of relative density, which is still consistent with the presented intuition, is the *local outlier factor* (LOF) algorithm [21].

2.7 Information Retrieval

Information retrieval (IR) refers to a vast field of research and practice which may be defined as "finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within large collections (usually stored on computers)" [38]. Over the span of several decades, IR has moved from a somewhat arcane activity performed by "specialist searchers" (for example, librarians), and studied by a small community of dedicated researchers, to an everyday task performed by computer users of all profiles and backgrounds by virtue of popular Web search engines like Google and Yahoo.

From the large body of tasks and techniques that form the field of IR, this section will review the document weighting schemes and scoring (similarity) metrics pertaining to the task of *ad hoc* retrieval [74, 38]. Ad hoc retrieval involves a (relatively) static document collection, which is searched in order to find documents that are most relevant to a user query, and presented to the user in a ranked order. Prior to search, documents in the collection are indexed, that is, converted to a data structure which facilitates fast retrieval of documents that exhibit high scores when compared with the query. For details on indexing, as well as other IR tasks and methods not covered in this section, the interested reader is referred to the recently published introductory textbooks by Grossman and Frieder [74], and Manning et al. [38].

2.7.1 The Vector Space Model

In the vector space model (VSM), first introduced by Salton et al. [185], both documents and queries are represented as vectors in a high-dimensional space, in the manner that was already described in Section 2.1.1 as the bag-of-words representation. Section 2.1.1 also discusses many of the basic term-weighting schemes used in VSM [196, 38]. However, the vector space model allows the weighting schemes used for documents and queries to be different, and chosen independently from one another.

The function used to assign scores and rank documents with respect to a query can be the cosine similarity measure described in Section 2.2.5. A more general approach is to use only the *dot product* to express similarity: for two vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$, the dot product is defined as

$$\mathrm{dotprod}(\mathbf{x}, \mathbf{y}) = \langle \mathbf{x}, \mathbf{y} \rangle = \sum_{i=1}^{d} x_i y_i \,.$$

By allowing *cosine normalization*, that is, division of every vector component by the norm of the vector, to be performed in the term-weighting phase, dot-product similarity can be made equivalent to the cosine, providing that both documents and queries are cosine-normalized prior to similarity computation. On the other hand, this separation allows other normalization schemes different from cosine normalization to be performed, which are integrated, for example, into the advanced weighting schemes described in Section 2.7.2. Some form of normalization needs to be introduced primarily to prevent long documents from being retrieved too often [196, 38].

2.7.2 Advanced Representations

This section will briefly review two of the most prominent advanced term-weighting schemes, Okapi BM25 and pivoted cosine, both of which take additional factors into account when constructing term weights. Both schemes can be viewed as assigning separate weights to documents and queries, and then employing the dot product to measure similarity between them.

Okapi BM25

The Okapi information-retrieval system has exhibited good empirical performance over the course of several Text Retrieval Conferences (TREC) [180, 179]. The BM25 weighting scheme, first implemented in the Okapi system and developed through a probabilistic theoretical framework, includes parameters that may be tuned to the specific retrieval task and collection in order to maximize performance. Providing that n is the total number of documents in the collection, df the term's document frequency, tf the term frequency, dl the document length (the total number of terms), and $avdl$ the average document length, in the basic version of the scheme, term weights of documents are given by

$$\log \frac{n - df + 0.5}{df + 0.5} \cdot \frac{(k_1 + 1)tf}{k_1((1 - b) + b\frac{dl}{avdl}) + tf},$$

while the term weights of queries are

$$\frac{(k_3 + 1)tf}{k_3 + tf},$$

where k_1, b, and k_3 are tunable parameters. The recommended values for the parameters are $k_1 = 1.2$, $b = 0.75$, and $k_3 = 7$. For additional information on the BM25 weighting scheme, including motivation and theoretical justification, see [200, 178].

Pivoted Cosine

Motivated by providing a better correspondence between the distribution of document length and document relevance, Singhal et al. [197, 198, 196] introduced several variants of the *pivoted cosine* weighting scheme. Using the notation from the previous subsection, in the variant from [198], term weights of documents (the *dtb* weighting [198]) are given by

$$\frac{\text{log1}(tf) \cdot \log\left(\frac{n+1}{df}\right)}{s + (1 - s)\frac{dl}{avdl}},$$

while the term weights of queries (the *dtn* weighting [198]) are

$$\text{log1}(tf) \cdot \log\left(\frac{n + 1}{df}\right),$$

where $\text{log1}(tf) = 1 + \log(1 + \log(tf))$ if $tf \neq 0$, otherwise $\text{log1}(tf) = 0$. The recommended value for the tunable parameter s is 0.8.

2.7.3 Evaluation of IR Systems

In order to evaluate an information-retrieval system, a test collection is needed, consisting of three main components [38]:

		Retrieved? yes	Retrieved? no
Relevant?	yes	True Positive (TP)	False Negative (FN)
	no	False Positive (FP)	True Negative (TN)

Table 2.4: Outcomes of information retrieval
Tabela 2.4: Rezultati *information retrieval*-a

1. A document collection,
2. A set of queries,
3. A set of relevance judgements, usually a binary assessment of *relevant* or *non-relevant* for each document-query pair.

Alternatively, labeled corpora typically used for evaluation of classification (see Section 2.3.3) can be employed in the context of IR, by making relevance judgements through label match and mismatch.

Two of the most common evaluation measures for IR, already discussed in Section 2.3.3 in the context of classification, are precision and recall. *Precision* is defined as the ratio of the number of relevant documents that were retrieved, and the total number of retrieved documents, while *recall* is the ratio between the number of relevant documents retrieved, and the total number of relevant documents. In terms of possible outcomes of retrieval with regards to one query, summarized in Table 2.4 (which is analogous to Table 2.3), precision and recall are computed as:

$$precision = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}}\,,$$

$$recall = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}\,.$$

To obtain precision and recall values for a given set of queries, the measurements for individual queries are averaged. For a fixed number of retrieved documents k, precision is also referred to as *precision at* k. As in Section 2.3.3, precision and recall can be combined to form the *F-measure*:

$$\mathrm{F}_\beta = \frac{(\beta^2+1)\cdot precision \cdot recall}{\beta^2 \cdot precision + recall}\,.$$

The measures described above are essentially set-based measures, and may not be entirely suitable for evaluation of *ranked* retrieval. In addition, they may depend on a predetermined number of retrieved results k. For this reason, several aggregate evaluation measures have been proposed, including the precision-recall curve, point-wise average precision, and mean average precision (MAP).

Plotting the precision of a set of queries at every recall level produces the basic *precision-recall curve*. Because of its jagged shape, interpolation can be performed to

obtain a smooth decreasing curve, referred to as the *interpolated precision-recall curve*. The interpolated precision, p_{interp}, at a specified recall level r, is defined as the highest precision for all recall levels r' higher than r [38]:

$$p_{interp}(r) = \max_{r' \geq r} precision(r') \,.$$

Recall levels for the interpolated precision-recall curve can be taken only at specified equidistant values between 0 and 1; typically 11 values are used: $r \in \{0, 0.1, 0.2 \ldots 1\}$. Moreover, by taking the average of the precision values for the specified recall values results in *point-wise average precision*. For the typical recall values given above, the measure is referred to as *11-point average precision*.

Finally, a common measure that provides a single aggregate value of precision across all recall levels is *mean average precision (MAP)*. For a given set of queries Q, and query $q_j \in Q$, let m_j denote the number of retrieved documents, and R_{jk} the set of first k ranked results. Then,

$$\mathrm{MAP}(Q) = \frac{1}{|Q|} \sum_{j=1}^{|Q|} \frac{1}{m_j} \sum_{k=1}^{m_j} precision(R_{jk}) \,.$$

The MAP measure roughly approximates the area under the uninterpolated precision-recall curve [38].

2.8 Dimensionality Reduction

One of the principal approaches for tackling the problems caused by the curse of dimensionality (see Chapter 1) is by application of dimensionality-reduction techniques to transform the data to a lower-dimensional representation.

Regarding text data in the bag-of-words representation, many preprocessing steps can be considered a limited form of dimensionality reduction. Such steps include the elimination of digits and special characters, and removal of words which appear too infrequently or too frequently in the document set with respect to some predefined thresholds (for example, excluding all words which appear in less than three or more than half of all documents). The removal of words which are too frequent (for example, "I," "the," "with," etc.) is also often done with regards to a predefined list of stop words. Such words generally inhibit ML, DM, and IR algorithms because they do not add useful information to the BOW model (for a majority of applications). A practically standard stop-word list for English is the one that was used in the SMART document retrieval system [184].

Stemming can be viewed as another technique for dimensionality reduction of text data. It transforms all forms of a word to the same stem, like "computer," "computing," and "computational" to "comput." Therefore, not only does it reduce the number of features, but it also captures correlations between some words by fusing them, that way

possibly improving the performance of certain techniques. The problem of algorithmically determining the stem of an arbitrary English word is satisfactorily solved for most applications, one of the widely used algorithms being the Porter stemmer [158]. However, for many languages in which words inflect more than in English, such solutions are not possible, and this problem may itself be tackled by machine-learning techniques.

Regarding time-series data, the preprocessing technique of scaling, such as uniform scaling [65], can reduce (but also extend) the length (that is, the number of dimensions) of a time series.

The above preprocessing methods are generally not expected to greatly reduce the dimensionality of text or time-series data, or effectively capture the intrinsic dimensionality. Numerous techniques for dimensionality reduction, generally independent of the type of data, have been proposed to date, and the following sections will review those methods relevant to the research presented in later chapters of this dissertation.

The problem of how to reduce dimensionality can be approached from two distinct angles: feature *selection*, where the resulting set of features is a subset of the original one, and feature *extraction*, which derives a new set of features of smaller cardinality.

2.8.1 Feature Selection

The question which feature selection tries to answer is this: for a given set of d features used to represent data, which of its 2^d subsets to choose in order to achieve optimal performance in a given task.

The Wrapper Approach

When class labels are present and the task in question is classification, a simple brute-force method which imposes itself consists of testing a classifier with every possible subset of the initial set of features, and selecting the one that performs best. Such an approach where a classifier is used to evaluate feature subsets is called the *wrapper* approach. Unfortunately, the method described above may be prohibitively slow even on small data sets, therefore different strategies are employed to reduce the search space. For example, starting with an empty (full) set of features, every possible feature may be added (removed) one at a time, the classifier trained and tested (see Section 2.3.3), and the best feature to add (remove) chosen. Even with this modification the wrapper approach may be too costly for high-dimensional data such as text, thus computationally less intensive methods, which do not rely on classifiers, are employed more often.

The Filter Approach

The *filter* approach attempts to determine the importance of a feature based on some measure which is relatively simple to compute. In the supervised setting, a feature is considered more "important" if it strongly correlates with the class feature (it is relevant), at the same not correlating with other features (it is not redundant). There exist

many ways to formalize this notion, including *information gain*, *gain ratio*, *symmetrical uncertainty*, *chi square*, *relief*, and, for textual data, *term frequency*.

Information-theoretic measures. Several useful feature-ranking measures originate from information theory, including *information gain*, *gain ratio*, and *symmetrical uncertainty*. The number of bits needed to express event x_i, which occurs with probability $\mathrm{P}(x_i)$, is called *information*, expressed as $\mathrm{I}(x_i) = -\log_2 \mathrm{P}(x_i)$. The expected value of I for a random variable containing events x_i is *entropy*:

$$\mathrm{H}(X) = \sum_i \mathrm{P}(x_i)\,\mathrm{I}(x_i) = -\sum_i \mathrm{P}(x_i)\log_2 \mathrm{P}(x_i),$$

and the conditional entropy of X after observing Y:

$$\mathrm{H}(X|Y) = -\sum_j \mathrm{P}(y_j)\sum_i \mathrm{P}(x_i|y_j)\log_2 \mathrm{P}(x_i|y_j).$$

The reduction in entropy of X before and after observing Y, that is, the average amount of information about X contained in Y, is referred to as *expected cross entropy* [137]:

$$\mathrm{CH}(X,Y) = \mathrm{H}(X) - \mathrm{H}(X|Y).$$

If we consider features as random variables, then the expected cross entropy of a fixed class attribute C and attribute A is known as the *information gain* of A:

$$\mathrm{IG}_C(A) = \mathrm{CH}(C,A) = \mathrm{H}(C) - \mathrm{H}(C|A).$$

The probabilities are calculated from a given data set, thus entropy can be viewed as a measure of (im)purity of the data set relative to the classification we wish to achieve [136] (p. 55). The usual approach is to rank every feature with regards to the class using the IG criterion and choose the best features. Note that the IG measure takes into account only the correlations of a feature with the class feature, ignoring its dependencies with other features.

One possible problem of information gain is its bias towards features with more values. This may be fixed by normalizing IG with the entropy of A, yielding *gain ratio*:

$$\mathrm{GR}_C(A) = \mathrm{IG}_C(A)/\,\mathrm{H}(A).$$

Another approach is used in the *symmetrical-uncertainty* measure:

$$\mathrm{SU}_C(A) = 2\mathrm{IG}_C(A)/(\mathrm{H}(C) + \mathrm{H}(A)).$$

The values of GR and SU lie between 0 and 1, with 0 meaning no correlation, and 1 denoting full. Since $\mathrm{H}(X) - \mathrm{H}(X|Y) = \mathrm{H}(Y) - \mathrm{H}(Y|X)$, IG and SU are considered symmetrical, because it is irrelevant which variable is observed and which one is ranked – the correlation works both ways. For GR this is clearly not the case. More details on these measures can be found in [78] and [137] (p. 42).

Chi-square. The χ^2 measure from statistics can also be used for estimating the correlation between features and the class feature. If n is the size of the data set, for the simplest binary version of BOW, where attribute $A \in \{a_0, a_1\}$, and binary classification ($C \in \{c_0, c_1\}$), the χ^2 metric is

$$\mathrm{CHI}_C(A) = \frac{n\big[\mathrm{P}(a_0, c_0)\,\mathrm{P}(a_1, c_1) - \mathrm{P}(a_0, c_1)\,\mathrm{P}(a_1, c_0)\big]^2}{\mathrm{P}(a_0)\,\mathrm{P}(a_1)\,\mathrm{P}(c_0)\,\mathrm{P}(c_1)}\,.$$

Relief. A different approach to feature ranking is used by the *relief* measure first introduced by Kira and Rendell [107]. Relief takes a random example from the data set and locates two of its nearest neighbors (with regards to some vector distance metric), one from the positive and one from the negative class, and uses the values of their features to update the relevance of each feature. The procedure is repeated a specified number of times, the more the better (sampling *every* example if possible). Relief was later extended to ReliefF (RF), with added support for multi-class and noisy data sets [110]. It handles noise by taking k nearest neighbors from the positive and negative class and averaging them. More information on the RF algorithm can be found in [77].

Term frequency. When dealing with text data in the bag-of-words representation, the number of documents a feature (term) occurs in, called *term frequency* (which we shall denote TFDR to differentiate term frequency as a dimensionality-reduction method from term frequency as a feature-weighing method in the BOW representation), may be a surprisingly effective way to rank features for selection. A variation of this measure is to count all occurrences of a term in the whole document set. Note that the removal of stop words is an important preprocessing step to take before using TFDR, otherwise many useless features will be retained (unless stop words are important in the particular application of the classifier).

A graphical representation. In a comprehensive study of feature-selection methods for text classification [57], Forman gives a graphical analysis of the methods' decision boundaries. The curves in Figure 2.13 are plotted with axes representing the numbers of positive and negative documents containing a word. The words inside the areas enclosed within the boundaries are to be eliminated for being shared by too many positive and negative documents, meaning that they are determined to be least discriminative. The graph includes IG and CHI metrics, as well as some others: *document frequency* (DFreq), *odds ratio* (OR), *bi-normal separation* (BNS), and *probability ratio* (PR). All metrics are set up to select exactly 100 features on the used Cora data set (see page 49), with the graph showing just how different (or similar) their strategies are.

In the context of text categorization, using some of the described measures and several similar ones, Yang and Pedersen have shown that feature selection can reduce the number of features by 90% up to 99% without significant loss of performance [227].

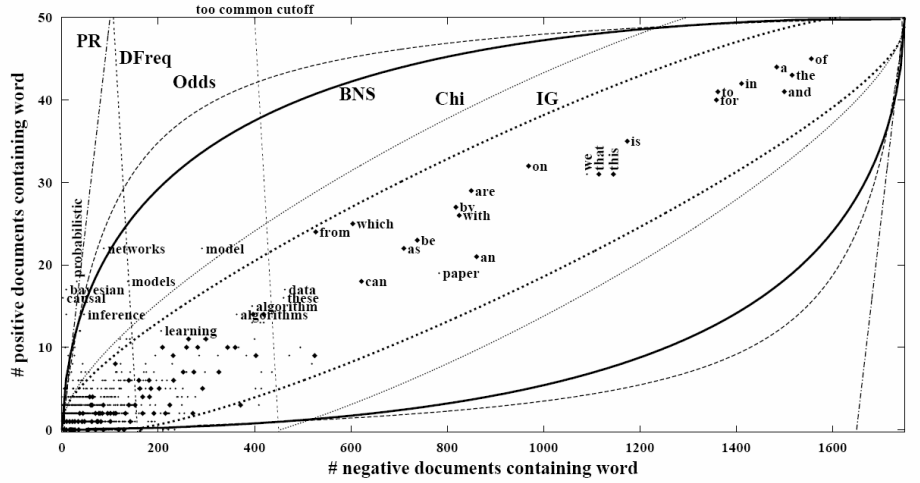


Figure 2.13: Decision boundary curves for various feature-selection methods
Slika 2.13: Granične krive različitih metoda za odabir atributa

Furthermore, some combinations of measures, reduction rates, and classifiers exhibited a performance increase over full feature sets.

Many of the aforementioned methods can be used in an unsupervised setting, to select features for later application of clustering – not by correlating them with the class feature, but only amongst themselves. However, high dimensionality of data generally makes such straightforward application infeasible. Several unsupervised feature-selection methods for text clustering have been compared in the study presented in [129], concluding that unsupervised methods perform worse than supervised ones, when the latter are applicable. The study introduced an approach that iteratively combines clustering and supervised feature selection (which considers generated clusters as classes). Still, the main difficulties of feature selection for (text) clustering lie in the question of how to assess feature relevance and relate it to clusters, and in the lack of a standard evaluation methodology (see Section 2.5.2).

2.8.2 Feature Extraction

Unlike feature selection, feature *extraction* (often referred to as dimensionality reduction in its own right) is concerned with engineering a completely different set of features based on the training data. The resulting features may seem completely counterintuitive when observed directly, but what is important is that they accommodate the technique that is to be applied to the transformed data set.

Generally, feature-extraction methods attempt to reduce the data to a space of lower dimensionality, while preserving valuable "information." What exactly is considered as

"information" gives rise to a plethora of different approaches and techniques for feature extraction – examples of "information" include variance in the data, pairwise distances, separability of classes, etc.

An important distinction of methods for feature extraction is into *linear* and *nonlinear* techniques. Linear feature-extraction techniques seek to find a transformation matrix G, which when multiplied with the data matrix X produces a matrix representing data points in a new space: $Y = XG$.[10] The number of columns of G can be significantly smaller than the number of columns of X, producing a data representation in Y of much lower dimensionality. The transformation matrix G is derived from the data matrix X, but may be applied in the same way to (matrices of row) vectors which were not used in the derivation. Nonlinear techniques, on the other hand, produce data transformations which cannot be expressed through multiplication with a single matrix. For other categorizations of feature-extraction techniques, see [117, 211].

We will describe the following approaches to feature extraction: *singular value decomposition (SVD)*, *principal component analysis (PCA)*, *independent component analysis (ICA)*, *multidimensional scaling (MDS)*, *stochastic neighbor embedding (SNE)*, *Isomap*, *diffusion maps*, the *discrete Fourier transform (DFT)*, and the *discrete wavelet transform (DWT)*.

Singular value decomposition. Singular value decomposition (SVD) is a linear algebra matrix decomposition technique which can be successfully applied for feature extraction. The SVD of matrix $X = U\Sigma V^T$, where the columns of U are orthogonal eigenvectors of XX^T, the columns of V are orthogonal eigenvectors of X^TX, and Σ is a diagonal matrix of singular values – the square roots of eigenvalues of XX^T. For SVD reduction, matrix G can be obtained by taking the columns of V that correspond to largest singular values from Σ [203].

Application of SVD to text data in the BOW representation has an intuitive interpretation – the new features correspond to combinations of original terms and represent semantic "concepts" ("topics") that were derived from co-occurrence relations of terms in different documents. In this context, the technique is generally referred to as *latent semantic indexing* (LSI). The main strength of LSI is the ability to encapsulate the small effects of many features which would have been considered redundant by feature-selection methods, but whose cumulative effect may be really relevant for a given task, like classification. The drawback lies in the opposite extreme – if there are only a few highly relevant features, their importance may be lost in the transformation [190].

Principal component analysis. The main idea behind principal component analysis (PCA) is that high information corresponds to high variance [15]. PCA is a linear technique, which relies on the observation that for a given data distribution, the direction of maximum variance is parallel to the eigenvector corresponding to the largest eigenvalue

[10] We assume that in data matrices, for example, $X_{n\times d}$, rows correspond to n data vectors, while columns represent d features. For individual vectors, for example, $\mathbf{x}$, we assume the column format $d \times 1$.

of the covariance matrix of the data distribution. Furthermore, of all directions orthogonal to the direction with the highest variance, the second highest variance is in the direction of the eigenvector corresponding to the second largest eigenvalue, and so on. In practice, sample covariance $\mathrm{Cov}(X)$ obtained from the data matrix X is used, since the true data distribution is not known.

If the eigenvalues of $\mathrm{Cov}(X)$ are $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_r \geq 0$, and their corresponding eigenvectors are $\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_r$, then the transformation matrix G used to obtain the transformed data $Y = XG$, is given by $G = [\mathbf{e}_1\ \mathbf{e}_2\ \ldots\ \mathbf{e}_r]$. The eigenvectors are referred to as *principal axes*, and are selected to be orthogonal and of length 1.

A useful property of the eigenvalues is that their diagonal matrix forms the sample covariance of transformed data Y: $\mathrm{Cov}(Y) = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_r)$. Therefore, the number of columns of G used to transform X, that is, the dimensionality m of the resulting data space, can be selected to account for a specified proportion of variance:

$$\frac{\sum_{i=1}^{m} \lambda_i}{\sum_{i=1}^{r} \lambda_i}.$$

For example, to account for 95% of variance, m should be selected to be the smallest such number so that the above ratio is greater than or equal to 0.95.

Independent component analysis. Independent component analysis (ICA) was originally developed to deal with problems related to the cocktail-party problem: reconstructing original speech signals from multiple speakers in a room from recordings made by microphones positioned at different places throughout the room [88]. The assumption is made that the speakers, that is, the true *sources* of the signals, are statistically independent. Using vector and matrix notation, if $\mathbf{x}$ denotes the multidimensional vector random variable corresponding to the observed data (for example, the microphone recordings), and $\mathbf{s}$ the random variable of the sources, the mixing model may be written as $\mathbf{x} = A\mathbf{s}$.

The main problem in ICA is recovering the mixing matrix A from the observations provided by $\mathbf{x}$. Assuming that $\mathbf{x}$ has been *whitened* to have its covariance matrix equal to the identity matrix (which can be achieved using SVD), and that $\mathbf{s}$ has identity covariance matrix (since the sources are statistically independent), it follows that A is orthogonal. Therefore, solving the ICA problem involves finding an orthogonal A such that the components of the vector random variable $\mathbf{s} = A^T\mathbf{x}$ are independent (and non-Gaussian) [82]. Once A and $\mathbf{s}$ have been recovered, the reduced representation of data can be obtained by restricting the number of components of $A\mathbf{s}$.

Multidimensional scaling. The term multidimensional scaling (MDS) refers to a family of nonlinear techniques which aim to preserve pairwise distances between points in a data set. One categorization of MDS methods is into *metric* and *nonmetric*. Metric MDS deals with quantitative distance measurements (for example, Euclidean distances), while nonmetric MDS operates with ordinal relations between data points. Another categorization differentiates *classical* MDS (CMDS), which deals with a single

distance matrix, from more advanced variants like *replicated* MDS and *weighted* MDS, which permit more than one distance matrix to be defined on the data [232].

For given data points $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$, classical MDS attempts to find their low-dimensional representation $\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_n$ which minimizes a cost function that expresses the degree of preservation of pairwise distances in the new representation. A commonly used cost function is the raw stress function, defined as $\sum_{ij}(dist(\mathbf{x}_i, \mathbf{x}_j) - dist(\mathbf{y}_i, \mathbf{y}_j))^2$, where $dist$ usually denotes Euclidean distance.

Stochastic neighbor embedding. Similarly to MDS, stochastic neighbor embedding (SNE) attempts to preserve pairwise distance between data points, putting more emphasis on distances between nearby points [86]. However, SNE uses a stochastic notion of distance, and with it a different formulation of the cost function.

Let p_{ij} denote the probability that data points $\mathbf{x}_i$ and $\mathbf{x}_j$ originate from the same Gaussian distribution. For every i, j, these probabilities are computed and stored in matrix P. Conversely, let q_{ij} denote the probability that $\mathbf{y}_i$ and $\mathbf{y}_j$ originate from the same Gaussian, stored in matrix Q. SNE minimizes the difference between distributions P and Q, measured by Kullback-Leibler divergence: $\sum_{ij} p_{ij} \log \frac{p_{ij}}{q_{ij}}$.

Isomap. If data points lie on a curved low-dimensional manifold like the "swiss roll" data set, MDS may consider two points to be close when they are actually far away within the underlying manifold. For this reason, Isomap [207] attempts to preserve *geodesic* distances between data points. Geodesic distances are obtained by constructing a k-nearest neighbor graph of the data with edges weighted by distances between points, and taking the sum of edge weights along the shortest path between two points in the graph. To obtain the low-dimensional representation, multidimensional scaling is then applied to the pairwise geodesic distances.

Diffusion maps. Diffusion maps [115, 140], which originate from the field of dynamical systems, are based on defining a Markov random walk on the graph derived from the data. The edge weights of the graph are assigned using the RBF (Gaussian) kernel function of Euclidean distances between data points:

$$w_{ij} = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}\right).$$

The random walk, repeated multiple times, produces a measure of distance between data points which may capture manifold structure in the data. The main goal of the diffusion-maps method is to preserve these *diffusion distances* in the low-dimensional data representation.

Discrete Fourier transform. The discrete Fourier transform (DFT) is a technique primarily used for representation and dimensionality reduction of signal-like data, such as time series [56]. DFT transforms a series from the time domain to the frequency

domain by expressing it with coefficients in a space of complex exponential functions that represent sinusoidal functions in the real domain. For a given series vector $\mathbf{x} = (x_1, x_2, \ldots, x_d)$, DFT produces complex vector $\mathbf{z} = (z_1, z_2, \ldots, z_d)$ that satisfies

$$z_k = \sum_{j=1}^{d} x_j \omega_n^{(j-1)(k-1)},$$

for every $k \in \{1, 2, \ldots, d\}$, where $\omega_n = \exp(-2\pi i/n)$. To reduce dimensionality, only a certain number of consecutive coefficients of $\mathbf{z}$ beginning with the first can be kept, since they carry most of the information about $\mathbf{x}$.

Discrete wavelet transform. The discrete wavelet transform (DWT) is another technique that can be used for dimensionality reduction of signal and time-series data [31]. DWT projects series $x(t)$ into the time-frequency domain whose basis functions are given by

$$\Psi_{j,k}(t) = 2^{j/2}\Psi(2^j t - k),$$

for integer numbers j and k, which are respectively referred to as indices of dilatation and translation, while Ψ (on the right-hand side) is called the *mother wavelet function*. Series $x(t)$ can be represented using the basis as

$$x(t) = \sum_{j,k \in \mathbb{Z}} a_{j,k} \Psi_{j,k}(t),$$

where $a_{j,k}$ are referred to as DWT coefficients of $x(t)$, expressed by

$$a_{j,k} = \langle \Psi_{j,k}(t), x(t) \rangle = \int_{-\infty}^{\infty} \Psi_{j,k}(t) x(t) dt.$$

A commonly used wavelet is the Haar wavelet, with the following mother function:

$$\Psi(t) = \begin{cases} 1, & 0 \leq t < 1/2, \\ -1, & 1/2 \leq t < 1, \\ 0, & \text{otherwise.} \end{cases}$$

2.9 Summary

The aim of the survey presented in this chapter was to describe the some of the important principles and techniques for machine learning, data mining, and information retrieval, with the emphasis on applications to high-dimensional data such as text and time series. Regarding text data, the survey was concerned with the "low-levels" of application – the basic transformations of raw (hyper)text to a different representation, and the algorithms applicable to such data. Nevertheless, multiple layers of application of ML or DM algorithms, that is, employing ML/DM techniques on data derived

using ML/DM techniques, may eventually be unavoidable in practice. For example, in many languages stemming may need to be handled using ML methods, because it is not feasible to solve the problem algorithmically (to a useful degree of accuracy) as is for English. The wrapper methods of feature selection are further examples. As for time-series data, common applications include both working with raw time series, and time series transformed to a different representation my means of, for example, dimensionality-reduction methods.

In writing this chapter, care was taken to present only the techniques, without delving too deep into descriptions of particular applications. Nevertheless, the key to understanding the behavior of techniques in practice is to examine how a concrete problem was solved, and grasp the reasons why particular features, representations, and algorithms were chosen. This chapter provided appropriate mention of various application tasks for the described techniques, and references for further reading. Many of the applications, however, need to handle data spaces of high dimensionality and the problems which arise in them. The following two parts of this dissertation will address two angles relevant to the issues raised by high dimensionality: (1) the behavior of distance (similarity) metrics, and the consequences on key techniques for machine learning, time-series classification, and information retrieval, and (2) feature-selection methods in the context of the text-categorization task.

Part II

Metrics

Chapter 3

The Concentration Phenomenon

The phenomenon of distance concentration refers to one aspect of the dimensionality curse which is manifested by the tendency of distances between all pairs of points in a high-dimensional data set to become almost equal. Distance concentration and the meaningfulness of the notion of nearest neighbors in high dimensions has been thoroughly explored for general distance measures [16, 51], and specifically for Minkowski and fractional distances [44, 85, 2, 61, 59, 87]. The effect of the phenomenon on machine learning was demonstrated, for example, in studies of the behavior of kernels in the context of support vector machines, lazy learning, and radial basis function (RBF) networks [55, 59].

After reviewing and illustrating some of the more important results concerning the concentration of distances (that is, norms) in Section 3.1, we will show, in Sections 3.2 and 3.3, that the cosine similarity measure also concentrates for a wide class of data distributions. Although this result may be interesting in its own right, in this dissertation it will primarily serve as support for the explanation of the hubness phenomenon in the context of cosine similarity and information-retrieval applications (Chapter 7).

3.1 Concentration of Distances

Concentration is typically measured by examining the ratio between some notion of spread and some notion of magnitude of distances between points in a particular setting. If this ratio converges to 0 as dimensionality increases, it is said that distances concentrate. Assuming data set $D \subset \mathbb{R}$ consists of d-dimensional points drawn according to some stochastic mechanism, a popular measure of concentration is *relative contrast*, which is defined as

$$\mathcal{C}_D = \frac{\max_{\mathbf{x} \in D} \|\mathbf{x}\| - \min_{\mathbf{x} \in D} \|\mathbf{x}\|}{\min_{\mathbf{x} \in D} \|\mathbf{x}\|}, \tag{3.1}$$

where $\|\cdot\|$ is an arbitrary norm.[1] The numerator of Equation 3.1 will be referred to as *contrast* [61]. We will assume that points are drawn from a random d-dimensional vector $\mathbf{X}_d$ with iid components.

Another measure of concentration, proposed (that is, made explicit) in [61] is *relative variance*, which is defined for d-dimensional random vector $\mathbf{X}_d$ as

$$\mathcal{V}_{\mathbf{X}_d} = \frac{\sqrt{\mathrm{Var}(\|\mathbf{X}_d\|)}}{\mathrm{E}(\|\mathbf{X}_d\|)}\,. \tag{3.2}$$

An early result which establishes a connection between the two measures of concentration is the following theorem by Beyer et al. [16].[2]

Theorem 1 [16, adapted] *Let $\mathbf{x}^{(j)} : 1 \leq j \leq n$ be n d-dimensional iid random vectors, and let $\|\cdot\|$ denote an arbitrary norm. If*

$$\lim_{d\to\infty} \mathrm{Var}\left(\frac{\|\mathbf{x}^{(j)}\|}{\mathrm{E}(\|\mathbf{x}^{(j)}\|)}\right) = 0, \tag{3.3}$$

then, for any $\epsilon > 0$,

$$\lim_{d\to\infty} \mathrm{P}\left[\frac{\max_j \|\mathbf{x}^{(j)}\| - \min_j \|\mathbf{x}^{(j)}\|}{\min_j \|\mathbf{x}^{(j)}\|} \leq \epsilon\right] = 1. \tag{3.4}$$

The theorem can be interpreted as stating that if, for a set of n d-dimensional points, the condition given by Equation 3.3 is satisfied, then the relative contrast ratio from Equation 3.4 becomes progressively smaller, that is, the difference between the maximal and minimal observed distance to the origin becomes negligible compared to the minimal distance. (The origin can be regarded as a query point without loss of generality [61].) A converse result to this theorem, reversing the direction of implication, is provided by Durrant and Kabán [51]. Based on Theorem 1, Beyer et al. question the very meaningfulness of nearest-neighbor queries in high-dimensional spaces [16], and explore settings which do and do not fulfill Equation 3.3. François et al. [61] prove that this condition holds for iid random data and l_p norms with $p > 0$ (see Theorem 5 at the end of the current section).

Following the findings of Beyer et al. [16], which are relevant to arbitrary distances, Hinneburg et al. [85] prove the following theorem for Minkowski norms.

[1] Most distance-concentration results that will be reviewed in this section (with the exception of [16]) examine only norms, possibly generalizing to pairwise distances by observing difference vectors $\mathbf{x} - \mathbf{y}$ instead of single vectors [61, 59].

[2] Besides examining pairwise distances, the original theorem is formulated in a more general setting that involves an infinite sequence of $m = 1, 2, \ldots$ data distributions, where m can be interpreted as dimensionality and distributions selected in such a way to comprise a sequence of multivariate d-dimensional distributions with iid components.

Theorem 2 [85, adapted] *Let $\mathbf{x}^{(j)} : 1 \leq j \leq n$ be n d-dimensional iid random vectors distributed over $[0,1]^d$, and let $\|\cdot\|_p$ denote the Minkowski norm with exponent p. There exists a constant C_p independent from the distribution of $\mathbf{x}^{(j)}$ such that*

$$C_p \leq \lim_{d\to\infty} \mathrm{E}\left(\frac{\max_j \|\mathbf{x}^{(j)}\|_p - \min_j \|\mathbf{x}^{(j)}\|_p}{d^{\frac{1}{p}-\frac{1}{2}}}\right) \leq (n-1)C_p\,. \tag{3.5}$$

The theorem states that the contrast, contained in the numerator of the fraction in Equation 3.5, asymptotically behaves like $d^{1/p-1/2}$ as d increases. This means that for $p = 1$ (Manhattan norm) the contrast grows like $\sqrt{d}$, for $p = 2$ (Euclidean norm) the contrast remains constant, while for $p \geq 3$ the contrast shrinks to 0. Based on this observation, Hinneburg et al. [85] conclude that nearest-neighbor search in high-dimensional spaces is meaningless for l_p norms with $p \geq 3$, since it becomes increasingly difficult to distinguish the nearest from the farthest neighbor.

Aggarwal et al. [2] provide additional results that consider fractional norms (that is, l_p norms with rational $p \in (0,1)$, see Section 2.2). Please note that the following theorem requires the data to be uniformly distributed.

Theorem 3 [2, adapted] *Let $\mathbf{x}^{(j)} : 1 \leq j \leq n$ be n d-dimensional iid random vectors uniformly distributed over $[0,1]^d$, and let $\|\cdot\|_p$ denote the fractional norm with rational exponent $p \in (0,1)$. There exists a constant C independent from p and from d such that*

$$C\sqrt{\frac{1}{2p+1}} \leq \lim_{d\to\infty} \mathrm{E}\left(\frac{\max_j \|\mathbf{x}^{(j)}\|_p - \min_j \|\mathbf{x}^{(j)}\|_p}{\min_j \|\mathbf{x}^{(j)}\|_p}\right)\sqrt{d} \leq (n-1)C\sqrt{\frac{1}{2p+1}}\,.$$

Based on this result, Aggarwal et al. [2] advocate the use of fractional distances due to their apparently slower concentration than Minkowski distances, and empirically demonstrated robustness to noise. However, this view can be challenged when non-uniformly distributed data is considered, as well as different types of noise [59].

The following results analyze the elements found in the ratio of relative variance (Equation 3.2). Let $\mathbf{X}_d = (X_1, X_2, \ldots, X_d)$ be a random d-dimensional vector with iid components: $X_i \sim \mathcal{F}$.

Theorem 4 [44, adapted]

$$\mathrm{E}(\|\mathbf{X}_d\|_2) = \sqrt{\alpha \cdot d - \beta} + O(1/d), \textit{ and}$$

$$\mathrm{Var}(\|\mathbf{X}_d\|_2) = \beta + O(1/\sqrt{d}),$$

where α and β are constants that depend on the distribution of $\mathbf{X}_d$ but do not depend on its dimensionality.

The theorem states that the expected value of the Euclidean norm asymptotically behaves like $\sqrt{d}$, while the variance remains constant. François et al. [61] prove a more general result, which holds for arbitrary $p > 0$, encompassing both Minkowski and fractional norms.

Theorem 5 [61, Theorem 5, adapted]

$$\lim_{d\to\infty} \frac{\sqrt{\mathrm{Var}(\|\mathbf{X}_d\|_p)}}{\mathrm{E}(\|\mathbf{X}_d\|_p)} = 0\,.$$

The proof of Theorem 5 is based on proving the following two limits, which we will formulate as lemmas. For random variables $|X_i|^p$, $p > 0$, let $\mu_{|\mathcal{F}|^p}$ and $\sigma^2_{|\mathcal{F}|^p}$ denote their mean and variance, respectively.

Lemma 1 [61, Equation 17, adapted]

$$\lim_{d\to\infty} \frac{\mathrm{E}(\|\mathbf{X}_d\|_p)}{d^{1/p}} = \mu_{|\mathcal{F}|^p}\,.$$

Lemma 2 [61, Equation 21, adapted]

$$\lim_{d\to\infty} \frac{\mathrm{Var}(\|\mathbf{X}_d\|_p)}{d^{2/p-1}} = \frac{\sigma^2_{|\mathcal{F}|^p}}{\left(p\cdot\mu^{(p-1)/p}_{|\mathcal{F}|^p}\right)^2}\,.$$

From the two lemmas it can be observed that although the expected value of the norm increases with d for any value of p (albeit at a different rate), the variances behave in various ways: they diverge for $p < 2$, remain constant for $p = 2$, and shrink to 0 for $p > 2$. In all cases, however, the ratio between variance and expectation, expressing the relative variance in Theorem 5, converges to 0 as dimensionality increases.

To illustrate the behavior of l_p norms discussed in this section, Figure 3.1 shows, for various p values and dimensionalities 1–100, from top to bottom: the maximal observed value, mean value plus one standard deviation, the mean value, mean value minus one standard deviation, and minimal observed value of l_p norms computed for $n = 2000$ iid uniformly distributed points.

3.2 Concentration of Cosine Similarity

Distance concentration, as explained in the previous section, refers to the tendency of the ratio between some notion of spread (for example, standard deviation) and some notion of magnitude (for example, the mean) of the distribution of all pairwise distances (or, equivalently, the norms) within a data set to converge to 0 as dimensionality increases. Hereby, we examine concentration in the context of cosine similarity, and provide a proof of this property by considering two random d-dimensional vectors $\mathbf{p}$ and $\mathbf{q}$ with iid components. Let $\cos(\mathbf{p},\mathbf{q})$ denote the cosine similarity between $\mathbf{p}$ and $\mathbf{q}$, defined in Equation 3.6:[3]

$$\cos(\mathbf{p},\mathbf{q}) = \frac{\langle \mathbf{p},\mathbf{q}\rangle}{\|\mathbf{p}\|\|\mathbf{q}\|}\,. \tag{3.6}$$

[3]Henceforth, $\|\cdot\|$ denotes the Euclidean (l_2) norm.

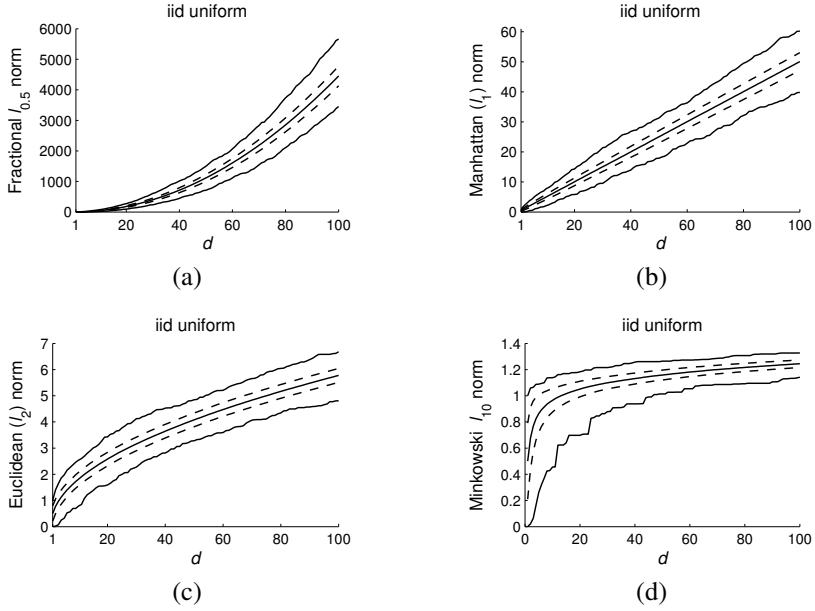


Figure 3.1: Concentration of l_p norms for iid uniform random data: (a) fractional $l_{0.5}$ norm, (b) Manhattan norm ($p = 1$), (c) Euclidean norm ($p = 2$), and (d) l_{10} norm
Slika 3.1: Koncentracija l_p norme za *iid* uniforman skup slučajnih tačaka: (a) razlomljena $l_{0.5}$ norma, (b) Menhetn norma ($p = 1$), (c) euklidska norma ($p = 2$), i (d) l_{10} norma

From the extension of Pythagoras' theorem we obtain Equation 3.7, which relates $\cos(\mathbf{p}, \mathbf{q})$ with the Euclidean distance between $\mathbf{p}$ and $\mathbf{q}$:

$$\cos(\mathbf{p}, \mathbf{q}) = \frac{\|\mathbf{p}\|^2 + \|\mathbf{q}\|^2 - \|\mathbf{p} - \mathbf{q}\|^2}{2\|\mathbf{p}\|\|\mathbf{q}\|} \,. \tag{3.7}$$

Define the following random variables: $X = \|\mathbf{p}\|$, $Y = \|\mathbf{q}\|$, and $Z = \|\mathbf{p} - \mathbf{q}\|$. Since $\mathbf{p}$ and $\mathbf{q}$ have iid components, it follows that X and Y are independent of each other, but not of Z. Let C be the random variable that denotes the value of $\cos(\mathbf{p}, \mathbf{q})$. From Equation 3.7, with simple algebraic manipulations and substitution of the norms with the corresponding random variables, we obtain Equation 3.8:

$$C = \frac{1}{2}\left(\frac{X}{Y} + \frac{Y}{X} - \frac{Z^2}{XY}\right). \tag{3.8}$$

Let $\mathrm{E}(C)$ and $\mathrm{Var}(C)$ denote the expectation and variance of C, respectively. An established way to demonstrate concentration is by examining the asymptotic relation between $\sqrt{\mathrm{Var}(C)}$ and $\mathrm{E}(C)$ when dimensionality d tends to infinity (relative variance,

Equation 3.2). To express this asymptotic relation, we first need to express the asymptotic behavior of $\mathrm{E}(C)$ and $\mathrm{Var}(C)$ with regards to d. Since, from Equation 3.8, C is related to functions of X, Y, and Z, we start by studying the expectations and variances of these random variables. The following theorem adapts Lemmas 1 and 2 by taking $p = 2$ and adjusting the notation.

Theorem 6 [61, adapted]
$\lim_{d\to\infty}\big(\mathrm{E}(X)/\sqrt{d}\,\big) = const$, *and* $\lim_{d\to\infty}\mathrm{Var}(X) = const$. *The same holds for* Y.

Corollary 1
$\lim_{d\to\infty}\big(\mathrm{E}(Z)/\sqrt{d}\,\big) = const$, *and* $\lim_{d\to\infty}\mathrm{Var}(Z) = const$.

Proof Follows directly from Theorem 6 and the fact that, since vectors $\mathbf{p}$ and $\mathbf{q}$ have iid components, vector $\mathbf{p} - \mathbf{q}$ also has iid components. □

Corollary 2
$\lim_{d\to\infty}(\mathrm{E}(X^2)/d) = const$, *and* $\lim_{d\to\infty}(\mathrm{Var}(X^2)/d) = const$. *The same holds for random variables* Y^2 *and* Z^2.

Proof From Theorem 6 and the equation $\mathrm{E}(X^2) = \mathrm{Var}(X) + \mathrm{E}(X)^2$ it follows that

$$\lim_{d\to\infty}(\mathrm{E}(X^2)/d) = const.$$

The same holds for $\mathrm{E}(Y^2)$ and, taking into account Corollary 1, for $\mathrm{E}(Z^2)$. By using the delta method to approximate the moments of a function of a random variable with Taylor expansions [24], we have

$$\mathrm{Var}(X^2) \sim (2\,\mathrm{E}(X))^2\,\mathrm{Var}(X)\,.$$

From Theorem 6 it now follows that

$$\lim_{d\to\infty}(\mathrm{Var}(X^2)/d) = const\,.$$

Analogous derivations hold for $\mathrm{Var}(Y^2)$ and $\mathrm{Var}(Z^2)$. □

Based on the above results, the following two theorems show that $\sqrt{\mathrm{Var}(C)}$ reduces asymptotically to 0, while $\mathrm{E}(C)$ remains asymptotically constant (proofs are given in the next section).

Theorem 7 $\lim_{d\to\infty}\sqrt{\mathrm{Var}(C)} = 0$.

Theorem 8 $\lim_{d\to\infty}\mathrm{E}(C) = const$.

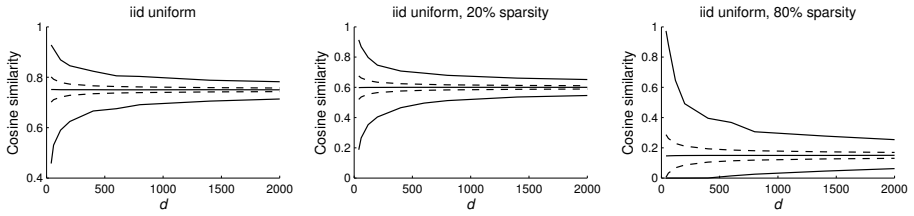


Figure 3.2: Concentration of cosine similarity for iid uniform random data that is (a) dense, (b) 20% sparse, and (c) 80% sparse
Slika 3.2: Koncentracija kosinusne mere sličnosti za *iid* uniforman skup slučajnih tačaka koje su (a) guste, (b) 20% retke, i (c) 80% retke

Figure 3.2 illustrates the described findings for random data sets with $n = 2000$ d-dimensional points, whose components are independently drawn from the uniform distribution in range $[0, 1]$. Figure 3.2(a) depicts unmodified (dense) data, while Figures 3.2(b) and (c) show random data which was made sparse by setting a certain percentage of randomly selected component values to 0. With respect to the distribution of all pairwise similarities, the plots include, from top to bottom: maximal observed value, mean value plus one standard deviation, the mean value, mean value minus one standard deviation, and minimal observed value. The figures illustrate that, with increasing dimensionality, expectation becomes constant and variance shrinks. It is also evident that the property holds regardless of the sparsity of data, suggesting that the phenomenon is relevant to real sparse data such as text, where cosine similarity is often employed.

It is worth noting that the concentration of cosine similarity results from different reasons than the concentration of Euclidean (l_2) distance. For the latter, its standard deviation converges to a constant [61], whereas its expectation asymptotically increases with d. Nevertheless, in both cases the relative relationship between the standard deviation and the expectation is similar.

3.3 Proofs of Theorems 7 and 8

Theorem 7 $\lim_{d\to\infty} \sqrt{\mathrm{Var}(C)} = 0.$

Proof From Equation 3.8 we get:

$$\begin{aligned} 4\,\mathrm{Var}(C) \;=\; & \mathrm{Var}\left(\frac{X}{Y}\right) + \mathrm{Var}\left(\frac{Y}{X}\right) + \mathrm{Var}\left(\frac{Z^2}{XY}\right) + \\ & 2\,\mathrm{Cov}\left(\frac{X}{Y}, \frac{Y}{X}\right) - 2\,\mathrm{Cov}\left(\frac{X}{Y}, \frac{Z^2}{XY}\right) - 2\,\mathrm{Cov}\left(\frac{Y}{X}, \frac{Z^2}{XY}\right). \end{aligned} \tag{3.9}$$

For the first term, using the delta method [24] and the fact that X and Y are independent:

$$\mathrm{Var}\left(\frac{X}{Y}\right) \sim \frac{\mathrm{Var}(X)}{\mathrm{E}^2(Y)} + \frac{\mathrm{E}^2(X)}{\mathrm{E}^4(Y)}\,\mathrm{Var}(Y)\,,$$

from which it follows, based on Theorem 6, that $\mathrm{Var}(\frac{X}{Y})$ is $O(1/d)$ (for brevity, we resort to oh notation in this proof). In the same way, $\mathrm{Var}(\frac{Y}{X})$ is also $O(1/d)$.

For the third term of Equation 3.8, again from the delta method:

$$\begin{aligned}\mathrm{Var}\left(\frac{Z^2}{XY}\right) \quad \sim \quad & \frac{\mathrm{Var}(Z^2)}{\mathrm{E}^2(X)\,\mathrm{E}^2(Y)} - \\ & \frac{2\,\mathrm{E}(Z^2)}{\mathrm{E}^3(X)\,\mathrm{E}^3(Y)}\,\mathrm{Cov}(Z^2, XY) + \frac{\mathrm{E}^2(Z^2)}{\mathrm{E}^4(X)\,\mathrm{E}^4(Y)}\,\mathrm{Var}(XY)\,.\end{aligned} \tag{3.10}$$

In Equation 3.10, based on Theorem 6 and Corollary 2, the first term is $O(1/d)$. Since variables X and Y are independent [73]:

$$\mathrm{Var}(XY) = \mathrm{E}^2(X)\,\mathrm{Var}(Y) + \mathrm{E}^2(Y)\,\mathrm{Var}(X) + \mathrm{Var}(X)\,\mathrm{Var}(Y)\,,$$

from which it follows that $\mathrm{Var}(XY)$ is $O(d)$, thus the third term is $O(1/d)$, too. $\mathrm{Cov}(Z^2, XY)$ is $O(d)$, because from the definition of the correlation coefficient,

$$|\,\mathrm{Cov}(Z^2, XY)| \leq \max(\mathrm{Var}(Z^2), \mathrm{Var}(XY))\,.$$

Thus, the second term of Equation 3.10 is $O(1/d)$. Since all its terms are $O(1/d)$, $\mathrm{Var}(\frac{Z^2}{XY})$ is $O(1/d)$.

Returning to Equation 3.9 and its fourth term, from the definition of the correlation coefficient it follows that

$$\left|\mathrm{Cov}\left(\frac{X}{Y}, \frac{Y}{X}\right)\right| \leq \max\left(\mathrm{Var}\left(\frac{X}{Y}\right), \mathrm{Var}\left(\frac{Y}{X}\right)\right),$$

thus $\mathrm{Cov}(\frac{X}{Y}, \frac{Y}{X})$ is $O(1/d)$. For the fifth term, again from the definition of the correlation coefficient we have

$$\left|\mathrm{Cov}\left(\frac{X}{Y}, \frac{Z^2}{XY}\right)\right| \leq \max\left(\mathrm{Var}\left(\frac{X}{Y}\right), \mathrm{Var}\left(\frac{Z^2}{XY}\right)\right).$$

Based on the previously expressed $\mathrm{Var}(\frac{X}{Y})$ and $\mathrm{Var}(\frac{Z^2}{XY})$, we get that $\mathrm{Cov}(\frac{X}{Y}, \frac{Z^2}{XY})$ is $O(1/d)$. Similarly, the sixth term, $\mathrm{Cov}(\frac{Y}{X}, \frac{Z^2}{XY})$, is $O(1/d)$. Having determined all 6 terms, $4\,\mathrm{Var}(C)$, and thus $\mathrm{Var}(C)$, is $O(1/d)$. It follows that $\lim_{d\to\infty} \sqrt{\mathrm{Var}(C)} = 0$. □

Theorem 8 $\lim_{d\to\infty} \mathrm{E}(C) = const.$

Proof From Equation 3.8 we get:

$$2\,\mathrm{E}(C) = \mathrm{E}\left(\frac{X}{Y}\right) + \mathrm{E}\left(\frac{Y}{X}\right) - \mathrm{E}\left(\frac{Z^2}{XY}\right)\,. \tag{3.11}$$

For the first term, using the delta method [24] and the fact that X and Y are independent:

$$\mathrm{E}\left(\frac{X}{Y}\right) \sim \frac{\mathrm{E}(X)}{\mathrm{E}(Y)}\left(1+\mathrm{Var}(Y)\right).$$

Based on limits for $\mathrm{E}(X)/\sqrt{d}$, $\mathrm{E}(Y)/\sqrt{d}$, and $\mathrm{Var}(Y)$ in Theorem 6, it follows that $\lim_{d\to\infty}\mathrm{E}(\frac{X}{Y}) = const$. For the second term, analogously, $\lim_{d\to\infty}\mathrm{E}(\frac{Y}{X}) = const$.

For the third term in Equation 3.11, again from the delta method:

$$\mathrm{E}\left(\frac{Z^2}{XY}\right) \sim \frac{\mathrm{E}(Z^2)}{\mathrm{E}(X)\,\mathrm{E}(Y)} - \frac{\mathrm{Cov}(Z^2,XY)}{\mathrm{E}^2(X)\,\mathrm{E}^2(Y)} + \frac{\mathrm{E}(Z^2)}{\mathrm{E}^3(X)\,\mathrm{E}^3(Y)}\,\mathrm{Var}(XY). \quad (3.12)$$

In Equation 3.12, from the limits derived in Theorem 6 and Corollary 2 follows the limit of the first term,

$$\lim_{d\to\infty}\frac{\mathrm{E}(Z^2)}{\mathrm{E}(X)\,\mathrm{E}(Y)} = const.$$

The limit of the second term in Equation 3.12 can be obtained by multiplying and dividing by d^2, yielding

$$\lim_{d\to\infty}\frac{\mathrm{Cov}(Z^2,XY)}{d^2}\left(\lim_{d\to\infty}\frac{\mathrm{E}^2(X)\,\mathrm{E}^2(Y)}{d^2}\right)^{-1}.$$

From the definition of the correlation coefficient we have:

$$\left|\lim_{d\to\infty}\frac{\mathrm{Cov}(Z^2,XY)}{d^2}\right| \le \sqrt{\lim_{d\to\infty}\frac{\mathrm{Var}(Z^2)}{d^2}}\sqrt{\lim_{d\to\infty}\frac{\mathrm{Var}(XY)}{d^2}}.$$

From $\mathrm{Var}(XY) = \mathrm{E}^2(X)\,\mathrm{Var}(Y) + \mathrm{E}^2(Y)\,\mathrm{Var}(X) + \mathrm{Var}(X)\,\mathrm{Var}(Y)$ [73], based on Theorem 6 and Corollary 2, we find that both limits on the right side are equal to 0, implying that

$$\lim_{d\to\infty}\frac{\mathrm{Cov}(Z^2,XY)}{d^2} = 0.$$

On the other hand, from Theorem 6 we have

$$\lim_{d\to\infty}\frac{\mathrm{E}^2(X)\,\mathrm{E}^2(Y)}{d^2} = const.$$

The preceding two limits provide us with the limit for the second term of Equation 3.12:

$$\lim_{d\to\infty}\frac{\mathrm{Cov}(Z^2,XY)}{\mathrm{E}^2(X)\,\mathrm{E}^2(Y)} = 0.$$

Finally, for the third term of Equation 3.12, again based on the limits given in Theorem 6 and Corollary 2, and the previously derived limit for $\mathrm{Var}(XY)/d^2$, we obtain

$$\lim_{d\to\infty}\frac{\mathrm{E}(Z^2)}{\mathrm{E}^3(X)\,\mathrm{E}^3(Y)}\,\mathrm{Var}(XY) = 0.$$

Summing up all partial limits, it follows that $\lim_{d\to\infty} 2\,\mathrm{E}(C) = const$, and thus $\lim_{d\to\infty}\mathrm{E}(C) = const$. □

Chapter 4

The Hubness Phenomenon

The curse of dimensionality, a term originally introduced by Bellman [10], is nowadays commonly used in many fields to refer to challenges posed by high dimensionality of data space. In the fields of machine learning and data mining, affected methods and tasks include Bayesian modeling [18], nearest-neighbor prediction [82] and search [111], neural networks [17], etc. (See Chapter 1.)

There exists an aspect of the curse of dimensionality that is related to nearest neighbors (NNs), which we will refer to as *hubness*. Let $D \subset \mathbb{R}^d$ be a set of d-dimensional points and $N_k(\mathbf{x})$ the number of *k-occurrences* of each point $\mathbf{x} \in D$, that is, the number of times $\mathbf{x}$ occurs among the k nearest neighbors of all other points in D, according to some distance measure. Under widely applicable conditions, as dimensionality increases, the distribution of N_k becomes considerably skewed to the right, resulting in the emergence of *hubs*, that is, points which appear in many more k-NN lists than other points, effectively making them "popular" nearest neighbors. Unlike distance concentration, hubness and its influence on machine learning, data mining, and information retrieval have not been explored in depth. In this chapter we will study the manifestations and causes of this aspect of the dimensionality curse. Subsequent chapters will examine the implications of hubness on various tasks in machine learning (Chapter 5), time-series analysis (Chapter 6), and information retrieval (Chapter 7).

As will be described in Section 4.3, the phenomena of distance concentration and hubness are related, but distinct. Traditionally, distance concentration is studied through asymptotic behavior of norms, that is, distances to the origin, with increasing dimensionality. The obtained results trivially extend to reference points other than the origin, and to pairwise distances between all points. However, the asymptotic tendencies of distances of all points to different reference points do not necessarily occur at the same *speed*, which will be shown for normally distributed data by our main theoretical result outlined in Section 4.3.2, and given with full details in Section 4.4. The main consequence of the analysis, which is further discussed in Section 4.5 and supported by theoretical results from [144] and [143], is that the hubness phenomenon is an inherent

property of data distributions in high-dimensional space under widely used assumptions, and not an artefact of a finite sample or specific properties of a particular data set.

The above result is relevant to machine learning because many families of ML algorithms, regardless of whether they are supervised, semi-supervised, or unsupervised, directly or indirectly make use of distances between data points (and, with them, k-NN graphs) in the process of building a model. The same observation can be made with regards to algorithms and models in data mining and information retrieval. Moreover, the hubness phenomenon recently started to be observed in application fields like music retrieval [9], speech recognition [49], and fingerprint identification [84], where it is described as a problematic situation, but little or no insight is offered into the origins of the phenomenon. This dissertation presents a unifying view of the hubness phenomenon. In this regard, the current chapter will perform a through theoretical analysis of data distributions, and empirical investigation including various synthetic data sets, explaining the origins of the phenomenon and the mechanism through which hubs emerge. Chapter 5 will extend the discussion to real data sets from numerous application fields, analyze the role of *antihubs* (points which appear in very few, if any, k-NN lists of other points) and the interaction of hubness with information provided by class labels, and study the impact of the phenomenon on various machine-learning algorithms.

After discussing related work in the next section, we make the following contributions. First, we demonstrate the emergence of hubness on synthetic data in Section 4.2. The following section provides a comprehensive explanation of the origins of the phenomenon, through empirical and theoretical analysis of artificial data distributions,[1] linking hubness with the dimensionality of data. Section 4.4 presents the details of our main theoretical result which describes the mechanism through which hubs emerge as dimensionality increases. Finally, Section 4.5 provides discussion and further illustration of the behavior of nearest-neighbor relations in high dimensions, connecting our findings with existing theoretical results.

4.1 Related Work

The hubness phenomenon has been recently observed in several application areas involving sound and image data [9, 49, 84], where it was perceived as a problematic situation, but without establishing a connection with high dimensionality of data. One recent work that established a connection between hubness and high dimensionality is the dissertation by Berenzweig [12], who identified high dimensionality as a cause of hubness, but did not provide practical or theoretical support that would explain the mechanism through which hubness emerges in real high-dimensional data.

The distribution of N_1 has been explicitly studied in the applied probability community [144, 131, 143, 229], and by mathematical psychologists [209, 208]. In the vast majority of studied settings (for example, Poisson process, d-dimensional torus), coupled with Euclidean distance, it was shown that the distribution of N_1 converges

[1] Generalization of the explanation to real data sets will be given in Chapter 5.

to the Poisson distribution with mean 1, as the number of points n and dimensionality d go to infinity. Moreover, from the results in [229] it immediately follows that, in the Poisson process case, the distribution of N_k converges to the Poisson distribution with mean k, for any $k \geq 1$. All these results imply that no hubness is to be expected within the settings in question. On the other hand, in the case of continuous distributions with iid components, for the following specific order of limits it was shown that $\lim_{n\to\infty} \lim_{d\to\infty} \mathrm{Var}(N_1) = \infty$, while $\lim_{n\to\infty} \lim_{d\to\infty} N_1 = 0$, in distribution [144, p. 730, Theorem 7], with a more general result given in [143]. According to the interpretation in [209], this suggests that if the number of dimensions is large relative to the number of points one may expect a small proportion of points to become hubs. However, the importance of this finding was downplayed to a certain extent [209, 143], citing empirically observed slow convergence [131], with the attention of the authors shifting more towards similarity measurements obtained directly from psychological and cognitive experiments [209, 208] that do not involve vector-space data. In Section 4.5 we will discuss the above results in more detail, as well as their relations with our theoretical and empirical findings.

It is worth noting that in ϵ-neighborhood graphs, that is, graphs where two points are connected if the *distance* between them is less than a given limit ϵ, the hubness phenomenon does not occur. Settings involving randomly generated points forming ϵ-neighborhood graphs are typically referred to as random geometric graphs, and are discussed in detail by Penrose [155].

4.2 Observing Hubness

At the beginning of the chapter we gave a simple set-based deterministic definition of N_k. To complement this definition and introduce N_k into a probabilistic setting that will also be considered in this dissertation, let $\mathbf{x}, \mathbf{x}_1, \ldots, \mathbf{x}_n$, be $n+1$ random vectors drawn from the same continuous probability distribution with support $\mathcal{S} \subseteq \mathbb{R}^d$, $d \in \{1, 2, \ldots\}$, and let $dist$ be a distance function defined on $\mathbb{R}^d$ (not necessarily a metric). Let functions $p_{i,k}$, where $i, k \in \{1, 2, \ldots, n\}$, be defined as

$$p_{i,k}(\mathbf{x}) = \begin{cases} 1, & \text{if } \mathbf{x} \text{ is among the } k \text{ nearest neighbors of } \mathbf{x}_i, \text{ according to } dist, \\ 0, & \text{otherwise.} \end{cases}$$

In this setting, we define $N_k(\mathbf{x}) = \sum_{i=1}^{n} p_{i,k}(\mathbf{x})$, that is, $N_k(\mathbf{x})$ is the random number of vectors from $\mathbb{R}^d$ that have $\mathbf{x}$ included in their list of k nearest neighbors.

We start our investigation with an illustrative experiment which demonstrates the changes in the distribution of N_k with varying dimensionality. Let us consider a random data set consisting of 10000 d-dimensional points, whose components are independently drawn from the uniform distribution in range $[0, 1]$, and the following distance functions: Euclidean (l_2), fractional $l_{0.5}$ (proposed for high-dimensional data in [2]), and cosine. Figure 4.1(a–c) shows the empirically observed distributions of N_k, with $k = 5$, for dimensionalities (a) $d = 3$, (b) $d = 20$, and (c) $d = 100$. In the same

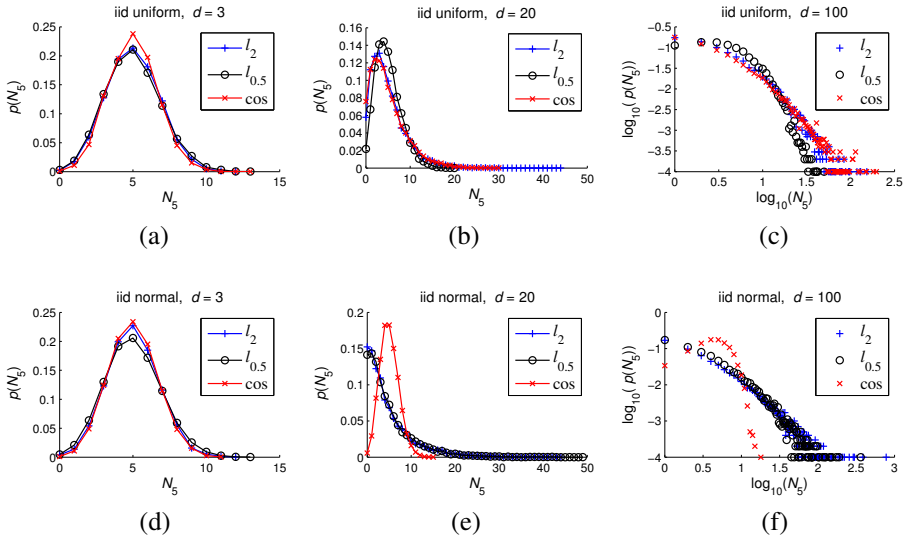


Figure 4.1: Empirical distribution of N_5 for Euclidean, $l_{0.5}$, and cosine distances on (a–c) iid uniform, and (d–f) iid normal random data sets with $n = 10000$ points and dimensionality (a, d) $d = 3$, (b, e) $d = 20$, and (c, f) $d = 100$ (log-log plot)
Slika 4.1: Empirijska distribucija N_5 za euklidsku, $l_{0.5}$, i kosinusnu udaljenost na (a–c) *iid* uniformnom, i (d–f) *iid* normalnom skupu $n = 10000$ slučajnih tačaka dimenzionalnosti (a, d) $d = 3$, (b, e) $d = 20$, i (c, f) $d = 100$ (log-log grafikon)

way, Figure 4.1(d–f) depicts the empirically observed N_k for data points randomly drawn from the iid normal distribution.

For $d = 3$ the empirical distributions of N_5 for the three distance functions (Figure 4.1(a, d)) are consistent with the binomial distribution. This is expected by considering k-occurrences as node in-degrees in the k-nearest neighbor digraph. For uniformly distributed points in low dimensions, the degree distributions of the digraphs closely resemble the degree distribution of the Erdős-Rényi (ER) random graph model, which is is binomial and Poisson in the limit [54].

As dimensionality increases, the observed distributions of N_5 depart from the random graph model and become more skewed to the right (Figure 4.1(b, c), and Figure 4.1(e, f) for l_2 and $l_{0.5}$ distances). We verified this by being able to fit major right portions (that is, tails) of the observed distributions with the log-normal distribution, which is highly skewed.[2] We made similar observations with various k values (generally focusing on the common case $k \ll n$, where n is the number of points in a data set),

[2]Fits were supported by the χ^2 goodness-of-fit test at significance level 0.05, where bins represent the number of observations of individual N_k values. These empirical distributions were compared with the expected output of a (discretized) log-normal distribution, making sure that counts in the bins do not fall below 5 by pooling the rightmost bins together.

distance measures (l_p-norm distances for both $p \geq 1$ and $0 < p < 1$, Bray-Curtis, normalized Euclidean, and Canberra), and data distributions. In virtually all these cases, skewness exists and produces hubs, that is, points with high k-occurrences. One exception visible in Figure 4.1 is the combination of cosine distance and normally distributed data. In most practical settings, however, such situations are not expected, and a thorough discussion of the necessary conditions for hubness to occur in high dimensions will be given in Section 4.5.

4.3 Explaining Hubness

This section moves on to exploring the causes of hubness and the mechanisms through which hubs emerge. Section 4.3.1 investigates the relationship between the position of a point in data space and hubness. Then, Section 4.3.2 explains the mechanism through which hubs emerge as points in high-dimensional space that become closer to all other points than their low-dimensional counterparts, outlining our main theoretical result.

4.3.1 The Position of Hubs

Let us consider again the iid uniform and iid normal random data examined in the previous section. We will demonstrate that the position of a point in data space has a significant effect on its k-occurrences value, by observing the sample mean of the data distribution as a point of reference. Figure 4.2 plots, for each point $\mathbf{x}$, its $N_5(\mathbf{x})$ against its Euclidean distance from the empirical data mean, for $d = 3,\ 20,\ 100$. As dimensionality increases, stronger correlation emerges, implying that points closer to the mean tend to become hubs. We made analogous observations with other values of k, and combinations of data distributions and distance measures for which hubness occurs. It is important to note that proximity to one global data-set mean correlates with hubness in high dimensions when the underlying data distribution is *unimodal*. For multimodal data distributions, for example those obtained through a mixture of unimodal distributions, hubs tend to appear close to the means of individual component distributions of the mixture. In the discussion that follows in Section 4.3.2 we will assume a unimodal data distribution, and defer the analysis of multimodal distributions until Section 5.3, which studies real data.

4.3.2 Mechanisms Behind Hubness

Although one may expect that some random points are closer to the data-set mean than others, in order to explain the mechanism behind hub formation we need to (1) understand the geometrical and distributional setting in which some points tend to be closer

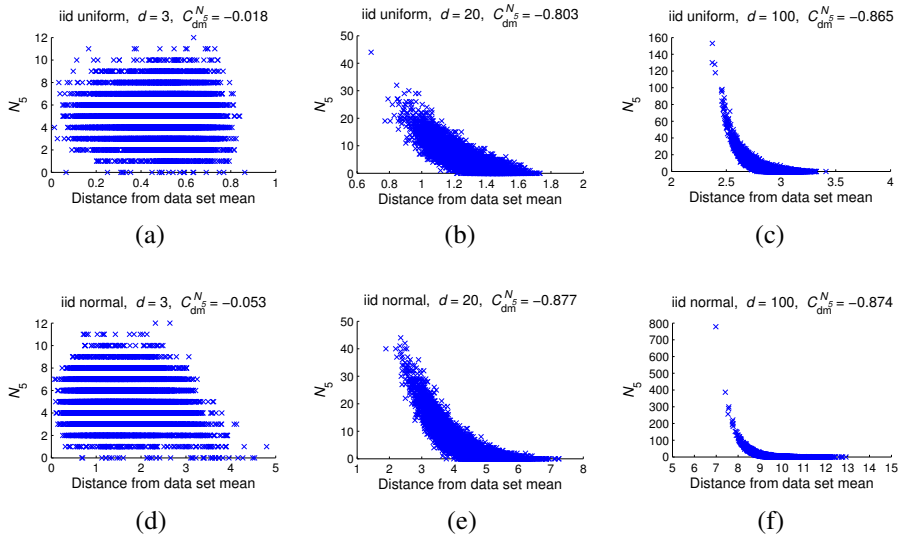


Figure 4.2: Scatter plots and Spearman correlation of $N_5(\mathbf{x})$ against the Euclidean distance of point $\mathbf{x}$ to the sample data-set mean for (a–c) iid uniform and (d–f) iid normal random data sets with (a, d) $d = 3$, (b, e) $d = 20$, and (c, f) $d = 100$
Slika 4.2: Grafikon i Spermanova korelacija $N_5(\mathbf{x})$ sa euklidskom udaljenošću tačke $\mathbf{x}$ od empirijskog centra skupa tačaka za (a–c) *iid* uniformne i (d–f) *iid* normalne skupove slučajnih tačaka sa (a, d) $d = 3$, (b, e) $d = 20$, i (c, f) $d = 100$

to the mean than others, and then (2) understand why such points become hubs in higher dimensions.[3]

Hubness appears to be related to the phenomenon of distance concentration. Based on existing theoretical results discussed in Section 3.1 (Theorems 1–5), it can be said that high-dimensional points are approximately lying on a hypersphere centered at the data-set mean. Moreover, the results in [44] and [61] (Theorem 4 and Lemma 2 from Section 3.1) specify that the distribution of distances to the data-set mean has a non-negligible variance for any finite d.[4] Hence, the existence of a non-negligible number of points closer to the data-set mean is *expected* in high dimensions.

To illustrate the above discussion, Figure 4.3 depicts, for iid normal data, the distribution of Euclidean distances of all points to the true data mean (the origin) for several d values. By definition, the distribution of distances is actually the Chi distribution

[3]We will assume that random points originate from a unimodal data distribution. In the multimodal case, it can be said that the observations which follow are applicable around one of the "peaks" in the pdf of the data distribution.

[4]These results apply to l_p-norm distances, but our numerical simulations suggest that other distance functions mentioned in Section 4.2 behave similarly. Moreover, any point can be used as a reference instead of the data mean, but we observe the data mean since it plays a special role with respect to hubness.

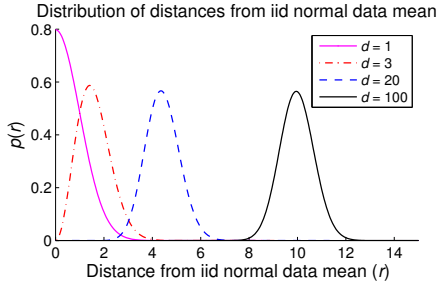


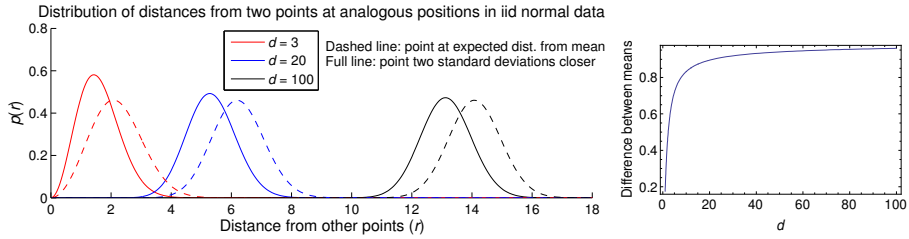
Figure 4.3: Probability density function of observing a point at distance r from the mean of a multivariate d-dimensional normal distribution, for $d = 1,\ 3,\ 20,\ 100$
Slika 4.3: Funkcija gustine verovatnoće da se uoči tačka na udaljenosti r od centra d-dimenzionalne normalne distribucije, za $d = 1,\ 3,\ 20,\ 100$

with d degrees of freedom (as the square root of the sum of squares of iid normal variables [48, 98]).[5] In this setting, distance concentration refers to the fact that the standard deviation of distance distributions is asymptotically constant with respect to increasing d, while the means of the distance distributions asymptotically behave like $\sqrt{d}$ (a direct consequence of the results by François et al. [61] reviewed in Section 3.1, and discussed further in Section 4.4). On the other hand, for l_p-norm distances with $p > 2$, the standard deviation would tend to 0 (Lemma 2). However, for any finite d, existing variation in the values of random coordinates causes some points to become closer to the distribution mean than others. This happens despite the fact that all distance values, in general, may be increasing together with d.

To understand why points closer to the data mean become hubs in high dimensions, let us consider the following example. We observe, within the iid normal setting, two points drawn from the data, but at specific positions with respect to the origin: point $\mathbf{b}_d$ which is at the expected distance from the origin, and point $\mathbf{a}_d$ which is two standard deviations closer. In light of the above, the distances of $\mathbf{a}_d$ and $\mathbf{b}_d$ from the origin change with increasing d, and it could be said that different $\mathbf{a}_d$-s (and $\mathbf{b}_d$-s) occupy analogous positions in the data spaces, with respect to changing d. The distances of $\mathbf{a}_d$ (and $\mathbf{b}_d$) to all other points, again following directly from the definition [48, 149], are distributed according to the *noncentral* Chi distributions with d degrees of freedom and noncentrality parameter λ equaling the distance of $\mathbf{a}_d$ ($\mathbf{b}_d$) to the origin. Figure 4.4(a) plots the probability density functions of these distributions for several values of d. It can be seen that, as d increases, the distance distributions for $\mathbf{a}_d$ and $\mathbf{b}_d$ move away from each other. This tendency is depicted more clearly in Figure 4.4(b) which plots the difference between the means of the two distributions with respect to d.

It is known, and expected, for points that are closer to the mean of the data distribution to be closer, on average, to all other points, for any value of d. However, the

[5] For this reason, in Figure 4.3 we plot the known pdf, not the empirically obtained distribution.



(a) (b)

Figure 4.4: (a) Distribution of distances to other points from iid normal random data for a point at the expected distance from the origin (dashed line), and a point two standard deviations closer (full line). (b) Difference between the means of the two distributions, with respect to increasing d

Slika 4.4: (a) Distribucija udaljenosti do ostalih tačaka iz *iid* normalnog skupa slučajnih tačaka za tačku na očekivanoj udaljenosti od koordinatnog početka (isprekidana linija), i za tačku koja je za dve standardne devijacije bliža (puna linija). (b) Razlika između očekivanih vrednosti dve distribucije, u odnosu na rastuće d

above analysis indicates that this tendency is amplified by high dimensionality, making points that reside in the proximity of the data mean become closer (in relative terms) to all other points than their low-dimensional analogues are. This tendency causes high-dimensional points that are closer to the mean to have increased inclusion probability into k-NN lists of other points, even for small values of k. We will discuss this relationship further in Section 4.5.

In terms of the notion of node *centrality* typically used in network analysis [189], the above discussion indicates that high dimensionality amplifies what can be called the *spatial centrality* of a point (by increasing its proximity to other points), which, in turn, affects the *degree centrality* of the corresponding node in the k-NN graph (by increasing its degree, that is, N_k). Other notions of node centrality, and the structure of the k-NN graph in general, will be studied in more detail in Section 4.5.1. The rest of this section will focus on describing the mechanism of the observed spatial centrality amplification.

In the preceding discussion we selected two points from iid normal data with specific distances from the origin expressed in terms of expected distance and deviation from it, and tracked the analogues of the two points for increasing values of dimensionality d. Generally, we can express the distances of the two points to the origin in terms of "offsets" from the expected distance measured by standard deviation, which in the case of iid normal random data would be $\lambda_{d,1} = \mu_{\chi(d)} + c_1\sigma_{\chi(d)}$ and $\lambda_{d,2} = \mu_{\chi(d)} + c_2\sigma_{\chi(d)}$, where $\lambda_{d,1}$ and $\lambda_{d,2}$ are the distances of the first and second point to the origin, $\mu_{\chi(d)}$ and $\sigma_{\chi(d)}$ are the mean and standard deviation of the Chi distribution with d degrees of freedom, and c_1 and c_2 are selected constants (the offsets). In the preceding example involving points $\mathbf{a}_d$ and $\mathbf{b}_d$, we set $c_1 = -2$ and $c_2 = 0$, respectively. However, anal-

ogous behavior can be observed with arbitrary two points whose distance to the data mean is below the expected distance, that is, for $c_1, c_2 \leq 0$. We describe this behavior by introducing the following notation: $\Delta\mu_d(\lambda_{d,1}, \lambda_{d,2}) = |\mu_{\chi(d,\lambda_{d,2})} - \mu_{\chi(d,\lambda_{d,1})}|$, where $\mu_{\chi(d,\lambda_{d,i})}$ is the mean of the noncentral Chi distribution with d degrees of freedom and noncentrality parameter $\lambda_{d,i}$ ($i \in \{1, 2\}$). In the following theorem, proven in Section 4.4, we show that $\Delta\mu_d(\lambda_{d,1}, \lambda_{d,2})$ increases with increasing values of d.

Theorem 9 *Let $\lambda_{d,1} = \mu_{\chi(d)} + c_1\sigma_{\chi(d)}$ and $\lambda_{d,2} = \mu_{\chi(d)} + c_2\sigma_{\chi(d)}$, where $d \in \mathbb{N}^+$, $c_1, c_2 \leq 0$, $c_1 < c_2$, and $\mu_{\chi(d)}$ and $\sigma_{\chi(d)}$ are the mean and standard deviation of the Chi distribution with d degrees of freedom, respectively. Define*

$$\Delta\mu_d(\lambda_{d,1}, \lambda_{d,2}) = \mu_{\chi(d,\lambda_{d,2})} - \mu_{\chi(d,\lambda_{d,1})}\,,$$

where $\mu_{\chi(d,\lambda_{d,i})}$ is the mean of the noncentral Chi distribution with d degrees of freedom and noncentrality parameter $\lambda_{d,i}$ ($i \in \{1, 2\}$).

There exists $d_0 \in \mathbb{N}$ such that for every $d > d_0$,

$$\Delta\mu_d(\lambda_{d,1}, \lambda_{d,2}) > 0\,,$$

and

$$\Delta\mu_{d+2}(\lambda_{d+2,1}, \lambda_{d+2,2}) > \Delta\mu_d(\lambda_{d,1}, \lambda_{d,2})\,. \tag{4.1}$$

The main statement of the theorem is given by Equation 4.1, which expresses the tendency of the difference between the means of the two distance distributions to increase with increasing dimensionality d. It is important to note that this tendency is obtained through analysis of *distributions* of data and distances, implying that the behavior is an inherent property of data distributions in high-dimensional space, rather than an artefact of other factors, such as finite sample size, etc. Through simulation involving randomly generated points we verified the behavior for iid normal data by replicating very closely the chart shown in Figure 4.4(b). Furthermore, simulations suggest that the same behavior emerges in iid uniform data,[6] as well as numerous other unimodal random data distributions, producing charts of the same shape as in Figure 4.4(b). Real data, on the other hand, tends to be clustered, and can be viewed as originating from a *mixture* of distributions resulting in a multimodal distribution of data. In this case, the behavior described by Theorem 9, and illustrated in Figure 4.4(b), is manifested primarily on the individual component distributions of the mixture, that is, on clusters of data points. The hubness phenomenon in real data sets is investigated more closely in Chapter 5.

4.4 Proof of Theorem 9

In this section we analyze the behavior of distances that provides the mechanism for the formation of hubs, introduced in Section 4.3.2, culminating with the proof of The-

[6] The uniform cube setting will be discussed in more detail in Section 4.5, in the context of results from related work [143].

orem 9. Section 4.4.1 reviews distance concentration results from [61], while Section 4.4.2 discusses distance distributions in iid normal random data and extended interpretations of the results from [61] in this setting. The notion of asymptotic equivalence that will be used in the proof of Theorem 9 is the subject of Section 4.4.3. The expectation of the noncentral Chi distribution, which is a key feature in the analysis of distance distributions in iid normal random data, is defined in Section 4.4.4, together with the generalized Laguerre function on which it relies. Properties of the generalized Laguerre function that will be used in the proof of Theorem 9 are presented in Section 4.4.5. Finally, the proof of Theorem 9 is given in Section 4.4.6.

4.4.1 Distance Concentration Results

We begin by reviewing the main results of François et al. [61] regarding distance concentration. Let $\mathbf{X}_d = (X_1, X_2, \ldots, X_d)$ be a random d-dimensional variable with iid components: $X_i \sim \mathcal{F}$, and let $\|\mathbf{X}_d\|$ denote its Euclidean norm. For random variables $|X_i|^2$, let $\mu_{|\mathcal{F}|^2}$ and $\sigma^2_{|\mathcal{F}|^2}$ signify their mean and variance, respectively. François et al. [61] prove the following two lemmas.[7]

Lemma 3 [61, Equation 17, adapted]

$$\lim_{d\to\infty} \frac{\mathrm{E}(\|\mathbf{X}_d\|)}{\sqrt{d}} = \mu_{|\mathcal{F}|^2}\,.$$

Lemma 4 [61, Equation 21, adapted]

$$\lim_{d\to\infty} \mathrm{Var}(\|\mathbf{X}_d\|) = \frac{\sigma^2_{|\mathcal{F}|^2}}{4\mu_{|\mathcal{F}|^2}}\,.$$

The above lemmas imply that, for iid random data, the expectation of the distribution of Euclidean distances to the origin (Euclidean norms) asymptotically behaves like $\sqrt{d}$, while the standard deviation is asymptotically constant. From now on, we will denote the mean and variance of random variables that are distributed according to some distribution $\mathcal{F}$ by $\mu_{\mathcal{F}}$ and $\sigma^2_{\mathcal{F}}$, respectively.

4.4.2 Distances in iid Normal Data

We now observe more closely the behavior of distances in iid normal random data. Let $\mathbf{Z}_d = (Z_1, Z_2, \ldots, Z_d)$ be a random d-dimensional vector whose components independently follow the standard normal distribution: $Z_i \sim \mathcal{N}(0;1)$, for every $i \in \{1, 2, \ldots, d\}$. Then, by definition, random variable $\|\mathbf{Z}_d\|$ is distributed according to the Chi distribution with d degrees of freedom: $\|\mathbf{Z}_d\| \sim \chi(d)$. In other words, $\chi(d)$ is the distribution of Euclidean distances of vectors drawn from $\mathbf{Z}_d$ to the origin. If one were

[7] François et al. [61] provide a more general result for l_p norms with arbitrary $p > 0$, which is discussed in Section 3.1 (Lemmas 1 and 2).

to fix another reference vector $\mathbf{x}_d$ instead of the origin, the distribution of distances of vectors drawn from $\mathbf{Z}_d$ to $\mathbf{x}_d$ would be completely determined by $\|\mathbf{x}_d\|$ since, again by definition, random variable $\|\mathbf{Z}_d - \mathbf{x}_d\|$ follows the noncentral Chi distribution with d degrees of freedom and noncentrality parameter $\lambda = \|\mathbf{x}_d\|$: $\|\mathbf{Z}_d - \mathbf{x}_d\| \sim \chi(d, \|\mathbf{x}_d\|)$.

In light of the above, let us observe two points, $\mathbf{x}_{d,1}$ and $\mathbf{x}_{d,2}$, drawn from $\mathbf{Z}_d$, and express their distances from the origin in terms of offsets from the expected distance, with the offsets described using standard deviations: $\|\mathbf{x}_{d,1}\| = \mu_{\chi(d)} + c_1\sigma_{\chi(d)}$ and $\|\mathbf{x}_{d,2}\| = \mu_{\chi(d)} + c_2\sigma_{\chi(d)}$, where $c_1, c_2 \leq 0$. We will assume $c_1 < c_2$, that is, $\mathbf{x}_{d,1}$ is closer to the data distribution mean (the origin) than $\mathbf{x}_{d,2}$. By treating c_1 and c_2 as constants, and varying d, we observe analogues of two points in spaces of different dimensionalities (roughly speaking, points $\mathbf{x}_{d,1}$ have identical "probability" of occurrence at the specified distance from the origin for every d, and the same holds for $\mathbf{x}_{d,2}$). Let $\lambda_{d,1} = \|\mathbf{x}_{d,1}\|$ and $\lambda_{d,2} = \|\mathbf{x}_{d,2}\|$. Then, the distributions of distances of points $\mathbf{x}_{d,1}$ and $\mathbf{x}_{d,2}$ to all points from the data distribution $\mathbf{Z}_d$ (that is, the distributions of random variables $\|\mathbf{Z}_d - \mathbf{x}_{d,1}\|$ and $\|\mathbf{Z}_d - \mathbf{x}_{d,2}\|$) are noncentral Chi distributions $\chi(d, \lambda_{d,1})$ and $\chi(d, \lambda_{d,2})$, respectively. We study the behavior of these two distributions with increasing values of d.

Lemmas 3 and 4, taking $\mathcal{F}$ to be the standard normal distribution and translating the space so that $\mathbf{x}_{d,1}$ or $\mathbf{x}_{d,2}$ become the origin, imply that both $\mu_{\chi(d,\lambda_{d,1})}$ and $\mu_{\chi(d,\lambda_{d,2})}$ asymptotically behave like $\sqrt{d}$ as $d \to \infty$, while $\sigma^2_{\chi(d,\lambda_{d,1})}$ and $\sigma^2_{\chi(d,\lambda_{d,2})}$ are both asymptotically constant.[8] However, for $\mathbf{x}_{d,1}$ and $\mathbf{x}_{d,2}$ placed at different distances form the origin ($\lambda_{d,1} \neq \lambda_{d,2}$, that is, $c_1 \neq c_2$), these asymptotic tendencies do not occur at the same *speed*. In particular, we will show that as d increases, the difference between $\mu_{\chi(d,\lambda_{d,1})}$ and $\mu_{\chi(d,\lambda_{d,2})}$ actually *increases*. If we take $\mathbf{x}_{d,1}$ to be closer to the origin than $\mathbf{x}_{d,2}$ ($c_1 < c_2$), this means that $\mathbf{x}_{d,1}$ becomes closer to all other points from the data distribution $\mathbf{Z}_d$ than $\mathbf{x}_{d,2}$, simply by virtue of increasing dimensionality, since for different values of d we place the two points at analogous positions in the data space with regards to the distance from the origin.

4.4.3 Asymptotic Equivalence

Before describing our main theoretical result, we present several definitions and lemmas, beginning with the notion of asymptotic equivalence that will be relied upon.

Definition 1 *Two real-valued functions $f(x)$ and $g(x)$ are* asymptotically equal, $f(x) \approx g(x)$, *iff for every $\epsilon > 0$ there exists $x_0 \in \mathbb{R}$ such that for every $x > x_0$, $|f(x) - g(x)| < \epsilon$.*

[8] More precisely, the lemmas can be applied only for points $\mathbf{x}'_{d,i}$ that have equal values of all components, since after translation data components need to be iid. Because of the symmetry of the Gaussian distribution, the same expectations and variances of distance distributions are obtained, for every d, with any $\mathbf{x}_{d,i}$ that has the same norm as $\mathbf{x}'_{d,i}$, thereby producing identical asymptotic results.

Equivalently, $f(x) \approx g(x)$ iff $\lim_{x\to\infty} |f(x) - g(x)| = 0$. Note that the $\approx$ relation is different from the divisive notion of asymptotic equivalence, where $f(x) \sim g(x)$ iff $\lim_{x\to\infty} f(x)/g(x) = 1$.

The following two lemmas describe approximations that will be used in the proof of Theorem 9, based on the $\approx$ relation.

Lemma 5 *For any constant $c \in \mathbb{R}$, let $f(d) = \sqrt{d+c}$, and $g(d) = \sqrt{d}$. Then, $f(d) \approx g(d)$.*

Proof

$$\lim_{d\to\infty} \left|\sqrt{d+c} - \sqrt{d}\right| = \lim_{d\to\infty} \left|\left(\sqrt{d+c} - \sqrt{d}\right) \frac{\sqrt{d+c} + \sqrt{d}}{\sqrt{d+c} + \sqrt{d}}\right| = \lim_{d\to\infty} \left|\frac{c}{\sqrt{d+c} + \sqrt{d}}\right| = 0\,.$$

□

Lemma 6 *$\mu_{\chi(d)} \approx \sqrt{d}$, and $\sigma_{\chi(d)} \approx 1/\sqrt{2}$.*

Proof Observe the expression for the mean of the $\chi(d)$ distribution,

$$\mu_{\chi(d)} = \sqrt{2}\,\frac{\Gamma\left(\frac{d+1}{2}\right)}{\Gamma\left(\frac{d}{2}\right)}\,.$$

The equality $x\Gamma(x) = \Gamma(x+1)$ and the convexity of $\log\Gamma(x)$ yield [76, p. 237]:

$$\Gamma\left(\frac{d+1}{2}\right)^2 \leq \Gamma\left(\frac{d}{2}\right)\Gamma\left(\frac{d+2}{2}\right) = \frac{d}{2}\,\Gamma\left(\frac{d}{2}\right)^2,$$

and

$$\Gamma\left(\frac{d+1}{2}\right)^2 = \frac{d-1}{2}\,\Gamma\left(\frac{d-1}{2}\right)\Gamma\left(\frac{d+1}{2}\right) \geq \frac{d-1}{2}\,\Gamma\left(\frac{d}{2}\right)^2,$$

from which we have

$$\sqrt{d-1} \leq \sqrt{2}\,\frac{\Gamma\left(\frac{d+1}{2}\right)}{\Gamma\left(\frac{d}{2}\right)} \leq \sqrt{d}\,.$$

From Lemma 5 it now follows that $\mu_{\chi(d)} \approx \sqrt{d}$.

Regarding the standard deviation of the $\chi(d)$ distribution, from Lemma 4, taking $\mathcal{F}$ to be the standard normal distribution, we obtain

$$\lim_{d\to\infty} \sigma^2_{\chi(d)} = \frac{\sigma^2_{\chi^2(1)}}{4\mu_{\chi^2(1)}} = \frac{1}{2}\,,$$

since the square of a standard normal random variable follows the chi-square distribution with one degree of freedom, $\chi^2(1)$, whose mean and variance are known: $\mu_{\chi^2(1)} = 1$, $\sigma^2_{\chi^2(1)} = 2$. It now directly follows that $\sigma_{\chi(d)} \approx 1/\sqrt{2}$. □

4.4.4 Expectation of the Noncentral Chi Distribution

The central notion in Theorem 9 is the noncentral Chi distribution. To express the expectation of the noncentral Chi distribution, the following two definitions are needed, introducing the Kummer confluent hypergeometric function ${}_1F_1$, and the generalized Laguerre function.

Definition 2 [90, p. 1799, Appendix A, Table 19.I]
For $a, b, z \in \mathbb{R}$, the Kummer confluent hypergeometric function ${}_1F_1(a; b; z)$ is given by

$$ {}_1F_1(a; b; z) = \sum_{k=0}^{\infty} \frac{(a)_k}{(b)_k} \cdot \frac{z^k}{\Gamma(k+1)} \,, $$

where $(\cdot)_k$ is the Pochhammer symbol, $(x)_k = \frac{\Gamma(x+k)}{\Gamma(x)}$.

Definition 3 [90, p. 1811, Appendix A, Table 20.VI]
For $\nu, \alpha, z \in \mathbb{R}$, the generalized Laguerre function $L_\nu^{(\alpha)}(z)$ is given by

$$ L_\nu^{(\alpha)}(z) = \frac{\Gamma(\nu+\alpha+1)}{\Gamma(\nu+1)} \cdot \frac{{}_1F_1(-\nu; \alpha+1; z)}{\Gamma(\alpha+1)} \,. $$

The expectation of the noncentral Chi distribution with d degrees of freedom and noncentrality parameter λ, denoted by $\mu_{\chi(d,\lambda)}$, can now be expressed via the generalized Laguerre function [48, 149]:

$$ \mu_{\chi(d,\lambda)} = \sqrt{\frac{\pi}{2}} L_{1/2}^{(d/2-1)} \left(-\frac{\lambda^2}{2} \right) . \tag{4.2} $$

4.4.5 Properties of the Generalized Laguerre Function

The proof of Theorem 9 will rely on several properties of the generalized Laguerre function. In this section we will review two known properties and prove several additional ones as lemmas.

An important property of the generalized Laguerre function is its infinite differentiability in z, with the result of differentiation again being a generalized Laguerre function:

$$ \frac{\partial}{\partial z} L_\nu^{(\alpha)}(z) = -L_{\nu-1}^{(\alpha+1)}(z) \,. \tag{4.3} $$

Another useful property is the following recurrence relation:

$$ L_\nu^{(\alpha)}(z) = L_{\nu-1}^{(\alpha)}(z) + L_\nu^{(\alpha-1)}(z) \,. \tag{4.4} $$

Lemma 7 *For $\alpha > 0$ and $z < 0$:*

(a) *$L_{-1/2}^{(\alpha)}(z)$ is a positive monotonically increasing function in z, while*

(b) *$L_{1/2}^{(\alpha)}(z)$ is a positive monotonically decreasing concave function in z.*

Proof (a) From Definition 3,

$$L^{(\alpha)}_{-1/2}(z) = \frac{\Gamma(\alpha+1/2)}{\Gamma(1/2)} \cdot \frac{{}_1F_1(1/2;\alpha+1;z)}{\Gamma(\alpha+1)}\,.$$

Since $\alpha > 0$ all three terms involving the Gamma function are positive. We transform the remaining term using the equality [90, p. 1799, Appendix A, Table 19.I]:

$${}_1F_1(a;b;z) = e^z\,{}_1F_1(b-a;b;-z)\,, \tag{4.5}$$

which holds arbitrary $a, b, z \in \mathbb{R}$, obtaining

$${}_1F_1(1/2;\alpha+1;z) = e^z\,{}_1F_1(\alpha+1/2;\alpha+1;-z)\,.$$

From Definition 2 it now directly follows that $L^{(\alpha)}_{-1/2}(z)$ is positive for $\alpha > 0$ and $z < 0$.

To show that $L^{(\alpha)}_{-1/2}(z)$ is monotonically increasing in z, from Equation 4.3 and Definition 3 we have

$$\frac{\partial}{\partial z} L^{(\alpha)}_{-1/2}(z) = -L^{(\alpha+1)}_{-3/2}(z) = -\frac{\Gamma(\alpha+1/2)}{\Gamma(-1/2)} \cdot \frac{{}_1F_1(3/2;\alpha+2;z)}{\Gamma(\alpha+2)}\,.$$

For $\alpha > 0$ and $z < 0$, from Equation 4.5 it follows that ${}_1F_1(3/2;\alpha+2;z) > 0$. Since $\Gamma(-1/2) < 0$ and all remaining terms are positive, it follows that $-L^{(\alpha+1)}_{-3/2}(z) > 0$. Thus, $L^{(\alpha)}_{-1/2}(z)$ is monotonically increasing in z.

(b) Proofs that $L^{(\alpha)}_{1/2}(z)$ is positive and monotonically decreasing are very similar to the proofs in part (a), and will be omitted. To address concavity, we observe the second derivative of $L^{(\alpha)}_{1/2}(z)$:

$$\frac{\partial^2}{\partial z} L^{(\alpha)}_{1/2}(z) = L^{(\alpha+2)}_{-3/2}(z) = \frac{\Gamma(\alpha+3/2)}{\Gamma(-1/2)} \cdot \frac{{}_1F_1(3/2;\alpha+3;z)}{\Gamma(\alpha+3)}\,.$$

Similarly to part (a), from Equation 4.5, Definition 2, and basic properties of the gamma function it follows that $L^{(\alpha+2)}_{-3/2}(z) < 0$ for $\alpha > 0$ and $z < 0$. Therefore, $L^{(\alpha)}_{1/2}(z)$ is concave in z. □

Lemma 8 *For $\alpha > 0$ and $z < 0$, $L^{(\alpha+1)}_{1/2}(z) \approx L^{(\alpha)}_{1/2}(z)$.*

Proof From the recurrence relation in Equation 4.4 we obtain

$$L^{(\alpha+1)}_{1/2}(z) = L^{(\alpha+1)}_{-1/2}(z) + L^{(\alpha)}_{1/2}(z)\,.$$

Therefore, to prove the lemma it needs to be shown that for $z < 0$,

$$\lim_{\alpha\to\infty} L^{(\alpha)}_{-1/2}(z) = 0\,. \tag{4.6}$$

From Definition 3 and Equation 4.5 we have

$$L^{(\alpha)}_{-1/2}(z) = \frac{e^{-z}}{\Gamma(1/2)} \cdot \frac{\Gamma(\alpha+1/2)}{\Gamma(\alpha+1)} \cdot {}_1F_1(\alpha+1/2;\alpha+1;-z)\,.$$

From the asymptotic expansion in [66, p. 16, adapted],

$${}_1F_1\left(\frac{1}{2}n;\frac{1}{2}(n+b);x\right) = e^x\left(1+O(n^{-1})\right), \tag{4.7}$$

where n is large and $x \geq 0$, it follows that $\lim_{\alpha\to\infty} {}_1F_1(\alpha+1/2;\alpha+1;-z) < \infty$. Thus, to prove Equation 4.6 and the lemma it remains to be shown that

$$\lim_{\alpha\to\infty} \frac{\Gamma(\alpha+1/2)}{\Gamma(\alpha+1)} = 0\,. \tag{4.8}$$

As in the proof of Lemma 6, from the inequalities derived in [76] we have

$$\sqrt{\beta-1} \leq \sqrt{2}\,\frac{\Gamma\left(\frac{\beta+1}{2}\right)}{\Gamma\left(\frac{\beta}{2}\right)} \leq \sqrt{\beta}\,,$$

where $\beta > 1$. Applying inversion and substituting β with $2\alpha+1$ yields the limit in Equation 4.8. □

Lemma 9 *For $\alpha > 1/2$ and $z < 0$:*

(a) $\lim_{z\to-\infty} L^{(\alpha)}_{-3/2}(z) = 0$*, and*

(b) $\lim_{\alpha\to\infty} L^{(\alpha)}_{-3/2}(z) = 0$.

Proof (a) From Definition 3 we have

$$L^{(\alpha)}_{-3/2}(z) = \frac{\Gamma(\alpha-1/2)}{\Gamma(-1/2)} \cdot \frac{{}_1F_1(3/2;\alpha+1;z)}{\Gamma(\alpha+1)}\,. \tag{4.9}$$

The following property [1, p. 504, Equation 13.1.5],

$${}_1F_1\left(a;b;z\right) = \frac{\Gamma(b)}{\Gamma(b-a)}(-z)^a\left(1+O(|z|^{-1})\right) \quad (z<0),$$

when substituted into Equation 4.9, taking $a = 3/2$ and $b = \alpha+1$, yields

$$L^{(\alpha)}_{-3/2}(z) = \frac{1}{\Gamma(-1/2)}(-z)^{-3/2}\left(1+O(|z|^{-1})\right). \tag{4.10}$$

From Equation 4.10 the desired limit directly follows.

(b) The proof of part (b) is analogous to the proof of Lemma 8, that is, Equation 4.6. From Definition 3 and Equation 4.5, after applying the expansion from [66] given in Equation 4.7, it remains to be shown that

$$\lim_{\alpha\to\infty} \frac{\Gamma(\alpha - 1/2)}{\Gamma(\alpha + 1)} = 0\,. \tag{4.11}$$

Since $\Gamma(\alpha - 1/2) < \Gamma(\alpha + 1/2)$ for every $\alpha \geq 2$, the desired limit in Equation 4.11 follows directly from Equation 4.8. □

4.4.6 The Main Result

This section restates and proves our main theoretical result.

Theorem 9 *Let $\lambda_{d,1} = \mu_{\chi(d)} + c_1\sigma_{\chi(d)}$ and $\lambda_{d,2} = \mu_{\chi(d)} + c_2\sigma_{\chi(d)}$, where $d \in \mathbb{N}^+$, $c_1, c_2 \leq 0$, $c_1 < c_2$, and $\mu_{\chi(d)}$ and $\sigma_{\chi(d)}$ are the mean and standard deviation of the Chi distribution with d degrees of freedom, respectively. Define*

$$\Delta\mu_d(\lambda_{d,1}, \lambda_{d,2}) = \mu_{\chi(d,\lambda_{d,2})} - \mu_{\chi(d,\lambda_{d,1})}\,,$$

where $\mu_{\chi(d,\lambda_{d,i})}$ is the mean of the noncentral Chi distribution with d degrees of freedom and noncentrality parameter $\lambda_{d,i}$ ($i \in \{1, 2\}$).

There exists $d_0 \in \mathbb{N}$ such that for every $d > d_0$,

$$\Delta\mu_d(\lambda_{d,1}, \lambda_{d,2}) > 0\,, \tag{4.12}$$

and

$$\Delta\mu_{d+2}(\lambda_{d+2,1}, \lambda_{d+2,2}) > \Delta\mu_d(\lambda_{d,1}, \lambda_{d,2})\,. \tag{4.13}$$

Proof To prove Equation 4.12, we observe that for $d > 2$,

$$\begin{aligned}
\Delta\mu_d(\lambda_{d,1}, \lambda_{d,2}) &= \mu_{\chi(d,\lambda_{d,2})} - \mu_{\chi(d,\lambda_{d,1})} \\
&= \sqrt{\frac{\pi}{2}}\; L_{1/2}^{(\frac{d}{2}-1)}\left(-\frac{\lambda_{d,2}^2}{2}\right) - \sqrt{\frac{\pi}{2}}\; L_{1/2}^{(\frac{d}{2}-1)}\left(-\frac{\lambda_{d,1}^2}{2}\right) \\
&> 0\,,
\end{aligned}$$

where the last inequality holds for $d > 2$ because $\lambda_{d,1} < \lambda_{d,2}$, and $L_{1/2}^{(d/2-1)}(z)$ is a monotonically decreasing function in $z < 0$ for $d/2 - 1 > 0$ (Lemma 7).

In order to prove Equation 4.13, we will use approximate values of noncentrality parameters $\lambda_{d,1}$ and $\lambda_{d,2}$. Let $\widehat{\lambda}_{d,1} = \sqrt{d} + c_1/\sqrt{2}$, and $\widehat{\lambda}_{d,2} = \sqrt{d} + c_2/\sqrt{2}$. From Lemma 6 it follows that $\widehat{\lambda}_{d,1} \approx \lambda_{d,1}$ and $\widehat{\lambda}_{d,2} \approx \lambda_{d,2}$. Thus, by proving that there exists $d_2 \in \mathbb{N}$ such that for every $d > d_2$,

$$\Delta\mu_{d+2}(\widehat{\lambda}_{d+2,1}, \widehat{\lambda}_{d+2,2}) > \Delta\mu_d(\widehat{\lambda}_{d,1}, \widehat{\lambda}_{d,2})\,, \tag{4.14}$$

we prove that there exists $d_1 \in \mathbb{N}$ such that for every $d > d_1$ Equation 4.13 holds. The existence of such d_1, when approximations are used as function arguments, is ensured by the fact that $L_{1/2}^{(\alpha)}(z)$ is a monotonically decreasing *concave* function in z (Lemma 7), and by the transition from α to $\alpha + 1$ having an insignificant impact on the value of the Laguerre function for large enough α (Lemma 8). Once Equation 4.14 is proven, Equations 4.12 and 4.13 will hold for every $d > d_0$, where $d_0 = \max(2, d_1)$.

To prove Equation 4.14, from Equation 4.2 it follows we need to show that

$$\begin{aligned} & L_{1/2}^{(d/2)}\left(-\frac{1}{2}\left(\sqrt{d+2}+c_2/\sqrt{2}\right)^2\right) - L_{1/2}^{(d/2)}\left(-\frac{1}{2}\left(\sqrt{d+2}+c_1/\sqrt{2}\right)^2\right) \quad (4.15) \\ > \; & L_{1/2}^{(d/2-1)}\left(-\frac{1}{2}\left(\sqrt{d}+c_2/\sqrt{2}\right)^2\right) - L_{1/2}^{(d/2-1)}\left(-\frac{1}{2}\left(\sqrt{d}+c_1/\sqrt{2}\right)^2\right). \end{aligned}$$

We observe the second derivative of $L_{1/2}^{(\alpha)}(z)$:

$$\frac{\partial^2}{\partial z} L_{1/2}^{(\alpha)}(z) = L_{-3/2}^{(\alpha+2)}(z)\,.$$

Since $L_{-3/2}^{(\alpha+2)}(z)$ tends to 0 as $z \to -\infty$, and tends to 0 also as $\alpha \to \infty$ (Lemma 9), it follows that the two Laguerre functions on the left side of Equation 4.15 can be approximated by a linear function with an arbitrary degree of accuracy for large enough d. More precisely, since $L_{1/2}^{(\alpha)}(z)$ is monotonically decreasing in z (Lemma 7) there exist $a, b \in \mathbb{R}$, $a > 0$, such that the left side of Equation 4.15, for large enough d, can be replaced by

$$\begin{aligned} & -a\left(-\frac{1}{2}\left(\sqrt{d+2}+c_2/\sqrt{2}\right)^2\right) + b \; - \left(-a\left(-\frac{1}{2}\left(\sqrt{d+2}+c_1/\sqrt{2}\right)^2\right) + b\right) \\ = \; & \frac{a}{2}\left(\sqrt{d+2}+c_2/\sqrt{2}\right)^2 - \frac{a}{2}\left(\sqrt{d+2}+c_1/\sqrt{2}\right)^2. \quad (4.16) \end{aligned}$$

From Lemma 8 it follows that the same linear approximation can be used for the right side of Equation 4.15, replacing it by

$$\frac{a}{2}\left(\sqrt{d}+c_2/\sqrt{2}\right)^2 - \frac{a}{2}\left(\sqrt{d}+c_1/\sqrt{2}\right)^2. \quad (4.17)$$

After substituting the left and right side of Equation 4.15 with Equations 4.16 and 4.17, respectively, it remains to be shown that

$$\begin{aligned} & \frac{a}{2}\left(\sqrt{d+2}+c_2/\sqrt{2}\right)^2 - \frac{a}{2}\left(\sqrt{d+2}+c_1/\sqrt{2}\right)^2 \\ & > \frac{a}{2}\left(\sqrt{d}+c_2/\sqrt{2}\right)^2 - \frac{a}{2}\left(\sqrt{d}+c_1/\sqrt{2}\right)^2. \quad (4.18) \end{aligned}$$

Multiplying both sides by $\sqrt{2}/a$, moving the right side to the left, and applying algebraic simplification reduces Equation 4.18 to

$$\Big(c_2 - c_1\Big)\Big(\sqrt{d+2} - \sqrt{d}\Big) > 0,$$

which holds for $c_1 < c_2$, thus concluding the proof. □

4.5 Discussion

This section will discuss several additional considerations and related work regarding the geometry of high-dimensional spaces and the behavior of data distributions within them. First, let us consider the geometric upper limit to the number of points that point $\mathbf{x}$ can be a nearest neighbor of, in Euclidean space. In one dimension, this number is 2, in two dimensions it is 5, while in 3 dimensions it equals 11 [208]. Generally, for Euclidean space of dimensionality d this number is equal to the *kissing number*, which is the maximal number of hyperspheres that can be placed to touch a given hypersphere without overlapping, with all hyperspheres being of the same size.[9] Exact kissing numbers for arbitrary d are generally not known, however there exist bounds which imply that they progress exponentially with d [150, 234]. Furthermore, when considering k nearest neighbors for $k > 1$, the bounds become even larger. Therefore, only for very low values of d geometrical constraints of vector space prevent hubness. On the other hand, for higher values of d hubness may or may not occur, and the geometric bounds, besides providing "room" for hubness (even for values of k as low as 1) do not contribute much in fully characterizing the hubness phenomenon. Therefore, in high dimensions the behavior of *data distributions* needed to be studied.

We focus the rest of the discussion around the following important result,[10] drawing parallels with our results and analysis, and extending existing interpretations.

Theorem 10 [143, p. 803, Theorem 3, adapted]
Let $\mathbf{x}^{(i)} = (x_1^{(i)}, \ldots, x_d^{(i)})$, $i = 0, \ldots, n$ *be a sample of* $n+1$ *iid points from a distribution* $\mathbf{F}(\mathbf{X})$, $\mathbf{X} = (X_1, \ldots, X_d) \in \mathbb{R}^d$. *Assume that* $\mathbf{F}$ *is of the form* $\mathbf{F}(\mathbf{X}) = \prod_{k=1}^{d} \mathcal{F}(X_k)$, *that is, the coordinates* $X_1, \ldots, X_d$ *are iid. Let the distance measure be of the form* $D(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) = \sum_{k=1}^{d} g(x_k^{(i)}, x_k^{(j)})$. *Let* $N_1^{n,d}$ *denote the number of points among* $\{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(n)}\}$ *whose nearest neighbor is* $\mathbf{x}^{(0)}$.

Suppose $0 < \mathrm{Var}(g(X,Y)) < \infty$ *and set*

$$\beta = \mathrm{Correlation}(g(X,Y), g(X,Z))\,, \tag{4.19}$$

where X, Y, Z *are iid with common distribution* $\mathcal{F}$ *(the marginal distribution of* X_k*).*

[9] If ties are disallowed, it may be necessary to subtract 1 from the kissing number to obtain the maximum of N_1.

[10] A theorem that is effectively a special case of this result was proven previously [144, Theorem 7] for continuous distributions with finite kurtosis and Euclidean distance.

(a) *If* $\beta = 0$ *then*

$$\lim_{n\to\infty} \lim_{d\to\infty} N_1^{n,d} = \text{Poisson}(\lambda = 1) \;\; \textit{in distribution} \tag{4.20}$$

and

$$\lim_{n\to\infty} \lim_{d\to\infty} \text{Var}(N_1^{n,d}) = 1\,. \tag{4.21}$$

(b) *If* $\beta > 0$ *then*

$$\lim_{n\to\infty} \lim_{d\to\infty} N_1^{n,d} = 0 \;\; \textit{in distribution} \tag{4.22}$$

while

$$\lim_{n\to\infty} \lim_{d\to\infty} \text{Var}(N_1^{n,d}) = \infty\,. \tag{4.23}$$

What is exceptional in this theorem are Equations 4.22 and 4.23. According to the interpretation in [209], they suggest that if the number of dimensions is large relative to the number of points, one may expect to have a large proportion of points with N_1 equaling 0, and a small proportion of points with high N_1 values, that is, hubs.[11] Trivially, Equation 4.23 also holds for N_k with $k > 1$, since for any point $\mathbf{x}$, $N_k(\mathbf{x}) \geq N_1(\mathbf{x})$.

The setting involving iid normal random data and Euclidean distance, used in our Theorem 9 (and, generally, any iid random data distribution with Euclidean distance), fulfills the conditions of Theorem 10 for Equations 4.22 and 4.23 to be applied, since the correlation parameter $\beta > 0$. This correlation exists because, for example, if we view vector component variable X_j ($j \in \{1, 2, \ldots, d\}$) and the distribution of data points within it, if a random point drawn from X_j is closer to the mean of X_j it is more likely to be close to other random points drawn from X_j, and vice versa, producing the case $\beta > 0$.[12] Therefore, Equations 4.22 and 4.23 from Theorem 10 directly apply to the setting studied in Theorem 9, providing asymptotic evidence for hubness. However, since the proof of Equations 4.22 and 4.23 in Theorem 10 relies on applying the central limit theorem to the (normalized) distributions of pairwise distances between vectors $\mathbf{x}^{(i)}$ and $\mathbf{x}^{(j)}$ ($0 \leq i \neq j \leq n$) as $d \to \infty$ (obtaining limit distance distributions which are Gaussian), the results of Theorem 10 are inherently asymptotic in nature. Theorem 9, on the other hand, describes the behavior of distances in finite dimensionalities,[13] providing the means to characterize the behavior of N_k in high, but finite-dimensional space. What remains to be done is to formally connect Theorem 9 with the skewness of N_k in finite dimensionalities, for example by observing point $\mathbf{x}$

[11] Reversing the order of limits, which corresponds to having a large number of points relative to the number of dimensions, produces the same asymptotic behavior as in Equations 4.20 and 4.21, that is, no hubness, in all studied settings.

[12] Similarly to Section 4.3.2, this argument holds for unimodal distributions of component variables; for multimodal distributions the driving force behind nonzero β is the proximity to a peak in the probability density function.

[13] Although the statement of Theorem 9 relies on dimensionality being greater than some d_0 which is finite, but can be arbitrarily high, empirical evidence suggests that actual d_0 values are low, often equaling 0.

with a fixed position relative to the data distribution mean (the origin) across dimensionalities, in terms of being at distance $\mu_{\chi(d)} + c\,\sigma_{\chi(d)}$ from the origin, and expressing how the probability of $\mathbf{x}$ to be the nearest neighbor (or among the k nearest neighbors) of a randomly drawn point changes with increasing dimensionality.[14] We address this investigation as a point of future work.

Returning to Theorem 10 and the value of β from Equation 4.19, as previously discussed, $\beta > 0$ signifies that the position of a vector component value makes a difference when computing distances between vectors, causing some component values to be more "special" than others. Another contribution of Theorem 9 is that it illustrates how the individual differences in component values combine to make positions of whole vectors more special (by being closer to the data center). On the other hand, if $\beta = 0$ no point can have a special position with respect to all others. In this case, Equations 4.20 and 4.21 hold, which imply there is no hubness. This setting is relevant, for example, to points being generated by a Poisson process which spreads the vectors uniformly over $\mathbb{R}^d$, where all positions within the space (both at component-level and globally) become basically equivalent. Although it does not directly fit into the framework of Theorem 10, the same principle can be used to explain the absence of hubness for normally distributed data and cosine distance from Section 4.2: in this setting no vector is more spatially central than any other. Equations 4.20 and 4.21, which imply no hubness, hold for many more "centerless" settings, including random graphs, settings with exchangeable distances, and d-dimensional toruses [143].

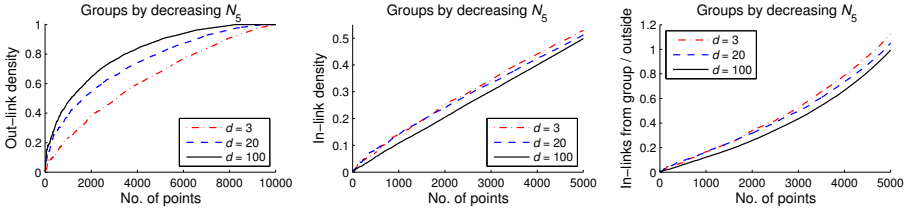
The following two subsections will address several additional issues concerning the interpretation of Theorem 10.

4.5.1 Nearest-Neighbor Graph Structure

The interpretation of Theorem 10 from [209] may be understood in the sense that, with increasing dimensionality, *very few* exceptional points become hubs, while all others are relegated to antihubs. In this section we will empirically examine the structural change of the k-NN graph as the number of dimensions increases. We will also discuss and consolidate different notions node centrality in the k-NN graph, and their dependence on the dimensionality of data.

First, as in Section 4.2, we consider $n = 10000$ iid uniform random data points of different dimensionality. Let us observe hubs, that is, points with highest N_5, collected in groups of progressively increasing size: $5, 10, 15, \ldots, 10000$. In analogy with the notion of network density from social network analysis [189], we define group *out-link density* as the proportion of the number of arcs that originate and end in nodes from the group, and the total number of arcs that originate from nodes in the group. Conversely, we define group *in-link density* as the proportion of the number of arcs that originate and end in nodes from the group, and the total number of arcs that *end* in nodes from the group. Figure 4.5(a, b) shows the out-link and in-link densities of groups

[14] For $c < 0$ we expect this probability to increase since $\mathbf{x}$ is closer to the data distribution mean, and becomes closer to other points as dimensionality increases.



(a) (b) (c)

Figure 4.5: (a) Out-link, and (b) in-link densities of groups of hubs with increasing size; (c) ratio of the number of in-links originating from points within the group and in-links originating from outside points, for iid uniform random data with dimensionality $d = 3,\ 20,\ 100$

Slika 4.5: Gustina (a) izlaznih i (b) ulaznih grana za grupe habova rastućih kardinaliteta; (c) odnos između broja ulaznih grana koje polaze iz tačaka unutar grupe i broja ulaznih grana koje polaze iz spoljašnjih tačaka, za *iid* uniforman skup slučajnih tačaka dimenzionalnosti $d = 3,\ 20,\ 100$

of strongest hubs in iid uniform random data (similar tendencies can be observed with other synthetic data distributions). It can be seen in Figure 4.5(a) that hubs are more cohesive in high dimensions, with more of their out-links leading to other hubs. On the other hand, Figure 4.5(b) suggests that hubs also receive more in-links from non-hub points in high dimensions than in low dimensions. Moreover, Figure 4.5(c), which plots the ratio of the number of in-links that originate within the group, and the number of in-links which originate outside, shows that hubs receive a larger proportion of in-links from non-hub points in high dimensions than in low dimensions. We have reported our findings for $k = 5$, however similar results are obtained with other values of k.

Overall, it can be said that in high dimensions hubs receive more in-links than in low dimensions from both hubs and non-hubs, and that the range of influence of hubs gradually widens as dimensionality increases. We can therefore conclude that the transition of hubness from low to high dimensionalities is "smooth," both in the sense of the change in the overall distribution of N_k, and the change in the degree of influence of data points, as expressed by the above analysis of links.

So far, we have viewed hubs primarily through their exhibited high values of N_k, that is, high degree centrality in the k-NN directed graph. However, (scale-free) network analysis literature often attributes other properties to hubs [5], viewing them as nodes that are important for preserving network structure due to their central positions within the *graph*, indicated, for example, by their *betweenness centrality* [189]. On the other hand, as discussed in [125], in both synthetic and real-world networks high-degree nodes do not necessarily need to correspond to nodes that are central in the graph, that is, high-degree nodes can be concentrated at the *periphery* of the network and bear little structural significance. For this reason, we computed the betweenness centrality of nodes in k-NN graphs of synthetic data sets studied in this chapter, and

calculated its Spearman correlation with node degree, denoting the measure by $C_{\mathrm{BC}}^{N_k}$. For iid uniform data ($k = 5$), when $d = 3$ the measured correlation is $C_{\mathrm{BC}}^{N_5} = 0.311$, when $d = 20$ the correlation is $C_{\mathrm{BC}}^{N_5} = 0.539$, and finally when $d = 100$ the correlation rises to $C_{\mathrm{BC}}^{N_5} = 0.647$.[15] This suggests that with increasing dimensionality the centrality of nodes increases not only in the sense of higher node degree or spatial centrality of vectors (as discussed in Section 4.3.2), but also in the structural graph-based sense. In Section 5.3 we will provide further support for this observation on real data.

4.5.2 Rate of Convergence and the Role of Boundaries

On several occasions, the authors of Theorem 10 have somewhat downplayed the significance of equations 4.22 and 4.23 [209, 143], citing empirically observed slow convergence [131], even to the extent of not observing significant differences between hubness in the Poisson process and iid uniform cube settings. However, results in the preceding sections of this chapter suggest that this convergence is fast enough to produce notable hubness in high-dimensional data. In order to directly illustrate the difference between a setting which provides no possibility for spatial centrality of points, and one that does, we will observe the Poisson process vs. the iid unit cube setting. We will be focusing on the location of the nearest neighbor of a point from the cube, that is, on determining whether it stays within the boundaries of the cube as dimensionality increases.

Lemma 10 *Let points be spread in $\mathbb{R}^d$ according to a Poisson process with constant intensity $\lambda > 1$. Observe a unit hypercube $C \subset \mathbb{R}^d$, and an arbitrary point $\mathbf{x} = (x_1, x_2, \ldots, x_d) \in C$, generated by the Poisson process. Let $p_{\lambda,d}$ denote the probability that the nearest neighbor of $\mathbf{x}$, with respect to Euclidean distance, is not situated in C. Then,*

$$\lim_{d\to\infty} p_{\lambda,d} = 1\,.$$

Proof Out of the $3^d - 1$ unit hypercubes that surround C, let us observe only the $2d$ hypercubes that differ from C only in one coordinate. We will restrict the set of considered points to these $2d$ cubes, and prove that the probability that the nearest neighbor of $\mathbf{x}$ comes from one of the $2d$ cubes, $\widehat{p}_{\lambda,d}$, converges to 1 as $d \to \infty$. From this, the convergence of $p_{\lambda,d}$ directly follows, since $p_{\lambda,d} \geq \widehat{p}_{\lambda,d}$.

Let $\widehat{p}_{\lambda,d}(i)$ denote the probability that the nearest neighbor of $\mathbf{x}$ comes from one of the two hypercubes which differ from C only in the ith coordinate, $i \in \{1, 2, \ldots, d\}$. For a given coordinate i and point $\mathbf{x}$, let the *1-dimensional nearest neighbor* of x_i denote the closest y_i value of all other points $\mathbf{y}$ from the Poisson process, observed when all coordinates except i are disregarded. Conversely, for a given coordinate i and point $\mathbf{x}$, let the *$(d-1)$-dimensional nearest neighbor* of $(x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_d)$ denote the closest point $(y_1, \ldots, y_{i-1}, y_{i+1}, \ldots, y_d)$, obtained when coordinate i is disregarded.

[15] Betweenness centrality is computed on directed k-NN graphs. We obtained similar correlations when undirected graphs were used.

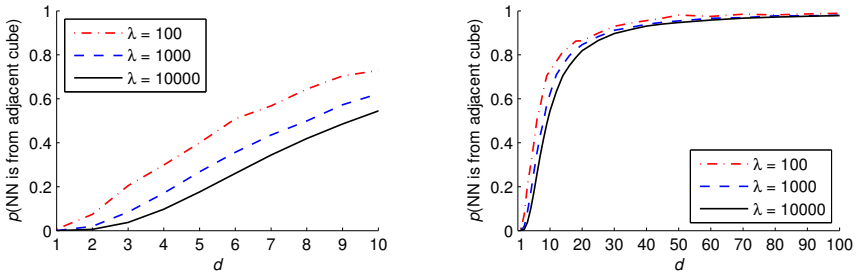


Figure 4.6: Probability that the nearest neighbor of a point from the unit hypercube originates from one of the adjacent hypercubes, for Poisson processes with $\lambda = 100, 1000$, and 10000 expected points per hypercube, obtained through simulation and averaged over 10 runs

Slika 4.6: Verovatnoća da najbliži sused tačke iz jedinične hiperkocke pripada nekoj od susednih hiperkocki, za Poasonove procese sa $\lambda = 100,\ 1000$ i 10000 očekivanih tačaka po hiperkocki, dobijena kao prosek 10 simulacija

Observing the ith coordinate only, the probability for the 1-dimensional nearest neighbor of x_i to come from one of the surrounding unit intervals is $\widehat{p}_{\lambda,1}$. Although small, $\widehat{p}_{\lambda,1} > 0$. Assuming this event has occurred, let $\mathbf{y} \in \mathbb{R}^d$ be the point whose component y_i is the 1-dimensional nearest neighbor of x_i that is not within the unit interval containing x_i. Let r_λ denote the probability that the remaining coordinates of $\mathbf{y}$, $(y_1, \ldots, y_{i-1}, y_{i+1}, \ldots, y_d)$, constitute a $(d-1)$-dimensional nearest neighbor of $(x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_d)$, within the confines of C. It can be observed that r_λ is strictly greater than 0, inversely proportional to λ (roughly equaling $1/(\lambda - 1)$), and independent of d. Thus, $\widehat{p}_{\lambda,d}(i) \geq \widehat{p}_{\lambda,1} \cdot r_\lambda > 0$.

Let $\widehat{q}_{\lambda,d} = 1 - \widehat{p}_{\lambda,d}$, the probability that the nearest neighbor of $\mathbf{x}$ comes from C (recall the restriction of the location of the nearest neighbor to C and its immediately surrounding $2d$ hypercubes). In light of the above, $\widehat{q}_{\lambda,d} = \prod_{i=1}^{d} \widehat{q}_{\lambda,d}(i) = \prod_{i=1}^{d}(1 - \widehat{p}_{\lambda,d}(i))$. Since each $\widehat{p}_{\lambda,d}(i)$ is bounded from below by a constant strictly greater than 0 (which depends only on λ), each $\widehat{q}_{\lambda,d}(i)$ is bounded from above by a constant strictly smaller than 1. It follows that $\lim_{d\to\infty} \widehat{q}_{\lambda,d} = 0$, and therefore $\lim_{d\to\infty} \widehat{p}_{\lambda,d} = 1$. □

To illustrate the rate of convergence in Lemma 10, Figure 4.6 plots the empirically observed probabilities that the nearest neighbor of a point from a unit hypercube originates in one of the $2d$ immediately adjacent unit hypercubes ($\widehat{p}_{\lambda,d}$). It can be seen that the probability that the nearest neighbor comes from outside of the cube quickly becomes close to 1 as dimensionality increases. Please note that due to feasibility of simulation the plots in Figure 4.6 represent the empirical lower bounds (that is, the empirical estimates of $\widehat{p}_{\lambda,d}$) of the true probabilities from Lemma 10 ($p_{\lambda,d}$) by considering only the $2d$ immediately adjacent hypercubes.

The above indicates that when boundaries are introduced in high dimensions, the setting completely changes, in the sense that new nearest neighbors need to be located inside the boundaries. Under such circumstances, points which are closer to the center have a better chance of becoming nearest neighbors, with the mechanism described in previous sections. Another implication of the above observations, stemming from Figure 4.6, is that the number of dimensions does not need to be very large compared to the number of points for the setting to change. As for boundaries, they can be viewed as a dual notion to spatial centrality discussed earlier. With Poisson processes and cubes this duality is rather straightforward, however for continuous distributions in general there exist no boundaries in a strict mathematical sense. Nevertheless, since data sets contain a finite number of points, it can be said that "practical" boundaries exist in this case as well.

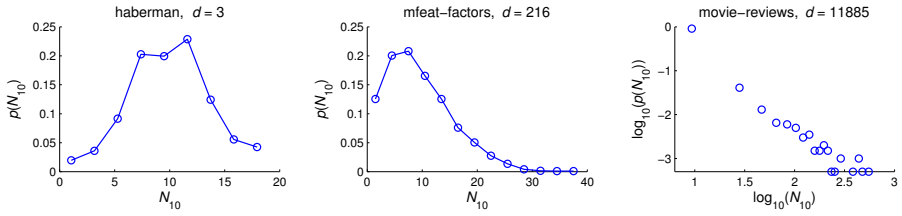
Chapter 5

Hubness and Machine Learning

This chapter will continue the analysis of the hubness phenomenon from Chapter 4 by generalizing its conclusions to real data sets from various application areas, also considering points with low k-occurrences (the *antihubs*) and the interaction of hubness with information provided by class labels, and discussing the impact of the phenomenon on various machine-learning algorithms. After reviewing related work in the next section, the hubness phenomenon is observed and measured on real data in Section 5.2, introducing the large collection of real-world data sets from well-known repositories that is used in the subsequent experiments and demonstrations. Section 5.3 explains hubness in real data sets, linking it with the *intrinsic* dimensionality of data, and supporting the presented analysis with empirical evidence obtained from the data collection. Section 5.4 discusses hubs and their opposites – antihubs – and the relationships between hubs, antihubs, and different notions of outliers. Section 5.5 provides further support for the link between intrinsic dimensionality and hubness, and demonstrates that dimensionality reduction may not constitute an easy mitigation of the phenomenon. Section 5.6 explores the impact of hubness on common supervised, semi-supervised, and unsupervised machine-learning algorithms, showing that the information provided by hubness can be used to significantly affect the success of the generated models. Section 5.7 summarizes the main points of the chapter, and the possibilities for future work.

5.1 Related Work

As was already mentioned in the previous chapter, the hubness phenomenon has been recently observed in several application areas involving sound and image data [12, 9, 49, 84]. Regarding machine-learning and data-mining tasks, hubness was briefly mentioned in the context of graph construction for semi-supervised learning [93]. In addition, there have been attempts to explicitly avoid the influence of hubs in 1-NN time-series classification, apparently without clear awareness about the existence of the phenomenon [89], and to account for possible skewness of the distribution of N_1 in re-



(a) (b) (c)

Figure 5.1: Empirical distribution of N_{10} for three real data sets of different dimensionalities
Slika 5.1: Empirijska distribucija N_{10} za tri skupa stvarnih podataka različite dimenzionalnosti

verse nearest-neighbor search [195],[1] where $N_k(\mathbf{x})$ denotes the number of times point $\mathbf{x}$ occurs among the k nearest neighbors of all other points in the data set. None of the mentioned papers, however, successfully analyze the causes of hubness or generalize it to other applications.

5.2 Observing Hubness in Real Data

To illustrate the hubness phenomenon on real data, let us consider the empirical distribution of N_k ($k = 10$) for three real data sets, given in Figure 5.1. As in the previous section, a considerable increase in the skewness of the distributions can be observed with increasing dimensionality.

In all, we examined 50 real data sets from well known sources, belonging to three categories: UCI multidimensional data, gene expression data, and textual data in the bag-of-words representation,[2] listed in Table 5.1. The table includes columns that describe data set sources (2nd column), basic statistics (data transformation (3rd column): whether standardization was applied, or for textual data the used bag-of-words document representation; the number of points (4th column, n); dimensionality (5th column, d); the number of classes (7th column)), and the distance metric used (Euclidean or cosine, 8th column). We took care to ensure that the choice of distance measure

[1] Reverse nearest-neighbor queries retrieve data points that have the query point $\mathbf{q}$ as their nearest neighbor.

[2] Data sources are the University of California, Irvine (UCI) Machine Learning Repository, Kent Ridge (KR) Bio-Medical Data Set Repository, dmoz Open Directory, and `www.cs.cornell.edu/People/pabo/movie-review-data/` (PaBo). We used the movie review polarity data set v2.0 initially introduced in [152], while the computers and sports data sets are described in Chapter 8. Preprocessing of all text data sets (except dexter, which is already preprocessed) involved stop-word removal and stemming using the Porter stemmer [158]. Documents were transformed into the bag-of-words representation with word weights being either term frequencies (`tf`), or term frequencies multiplied by inverse document frequencies (`tfidf`), with the choice based on independent experiments involving several classifiers. All term-frequency vectors were normalized to average document length.

Name	Src.	Trans.	n	d	d_{mle}	Cls.	Dist.	$S_{N_{10}}$	$S^S_{N_{10}}$	Clu.	$C^{N_{10}}_{\mathrm{dm}}$	$C^{N_{10}}_{\mathrm{cm}}$	$\widetilde{BN}_{10}$	$C^{N_{10}}_{BN_{10}}$	CAV
abalone	UCI	stan	4177	8	5.39	29	l_2	0.277	0.235	62	−0.047	−0.526	0.804	0.934	0.806
arcene	UCI	stan	100	10000	22.85	2	l_2	0.634	2.639	2	−0.559	−0.684	0.367	0.810	0.455
arrhythmia	UCI	stan	452	279	21.63	16	l_2	1.984	6.769	8	−0.867	−0.892	0.479	0.898	0.524
breast-w	UCI	stan	699	9	5.97	2	l_2	1.020	0.667	7	−0.062	−0.240	0.052	0.021	0.048
diabetes	UCI	stan	768	8	6.00	2	l_2	0.555	0.486	15	−0.479	−0.727	0.322	0.494	0.337
dorothea	UCI	none	800	100000	201.11	2	cos	2.355	1.016	19	−0.632	−0.672	0.108	0.236	0.092
echocardiogram	UCI	stan	131	7	4.92	2	l_2	0.735	0.438	5	−0.722	−0.811	0.372	0.623	0.337
ecoli	UCI	stan	336	7	4.13	8	l_2	0.116	0.208	8	−0.396	−0.792	0.223	0.245	0.193
gisette	UCI	none	6000	5000	149.35	2	cos	1.967	4.671	76	−0.667	−0.854	0.045	0.367	0.241
glass	UCI	stan	214	9	4.37	7	l_2	0.154	0.853	11	−0.430	−0.622	0.414	0.542	0.462
haberman	UCI	stan	306	3	2.89	2	l_2	0.087	−0.316	11	−0.330	−0.573	0.348	0.305	0.360
ionosphere	UCI	stan	351	34	13.57	2	l_2	1.717	2.051	18	−0.639	−0.832	0.185	0.464	0.259
iris	UCI	stan	150	4	2.96	3	l_2	0.126	−0.068	4	−0.275	−0.681	0.087	0.127	0.147
isolet1	UCI	stan	1560	617	13.72	26	l_2	1.125	6.483	38	−0.306	−0.760	0.283	0.463	0.352
mfeat-factors	UCI	stan	2000	216	8.47	10	l_2	0.826	5.493	44	−0.113	−0.688	0.063	0.001	0.145
mfeat-fourier	UCI	stan	2000	76	11.48	10	l_2	1.277	4.001	44	−0.350	−0.596	0.272	0.436	0.415
mfeat-karhunen	UCI	stan	2000	64	11.82	10	l_2	1.250	8.671	40	−0.436	−0.788	0.098	0.325	0.205
mfeat-morph	UCI	stan	2000	6	3.22	10	l_2	−0.153	0.010	44	−0.039	−0.424	0.324	0.306	0.397
mfeat-pixel	UCI	stan	2000	240	11.83	10	l_2	1.035	3.125	44	−0.210	−0.738	0.049	0.085	0.107
mfeat-zernike	UCI	stan	2000	47	7.66	10	l_2	0.933	3.389	44	−0.185	−0.657	0.235	0.252	0.400
musk1	UCI	stan	476	166	6.74	2	l_2	1.327	3.845	17	−0.376	−0.752	0.237	0.621	0.474
optdigits	UCI	stan	5620	64	9.62	10	l_2	1.095	3.789	74	−0.223	−0.601	0.044	0.097	0.168
ozone-eighthr	UCI	stan	2534	72	12.92	2	l_2	2.251	4.443	49	−0.216	−0.655	0.086	0.300	0.138
ozone-onehr	UCI	stan	2536	72	12.92	2	l_2	2.260	5.798	49	−0.215	−0.651	0.046	0.238	0.070
page-blocks	UCI	stan	5473	10	3.73	5	l_2	−0.014	0.470	72	−0.063	−0.289	0.049	−0.046	0.068
parkinsons	UCI	stan	195	22	4.36	2	l_2	0.729	1.964	8	−0.414	−0.649	0.166	0.321	0.256
pendigits	UCI	stan	10992	16	5.93	10	l_2	0.435	0.982	104	−0.062	−0.513	0.014	−0.030	0.156
segment	UCI	stan	2310	19	3.93	7	l_2	0.313	1.111	48	−0.077	−0.453	0.089	0.074	0.332
sonar	UCI	stan	208	60	9.67	2	l_2	1.354	3.053	8	−0.550	−0.771	0.286	0.632	0.461
spambase	UCI	stan	4601	57	11.45	2	l_2	1.916	2.292	49	−0.376	−0.448	0.139	0.401	0.271
spectf	UCI	stan	267	44	13.83	2	l_2	1.895	2.098	11	−0.616	−0.729	0.300	0.595	0.366
spectrometer	UCI	stan	531	100	8.04	10	l_2	0.591	3.123	17	−0.269	−0.670	0.200	0.225	0.242
vehicle	UCI	stan	846	18	5.61	4	l_2	0.603	1.625	25	−0.162	−0.643	0.358	0.435	0.586
vowel	UCI	stan	990	10	2.39	11	l_2	0.766	0.935	27	−0.252	−0.605	0.313	0.691	0.598
wdbc	UCI	stan	569	30	8.26	2	l_2	0.815	3.101	16	−0.449	−0.708	0.065	0.170	0.129
wine	UCI	stan	178	13	6.69	3	l_2	0.630	1.319	3	−0.589	−0.874	0.076	0.182	0.084
wpbc	UCI	stan	198	33	8.69	2	l_2	0.863	2.603	6	−0.688	−0.878	0.340	0.675	0.360
yeast	UCI	stan	1484	8	5.42	10	l_2	0.228	0.105	34	−0.421	−0.715	0.527	0.650	0.570
AMLALL	KR	none	72	7129	31.92	2	l_2	1.166	1.578	2	−0.868	−0.927	0.171	0.635	0.098
colonTumor	KR	none	62	2000	11.22	2	l_2	1.055	1.869	3	−0.815	−0.781	0.305	0.779	0.359
DLBCL	KR	none	47	4026	16.11	2	l_2	1.007	1.531	2	−0.942	−0.947	0.338	0.895	0.375
lungCancer	KR	none	181	12533	59.66	2	l_2	1.248	3.073	6	−0.537	−0.673	0.052	0.262	0.136
MLL	KR	none	72	12582	28.42	3	l_2	0.697	1.802	2	−0.794	−0.924	0.211	0.533	0.148
ovarian-61902	KR	none	253	15154	9.58	2	l_2	0.760	3.771	10	−0.559	−0.773	0.164	0.467	0.399
computers	dmoz	tf	697	1168	190.33	2	cos	2.061	2.267	26	−0.566	−0.731	0.312	0.699	0.415
dexter	UCI	none	300	20000	160.78	2	cos	3.977	4.639	13	−0.760	−0.781	0.301	0.688	0.423
mini-newsgroups	UCI	tfidf	1999	78273	226.43	20	cos	1.980	1.765	44	−0.422	−0.704	0.524	0.701	0.526
movie-reviews	PaBo	tf	2000	11885	54.95	2	cos	8.796	7.247	44	−0.604	−0.739	0.398	0.790	0.481
reuters-transcribed	UCI	tfidf	201	3029	234.68	10	cos	1.165	1.693	3	−0.781	−0.763	0.642	0.871	0.595
sports	dmoz	tf	752	1185	250.24	2	cos	1.629	2.543	27	−0.584	−0.736	0.260	0.604	0.373

Table 5.1: Real data sets
Tabela 5.1: Skupovi stvarnih podataka

and preprocessing (transformation) corresponds to a realistic scenario for the particular data set.

To characterize the asymmetry of N_k we use the standardized third moment of the distribution of k-occurrences,

$$S_{N_k} = \frac{\mathrm{E}(N_k - \mu_{N_k})^3}{\sigma_{N_k}^3},$$

where μ_{N_k} and σ_{N_k} are the mean and standard deviation of N_k, respectively. The corresponding (9th) column of Table 5.1, which shows the empirical $S_{N_{10}}$ values for the real data sets, indicates that the distributions of N_{10} for most examined data sets are skewed to the right.[3] The value of k is fixed at 10, but analogous observations can be made with other values of k.

It can be observed that some S_{N_k} values in Table 5.1 are quite high, indicating strong hubness in the corresponding data sets.[4] Moreover, computing the Spearman correlation between d and S_{N_k} over all 50 data sets reveals it to be strong (0.62), signifying that the relationship between dimensionality and hubness extends from synthetic to real data in general. On the other hand, careful scrutiny of the charts in Figure 5.1 and S_{N_k} values in Table 5.1 reveals that for real data the impact of dimensionality on hubness may not be as strong as could be expected after viewing hubness on synthetic data in Figure 4.1. Explanations for this observation will be given in the next section.

5.3 Explaining Hubness in Real Data

Results describing the origins of hubness given in the previous chapter were obtained by examining data sets that follow specific distributions and generated as iid samples from these distributions. To extend these results to real data, we need to take into account two additional factors: (1) real data sets usually contain dependent attributes, and (2) real data sets are usually clustered, that is, points are organized into groups produced by a mixture of distributions instead of originating from a single (unimodal) distribution.

To examine the first factor (dependent attributes), we adopt the approach from [61] used in the context of distance concentration. For each data set we randomly permute the elements within every attribute. This way, attributes preserve their individual distributions, but the dependencies between them are lost and the *intrinsic dimensionality* of data sets increases, becoming equal to their embedding dimensionality d [61]. In Table 5.1 (10th column) we give the empirical skewness, denoted as $S^S_{N_k}$, of the shuffled data. For the vast majority of high-dimensional data sets, $S^S_{N_k}$ is considerably higher than S_{N_k}, indicating that hubness actually depends on the intrinsic rather than embedding dimensionality. This provides an explanation for the apparent weaker influence of d on hubness in real data than in synthetic data sets, observed in Section 5.2.

[3] If $S_{N_k} = 0$ there is no skewness, positive (negative) values signify skewness to the right (left).

[4] For comparison, sample skewness values for iid uniform random data and Euclidean distance, shown in Figure 4.1(a–c), are 0.121, 1.541, and 5.445 for dimensionalities 3, 20, and 100, respectively. The values for iid normal data from Figure 4.1(d–f) are 0.118, 2.055, and 19.210.

	d	d_{mle}	$S_{N_{10}}$	$C_{\mathrm{dm}}^{N_{10}}$	$C_{\mathrm{cm}}^{N_{10}}$	$\widetilde{BN}_{10}$	$C_{BN_{10}}^{N_{10}}$
d_{mle}	0.87						
$S_{N_{10}}$	0.62	0.80					
$C_{\mathrm{dm}}^{N_{10}}$	−0.52	−0.60	−0.42				
$C_{\mathrm{cm}}^{N_{10}}$	−0.43	−0.48	−0.31	0.82			
$\widetilde{BN}_{10}$	−0.05	0.03	−0.08	−0.32	−0.18		
$C_{BN_{10}}^{N_{10}}$	0.32	0.39	0.29	−0.61	−0.46	0.82	
CAV	0.03	0.03	0.03	−0.14	−0.05	0.85	0.76

Table 5.2: Spearman correlations over 50 real data sets
Tabela 5.2: Spermanove korelacije nad 50 skupova stvarnih podataka

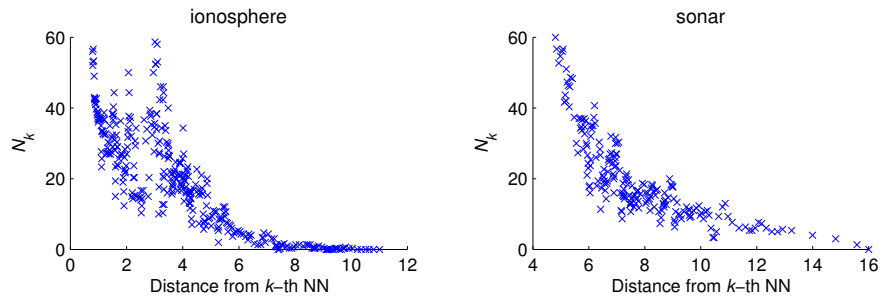
To examine the second factor (many groups), for every data set we measured: (i) the Spearman correlation, denoted as $C_{\mathrm{dm}}^{N_{10}}$ (12th column), of the observed N_k and distance from the data-set mean, and (ii) the correlation, denoted as $C_{\mathrm{cm}}^{N_{10}}$ (13th column), of the observed N_k and distance to the closest group mean. Groups are determined with K-means clustering, where the number of clusters for each data set, given in column 11 of Table 5.1, was determined by exhaustive search of values between 2 and $\lfloor\sqrt{n}\rfloor$, to maximize $C_{\mathrm{cm}}^{N_{10}}$.[5] In most cases, $C_{\mathrm{cm}}^{N_{10}}$ is considerably stronger than $C_{\mathrm{dm}}^{N_{10}}$. Consequently, in real data, hubs tend to be closer than other points to their respective cluster centers (which we verified by examining the individual scatter plots).

To further support the above findings, we include the 6th column (d_{mle}) to Table 5.1, corresponding to intrinsic dimensionality measured by the maximum likelihood estimator [120]. Next, we compute Spearman correlations between various measurements from Table 5.1 over all 50 examined data sets, given in Table 5.2. The observed skewness of N_k, besides being strongly correlated with d, is even more strongly correlated with the intrinsic dimensionality d_{mle}. Moreover, intrinsic dimensionality positively affects the correlations between N_k and the distance to the data-set mean / closest cluster mean, implying that in higher (intrinsic) dimensions the positions of hubs become increasingly localized to the proximity of centers.

Section 5.5, which discusses the interaction of hubness with dimensionality reduction, will provide even more support to the observation that hubness depends on intrinsic, rather than embedding dimensionality.

Recalling the discussion about the relationships between different notions of centrality in Section 4.5.1, we provide further support to the observation that the correlation between the N_k value and betweenness centrality of a point, $C_{\mathrm{BC}}^{N_{10}}$, increases with increasing (intrinsic) dimensionality by computing, over the 50 real data sets listed in

[5] We report averages of $C_{\mathrm{cm}}^{N_{10}}$ over 10 runs of K-means clustering with different random seeding, in order to reduce the effects of chance.



(a) (b)

Figure 5.2: Correlation between low N_k and outlier score ($k = 20$)
Slika 5.2: Korelacija između niskih vrednosti N_k i *outlier* koeficijenta ($k = 20$)

Table 5.1, the correlation between $C_{\mathrm{BC}}^{N_{10}}$ and $S_{N_{10}}$, finding it to be significant: 0.548.[6] This indicates that real data sets that exhibit strong skewness in the distribution of N_{10} also tend to have strong correlation between N_{10} and betweenness centrality of nodes, giving hubs a broader significance for the structure of the k-NN graphs constructed from the data.

5.4 Hubs and Outliers

The non-negligible variance of the distribution of distances to the data mean described in Section 4.3.2 has an additional "side": we also expect points farther from the mean and, therefore, with much lower observed N_k than the rest.[7] Such points correspond to the bottom-right parts of Figure 4.2(b, c, e, f), and will be referred to as *antihubs*. Since antihubs are far away from all other points, in high dimensions they can be regarded as distance-based *outliers* [203].

To further support the connection between antihubs and distance-based outliers, let us consider a common outlier score of a point as the distance from its kth nearest neighbor [203]. Low N_k values and high outlier scores are correlated as exemplified in Figure 5.2(a, b) (in their lower-right parts) for two data sets from Table 5.1.

Next, let us recall the iid normal random data setting, and the probability density function corresponding to observing a point at a specified distance from the mean, plotted in Figure 4.3. An analogous chart for real data is given in Figure 5.3, which shows the empirical distributions of distances from the closest cluster mean for three real data sets, as described in Section 5.3. In both figures it can be seen that in low dimensions

[6] This value is computed on directed k-NN graphs; when betweenness centrality is computed on undirected graphs, the correlation is even stronger: 0.585.

[7] Assuming the presence of hubs, the existence of points with low N_k is implied by the constant-sum property of N_k: for any data set D, $\sum_{\mathbf{x} \in D} N_k(\mathbf{x}) = k|D|$ [9].

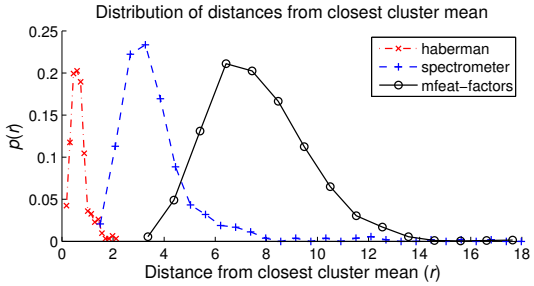


Figure 5.3: Probability density function of observing a point at distance r from the closest cluster mean for three real data sets
Slika 5.3: Funkcija gustine verovatnoće da se uoči tačka na udaljenosti r od najbližeg centra klastera, za tri skupa stvarnih podataka

the probability of observing a point near a center is quite high, while as dimensionality increases it becomes close to zero. If we now consider a *probabilistic* definition of an outlier as a point with a low probability of occurrence [203], in high dimensions hubs actually *are* outliers, as points closer to the distribution (or cluster) mean than the majority of other points. Therefore, somewhat counterintuitively, it can be said that hubs are points that reside in low-density regions of the high-dimensional data space and are at the same time close to many other points. On the other hand, distance-based outliers correspond to probabilistic outliers that are farther away from the center(s). It follows that the definitions of distance-based and probabilistic outliers significantly diverge from one another in high dimensions, with distance-based outliers only partially corresponding to probabilistic outliers. To prevent confusion in the rest of the dissertation, we shall continue to refer to "hubs" and "outliers" in the distance-based sense. Outliers will be analyzed further in Section 5.6.3.

5.5 Hubness and Dimensionality Reduction

In this section we elaborate further on the interplay of skewness of N_k and intrinsic dimensionality by considering dimensionality reduction (DR) techniques. The main question motivating the discussion in this section is whether dimensionality reduction can alleviate the issue of the skewness of k-occurrences altogether. We leave a more detailed and general investigation of the interaction between hubness and dimensionality reduction as a point of future work.

We examined the following methods: principal component analysis – PCA [99], independent component analysis – ICA [88], stochastic neighbor embedding – SNE [86], isomap [207], and diffusion maps [115, 140]. Figure 5.4 depicts the relationship between the percentage of the original number of features maintained by the DR methods and S_{N_k}, for several high-dimensional real data sets (musk1, mfeat-factors, and spec-

trometer; see Table 5.1) and iid uniform random data (with Euclidean distance, $k = 10$, and the same number of neighbors used for isomap and diffusion maps). For PCA, ICA, SNE, and the real data sets, looking from right to left, S_{N_k} stays relatively constant until a small percentage of features is left, after which it suddenly drops (Figure 5.4(a–c)). It can be said that this is the point where the intrinsic dimensionality of data sets is reached, and further reduction of dimensionality may incur loss of valuable information. Such behavior is in contrast with the case of iid uniform random data (full black line in Figure 5.4(a–c)), where S_{N_k} steadily and steeply reduces with the decreasing number of randomly selected features (dimensionality reduction is not meaningful in this case), because intrinsic and embedded dimensionalities are equal. Since PCA is equivalent to metric multidimensional scaling (MDS) when Euclidean distances are used [207], and SNE is a variant of MDS which favors the preservation of distances between nearby points, we can roughly regard the notion of intrinsic dimensionality used in this dissertation as the minimal number of features needed to account for all *pairwise distances* within a data set. Although ICA does not attempt to explicitly preserve pairwise distances, combinations of independent components produce skewness of N_k which behaves in a way that is similar to the skewness observed with PCA and SNE.

On the other hand, isomap and diffusion maps replace the original distances with distances derived from a neighborhood graph. It can be observed in Figure 5.4(d, e) that such replacement generally reduces S_{N_k}, but in most cases does not alleviate it completely. With the decreasing number of features, however, S_{N_k} of real data sets in Figure 5.4(d, e) still behaves in a manner more similar to S_{N_k} of real data sets for PCA, ICA, and SNE (Figure 5.4(a–c)) than iid random data (dash-dot black line in Figure 5.4(d, e)).

The above observations signify that, if distances are not explicitly altered (as with isomap and diffusion maps DR methods), that is, if one cares about preserving the original distances, dimensionality reduction may not have a significant effect on hubness when the number of features is above the intrinsic dimensionality. This observation is useful in most practical cases because if dimensionality is reduced below intrinsic dimensionality, loss of information can occur. If one still chooses to apply aggressive dimensionality reduction and let the resulting number of features fall below intrinsic dimensionality, it can be expected of pairwise distances and nearest-neighbor relations between points in the data set to be altered, and hubness to be reduced or even lost. Whether these effects should be actively avoided or sought really depends on the application domain and task at hand, that is, whether and to what degree the original pairwise distances represent valuable information, and how useful are the new distances and neighborhoods after dimensionality reduction.

5.6 Impact of Hubness on Machine Learning

The impact of hubness on machine-learning applications has not been thoroughly investigated so far. In this section we examine a wide range of commonly used machine-

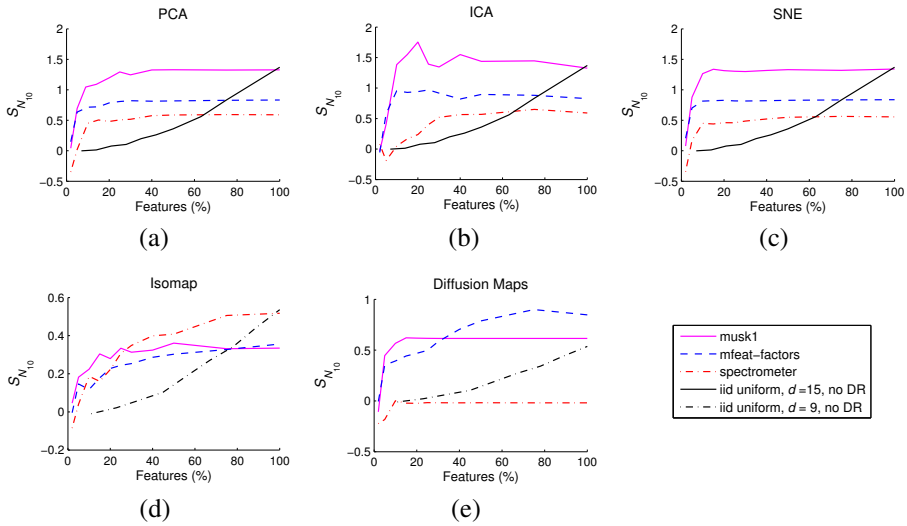


Figure 5.4: Skewness of N_{10} in relation to the percentage of the original number of features maintained by dimensionality reduction, for real and iid uniform random data and (a) principal component analysis – PCA, (b) independent component analysis – ICA, (c) stochastic neighbor embedding – SNE, (d) isomap, and (e) diffusion maps
Slika 5.4: Koeficijent asimetrije N_{10} u odnosu na procenat originalnog broja atributa zadržanog pri redukciji dimenzionalnosti, za prave i *iid* uniformne slučajne skupove tačaka i (a) *principal component analysis* – PCA, (b) *independent component analysis* – ICA, (c) *stochastic neighbor embedding* – SNE, (d) isomap, i (e) *diffusion maps*

learning methods for supervised (Section 5.6.1), semi-supervised (Section 5.6.2), and unsupervised learning (Section 5.6.3), that either directly or indirectly use distances in the process of building a model. Our main objective is to demonstrate that hubs (as well as their opposites, anitihubs) can have a significant effect on these methods. The presented results highlight the need to take hubs into account in a way equivalent to other factors, such as the existence of outliers, the role of which has been well studied.

5.6.1 Supervised Learning

To investigate possible implications of hubness on supervised learning, we first study the interaction of k-occurrences with information provided by labels. We then move on to examine the effects of hubness on several well-known classification algorithms.

"Good" and "Bad" k-Occurrences

When labels are present, k-occurrences can be distinguished based on whether labels of neighbors match. We define the number of *"bad" k-occurrences* of $\mathbf{x}$, $BN_k(\mathbf{x})$, as

the number of points from data set D for which $\mathbf{x}$ is among the first k NNs, and the labels of $\mathbf{x}$ and the points in question *do not match*. Conversely, $GN_k(\mathbf{x})$, the number of *"good" k-occurrences* of $\mathbf{x}$, is the number of such points where labels do match. Naturally, for every $\mathbf{x} \in D$, $N_k(\mathbf{x}) = BN_k(\mathbf{x}) + GN_k(\mathbf{x})$.

To account for labels, Table 5.1 includes $\widetilde{BN}_{10}$ (14th column), the sum of all observed "bad" k-occurrences of a data set normalized by $\sum_{\mathbf{x}} N_{10}(\mathbf{x}) = 10n$. This measure is intended to express the total amount of "bad" k-occurrences within a data set. Also, to express the amount of information that "regular" k-occurrences contain about "bad" k-occurrences in a particular data set, $C_{BN_{10}}^{N_{10}}$ (15th column) denotes the Spearman correlation between BN_{10} and N_{10} vectors. The motivation behind this measure is to express the degree to which BN_k and N_k follow a similar distribution.

"Bad" hubs, that is, points with high BN_k, are of particular interest to supervised learning because they carry more information about the location of the decision boundaries than other points, and affect classification algorithms in different ways. To understand the origins of "bad" hubs in real data, we rely on the notion of the *cluster assumption* from semi-supervised learning [33], which roughly states that most pairs of points in a high density region (cluster) should be of the same class. To measure the degree to which the cluster assumption is violated in a particular data set, we simply define the *cluster assumption violation* (CAV) coefficient as follows. Let a be the number of pairs of points which are in different classes but in the same cluster, and b the number of pairs of points which are in the same class and cluster. Then, we define

$$\mathrm{CAV} = \frac{a}{a+b}\,,$$

which gives a number in the $[0, 1]$ range, higher if there is more violation. To reduce the sensitivity of CAV to the number of clusters (too low and it will be overly pessimistic, too high and it will be overly optimistic), we choose the number of clusters to be 3 times the number of classes of a particular data set. Clustering is performed with K-means.

For all examined real data sets, we computed the Spearman correlation between the total amount of "bad" k-occurrences, $\widetilde{BN}_{10}$, and CAV (16th column of Table 5.1), and found it strong (0.85, see Table 5.2). Another significant correlation (0.39) is observed between $C_{BN_{10}}^{N_{10}}$ and intrinsic dimensionality. In contrast, $\widetilde{BN}_{10}$ and CAV are not correlated with intrinsic dimensionality nor with the skewness of N_{10}. The latter fact indicates that high dimensionality and skewness of N_k are not sufficient to induce "bad" hubs. Instead, based on the former fact, we can argue that there are two, mostly independent, forces at work: violation of the cluster assumption on one hand, and high intrinsic dimensionality on the other. "Bad" hubs originate from putting the two together; that is, the consequences of violating the cluster assumption can be more severe in high dimensions than in low dimensions, not in terms of the total amount of "bad" k-occurrences, but in terms of their distribution, since strong regular hubs are now more prone to "pick up" bad k-occurrences than non-hub points. This is supported by the positive correlation between $C_{BN_{10}}^{N_{10}}$ and d_{mle}, meaning that in high dimensions BN_k tends to follow a more similar distribution to N_k than in low dimensions.

Influence on Classification Algorithms

We now examine how skewness of N_k and the existence of ("bad") hubs affects well-known classification techniques, focusing on the k-nearest neighbor classifier (k-NN), support vector machines (SVM), and AdaBoost. We demonstrate our findings on a selection of data sets from Table 5.1 which have relatively high (intrinsic) dimensionality, and a non-negligible amount of "badness" ($\widetilde{BN}_k$) and cluster assumption violation (CAV). Generally, the examined classification algorithms (including semi-supervised learning from Section 5.6.2) exhibit similar behavior on other data sets from Table 5.1 with the aforementioned properties, and also with various different values of k (in the general range 1–50, as we focused on values of k which are significantly smaller than the number of points in a data set).

k-nearest neighbor classifier. The k-nearest neighbor classifier is negatively affected by the presence of "bad" hubs, because they provide erroneous class information to many other points. To validate this assumption, we devised a simple weighting scheme. For each point $\mathbf{x}$, we compute its standardized "bad" hubness score:

$$h_B(\mathbf{x}, k) = \frac{BN_k(\mathbf{x}) - \mu_{BN_k}}{\sigma_{BN_k}},$$

where μ_{BN_k} and σ_{BN_k} are the mean and standard deviation of BN_k, respectively. During majority voting in the k-NN classification phase, when point $\mathbf{x}$ participates in the k-NN list of the point being classified, the vote of $\mathbf{x}$ is weighted by

$$w_k(\mathbf{x}) = \exp(-h_B(\mathbf{x}, k)),$$

thus lowering the influence of "bad" hubs on the classification decision. Figure 5.5 compares the resulting accuracy of k-NN classifier with and without this weighting scheme for six data sets from Table 5.1. Leave-one-out cross-validation is performed, with Euclidean distance being used for determining nearest neighbors. The k value for N_k is naturally set to the k value used by the k-NN classifier, and $h_B(\mathbf{x}, k)$ is recomputed for the training set of each fold. The reduced accuracy of the unweighted scheme signifies the negative influence of "bad" hubs.

Although "bad" hubs tend to carry more information about the location of class boundaries than other points, the "model" created by the k-NN classifier places the emphasis on describing non-borderline regions of the space occupied by each class. For this reason, it can be said that "bad" hubs are truly bad for k-NN classification, creating the need to penalize their influence on the classification decision. On the other hand, for classifiers that explicitly model the borders between classes, such as support vector machines, "bad" hubs can represent points which contribute information to the model in a positive way, as will be discussed next.

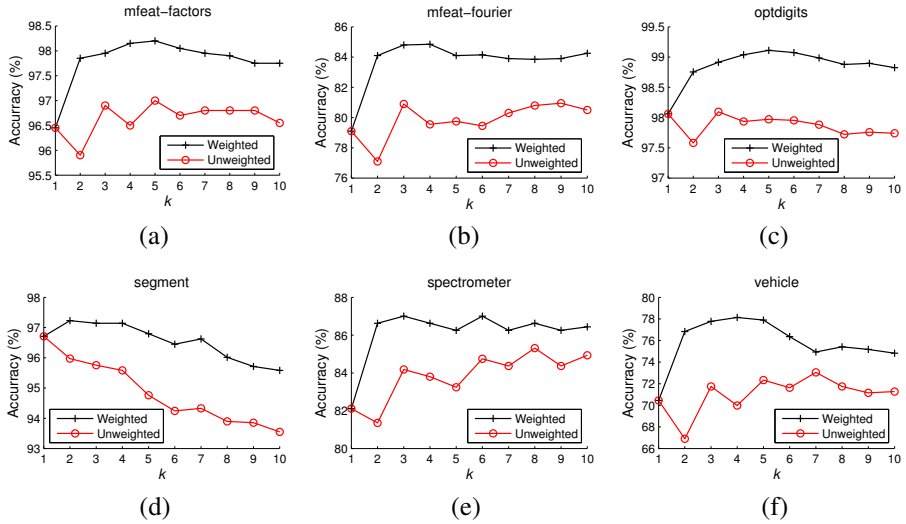


Figure 5.5: Accuracy of k-NN classifier with and without the weighting scheme
Slika 5.5: Tačnost k-NN klasifikatora sa i bez upotrebe težina

Support vector machines. We consider SVMs with the RBF (Gaussian) kernel of the form:

$$K(\mathbf{x}, \mathbf{y}) = \exp\left(-\gamma \|\mathbf{x} - \mathbf{y}\|^2\right),$$

where γ is a data-dependent constant. $K(\mathbf{x}, \mathbf{y})$ is a smooth monotone function of Euclidean distance between points. Therefore, N_k values in the kernel space are exactly the same as in the original space.[8] To examine the influence of "bad" hubs on SVMs, Figure 5.6 illustrates 10-fold cross-validation accuracy results of SVM trained using sequential minimal optimization [157, 101], when points are progressively removed from the training sets: (i) by decreasing BN_k ($k = 5$), and (ii) at random. Accuracy drops with removal by BN_k, indicating that bad hubs are important for SVMs. The difference in SVM accuracy between random removal and removal by BN_k becomes consistently significant at some stage of progressive removal, as denoted by the dashed vertical lines in the plots, according to the paired t-test at significance level 0.05.[9]

The reason behind the above observation is that for high-dimensional data, points with high BN_k can comprise good support vectors. Table 5.3 exemplifies this point by listing the normalized average ranks of support vectors in the 10-fold cross-validation models with regards to decreasing BN_k. The ranks are in the range $[0, 1]$, with the

[8] Centering the kernel matrix changes the N_k of points in the kernel space, but we observed that the overall distribution (that is, its skewness) does not become radically different. Therefore, the following arguments still hold for centered kernels, providing N_k is computed in the kernel space.

[9] Since random removal was performed in 20 runs, fold-wise accuracies for statistical testing were obtained in this case by averaging over the runs.

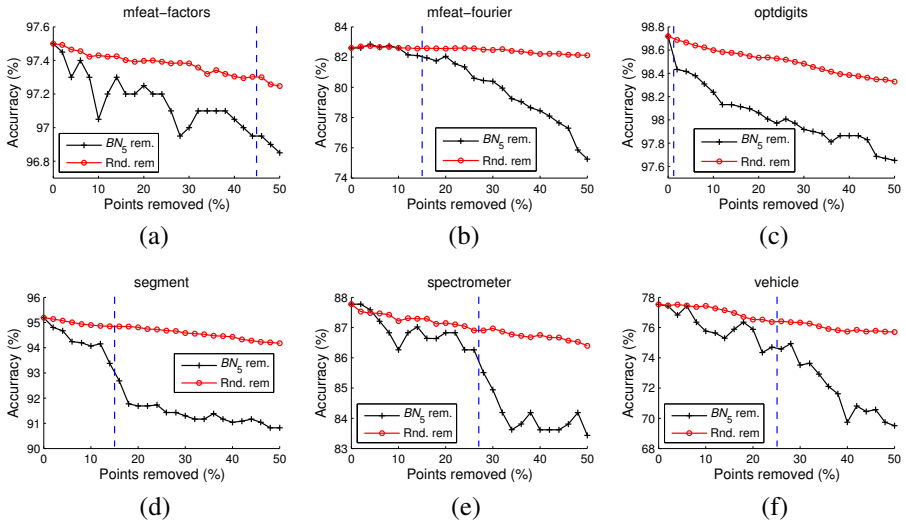


Figure 5.6: Accuracy of SVM with RBF kernel and points being removed from the training sets by decreasing BN_5, and at random (averaged over 20 runs)
Slika 5.6: Tačnost SVM-a sa RBF kernelom i tačkama koje se uklanjaju iz skupova obučavanja po opadajućem BN_5, i u slučajnom redosledu (prosek 20 eksperimenata)

Data set	γ	SV rank	Data set	γ	SV rank
mfeat-factors	0.005	0.218	segment	0.3	0.272
mfeat-fourier	0.02	0.381	spectrometer	0.005	0.383
optdigits	0.02	0.189	vehicle	0.07	0.464

Table 5.3: Normalized average support vector ranks with regards to decreasing BN_5
Tabela 5.3: Normalizovani srednji rangovi *support* vektora u odnosu na opadajući BN_5

value 0.5 expected from a random set of points. Lower values of the ranks indicate that the support vectors, on average, tend to have high BN_k. The table also lists the values of the γ parameter of the RBF kernel, as determined by independent experiments involving 9-fold cross-validation.

AdaBoost. Boosting algorithms take into account the "importance" of points in the training set for classification by weak learners, usually by assigning weights to individual points – the higher the weight, the more attention is to be paid to the point by subsequently trained weak learners. We consider the AdaBoost.MH algorithm [188] in conjunction with CART trees [20] with the maximal depth of three. (More precisely, we use the binary "Real AdaBoost" algorithm and the one-vs-all scheme to handle multi-

class problems, which is equivalent to the original AdaBoost.MH [64].) We define for each point $\mathbf{x}$ its *standardized hubness* score:

$$h(\mathbf{x}, k) = \frac{N_k(\mathbf{x}) - \mu_{N_k}}{\sigma_{N_k}}, \tag{5.1}$$

where μ_{N_k}, σ_{N_k} are the mean and standard deviation of N_k, respectively. We set the initial weight of each point $\mathbf{x}$ in the training set to

$$w_k(\mathbf{x}) = \frac{1}{1 + |h(\mathbf{x}, k)|},$$

normalized by the sum over all points, for an empirically determined value of k. The motivation behind the weighting scheme is to assign less importance to both hubs and outliers than other points (this is why we take the absolute value of $h(\mathbf{x}, k)$).

Figure 5.7 illustrates on six classification problems from Table 5.1 how the weighting scheme helps AdaBoost achieve better generalization in fewer iterations. Data sets were split into training, validation, and test sets with size ratio 2:1:1, parameter k was chosen based on classification accuracy on the validation sets, and accuracies on the test sets are reported. While it is known that AdaBoost is sensitive to outliers [175], improved accuracy suggests that hubs should be regarded in an analogous manner, that is, both hubs and antihubs are intrinsically more difficult to classify correctly, and the attention of the weak learners should initially be focused on "regular" points. The discussion from Section 5.4, about hubs corresponding to probabilistic outliers in high-dimensional data, offers an explanation for the observed good performance of the weighting scheme, as both hubs and antihubs can be regarded as (probabilistic) outliers.

To provide further support, Figure 5.8 depicts binned accuracies of unweighted AdaBoost trained in one fifth of the iterations shown in Figure 5.7, for points sorted by decreasing N_k. It illustrates how in earlier phases of ensemble training the generalization power with hubs and/or antihubs is worse than with regular points. Moreover, for the considered data sets it is actually the hubs that appear to cause more problems for AdaBoost than antihubs (that is, distance-based outliers).

5.6.2 Semi-Supervised Learning

Semi-supervised learning algorithms make use of data distribution information provided by unlabeled examples during the process of building a classifier model. An important family of approaches are graph-based methods, which represent data as nodes of a graph, the edges of which are weighted by pairwise distances of incident nodes [33].

We consider the well-known algorithm by Zhu et al. [238], whose strategy involves computing a real-valued function f on graph nodes, and assign labels to nodes based on its values (see Section 2.4). Taking into account the properties of hubs and antihubs discussed previously, for high-dimensional data sets it can be expected of hubs to be closer to many other points than "regular" points are, and thus carry larger edge

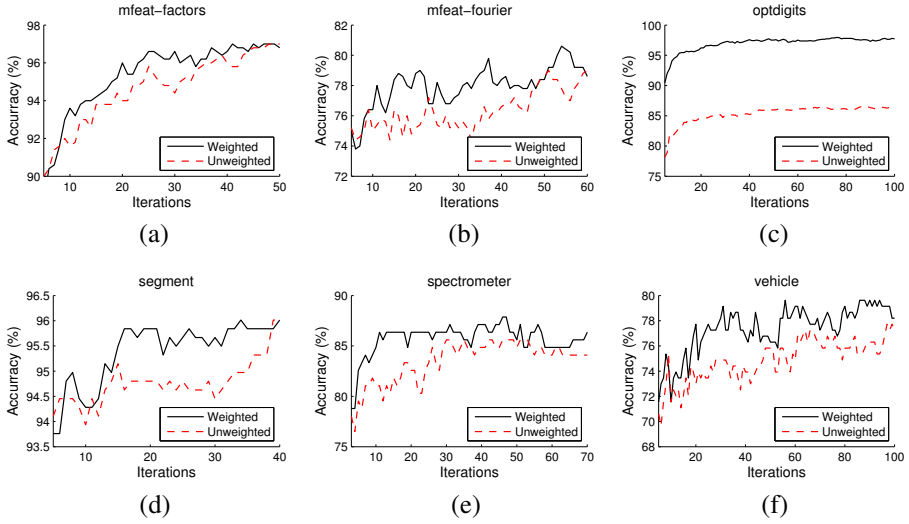


Figure 5.7: Accuracy of AdaBoost with and without the weighting scheme: (a) $k = 20$, (b) $k = 15$, (c) $k = 10$, (d) $k = 20$, (e) $k = 20$, (f) $k = 40$
Slika 5.7: Tačnost AdaBoost-a sa i bez upotrebe težina: (a) $k = 20$, (b) $k = 15$, (c) $k = 10$, (d) $k = 20$, (e) $k = 20$, (f) $k = 40$

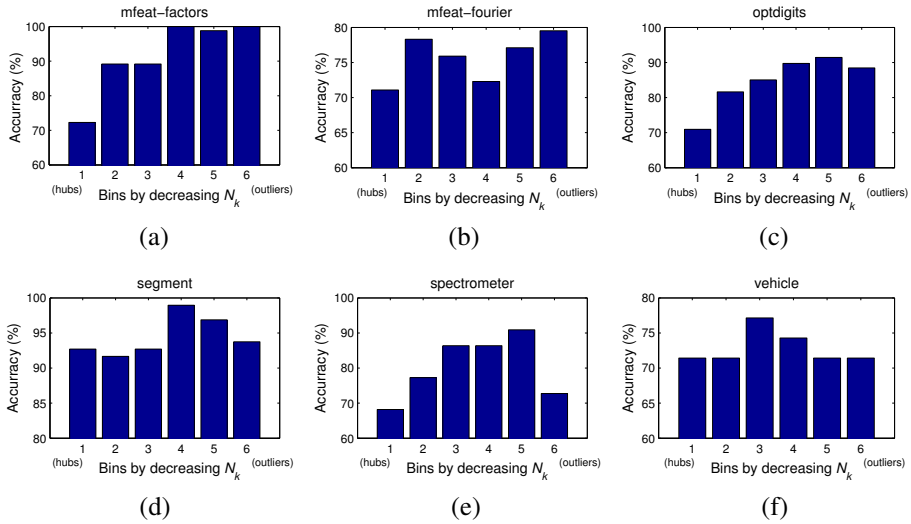


Figure 5.8: Binned accuracy of AdaBoost by decreasing N_k, at one fifth of the iterations shown in Figure 5.7
Slika 5.8: Binovana tačnost AdaBoost-a u odnosu na opadajuće N_k, pri jednoj petini broja iteracija prikazanog u Slici 5.7

weights and be more influential in the process of determining the optimal function f. Conversely, antihubs are positioned farther away from other points, and are expected to bear less influence on the computation of f. Therefore, following an approach that resembles active learning, selecting the initial labeled point set from hubs could be more beneficial in terms of classification accuracy than arbitrarily selecting the initial points to be labeled. On the other extreme, picking the initial labeled point set from the ranks of antihubs could be expected to have a detrimental effect on accuracy.

To validate the above hypothesis, we evaluated the accuracy of the harmonic function algorithm [238] on multiple high-dimensional data sets from Table 5.1, for labeled set sizes ranging from 1% to 10% of the original data set size, with the test set consisting of all remaining unlabeled points. Because the setting is semi-supervised, we compute the N_k scores of points based on complete data sets, instead of only training sets which was the case in Section 5.6.1. Based on the N_k scores ($k = 5$), Figure 5.9 plots classification accuracies for labeled points selected in the order of decreasing N_5 (we choose to take hub labels first), in the order of increasing N_5 (antihub labels are taken first), and in random order (where we report accuracies averaged over 10 runs).[10] It can be seen that when the number of labeled points is low in comparison to the size of the data sets, taking hub labels first generally produces better classification accuracy. On the other hand, when assigning initial labels to antihubs, accuracy becomes significantly worse, with a much larger labeled set size required for the accuracy to reach that of randomly selected labeled points.

5.6.3 Unsupervised Learning

This section will discuss the interaction of the hubness phenomenon with unsupervised learning, specifically the tasks of clustering and outlier detection.

Clustering

The main objectives of (distance-based) clustering algorithms are to minimize intra-cluster distance and maximize inter-cluster distance. The skewness of k-occurrences in high-dimensional data influences both objectives.

Intra-cluster distance may be increased due to points with low k-occurrences. As discussed in Section 5.4, such points are far from all the rest, acting as distance-based outliers. Distance-based outliers and their influence on clustering are well-studied subjects [203]: outliers do not cluster well because they have high intra-cluster distance, thus they are often discovered and eliminated beforehand. The existence of outliers is attributed to various reasons (for example, erroneous measurements). Nevertheless, the skewness of N_k suggests that in high-dimensional data outliers are also expected due to inherent properties of vector space. The next section will provide further discussion on this point.

[10] We determined the σ values of the RBF function in a separate experiment involving 10 runs of random selection of points for the labeled set, the size of which is 10% of the original data set.

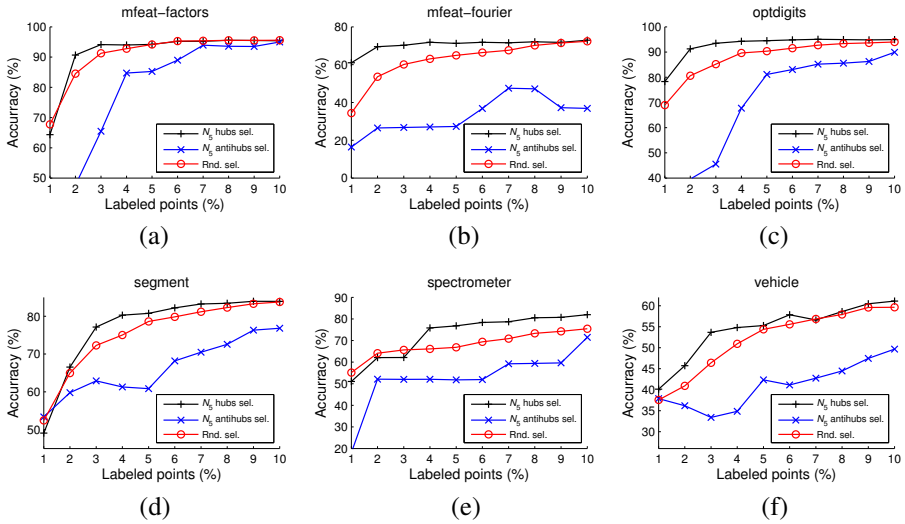


Figure 5.9: Accuracy of the semi-supervised algorithm from [238] with respect to the initial labeled set size as a percentage of the original data set size. Labeled points are selected in the order of decreasing N_5 (hubs first), increasing N_5 (antihubs first), and in random order. Sigma values of the RBF are: (a) $\sigma = 2.9$, (b) $\sigma = 2$, (c) $\sigma = 2.1$, (d) $\sigma = 0.9$, (e) $\sigma = 1.8$, (f) $\sigma = 0.7$

Slika 5.9: Tačnost semi-superviziranog algoritma iz [238] u odnosu na kardinalitet skupa tačaka sa poznatom klasom, izraženim kao procenat kardinaliteta originalnog skupa podataka. Tačke za poznatom klasom su birane po opadajućem redosledu N_5 (prvo habovi), po rastućem N_5 (prvo antihabovi), i po slučajnom redosledu. Vrednosti sigma parametra RBF-a su: (a) $\sigma = 2.9$, (b) $\sigma = 2$, (c) $\sigma = 2.1$, (d) $\sigma = 0.9$, (e) $\sigma = 1.8$, (f) $\sigma = 0.7$

Inter-cluster distance, on the other hand, may be reduced due to points with high k-occurrences, that is, hubs. Like outliers, hubs do not cluster well, but for a different reason: they have low inter-cluster distance, because they are close to many points, thus also to points from other clusters. In contrast to outliers, the influence of hubs on clustering has not attracted significant attention.

To examine the influence of both outliers and hubs, we used the popular silhouette coefficients (SC, see Section 2.5.2). We examined several clustering algorithms, and report results for the spectral algorithm from [135] and Euclidean distance, with similar results obtained for classical K-means, as well as the spectral clustering algorithm from [145] in conjunction with K-means and the algorithm from [135]. For a given data set, we set the number of clusters, K, to the number of classes (specified in Table 5.1). We select as hubs those points $\mathbf{x}$ with $h(\mathbf{x}, k) > 2$, that is, $N_k(\mathbf{x})$ more than two standard deviations higher than the mean (note that $h(\mathbf{x}, k)$, defined by Equation 5.1, ignores labels). Let n_h be the number of hubs selected. Next, we select as

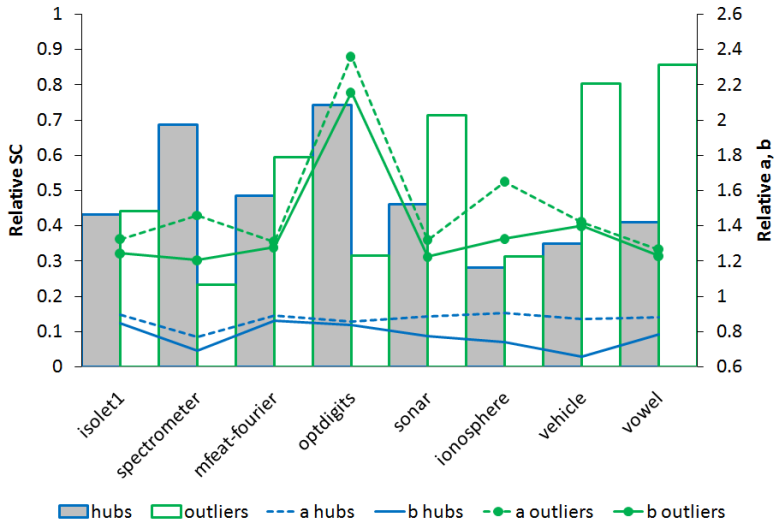


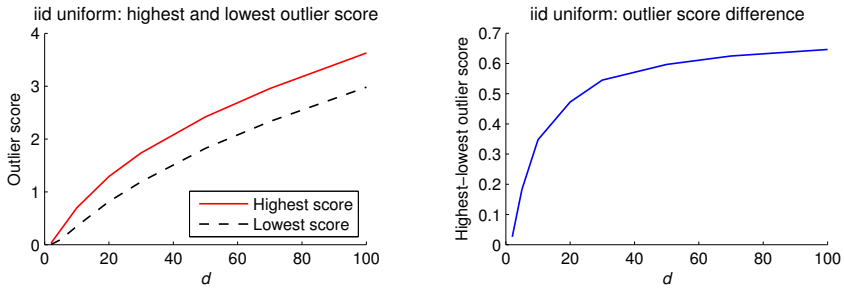
Figure 5.10: Relative silhouette coefficients for hubs (gray filled bars) and outliers (empty bars). Relative values for a and b coefficients are also plotted (referring to the right vertical axes)
Slika 5.10: Relativni koeficijenti siluete za habove (sivi stubovi) i *outlier*-e (beli stubovi). Relativne vrednosti koeficijenata a i b takođe su date (u odnosu na desnu vertikalnu osu)

outliers the n_h points with the lowest k-occurrences. Finally, we randomly select n_h points from the remaining points (we report averages for 100 different selections). To compare hubs and antihubs against random points, we measure the *relative SC* of hubs (antihubs): the mean SC of hubs (antihubs) divided by the mean SC of random points. For several data sets from Table 5.1, Figure 5.10 depicts with bars the relative silhouette coefficients.[11] As expected, outliers have relative SC lower than one, meaning that they cluster worse than random points. Notably, the same holds for hubs, too.[12]

To gain further insight, Figure 5.10 plots with lines (referring to the right vertical axes) the relative mean values of a_i and b_i for hubs and outliers (dividing with those of randomly selected points). Outliers have high relative a_i values, indicating higher intra-cluster distance. Hubs, in contrast, have low relative b_i values, indicating reduced inter-cluster distance. In conclusion, when clustering high-dimensional data, hubs should receive analogous attention as outliers.

[11] Data sets were selected due to their high (intrinsic) dimensionality – similar observations can be made with other high-dimensional data sets from Table 5.1.

[12] The statistical significance of differences between the SC of hubs and randomly selected points has been verified with the paired t-test at significance level 0.05.



(a) (b)

Figure 5.11: (a) Highest and lowest distance to the 5th nearest neighbor in iid uniform random data, with respect to increasing d. (b) The difference between the two distances
Slika 5.11: (a) Najveća i najmanja udaljenost do petog najbližeg suseda u *iid* uniformnom skupu slučajnih tačaka, u odnosu na rastuće d. (b) Razlika između dve udaljenosti

Outlier Detection

This section will discuss possible implications of high dimensionality on distance-based outlier detection, in light of the findings concerning the hubness phenomenon presented in previous sections. Section 5.4 already discussed the correspondence of antihubs with distance-based outliers in high dimensions, and demonstrated on real data the negative correlation between N_k and a commonly used outlier score – distance to the kth nearest neighbor. On the other hand, a prevailing view of the effect of high dimensionality on distance-based outlier detection is that, due to distance concentration, every point seems to be an equally good outlier, thus hindering outlier-detection algorithms [3]. Based on the observations regarding hubness and the behavior of distances discussed earlier, we believe that the true problem actually lies in the opposite extreme: high dimensionality induces antihubs that can represent "artificial" outliers. This is because, from the point of view of common distance-based outlier scoring schemes, antihubs may appear to be stronger outliers in high dimensions than in low dimensions, only due to the effects of increasing dimensionality of data.

To illustrate the above discussion, Figure 5.11(a) plots for iid uniform random data the highest and lowest outlier score (distance to the kth NN, $k = 5$) with respect to increasing d ($n = 10000$ points, averages over 10 runs are reported). In accordance with the asymptotic behavior of all pairwise distances discussed in previous sections, both scores increase with d. However, as Figure 5.11(b) shows, the *difference* between the two scores also increases. This implies that a point could be considered a distance-based outlier only because of high dimensionality, since outliers are not expected for any other reason in iid uniform data (we already demonstrated that such a point would most likely be an antihub). As a consequence, outlier-detection methods based on measuring distances between points may need to be adjusted to account for the (intrinsic) dimensionality of data, in order to prevent dimensionality-induced false positives.

5.7 Summary and Future Work

In this chapter, together with Chapter 4, we explored an aspect of the curse of dimensionality that is manifested through the phenomenon of hubness – the tendency of high-dimensional data sets to contain hubs, in the sense of popular nearest neighbors of other points. To the best of our knowledge, hubs and the effects they have on machine-learning techniques have not been thoroughly studied subjects. Through theoretical and empirical analysis involving synthetic and real data sets we demonstrated the emergence of the phenomenon and explained its origins, showing that it is an inherent property of data distributions in high-dimensional vector space that depends on the intrinsic, rather than embedding dimensionality of data. We also discussed the interaction of hubness with dimensionality reduction. Moreover, we explored the impact of hubness on a wide range of machine-learning tasks that directly or indirectly make use of distances between points, belonging to supervised, semi-supervised, and unsupervised learning families, demonstrating the need to take hubness into account in an equivalent degree to other factors, like the existence of outliers.

Besides application areas that involve audio and image data [9, 49, 12, 84], identifying hubness within data and methods from other fields can be considered an important aspect of future work, as well as designing application-specific methods to mitigate or take advantage of the phenomenon. We already established the existence of hubness in collaborative-filtering data with commonly used variants of cosine distance [141], time-series data sets in the context of k-NN classification involving dynamic time warping (DTW) distance (Chapter 6), text data within several variations of the classical vector space model for information retrieval (Chapter 7), and audio data for music information retrieval using spectral similarity measures [100]. In the immediate future we plan to perform a more detailed investigation of hubness in the field of outlier detection and image mining. Another application area that could directly benefit from an investigation into hubness are reverse k-NN queries, which retrieve data points that have the query point $\mathbf{q}$ as one of their k nearest neighbors [204].

One concern we elected not to include into the scope of this dissertation is the efficiency of computing N_k. It would be interesting to explore the interaction between approximate k-NN graphs [35] and hubness, in both directions: to what degree do approximate k-NN graphs preserve hubness information, and can hubness information be used to enhance the computation of approximate k-NN graphs for high-dimensional data (in terms of both speed and accuracy).

Possible directions for future work within different aspects of machine learning include a more formal and theoretical study of the interplay between hubness and various distance-based machine-learning models, possibly leading to approaches that account for the phenomenon at a deeper level. Supervised learning methods may deserve special attention, as it was also observed in another study [23] that the k-NN classifier and boosted decision trees can experience problems in high dimensions. Further directions of research may involve determining whether the phenomenon is applicable to probabilistic models, (unboosted) decision trees, and other techniques not explic-

itly based on distances between points; and also to algorithms that operate within general metric spaces. Since we determined that for K-means clustering of high-dimensional data hubs tend to be close to cluster centers, it would be interesting to explore whether this can be used to improve iterative clustering algorithms, like K-means or self-organizing maps [108]. Nearest-neighbor clustering [22] of high-dimensional data may also directly benefit from hubness information. Topics that could also be worth further study are the interplay of hubness with learned metrics [222] and dimensionality reduction, including supervised [219, 71], semi-supervised [235], and unsupervised approaches [211, 113]. Finally, as we determined high correlation between intrinsic dimensionality and the skewness of N_k, it would be interesting to see whether some measure of skewness of the distribution of N_k can be used for estimation of the intrinsic dimensionality of a data set.

Chapter 6

Hubness and Time Series

This chapter will study the hubness aspect of the dimensionality curse in the domain of time-series analysis. Although it has been suggested that, due to autocorrelation, time series typically have lower intrinsic dimensionality compared to their length [102], there exist problems where the effects of intrinsic dimensionality may not be negligible, for instance in time-series prediction [215]. In this chapter we study the impact hubness on the problem of time-series classification.

Time-series classification has been studied extensively by machine-learning and data-mining communities, resulting in a plethora of different approaches ranging from neural [156] and Bayesian networks [153] to genetic algorithms and support vector machines [52]. Somewhat surprisingly, the simple approach involving the 1-nearest neighbor (1-NN) classifier and some form of dynamic time warping (DTW) distance was shown to be competitive, if not superior, to many state-of-the art classification methods [47, 103].

In the preceding chapters we analyzed a new aspect of the dimensionality curse for general vector spaces and distance measures (for example, Euclidean and cosine) by observing the phenomenon of *hubness*: intrinsically high-dimensional data sets tend to contain *hubs*, that is, points that appear unexpectedly many times in the k-nearest neighbor lists of all other points. More precisely, let $D \subset \mathbb{R}^d$ be a set of points and $N_k(\mathbf{x})$ the number of *k-occurrences* of each point $\mathbf{x} \in D$, that is, the number of times $\mathbf{x}$ occurs among the k nearest neighbors of all other points in D. With increasing intrinsic dimensionality of D, the distribution of N_k becomes considerably skewed to the right, resulting in the emergence of hubs.

In Chapter 5 we have shown that the hubness phenomenon affects the task of classification of general vector-space data, notably the k-nearest neighbor (k-NN) classifier, support vector machines (SVM), and AdaBoost. Hubness affects the k-NN classifier by making some points (the hubs) substantially more influential on the classification decision than other points, thereby enabling certain points to misclassify others more frequently. This implies that in such situations the classification error may not be dis-

tributed uniformly but in a skewed way, with the responsibility for most of the classification error laying on a small part of the data set.

The phenomenon of hubness is relevant to the problem of time-series classification because, as will be shown, it impacts the performance of nearest-neighbor methods which were proven to be very effective for this task. In this chapter, we provide a detailed examination of how hubness influences classification of time series. We focus on the widely used k-NN classifier coupled with DTW distance, hoping that our findings will motivate a more general investigation of the impact of hubness on other classifiers for time series as a direction of future research. We use a collection of 35 data sets from the UCR repository [104] and from [47], which together comprise a large portion of the labeled time-series data sets publicly available for research purposes today.

To express the degree of hubness within data sets, we use the skewness measure (the standardized 3rd moment) of the distribution of N_k. We establish a link between hubness and classification by measuring the amount of class label variation in local neighborhoods. Based on these measurements we develop a framework to categorize different data sets. The framework allows identifying different degrees of hubness among the time-series data sets, determining for a significant number of them that classification can be improved by taking into account the hubness phenomenon. The latter fact is demonstrated through a simple, yet effective weighting scheme for the k-NN classifier, suggesting that consideration of hubness, in cases where it emerges, can allow the k-NN classifier (in general, with $k > 1$) to attain significantly better accuracy than the currently considered top-performing 1-NN classifier, which is not aware of hubness.

The rest of the chapter is organized as follows. The next section provides an overview of related work. Section 6.2 explores the hubness phenomenon, relating it with the intrinsic dimensionality of time-series data, and showing that there exist data sets with non-negligible amounts of hubness and relatively high intrinsic dimensionality. After illustrating the correspondence of hubness and intrinsic dimensionality in Section 6.2, a more thorough investigation into the causes of hubness in a large collection of time-series data sets is presented in Section 6.3. Then, the interplay between hubness and dimensionality reduction of time-series data is studied in Section 6.4. Section 6.5 discusses the impact of hubness on time-series classification, introducing the framework for relating hubness with k-NN performance. Section 6.6 provides experimental evidence demonstrating that strong hubness can be taken into account to improve the accuracy of k-NN classification, and Section 6.7 summarizes the chapter, providing guidelines for future work.

6.1 Related Work

Time-series classification is a well-studied topic of research, with successful state-of-the-art approaches including neural networks [156], Bayesian networks [153], genetic algorithms [52], and support vector machines [52]. Nevertheless, the simple method

combining the 1-NN classifier and some form of DTW distance was shown to be one of the best-performing time-series classification techniques [47, 103].

DTW is a classical distance measure well suited to the task of comparing time series [14]. It differs from Euclidean distance by allowing the vector components that are compared to "drift" from exactly corresponding positions, in order to minimize the distance and compensate for possible "stretching" and "shrinking" of parts of time series along the temporal axis (see Section 2.2.7).

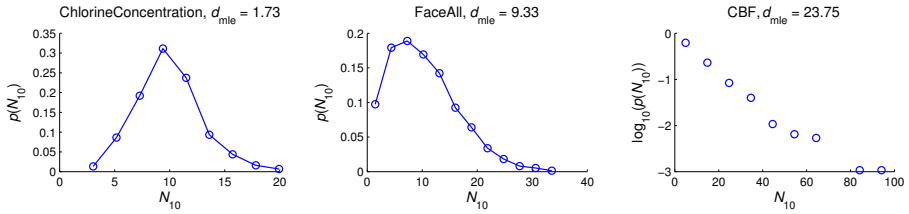
One downside of DTW distance is that finding the center of a group of time series is difficult [146, 147]. Several approaches have been proposed to date [75, 147], all based on averaging two time series along the same path in the matrix used to compute DTW distance by dynamic programming, with the differences in the order in which the time series are considered. The approach of sequential averaging of time series [75] will be used in later sections of this chapter to locate centers of different sets of time series. This is performed by taking the time series in some predetermined sequence, averaging the first two, then averaging the result with the third time series, and so on. After each averaging, uniform scaling [65, 147] will be applied to reduce the length of the average to the length of other time series in the set.

Hubness, as discussed in Chapter 4, was initially observed within different application areas, where it was perceived as a problematic situation, but without connecting it with intrinsic dimensionality of data. This connection was explored in Chapters 4 and 5, where hubness was also related to the phenomenon of distance concentration. Besides k-NN, it was shown that hubness affects other classifiers (SVM, AdaBoost), clustering, and outlier detection. Nevertheless, to our knowledge, no thorough investigation has been conducted so far concerning hubness and its consequences in the field of time-series classification.

An observation that some time series can misclassify others in 1-NN classification more frequently than expected was recently stated in [89], suggesting a heuristical method to consider the second and third neighbor in case the first neighbor misclassifies at least one instance from the training set. However, no study of the general hubness property was made, nor was the relation to intrinsic dimensionality established. In Section 6.5 we discuss correcting erroneous class information in k-NN classification with a more general approach than the heuristic scheme of [89].

6.2 Observing Hubness in Time Series

This section will first establish the relation between hubness and time series, and then explain the origins of hubness in this field in Section 6.3. For clarity of presentation, we initially focus our investigation on the unconstrained DTW distance, due to the simplicity of evaluation resulting from the lack of parameters that need to be tuned. Another reason is that the unconstrained DTW distance is among the best-performing distance measures for 1-NN classification [47]. However, analogous observations can be made for constrained DTW distance (CDTW) with varying tightness of the constraint



(a) (b) (c)

Figure 6.1: Distribution of N_{10} for DTW distance on time-series data sets with increasing estimates of intrinsic dimensionality (d_{mle})
Slika 6.1: Distribucija N_{10} za DTW udaljenosti na skupovima vremenskih serija sa rastućim ocenama latentne dimenzionalnosti (d_{mle})

parameter, and ultimately for Euclidean distance. We defer a more detailed discussion about these distance measures until Section 6.6.3.

The notation introduced in Chapter 4 will also be used in this chapter: let $D \subset \mathbb{R}^d$ be a set of points[1] and $N_k(\mathbf{x})$ the number of *k-occurrences* of each point $\mathbf{x} \in D$, that is, the number of times $\mathbf{x}$ occurs among the k nearest neighbors of all other points in D, with respect to some distance measure. $N_k(\mathbf{x})$ can also be viewed as the in-degree of node $\mathbf{x}$ in the k-nearest neighbor digraph made up of points from D.

We begin with an illustrative example demonstrating how hubness emerges with increasing intrinsic dimensionality of time-series data sets. The intrinsic dimensionality, denoted d_{mle}, has been approximated using the maximum likelihood estimator [120]. Figure 6.1 plots the distribution of N_k for the DTW distance on three time-series data sets (selected from the collection given in Table 6.1) with characteristic d_{mle} values that are low, medium, and high, respectively. In this example, N_k is measured for $k = 10$, but analogous results are obtained with other values of k. It can be seen that the increase of the value of d_{mle} in Figure 6.1(a) to Figure 6.1(c) corresponds to an increase of the right tail of the distribution of k-occurrences, causing some time series from the data set in Figure 6.1(b), and especially Figure 6.1(c), to have a significantly higher value of N_k than the expected value, which is equal to k. Therefore, for the considered three time-series data sets we observe that the increase of hubness closely follows the increase of intrinsic dimensionality.

6.3 Explaining Hubness in Time Series

In this section we will move on from illustrating to more rigorously establishing the positive correlation between hubness and intrinsic dimensionality in time-series data, through empirical measurements over a large collection of time-series data sets, using

[1]For convenience of using vector-space terminology, we shall refer to "time series" and "points" interchangeably.

the methodology established in the previous chapter. Through additional measurements it will be shown that, for intrinsically high-dimensional data sets, hubs tend to be located in the proximity of centers of high-density regions, that is, groups of points which can be determined by clustering. The reasons behind this tendency will be discussed in the exposition that follows.

First, as in Chapter 5, we express the degree of hubness in a data set by a single number – the skewness of the distribution of k-occurrences measured by its standardized 3rd moment: $S_{N_k} = \mathrm{E}(N_k - \mu_{N_k})^3/\sigma^3_{N_k}$, where μ_{N_k}, σ_{N_k} are the mean and standard deviation of N_k, respectively. We examine 35 time-series data sets from the UCR repository [104] and from [47], listed in Table 6.1.[2] First five columns specify the data set name, number of time series (n), time-series length, that is, embedding dimensionality (d), estimated intrinsic dimensionality (d_{mle}), and number of classes. Column 6 gives the skewness, S_{N_k}, in the data sets. We fix $k = 10$ for skewness and subsequent measurements given in Table 6.1. Results analogous to those presented in the following sections are obtained with other values of k.

The $S_{N_{10}}$ column of Table 6.1 shows that the distributions of N_{10} for all examined data sets are skewed to the right, with notable variations in the degrees of skewness.[3] The correspondence between hubness and intrinsic dimensionality is demonstrated by computing Spearman correlation between $S_{N_{10}}$ and d_{mle} over all 35 data sets, revealing it to be strong: 0.68. On the other hand, there is practically no correlation between $S_{N_{10}}$ and d: 0.05. This verifies the previously mentioned point that hubness emerges only with increasing intrinsic dimensionality, and that high dimensionality in itself is not sufficient since the intrinsic dimensionality can be significantly lower.

Although it is now evident that high intrinsic dimensionality creates hubs in time-series data sets, it is relevant to understand how exactly this happens. Explanation is provided by examining the location of hubs inside the data sets. In particular, for data sets with strong hubness it will be shown that hubs tend to be close to the centers of high density regions, that is, groups of similar points. In order to explain the mechanism through which hubness, intrinsic dimensionality, and groups of points interact, we introduce column 8 in Table 6.1.

$C_{\mathrm{cm}}^{N_{10}}$ (column 8) is the correlation between the distance of a point to the center of its own cluster, and N_{10}, over all points in a data set. Clusters were determined using agglomerative hierarchical clustering with complete linkage [203], with the number of clusters, determined by the refined L method [186], given in column 7 of Table 6.1. Since the arithmetic mean is not necessarily the best center of a group of points with respect to dynamic time warping distance, whenever a center was needed we performed 10 runs of sequential DTW averaging [75] (see Section 6.1) with different random permutations of points, considered in addition the arithmetic mean, and adopted as the center the point which was, on average, closest to all points from the group.[4]

[2] We omit three data sets (Beef, Coffee, OliveOil) from consideration due to their small size (60 time series or less), which renders the estimates of S_{N_k} (and other measures introduced later) unstable.

[3] If $S_{N_k} = 0$ there is no skewness, positive (negative) values signify skewness to the right (left).

[4] In the vast majority of cases, DTW averages were adopted as centers.

Name	n	d	d_{mle}	Cls.	$S_{N_{10}}$	Clu.	$C_{\mathrm{cm}}^{N_{10}}$	$\widetilde{BN}_{10}$	CAV
Car	120	577	5.99	4	1.628	9	−0.454	0.454	0.599
MALLAT	2400	1024	9.22	8	1.517	8	−0.280	0.026	0.087
Lighting7	143	319	9.03	7	1.313	9	−0.393	0.403	0.432
SyntheticControl	600	60	20.06	6	1.056	8	−0.412	0.017	0.070
Lighting2	121	637	7.57	2	0.962	10	−0.465	0.244	0.465
SwedishLeaf	1125	128	11.27	15	0.905	10	−0.255	0.271	0.769
Haptics	463	1092	9.77	5	0.901	6	−0.538	0.620	0.728
SonyAIBORobotSurface	621	70	12.57	2	0.899	9	−0.471	0.054	0.166
StarLightCurves	9236	1024	13.43	3	0.879	8	−0.312	0.080	0.223
Symbols	1020	398	6.22	6	0.843	6	−0.372	0.027	0.347
Fish	350	463	6.61	7	0.838	2	−0.426	0.356	0.604
ItalyPowerDemand	1096	24	9.01	2	0.833	10	−0.274	0.061	0.486
SonyAIBORobotSurfaceII	980	65	10.50	2	0.817	9	−0.436	0.069	0.316
FaceAll	2250	131	9.33	14	0.740	9	−0.286	0.073	0.371
50Words	905	270	7.26	50	0.670	6	−0.267	0.437	0.313
WordsSynonyms	905	270	7.26	25	0.670	6	−0.228	0.419	0.422
Yoga	3300	426	5.51	2	0.624	5	−0.091	0.128	0.478
OSULeaf	442	427	8.73	6	0.598	3	−0.368	0.457	0.623
Motes	1272	84	8.81	2	0.523	4	−0.270	0.101	0.373
ChlorineConcentration	4307	166	1.73	3	0.501	6	−0.012	0.316	0.600
Adiac	781	176	6.10	37	0.383	9	−0.307	0.520	0.644
GunPoint	200	150	3.75	2	0.373	5	−0.390	0.162	0.419
FaceFour	112	350	5.66	4	0.367	9	−0.467	0.151	0.154
InlineSkate	650	1882	5.89	7	0.350	8	−0.276	0.609	0.771
MedicalImages	1141	99	6.18	10	0.322	8	−0.272	0.312	0.460
DiatomSizeReduction	322	345	4.82	4	0.299	5	−0.135	0.013	0
Wafer	7164	152	3.54	2	0.269	8	−0.041	0.014	0.182
ECG200	200	96	9.03	2	0.234	5	−0.488	0.236	0.333
CinC	1420	1639	5.43	4	0.145	8	−0.238	0.079	0.677
ECGFiveDays	884	136	7.07	2	−0.027	5	−0.185	0.038	0.370
CBF	930	128	23.75	3	2.393	5	−0.387	0.001	0
TwoPatterns	5000	128	14.52	4	2.104	10	−0.488	0	0.002
Trace	200	275	11.05	4	0.680	6	−0.396	0.009	0.003
TwoLeadECG	1162	82	6.49	2	0.320	3	−0.188	0.002	0.468
Plane	210	144	4.01	7	−0.040	9	−0.593	0.003	0

Table 6.1: Time-series data sets from the UCR repository [104] and from [47]
Tabela 6.1: Skupovi vremenskih serija iz UCR repozitorijuma [104] i iz [47]

In the 8th column of Table 6.1, negative correlation can be observed for every data set. A stronger negative correlation indicates that time series closer to their respective cluster center tend to have higher N_{10}. Overall, the correlations in column 8 indicate that hubs tend to be close to the centers of high-density groups that are reflected by the clusters (the average value of $C_{\mathrm{cm}}^{N_{10}}$ is equal to -0.339). Moreover, the Spearman correlation over all 35 data sets between $S_{N_{10}}$ and $C_{\mathrm{cm}}^{N_{10}}$ is -0.38. This suggests that data sets with stronger hubness tend to have the hubs more localized to the proximity of cluster centers (which we verified by examining the individual scatter plots).

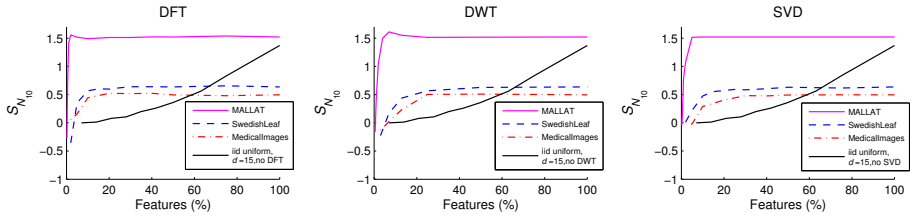
The reason why for intrinsically high-dimensional data sets hubs emerge in the proximity of group centers is related to the phenomenon of distance concentration [61, 85], and explained in detail for Euclidean distance in previous chapters. Since DTW can be viewed as a generalization of Euclidean distance which aims to reduce the distance between two time-series compared to Euclidean distance, the evidence from this chapter suggests that the same mechanisms of hub creation are also relevant to time-series data sets in the context of DTW.

6.4 Hubness and Dimensionality Reduction

In the preceding section it was shown that the skewness of N_k is strongly correlated with intrinsic dimensionality (d_{mle}) of time-series data. We elaborate further on the interplay of hubness and intrinsic dimensionality by considering dimensionality reduction (DR) techniques. The main question is whether DR can alleviate the issue of the skewness of k-occurrences altogether.

We examined three classic dimensionality-reduction techniques widely used on time-series data: discrete Fourier transform (DFT) [56], discrete wavelet transform (DWT) [31], and singular value decomposition (SVD) [56]. Figure 6.2 depicts for several real data sets the relationship between the percentage of features maintained by the DR methods, and S_{N_k}, for $k = 10$ and Euclidean distance. In addition, the plots show the behavior of skewness on synthetic vector-space data, where every vector component was drawn from an iid uniform distribution in the $[0, 1]$ range (2000 vectors were generated, and the average S_{N_k} over 20 runs with different random seeds reported).

For real data, observing the plots right to left (from high dimensionality to low) reveals that S_{N_k} remains relatively constant until a small percentage of features is reached, after which there is a sudden drop. This means that the distribution of k-occurrences remains considerably skewed for a wide range of dimensionalities, for which there exist time series with much higher N_k than the expected value (10). The point of the sudden drop in the value of S_{N_k} is where the intrinsic dimensionality is reached, and further dimensionality reduction may incur loss of information. The observed behavior for real data is in contrast with the case of iid uniform random data, where S_{N_k} steadily reduces with the decreasing number of (randomly) selected features (DR is not meaningful in this case), because intrinsic and embedding dimensionalities are equal. These findings indicate that dimensionality reduction may not have a signif-



(a) (b) (c)

Figure 6.2: Skewness of N_{10} in relation to the percentage of the original number of features maintained by dimensionality reduction
Slika 6.2: Koeficijent asimetrije N_{10} u odnosu na procenat originalnog broja atributa zadržanog pri redukciji dimenzionalnosti

icant effect on the skewness of N_k when the number of features is above the intrinsic dimensionality, a result that is useful in most practical cases since otherwise loss of valuable information may occur.

6.5 Impact of Hubness on Time-Series Classification

In this section we move on to determining how the information provided by labels interacts with hubness and intrinsic dimensionality, with the primary motivation of making the findings useful in the context of nearest-neighbor classification of time series. Section 6.5.1 defines the notions of "good" and "bad" k-occurrences based on whether the labels of neighbors match or not, and explains the mechanisms behind the emergence of "bad" hubs, that is, points with an unexpectedly high number of nearest-neighbor relationships with mismatched labels. Section 6.5.2 describes a framework to categorize time-series data sets based on measures of hubness and the distribution of label mismatches within a data set, allowing one to assess the merits of applying a simple weighting scheme for the k-NN classifier based on hubness, which is introduced in Section 6.5.3.

6.5.1 "Good" and "Bad" k-Occurrences

As in Chapter 5, when labels are present we distinguish k-occurrences based on whether labels of neighbors match. We define the number of *"bad" k-occurrences* of point $\mathbf{x}$, $BN_k(\mathbf{x})$, as the number of points from D for which $\mathbf{x}$ is among the first k nearest neighbors, and the labels of $\mathbf{x}$ and the points in question do not match. Conversely, $GN_k(\mathbf{x})$, the number of *"good" k-occurrences* of $\mathbf{x}$, is the number of such points where labels do match. Naturally, for every $\mathbf{x} \in D$, $N_k(\mathbf{x}) = BN_k(\mathbf{x}) + GN_k(\mathbf{x})$.

To account for labels, we introduce $\widetilde{BN}_k$, the sum of all "bad" k-occurrences of a data set normalized by dividing it with $\sum_{\mathbf{x}} N_k(\mathbf{x}) = kn$. Henceforth, we shall also

refer to this measure as the BN_k ratio. The motivation behind the measure is to express the total amount of "bad" k-occurrences within a data set. Table 6.1 includes $\widetilde{BN}_{10}$ (9th column).

"Bad" hubs, that is, points with high BN_k, are of particular interest to supervised learning since they affect k-NN classification more severely than other points. To understand the origins of "bad" hubs in real data, we rely on the notion of the *cluster assumption* from semi-supervised learning [33], which roughly states that most pairs of points in a high density region (cluster) should be of the same class.

To measure the degree to which the cluster assumption is violated in a particular data set, we refer to the definition of the *cluster assumption violation* (CAV) coefficient from Chapter 5. If a is the number of pairs of points which are in different classes but in the same cluster, and b the number of pairs of points which are in the same class and cluster, then $\mathrm{CAV} = a/(a+b)$, which gives a number in range $[0, 1]$, higher if there is more violation. To reduce the sensitivity of CAV to the number of clusters (too low and it will be overly pessimistic, too high and it will be overly optimistic), we select the number of clusters to be 5 times the number of classes of a particular time-series data set. As in Section 6.3, we use hierarchical agglomerative clustering with complete linkage [203].

For all 35 examined time-series data sets, we computed the Spearman correlation between $\widetilde{BN}_{10}$ and CAV (10th column of Table 6.1), and found it strong (0.74). In contrast, both $\widetilde{BN}_{10}$ and CAV are not correlated with the skewness of N_{10} (measured correlations are 0.01 and -0.07, respectively). The latter fact indicates that high intrinsic dimensionality and hubness are not sufficient to induce "bad" hubs. Instead, we can argue that there are two, mostly independent, factors at work: violation of the cluster assumption on one hand, and hubness induced by high intrinsic dimensionality on the other. "Bad" hubs originate from putting the two together; that is, the consequences of violating the cluster assumption can be more severe in high dimensions than in low dimensions, not in terms of the total amount of "bad" k-occurrences, but in terms of their distribution, since strong hubs are now more prone to "pick up" bad k-occurrences than non-hub points.

6.5.2 A Framework for Categorizing Time-Series Data Sets

Based on the conclusions of the previous subsection, we will now formulate a framework to categorize time-series data sets into 3 different cases. The examination of the 3 cases will divide the considered 35 time-series data sets into three zones, separated by horizontal lines in Table 6.1.[5] The motivation for using this framework is to help assess when hubness can play an important role in time-series classification.

A first observation regarding the collection of the data sets in Table 6.1 is that those contained in Zone 3 (at the bottom of Table 6.1) have extremely low $\widetilde{BN}_k$ (and thus, in

[5] Zone 1 contains data sets from Car to SonyAIBORobotSurfaceII, Zone 2 from FaceAll to ECGFiveDays, and Zone 3 from CBF to Plane.

most cases, the measured CAV) values, which are about one or two orders of magnitude smaller than the $\widetilde{BN}_k$ values of data sets in other zones. For Zone 3 data sets, the cluster assumption is hardly violated, that is, they contain an insignificant number of label mismatches between neighbors.[6] Please note that the data sets in Zone 3 have varying skewness (S_{N_k}) values, some of them being relatively high compared to those in other zones. This is in agreement with the discussion from Section 6.5.1 because when there is no violation of the cluster assumption, skewness cannot create "bad" hubs. Therefore, for data sets in Zone 3, which are practically trivial since the expected error rate will be close to 0, hubness cannot play a significant role.

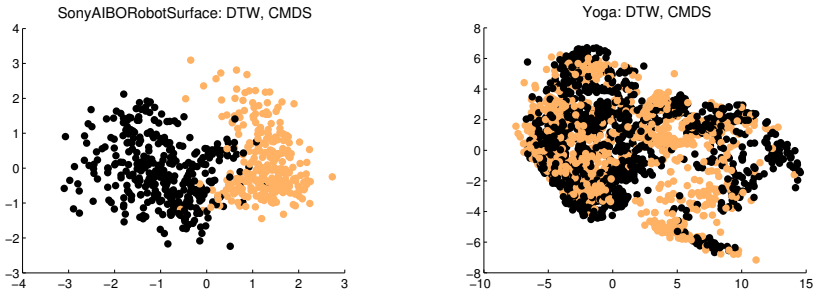
The remaining data sets, that is, those with non-negligible $\widetilde{BN}_k$ (and therefore, in most cases, the measured CAV), can be separated according to their skewness into two zones. In Zone 1 (at the top of Table 6.1) we placed data sets with relatively higher S_{N_k} values than those in Zone 2 (in the middle of Table 6.1). The separation between the two zones was made at approximately the middle value of S_{N_k}, because there exists a noticeable gap between the values of S_{N_k} that in a sense creates two natural clusters that correspond to the two zones.[7] From the discussion in Section 6.5.1 it follows that the data sets in Zone 1 have the potential to contain "bad" hubs, because they combine high skewness with cluster assumption violation, that is, the two factors that lead to the creation of "bad" hubs. Hubness can play a significant role in this case by having the "bad" hubs bear responsibility for most of the error, because the classification error is not distributed uniformly. As will be shown in Section 6.6, for data sets in Zone 1, hubness can be successfully taken into account in order to improve the performance of k-NN classification. This fact will be demonstrated in Section 6.5.3 by applying a simple weighting scheme that attempts to reduce the influence of "bad" hubs on the classification decision.

For the data sets in Zone 2, cluster assumption violation exists. However, the skewness of k-occurrences for these data sets is relatively low. Thus, according to Section 6.5.1, the data sets in Zone 2 are not expected to contain "bad" hubs that are strong enough to be responsible for most of the error that is created by cluster assumption violation. In this case, the error distributes more uniformly than in the case of Zone 1. Consequently, hubness is expected to have a less important role in Zone 2.

It is worth understanding further the case of Zone 2, in order to explain the cause of the non-negligible "badness" (BN_k), since it cannot be attributed to hubs. The class labels of data sets from Zone 2 are distributed in a mixed way that can be visualized as a checkerboard-like pattern. For example, Figure 6.3 plots two time-series data sets reduced down to two dimensions using classical multidimensional scaling (CMDS) [193]:

[6] We found the CAV measure to be somewhat unstable with respect to the choice of clustering algorithm and number of clusters. For this reason, we will rely mostly on the $\widetilde{BN}_k$ measure which is strongly correlated with CAV, but at the same time more stable and more clearly defined.

[7] In particular, within the two zones, the differences between consecutive values of S_{N_k} are mostly in the order of the second decimal digit, whereas this difference between SonyAIBORobotSurfaceII and 50Words is in the order of the first decimal digit. FaceAll is a boundary case that was assigned to the second zone because its S_{N_k} value is closer to that of 50Words than to that of SonyAIBORobotSurfaceII.



(a) (b)

Figure 6.3: Two time-series data sets reduced by classical MDS
Slika 6.3: Dva skupa vremenskih serija redukovana klasičnim MDS-om

SonyAIBORobotSurface (Figure 6.3(a)), which belongs to Zone 1, and Yoga (Figure 6.3(b)), which belongs to Zone 2 (for clarity of presentation, both cases contain 2 class labels). It can be seen that class labels in Figure 6.3(b) are considerably more mixed than in Figure 6.3(a). For a given point in the Yoga data set, this causes the neighbors other than the first to be much more likely to carry different class labels. At the same time, such behavior is not expected from the majority of points in the SonyAIBORobotSurface data set. In the case that the mixture of class labels is intense among neighbors, lower values of k for k-NN classification are expected to perform better, and when the mixture is very high, setting $k = 1$ can be viewed as the best option.[8]

To obtain a more quantitative evaluation supporting the aforementioned discussion, let us define a simple measure based on entropy, as follows. Assuming class labels in a data set take the values $1, 2, \ldots, K$, and $c \in \{1, 2, \ldots, K\}$, let $p_{c,k}(\mathbf{x})$ be the probability of observing class c among the k nearest neighbors of point $\mathbf{x}$, measured from the data set. We define the *k-entropy* of point $\mathbf{x}$ as

$$\mathrm{H}_k(\mathbf{x}) = -\sum_{c=1}^{K} p_{c,k}(\mathbf{x}) \log p_{c,k}(\mathbf{x}) \,,$$

where, by standard definition, we assume $0 \log 0 = 0$. A higher value of k-entropy for point $\mathbf{x}$ indicates a higher degree of mixture of class labels among the k nearest neighbors of $\mathbf{x}$. We define the k-entropy of a data set, $\overline{\mathrm{H}}_k$, as the average k-entropy of all its points. Figure 6.4 plots for increasing k the median values of $\overline{\mathrm{H}}_k$ computed separately for the 3 examined zones. For data sets in Zone 2 it is evident that $\overline{\mathrm{H}}_k$ steadily increases with increasing values of k. Thus, for these data sets the k-NN classifier is expected to deteriorate with increasing k. This suggests that the 1-NN classifier will be

[8] A more comprehensive explanation of this case is illustrated in [82] (p. 468) through an example of an "easy" and a "difficult" problem for the k-NN classifier. The difficult data set was synthetically generated with labels being assigned to points according to regions that form a 3-dimensional checkerboard pattern.

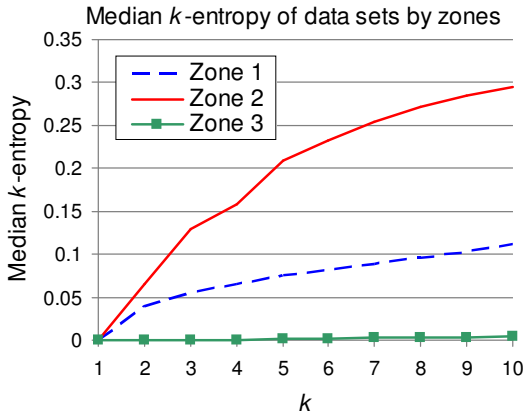


Figure 6.4: Median k-entropy for increasing values of k, computed over data sets in Zone 1, Zone 2, and Zone 3
Slika 6.4: Medijan k-entropije za rastuće vrednosti k, izračunate za skupove podataka u Zoni 1, Zoni 2, i Zoni 3

very competitive in this case and that it will be difficult for a weighting scheme similar to the one presented in the following section (which examines $k > 1$) to attain any improvement.

In summary, this section focused on the categorization of data sets according to the factors described in Section 6.5.1. The resulting framework allows one to identify a number of data sets – those in Zone 1 – for which hubness can play an important role in k-NN classification. This result motivates the development of the weighting scheme that will be presented in the following section and demonstrate the role of hubness.

6.5.3 Weighting Scheme for k-NN Classification

As explained above, for several data sets the k-NN classifier can be negatively affected by the presence of "bad" hubs, because they provide erroneous class information to many other points. To validate this assumption, we will evaluate a simple weighting scheme. For each point $\mathbf{x}$, we calculate its standardized "bad" hubness score:

$$h_B(\mathbf{x}, k) = \frac{BN_k(\mathbf{x}) - \mu_{BN_k}}{\sigma_{BN_k}},$$

where μ_{BN_k} and σ_{BN_k} are the mean and standard deviation of BN_k, respectively. During majority voting, when point $\mathbf{x}$ participates a k-NN list, its vote is weighted by

$$w_k(\mathbf{x}) = \exp(-h_B(\mathbf{x}, k)).$$

The effect of the weighting is that it decreases the influence of "bad" hubs on the classification decision. The k value for $w_k(\mathbf{x})$ is naturally set to the k value used by the k-NN classifier.

Note that the primary motivation for introducing this modification of the k-NN classifier is not to compete with the state-of-the-art time-series classifiers, but rather to illustrate the significance of the hubness phenomenon for time-series classification, and describe the circumstances in which hubness can be made useful. A more detailed examination and comparison of several different weighting schemes is addressed as a direction for future work.

6.6 Experimental Evaluation

The potential usefulness of hubness for k-NN time-series classification will be demonstrated in this section through an experimental comparison of weighted k-NN (k-WNN) described in Section 6.5.3 with regular 1-NN and k-NN classifiers.

6.6.1 The Experimental Setup

All 35 data sets from Table 6.1 are included in the experiments. For data sets of size 200 and larger, we performed 10 runs of 10-fold cross-validation, recording average error rates of 1-NN, k-WNN, and k-NN (for the later two we examined $2 \leq k \leq 10$). On data sets containing less than 200 time series, the classifiers were evaluated through leave-one-out cross-validation, which is a commonly followed practice for small data sets. As in previous sections, we consider the DTW distance.

6.6.2 k-NN Classification Results

The rightmost three columns of Table 6.2 show the average error rates of 1-NN, k-WNN, and k-NN, respectively. Since our primary goal is to provide a simple demonstration of possible merits of using hubness to improve k-NN time-series classification, we report weighted and unweighted k-NN error rates for the value of k for which k-WNN exhibited the smallest error. As in Table 6.1, separation between zones 1–3 described in Section 6.5.2 is signified by horizontal lines.

It can be observed that, as expected from the discussion in Section 6.5.2, in Zone 1 weighted k-NN outperforms 1-NN in the vast majority of cases, while the opposite is true in Zone 2. The smallest error rates of the three classifiers are highlighted in boldface.[9] Over both zones, weighted k-NN predominantly exhibits smaller error rates than the regular k-NN, but only in Zone 1 was the benefit of considering hubness large enough to yield improvement over 1-NN.

[9] In cases where 10×10-fold cross-validation was used, symbols $\bullet$/$\circ$ denote statistically significant improvement/degradation of error for k-WNN and k-NN with respect to 1-NN, according to the corrected resampled t-test [139] at significance level 0.1.

Name	n	$S_{N_{10}}$	$\widetilde{BN}_{10}$	Evaluation	k	1-NN	k-WNN	k-NN
Car	120	1.628	0.454	Leave-1-out	2	0.2333	**0.1750**	0.3083
MALLAT	2400	1.517	0.026	10×10-fold	4	0.0133	**0.0113**	0.0130
Lighting7	143	1.313	0.403	Leave-1-out	2	0.2797	**0.1748**	0.2378
SyntheticControl	600	1.056	0.017	10×10-fold	4	0.0080	**0.0018** •	**0.0018** •
Lighting2	121	0.962	0.244	Leave-1-out	4	0.0992	**0.0826**	0.1818
SwedishLeaf	1125	0.905	0.271	10×10-fold	5	0.1833	**0.1662**	0.1802
Haptics	463	0.901	0.620	10×10-fold	10	0.5783	**0.5107** •	0.5323 •
SonyAIBORobotSurface	621	0.899	0.054	10×10-fold	2	0.0245	**0.0151** •	**0.0151** •
StarLightCurves	9236	0.879	0.080	10×10-fold	9	0.0657	**0.0478** •	0.0646
Symbols	1020	0.843	0.027	10×10-fold	3	0.0183	**0.0177**	0.0187
Fish	350	0.838	0.356	10×10-fold	4	0.2014	0.2094	**0.1926**
ItalyPowerDemand	1096	0.833	0.061	10×10-fold	9	0.0474	**0.0355** •	0.0390
SonyAIBORobotSurfaceII	980	0.817	0.069	10×10-fold	2	0.0270	**0.0168** •	0.0170 •
FaceAll	2250	0.740	0.073	10×10-fold	2	**0.0225**	0.0302 ○	0.0314 ○
50Words	905	0.670	0.437	10×10-fold	5	**0.2793**	0.2959	0.2991 ○
WordsSynonyms	905	0.670	0.419	10×10-fold	3	**0.2656**	0.2838	0.2810
Yoga	3300	0.624	0.128	10×10-fold	2	**0.0592**	0.0651 ○	0.0702 ○
OSULeaf	442	0.598	0.457	10×10-fold	3	**0.2923**	0.3044	0.3178
Motes	1272	0.523	0.101	10×10-fold	2	**0.0448**	0.0588 ○	0.0645 ○
ChlorineConcentration	4307	0.501	0.316	10×10-fold	2	**0.0030**	0.0263 ○	0.0241 ○
Adiac	781	0.383	0.520	10×10-fold	2	**0.3343**	0.3600 ○	0.3599 ○
GunPoint	200	0.373	0.162	10×10-fold	2	**0.0835**	0.0865	0.0925
FaceFour	112	0.367	0.151	Leave-1-out	2	0.0536	**0.0446**	0.0893 ○
InlineSkate	650	0.350	0.609	10×10-fold	2	**0.4508**	0.4836 ○	0.5282 ○
MedicalImages	1141	0.322	0.312	10×10-fold	5	**0.1959**	0.1989	0.2003
DiatomSizeReduction	322	0.299	0.013	10×10-fold	2	0.0037	0.0037	0.0037
Wafer	7164	0.269	0.014	10×10-fold	3	**0.0059**	0.0073 ○	0.0078 ○
ECG200	200	0.234	0.236	10×10-fold	2	0.1646	**0.1622**	0.2064 ○
CinC	1420	0.145	0.079	10×10-fold	3	**0.0147**	0.0218 ○	0.0225 ○
ECGFiveDays	884	−0.027	0.038	10×10-fold	2	0.0090	**0.0077**	0.0082
CBF	930	2.393	0.001	10×10-fold	3	0	0	0
TwoPatterns	5000	2.104	0	10×10-fold	2	0	0	0
Trace	200	0.680	0.009	10×10-fold	2	0	0	0
TwoLeadECG	1162	0.320	0.002	10×10-fold	3	0.0009	0.0010	0.0010
Plane	210	−0.040	0.003	10×10-fold	2	0	0	0

Table 6.2: Error rates of 1-NN, weighted k-NN (k-WNN), and k-NN classifiers. Symbols •/○ denote statistically significant improvement/degradation of error for k-WNN and k-NN with respect to 1-NN

Tabela 6.2: Srednje greške 1-NN, težinskog k-NN (k-WNN), i k-NN klasifikatora. Simboli •/○ označavaju statistički značajno poboljšanje/pogoršanje greške k-WNN i k-NN u poređenju sa 1-NN

The above observations reinforce the categorization of data sets introduced in Section 6.5.2. On one hand, data sets from Zone 2 do not exhibit strong hubness, therefore the weighting scheme is not particularly successful at reducing the influence of "bad" hubs since there exist no significant hubs to begin with. On the other hand, as indicated in Figure 6.4, for data sets in Zone 2 there is a much stronger increase of k-entropy with successive values of k than for data sets in Zone 1, which suggests that considering additional neighbors in Zone 2 data sets does not provide correct and useful label information to the k-NN classifier. For these two reasons, with Zone 2 data sets 1-NN is expected to be the superior classifier in the majority of cases, which is confirmed by the results given in Table 6.2. In Zone 1, however, the combination of strong hubness and mild increase of k-entropy provides the weighted k-NN classifier with enough leverage to outperform 1-NN.

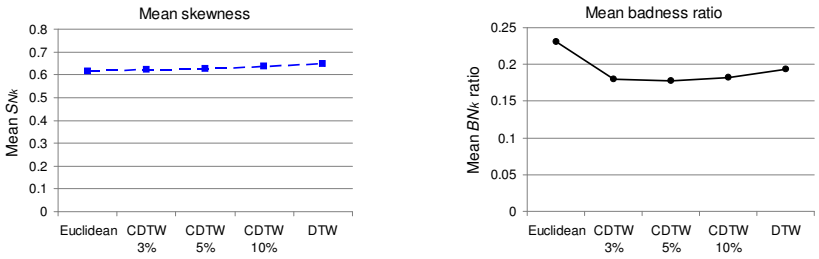
In addition, we believe that the described categorization offers an interesting byproduct. It helps to better understand why the 1-NN classifier has proven to be effective in so many cases in the context of time-series classification. For the data sets in Zone 3, the 1-NN classifier is effective because there is hardly room for improvement by any other classifier on those data sets. For the data sets in Zone 2, the 1-NN classifier is effective since the mixture of class labels in a checkerboard-like fashion deteriorates the performance of classifiers that consider more points beyond the immediate neighbor.

6.6.3 Other Distance Measures and Methods

The findings described in the preceding sections focused primarily on the unconstrained DTW distance. However, the phenomenon of hubness, which we showed to be a potentially significant factor for time-series classification, is also present when constrained versions of DTW distance are used, including Euclidean distance. To verify this, Figure 6.5(a) shows the average value of skewness of k-occurrences (for $k = 10$) over real data sets for constrained DTW distance with varying values of the constraint parameter: $r = 0\%$ (Euclidean distance), $r = 3\%, 5\%, 10\%$, and $r = 100\%$ (unconstrained DTW). It can be seen that the average skewness stays relatively constant.[10] From this, and also from observing the S_{N_k} values for individual data sets, we conclude that no single variant of the (C)DTW distance can be considered, in general terms, particularly prone to hubness.

On the other hand, over the 35 data sets we did detect a significant difference in the total amount of "badness" between different distance measures. Figure 6.5(b) plots the mean BN_k ratio ($\widetilde{BN}_k$) for $k = 10$ and (C)DTW distance with the same range of the constraint parameter r as in Figure 6.5(a). It can be seen that Euclidean distance exhibits considerably higher values of the BN_k ratio than CDTW distances with $r > 0\%$. Regarding the preceding discussions in this chapter, this means that the cluster structure imposed by Euclidean distance tends to correspond with the class labeling more weakly

[10] We removed from this measurement the 3 highest skewness values from data sets in Zone 3, since as previously explained these data sets are practically trivial for classification, and their skewness is not of particular relevance as it does not generate "bad" hubs.



(a) (b)

Figure 6.5: (a) Mean skewness of k-occurrences ($k = 10$), and (b) mean BN_k ratio, that is, $\widetilde{BN}_k$ ($k = 10$) over all considered time-series data sets, for varying values of the CDTW constraint parameter
Slika 6.5: (a) Srednji koeficijent asimetrije k-pojava ($k = 10$), i (b) srednji BN_k odnos, tj. $\widetilde{BN}_k$ ($k = 10$) za sve posmatrane skupove vremenskih serija, za različite vrednosti CDTW parametra ograničenja

than the cluster structures of CDTW with $r > 0\%$. In other words, for time-series data Euclidean distance tends to imply a higher degree of cluster assumption violation in a data set. This observation may represent an additional factor for the explanation of the superiority of DTW over Euclidean distance for time-series classification.

Finally, we implemented the modification of the 1-NN classifier from [89], which heuristically considers the labels of the second and third neighbor in the case that the first neighbor misclassifies at least one point from the training set. For the data sets of interest, that is, in Zone 1, this method tends to improve the performance of 1-NN. This is expected because the method also aims to correct the classification decisions of points which frequently misclassify others, despite the fact that the role of hubness was not recognized in [89]. For these data sets, our method produced significantly smaller error (at significance level 0.1) than the heuristic method from [89].

6.7 Summary and Future Work

Although time-series data sets tend not to have excessively high intrinsic dimensionality, in this chapter we demonstrated that it can be sufficient to induce hubness: a phenomenon where some points in a data set participate in unexpectedly many k-nearest neighbor lists of other points. After explaining the origins of hubness and its interaction with the information provided by labels, we formulated a framework which, based on hubness and the distribution of label mismatches within a data set, categorizes time-series data sets in a way that allows one to assess whether hubness can be used to improve the performance of the k-NN classifier.

In future work we plan to expand the set of considered distance measures beyond dynamic time warping with different values of the constraint parameter, and explore other state-of-the-art distances such as those which exhibited good performance in recent experiments with the 1-NN classifier [47]: longest common subsequence (LCSS) [218], edit distance on real sequence (EDR) [37], and edit distance with real penalty (ERP) [36]. It will be interesting to see whether the hubness phenomenon appears when these string-based measures are used, as opposed to vector-based ones like Euclidean and (C)DTW. An additional direction of future work is an examination of different weighting schemes for k-NN, in order to determine the most suitable scheme for incorporating hubness information into k-NN classification. Furthermore, time-series classification by methods other than k-NN may benefit from an investigation into the influence of hubness.

Another possible direction for future work is a more detailed exploration of hubness in the context of different time-series representation techniques. In this chapter we briefly considered DFT, DWT, and SVD. Besides these, a more detailed study could include, for example, piecewise aggregate approximation (PAA) [102], piecewise constant approximation (APCA) [26], and symbolic aggregate approximation (SAX) [127].

Finally, in the time-series domain hubness may be relevant to tasks other than classification. Interesting avenues for future research include assessing the influence of hubness on time-series clustering, indexing, and prediction.

Chapter 7

Hubness and Information Retrieval

The vector space model (VSM) [185] is a popular and widely applied information-retrieval (IR) model that represents each document as a vector of weighted term counts. A similarity measure is used to retrieve a list of documents relevant to a query document. VSM allows for many variations in the choice of term weights and similarity measure used, with prominent representatives including tfidf weighting and cosine similarity, as well as more recently proposed schemes Okapi BM25 [178] and pivoted cosine [196, 198].

Typically, the number of terms used in VSM is large, producing a high-dimensional vector space (with, for example, tens of thousands of dimensions). This high dimensionality has been identified as the source of several problems, such as susceptibility to noise and difficulty in capturing the underlying semantic structure. Such problems are commonly recognized as different aspects of the "curse of dimensionality" and their amelioration has attracted significant research effort, mainly based on dimensionality-reduction approaches.

In this chapter, we investigate the hubness aspect of the dimensionality curse, in the context of the vector space model in information retrieval. To our knowledge, hubness has not been thoroughly examined in connection to VSM and IR.

Hubness is worth studying in the context of information retrieval because it considerably impacts vector space models by causing hub documents to become obstinate results, that is, documents included in the search results of a large number of queries to which they are possibly irrelevant. This problem affects the performance of an IR system and the experience of its users, who may consistently observe the appearance of the same irrelevant results even for very different queries.

We commence our investigation in Section 7.1 by demonstrating the emergence of hubness in the context of IR. Section 7.2 continues with one of our main contributions, which is the explanation of the origins of the phenomenon, describing that it is mainly

a consequence of high intrinsic dimensionality of vector-space data and not of other factors, such as sparsity and skewness of the distribution of term frequencies (caused, for example, by differences in document lengths [196]). We link hubness with the behavior of similarity/distance measures in high-dimensional vector spaces and their *concentration*, that is, the tendency of all pair-wise similarity/distance values to become almost equal (see Chapter 3). To ease the presentation of hubness, our discussion first considers the classical VSM based on tfidf term weighting and cosine similarity, and then continues by demonstrating its generality on the more advanced variation Okapi BM25 [178], since hubness is an inherent characteristic of high-dimensional vector spaces that form the basis of various IR models. Moreover, Section 7.3 explains why hubness is not easily mitigated by dimensionality-reduction techniques.

Next, in Section 7.4 we proceed to examine how hubness affects IR applications by causing hubs to become frequently occurring but possibly irrelevant results to a large number of queries. For this purpose, we investigate the interaction between hubness and the notion of the cluster hypothesis [212], and propose a similarity adjustment scheme that takes into account the existence of hubs. The experimental evaluation of the proposed scheme on real data, in Section 7.4.2, indicates that significant performance improvements can be obtained through consideration of hubness. Finally, we provide the conclusions and directions for future work in Section 7.5.

7.1 Observing Hubness in Text Data

This section will demonstrate the existence of the hubness phenomenon, initially on dense and sparse synthetic data, and then on real text data, focusing on the classical tfidf weighting scheme and cosine similarity. A more advanced document representation, Okapi BM25, is discussed in Section 7.4.3.

To measure the existence of hubness, as in previous chapters, let $D \subset \mathbb{R}^d$ denote a set of vectors in a multidimensional vector space, and $N_k(\mathbf{x})$ the number of *k-occurrences* of each vector $\mathbf{x} \in D$, that is, the number of times $\mathbf{x}$ occurs among the k nearest neighbors of all other vectors in D, with respect to some similarity measure. $N_k(\mathbf{x})$ can also be viewed as the in-degree of node $\mathbf{x}$ in the k-nearest neighbor directed graph of vectors from D.

In a manner similar to Chapter 4, we begin by considering an illustrative example, the purpose of which is to demonstrate the existence of hubness in vector-space data, both dense and sparse, and its dependence on dimensionality. Let us consider a random data set of 2000 d-dimensional vectors (that is, points), whose components are independently drawn from the uniform distribution in range $[0, 1]$, and cosine similarity between them. Figure 7.1(a–c) shows the observed distribution of N_k ($k = 10$) with increasing dimensionality. For $d = 3$, the distribution of N_k in Figure 7.1(a) is consistent with the binomial distribution. Such behavior of N_k would also be expected if the graph was generated following a directed version of the Erdős-Rényi (ER) random graph model [54], where *neighbors* are randomly chosen instead of coordinates.

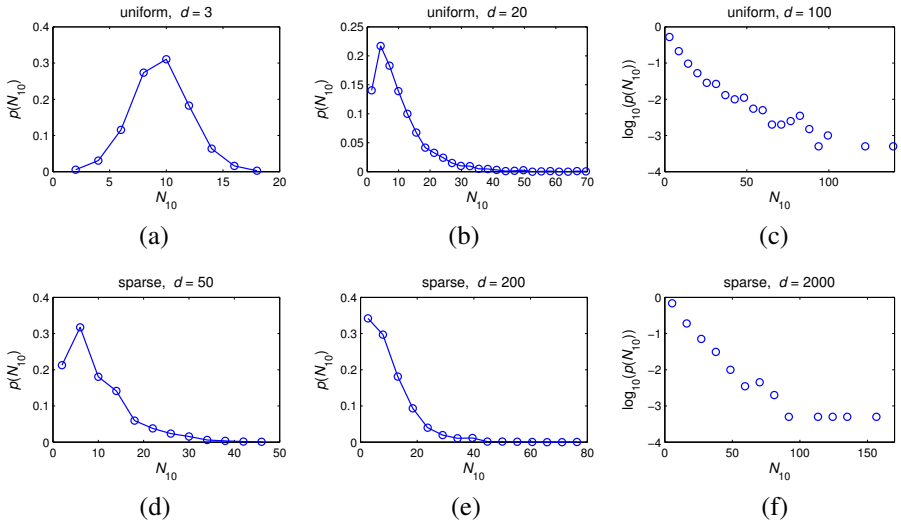
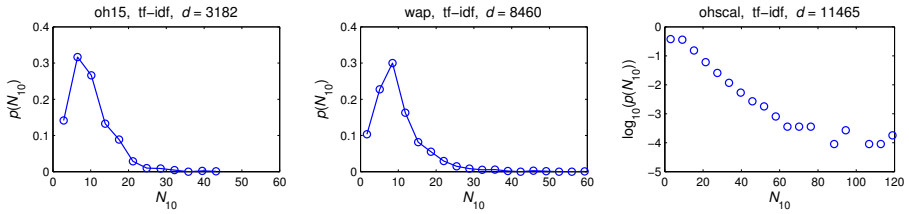


Figure 7.1: Distribution of N_{10} for cosine similarity on (a–c) iid uniform, and (d–f) skewed sparse random data with varying dimensionality (in c and f the vertical axis is in log scale)
Slika 7.1: Distribucija N_{10} za kosinusnu meru sličnosti na (a–c) *iid* uniformnim i (d–f) asimetričnim retkim skupovima slučajnih tačaka različitih dimenzionalnosti (kod c i f vertikalna osa je u logaritamskoj skali)

With increasing dimensionality, however, Figures 7.1(b) and (c) illustrate that the distribution of N_k departs for the random graph model and becomes more skewed to the right, producing vectors (called hubs) with N_k values much higher than the expected value k. The same behavior can be observed with other values of k and data distributions, and was explored in detail in Chapter 4 for Euclidean distance. The simple example with dense and uniformly distributed data is helpful to illustrate the connection between high dimensionality and hubness, since uniformity may not be intuitively expected to generate hubness for reasons other than high dimensionality. To illustrate hubness in a setting more reminiscent of text data that have sparsity and skewed distribution of term frequencies, we randomly generate 2000 vectors with the number of nonzero values for each coordinate ("term") being drawn from the $\mathrm{Lognormal}(5; 1)$ distribution (rounded to the nearest integer), and random numbers (drawn uniformly from $[0, 1]$) spread accordingly throughout the data matrix. Figures 7.1(d–f) demonstrate the increase of hubness with increasing dimensionality in this setting.

A commonly applied practice in IR research is to reduce the influence of long documents (having many nonzero term frequencies and/or high values of term frequencies), by using various normalization schemes [196] to prevent them from being similar to many other documents. However, as observed above and as will be analyzed in Sec-



(a) (b) (c)

Figure 7.2: Distribution of N_{10} for cosine similarity on text data sets with increasing dimensionality (c has log-scale vertical axis)
Slika 7.2: Distribucija N_{10} za kosinusnu sličnost, na tekstualnim skupovima podataka rastuće dimenzionalnosti (c sadrži vertikalnu osu u logaritamskoj skali)

tion 7.2, the high dimensionality that is an inherent characteristic of VSM is the main cause of hubness, as opposed to other data characteristics, since it emerges even when such normalization (cosine) is applied to sparse-skewed data, and also in the case of dense-uniform data where "long documents" are not expected.

Before elaborating on the mechanisms through which hubs form in the context of cosine similarity, we verify the existence of the phenomenon on real text data sets. Figure 7.2 shows the distribution of N_k ($k = 10$) for tfidf term weighting and cosine similarity on three text data sets selected with the criterion of having large difference in their dimensionality. Similarly to the synthetic data sets, it can be seen that hubness tends to become stronger as dimensionality increases, as observed in the longer "tails" of these distributions.

Table 7.1 summarizes the text data sets examined in this study.[1] As in previous chapters, besides basic statistics such as the number of points (n), dimensionality (d), and number of classes, the table also includes the skewness of the distribution of N_{10} ($S_{N_{10}}$). The $S_{N_{10}}$ values in Table 7.1 indicate a high degree of hubness in all data sets.

7.2 Explaining Hubness in Text Data

This section will discuss the reasons behind the emergence of hubness in text data, by first explaining the mechanisms of hub formation on synthetic data, including data reminiscent of real text data distributions (Section 7.2.1), and then exploring hubness in real text data sets (Section 7.2.2).

[1]The top 19 data sets [79], used in form released by Forman [58], include documents from TREC collections, the OHSUMED collection, Reuters and Los Angeles Times news stories, etc. The dmoz data set consists of a selection of short Web-page descriptions from 11 top-level categories from the dmoz Open Directory. The remaining reuters-transcribed and newsgroup data sets are available, for example, from the UCI Machine Learning Repository (for feasibility of analyzing pairwise distances, we split the 20-newsgroups data set into two parts). For all data sets, stop words were removed, and stemming was performed using the Porter stemmer [158]. As will be explained, we focus on data sets containing category labels.

Data set	n	d	Cls.	$S_{N_{10}}$	$S^S_{N_{10}}$	$C^{N_{10}}_{\mathrm{dm}}$	$C^{N_{10}}_{\mathrm{cm}}$	$C^{N_{10}}_{\mathrm{len}1}$	$C^{N_{10}}_{\mathrm{len}w}$	$\widetilde{BN}_{10}$	CAV
fbis	2463	2000	17	1.884	2.391	0.083	0.440	0.188	0.219	0.323	0.400
oh0	1003	3182	10	1.933	2.243	0.468	0.626	0.210	0.212	0.295	0.322
oh10	1050	3238	10	1.485	1.868	0.515	0.650	0.185	0.124	0.415	0.552
oh15	913	3100	10	1.337	2.337	0.477	0.624	0.180	0.146	0.410	0.588
oh5	918	3012	10	1.683	2.458	0.473	0.662	0.154	0.124	0.345	0.587
re0	1504	2886	13	1.421	2.048	0.310	0.493	−0.016	−0.021	0.332	0.512
re1	1657	3758	25	1.334	1.940	0.339	0.587	0.075	0.071	0.305	0.385
tr11	414	6429	9	2.957	0.593	0.348	0.658	0.193	0.157	0.257	0.199
tr12	313	5804	8	2.577	0.841	0.364	0.620	0.199	0.180	0.323	0.326
tr21	336	7902	6	5.016	2.852	0.213	0.572	0.369	0.352	0.172	0.176
tr23	204	5832	6	1.184	0.392	0.052	0.503	−0.057	−0.034	0.239	0.281
tr31	927	10128	7	1.843	2.988	0.218	0.448	0.118	0.109	0.132	0.117
tr41	878	7454	10	1.257	1.413	0.377	0.586	0.110	0.092	0.133	0.288
tr45	690	8261	10	1.490	1.060	0.304	0.638	0.077	0.089	0.175	0.203
wap	1560	8460	20	1.998	1.753	0.479	0.598	0.209	0.203	0.364	0.304
la1s	3204	13195	6	1.837	2.277	0.398	0.498	0.161	0.165	0.296	0.570
la2s	3075	12432	6	1.462	1.876	0.419	0.496	0.203	0.207	0.268	0.531
ohscal	11162	11465	10	3.016	5.150	0.223	0.315	0.052	0.077	0.521	0.793
new3s	9558	26832	44	2.795	2.920	0.146	0.424	0.120	0.129	0.338	0.640
reuters-transcribed	201	3029	11	1.165	1.187	0.671	0.537	0.185	0.140	0.642	0.627
dmoz	3918	10690	11	2.212	2.853	0.443	0.433	−0.100	−0.249	0.613	0.866
mini-newsgroups	1999	7827	20	1.980	1.243	0.388	0.603	0.168	0.152	0.524	0.832
20-newsgroups1	9996	19718	20	2.930	3.571	0.187	0.411	0.125	0.133	0.378	0.850
20-newsgroups2	9995	19644	20	2.716	3.424	0.204	0.405	0.127	0.133	0.375	0.868

Table 7.1: Text data sets
Tabela 7.1: Tekstualni skupovi podataka

7.2.1 The Mechanism of Hub Formation

To describe the mechanisms through which hubness emerges, we begin the discussion by considering again the random data introduced in Section 7.1, that is, the dense data matrix with iid uniform coordinates, and the sparse data set that simulates skewed term frequencies. For the same data sets and dimensionalities, Figure 7.3 shows the scatter plots of N_{10} against the similarity of each vector to the data-set mean, that is, its center. In the chart titles, we also give the corresponding Spearman correlations. It can be seen that, as dimensionality increases, this correlation becomes significantly stronger, to the point of almost perfect correlation of hubness with the proximity to the data center.

The existence of the described correlation provides the main reason for the formation of hubs: owing to the well-known property of vector spaces, vectors closer to the center tend to be closer, on average, to all other vectors. However, this tendency becomes amplified as dimensionality increases, making vectors in the proximity to the

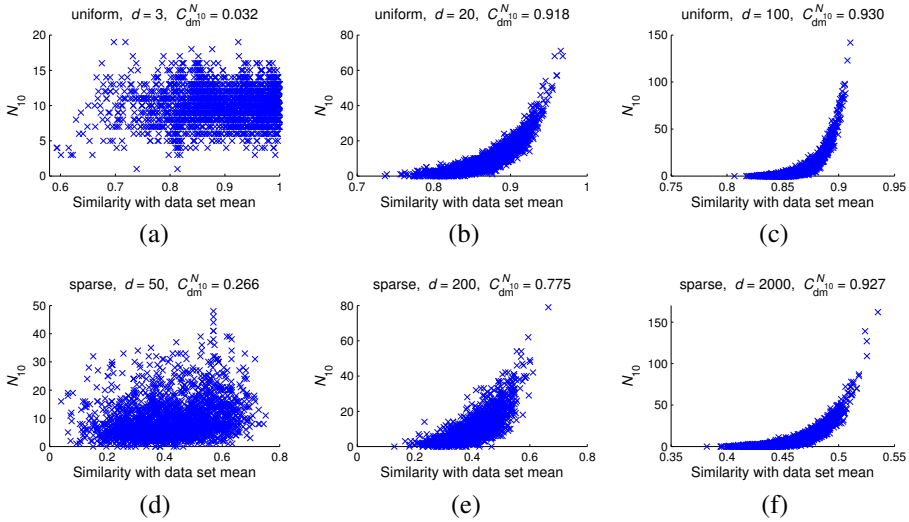
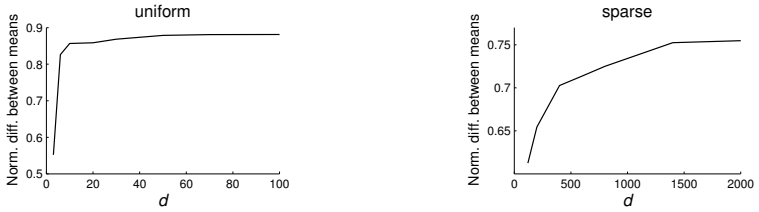


Figure 7.3: Scatter plots of $N_{10}(\mathbf{x})$ (and its Spearman correlation denoted in the chart titles by $C_{\mathrm{dm}}^{N_{10}}$) against the cosine similarity of each vector to the data-set center for (a–c) iid uniform and (d–f) sparse random data, and various dimensionalities (denoted as d in chart titles)

Slika 7.3: Grafikoni $N_{10}(\mathbf{x})$ (i Spermanove korelacije označene u naslovima grafikona sa $C_{\mathrm{dm}}^{N_{10}}$) u odnosu na kosinusnu sličnost svakog vektora sa centrom skupa tačaka za (a–c) *iid* uniformne i (d–f) retke skupove slučajnih tačaka, i različite dimenzionalnosti (označene sa d u naslovima grafikona)

data center become closer, in relative terms, to all other vectors, thus substantially raising their chances of being included in nearest-neighbor lists of other vectors.

To examine further the amplification caused by dimensionality, we compute separately for each of the two examined random data settings (dense-uniform and sparse-random) the distribution, S, of similarities between all vectors in the data set to the center of the data set. From each data set we select two vectors: $\mathbf{x}_0$ is selected to have similarity value to the data-set center exactly equal to the expected value $\mathrm{E}(S)$ of the computed distribution S (that is, at 0 standard deviations from $\mathrm{E}(S)$), whereas $\mathbf{x}_2$ is selected to have higher similarity to the data-set center, being equal to 2 standard deviations added to $\mathrm{E}(S)$ (we were able to select such vectors with negligible error compared to the similarities sought). Next, we compute the distributions of similarities of $\mathbf{x}_0$ and $\mathbf{x}_2$ to all other vectors, and the denote the means of these distributions $\mu_{\mathbf{x}_0}$ and $\mu_{\mathbf{x}_2}$, respectively. Figure 7.4 plots separately for the two examined cases of random data sets, the difference between the two similarity means, normalized (as explained in next paragraph) by dividing with the standard deviation, denoted σ_{all}, of all pairwise similarities, that is: $(\mu_{\mathbf{x}_2} - \mu_{\mathbf{x}_0})/\sigma_{\mathrm{all}}$. These figures show that, with increasing



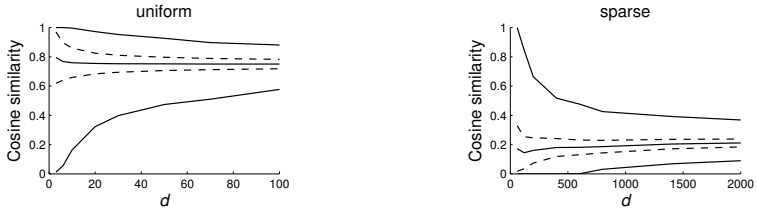
(a) (b)

Figure 7.4: Difference between the normalized means of two distributions of similarity with a point which has: (1) the expected similarity with the data center, and (2) similarity two standard deviations greater, for (a) uniform, and (b) sparse random data (right)
Slika 7.4: Razlika između normalizovanih očekivanih vrednosti dve distribucije sličnosti, sa tačkama koje imaju: (1) očekivanu sličnost sa centrom skupa podataka i (2) sličnost dve standardne devijacije veću, za (a) uniformne, i (b) retke skupove slučajnih tačaka

dimensionality, $\mathbf{x}_2$, which is more similar to the data center than $\mathbf{x}_0$, becomes progressively more similar (in relative terms) to all other vectors, a fact that demonstrates the aforementioned amplification, providing an empirical analogue to Theorem 9 and the discussion from Chapter 4, in the context of cosine similarity.

One question that remains is: in high-dimensional spaces, why is it expected to have some vectors closer to the center and thus become hubs? In Section 3.2 we analyzed the property of the cosine similarity measure, referred to as concentration, which in this case states that, as dimensionality tends to infinity, the expectation of pairwise similarities between all vectors tends to become constant, whereas their standard deviation (denoted above as σ_{all}) shrinks to zero. This means that the majority of vectors become about equally similar to each other, thus to the data center as well. However, high but finite dimensionalities, typical in IR, will result in a small but non-negligible standard deviation, which causes the existence of some vectors, that is, the hubs, that are closer to the center than other vectors. These facts also clarify the aforementioned normalization by σ_{all}, which comprises a way to account for concentration (shrinkage of σ_{all}) and meaningfully compare $\mu_{\mathbf{x}_0}$ and $\mu_{\mathbf{x}_2}$ across dimensionalities. In the case of cosine similarity, this normalization was necessary, as opposed to Euclidean distance discussed in Chapter 4, where the standard deviation of the distribution of all pairwise distances (σ_{all}) remains asymptotically constant across dimensionalities, making the differences $\mu_{\mathbf{x}_2} - \mu_{\mathbf{x}_0}$ directly comparable.

Figure 7.5 illustrates the concentration phenomenon on the uniform and sparse random data used in this chapter. With respect to the distribution of all pairwise similarities, the plots include, from top to bottom: maximal observed value, mean value plus one standard deviation, the mean value, mean value minus one standard deviation, and minimal observed value. The figures illustrate that, with increasing dimensionality, expectation becomes constant and variance shrinks.



(a) (b)

Figure 7.5: Concentration of cosine for (a) uniform, and (b) sparse random data
Slika 7.5: Koncentracija kosinusa za (a) uniformne, i (b) retke skupove slučajnih tačaka

Finally, we need to examine the relation between hubness and additional characteristics of text data sets, such sparsity and the skewed distribution of term frequencies in "long" documents (see Section 7.1). Since Figures 7.1 and 7.3 demonstrate hubness for both dense and sparse random data sets, sparsity on its own should not be considered as a key factor. Regarding the skewness in the distribution of term frequencies, we can consider two cases [196]: (a) more (in number) distinct terms, and (b) higher (in value) term frequencies. For the sparse data set with $d = 2000$ dimensions (Figure 7.3(f)) we measured the correlations of N_{10} with the number of nonzero simulated "terms" of a vector and with the total sum of term weights of a vector, and found both to be weak, 0.142 for case (a) and 0.19 for case (b), in comparison with correlation 0.927 (see title of Figure 7.3(f)) between N_{10} and the similarity with the data-set mean, which has been described as the main factor behind hubness. The weak correlations in cases (a) and (b), which will also be verified with real data (Section 7.2.2), are expected because normalization schemes (cosine in this example) are able to reduce the impact of long documents. What is, thus, important to note is that, even if the correlations of cases (a) and (b) are completely eliminated with another normalization scheme, the hubness phenomenon will still be present, since it is primarily caused by the inherent properties of high-dimensional vector space.

7.2.2 Hub Formation in Real Data

In the previous discussion we have used synthetic data that allow the control of important parameters. To verify the findings with real text data, analogously to Chapters 5 and 6, we need to take into account two additional factors: (1) real data sets usually contain dependent attributes, and (2) real data sets are usually clustered, that is, documents are organized into groups produced by a mixture of distributions instead of originating from one single distribution.

To examine the first factor (dependent attributes), as in Chapter 5 we adopt the approach of François et al. [61], used in the context of l_p-norm concentration. For each data set we randomly permute the elements within every attribute. This way, attributes preserve their individual distributions, but the dependencies between them

are lost and the *intrinsic dimensionality* of data sets increases [61]. In Table 7.1 we give the skewness, denoted $S^S_{N_{10}}$, of the modified data. In most cases $S^S_{N_{10}}$ is considerably higher than $S_{N_{10}}$, implying that hubness depends on the intrinsic rather than embedding (full) dimensionality of text data.

To examine the second factor (many groups), for every data set we measured: (i) Spearman correlation, denoted by $C^{N_{10}}_{\mathrm{dm}}$, of N_k and the similarity with the data-set center, and (ii) correlation, denoted by $C^{N_{10}}_{\mathrm{cm}}$, of N_k and the similarity with the closest group center. Groups are determined using K-means clustering, where the number of clusters was set to the number of document categories of the data set.[2] In most cases, $C^{N_{10}}_{\mathrm{cm}}$ is much stronger than $C^{N_{10}}_{\mathrm{dm}}$. Thus, generalizing the conclusion of Section 7.2.1 to the case of real data, hubs are more similar, compared with other vectors, to their respective cluster centers.

Regarding long documents (see Section 7.2.1), for each data set we computed the correlation between N_k and the number of nonzero term weights for a document, denoted by $C^{N_{10}}_{\mathrm{len1}}$, and also the correlation of N_k with the sum of term weights of a document, denoted by $C^{N_{10}}_{\mathrm{len}w}$. The corresponding columns of Table 7.1 signify that these correlations are weaker or nonexistent (on occasion even negative) compared to the correlation with the proximity to the closest cluster mean ($C^{N_{10}}_{\mathrm{cm}}$). The above observations are in accordance with the conclusions from the end of Section 7.2.1.

7.3 Hubness and Dimensionality Reduction

The attribute-shuffling experiment in Section 7.2.2 suggested that hubness is actually related more to the intrinsic dimensionality of data. We elaborate further on the interplay of hubness and intrinsic dimensionality by considering dimensionality reduction (DR) techniques. The main question is whether DR can alleviate the issue of hubness altogether in the text domain.

We examined the singular value decomposition (SVD) dimensionality-reduction method, which is widely used in IR through latent semantic indexing. Figure 7.6 depicts for several real data sets from Table 7.1 the relationship between the percentage of features (dimensions) maintained by SVD, and the skewness S_{N_k} ($k = 10$). All cases exhibit the same behavior: S_{N_k} stays relatively constant until a small percentage of features is left, after which it suddenly drops. This is the point where the intrinsic dimensionality is reached, and further reduction may incur loss of information. This observation indicates that, when the number of maintained features is above the intrinsic dimensionality, dimensionality reduction cannot significantly alleviate the skewness of k-occurrences, and thus hubness. This result is useful in most practical cases, because moving below the intrinsic dimensionality may cause loss of valuable information from the data.

[2] We report averages of $C^{N_{10}}_{\mathrm{cm}}$ over 10 runs of K-means clustering with different random seeding, in order to reduce the effects of chance.

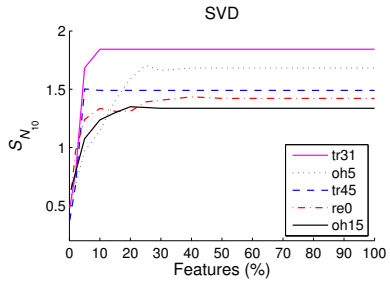


Figure 7.6: Skewness of N_{10} against the percentage of features kept by SVD
Slika 7.6: Koeficijent asimetrije N_{10} u odnosu na procenat broja atributa zadržanih od strane SVD-a

7.4 Impact of Hubness on Information Retrieval

This section examines the ways that hubness affects VSM with respect to the main objective of IR, which is to return relevant results for a query document. Section 7.4.1 explores the interaction of hubness with the notion of the cluster hypothesis. Based on this interaction, Section 7.4.2 proposes a similarity adjustment scheme whose aim is to show how consideration of hubness can be used to improve the precision of a VSM-based IR system, considering the classical tfidf weighting scheme and cosine similarity. Section 7.4.3 discusses how the findings can be generalized to more advanced VSM weighting schemes.

7.4.1 Hubness and the Cluster Hypothesis

In order to explore the interaction between hubness and the cluster hypothesis, we consider the commonly examined case of documents that belong to categories (for example, news categories, like sport or finance). However, a similar approach can be followed for other sources of information about documents, such as indication of their relevance to a set of predefined queries. In the presence of information about documents in the form of categories, k-occurrences can be distinguished based on whether category labels of neighbors match. As in previous chapters, we define the number of *"bad" k-occurrences* of document vector $\mathbf{x} \in D$, denoted $BN_k(\mathbf{x})$, as the number of vectors from D for which $\mathbf{x}$ is among the first k nearest neighbors and the labels of $\mathbf{x}$ and the vectors in question do not match. Conversely, $GN_k(\mathbf{x})$, the number of *"good" k-occurrences* of $\mathbf{x}$, is the number of such vectors where labels do match. For every $\mathbf{x} \in D$, $N_k(\mathbf{x}) = BN_k(\mathbf{x}) + GN_k(\mathbf{x})$.

We define $\widetilde{BN}_k$ as the sum of all "bad" k-occurrences of a data set normalized by dividing it with $\sum_{\mathbf{x}} N_k(\mathbf{x}) = kn$. The motivation behind the measure is to express the total amount of "bad" k-occurrences within a data set. Table 7.1 includes $\widetilde{BN}_{10}$. "Bad"

hubs, that is, documents with high BN_k, are of particular interest to IR, since they affect the precision of retrieval more severely than other documents by being among the k nearest neighbors (that is, in the result list) of many other documents with mismatching categories. To understand the origins of "bad" hubs in real data, we rely on the notion of the *cluster hypothesis* [212]. This hypothesis will be approximated by the *cluster assumption* from semi-supervised learning [33], which roughly states that most pairs of vectors in a high density region (cluster) should belong to the same category.

To measure the degree to which the cluster assumption is violated in a particular data set, we use the definition of the *cluster assumption violation* (CAV) coefficient from previous chapters. Let a be the number of pairs of documents which are in different category but in the same cluster, and b the number of pairs of documents which are in the same category and cluster. Then, we define $\mathrm{CAV} = a/(a+b)$, which gives a number in range $[0, 1]$, higher if there is more violation. To reduce the sensitivity of CAV to the number of clusters, we select the number of clusters to be 3 times the number of categories of a particular data set. As in Section 7.2.2, we use the K-means clustering algorithm.

For all examined text data sets, we computed the Spearman correlation between $\widetilde{BN}_{10}$ and CAV, and found it strong (0.844). In contrast, $\widetilde{BN}_{10}$ is not correlated with d nor with the skewness of N_{10} (measured correlations are -0.03 and 0.109, respectively). The latter indicates that high intrinsic dimensionality and hubness are not sufficient to induce "bad" hubs. Instead, we can argue that there are two, mostly independent, factors at work: violation of the cluster assumption on one hand, and hubness induced by high intrinsic dimensionality on the other. "Bad" hubs originate from putting the two together; that is, the consequences of violating the cluster assumption can be more severe in high dimensions than in low dimensions, not in terms of the total amount of "bad" k-occurrences, but in terms of their distribution, since strong hubs are now more prone to "pick up" bad k-occurrences than non-hubs.

7.4.2 A Similarity Adjustment Scheme

Based on the aforementioned conclusions about "bad" hubness, in this section we propose and evaluate a similarity adjustment scheme with the objective to show how its consideration can be used successfully for improving the precision of a VSM-based IR system. Our main goal is not to compete with the state-of-the-art methods for improving the precision and relevance of results obtained using baseline methods, but rather to demonstrate the practical significance of our findings in IR applications, and the need to account for hubness. Thus, the elaborate examination of more sophisticated methods is addressed as a point of future work.

Let D denote a set of documents, and Q a set of queries independent of D. We will also refer to D as the "training" set, and to Q as the "test" set, and by default compute N_k, BN_k and GN_k on D. We adjust the similarity measure used to compare document vector $\mathbf{x} \in D$ with query vector $\mathbf{q} \in Q$ by increasing the similarity in proportion with

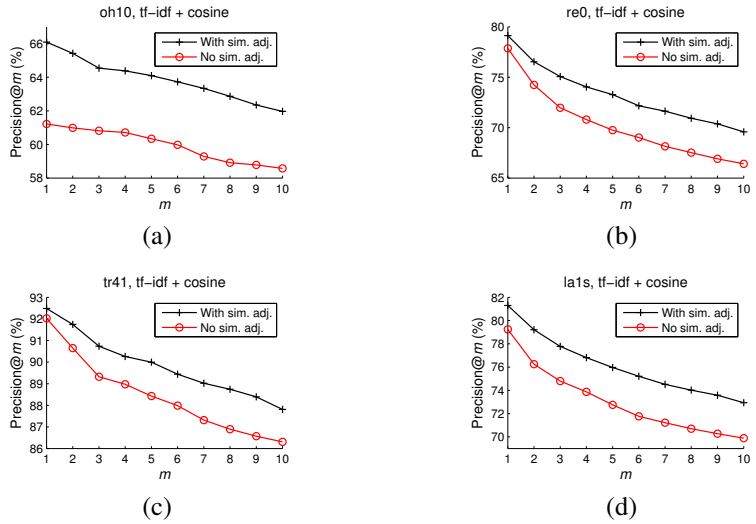


Figure 7.7: Precision at the number of retrieved results m, measured by 10-fold cross-validation
Slika 7.7: Preciznost po broju vraćenih rezultata m, izmerena 10-*fold* kros-validacijom

the "goodness" of $\mathbf{x}$ ($GN_k(\mathbf{x})$), and reducing it in proportion with the "badness" of $\mathbf{x}$ ($BN_k(\mathbf{x})$), both relative to the total hubness of $\mathbf{x}$ ($N_k(\mathbf{x})$), for a given k:

$$sim_a(\mathbf{x}, \mathbf{q}) = sim(\mathbf{x}, \mathbf{q}) + sim(\mathbf{x}, \mathbf{q})(GN_k(\mathbf{x}) - BN_k(\mathbf{x}))/N_k(\mathbf{x})\,.$$

The net effect of the adjustment is that strong "bad" hub documents become less similar to queries, reducing the chances of the document to be included in a list of retrieved results. To prevent documents from being excluded from retrieval too rigorously, the adjustment scheme also considers their "good" side and awards the presence of "good" k-occurrences in an analogous manner.

We experimentally evaluated the improvement gained by the proposed scheme compared to the standard tfidf representation and cosine similarity (all computations involving hubness use $k = 10$), through 10-fold cross-validation on data sets from Table 7.1. First, we focus on the impact of the adjustment scheme on the error introduced to the retrieval system by the strongest "bad" hubs. Let $W^{p\%}$ be the set of the top $p\%$ of documents with highest BN_k, as determined from the training set, and let $BN_k^{test}(\mathbf{x})$ and $N_k^{test}(\mathbf{x})$ be the ("bad") k-occurrences of document $\mathbf{x}$ from the training set, as determined from similarities with documents from the test set. We define the total "badness" of the strongest $p\%$ of "bad" hubs as

$$\mathrm{B}^{p\%} = \frac{\sum_{\mathbf{x} \in W^{p\%}} BN_k^{test}(\mathbf{x})}{\sum_{\mathbf{x} \in W^{p\%}} N_k^{test}(\mathbf{x})}\,,$$

Data set	$B_a^{5\%}$	$B^{5\%}$	$P@10_a$	P@10
fbis	47.73	68.58	72.10	67.59
oh0	49.47	55.93	71.85	70.03
oh10	64.17	70.58	61.97	58.58
oh15	56.21	68.71	62.96	59.19
oh5	51.63	56.68	67.85	64.84
re0	54.77	67.78	69.58	66.41
re1	59.47	69.37	72.04	68.98
tr11	44.86	43.38	74.70	74.06
tr12	65.60	64.15	69.23	67.11
tr21	23.89	27.65	83.70	82.90
tr23	45.03	52.50	75.94	75.60
tr31	44.36	55.50	88.43	86.33
tr41	35.34	49.04	87.81	86.31
tr45	37.40	52.05	84.08	81.88
wap	54.57	60.44	65.18	63.42
la1s	49.01	57.86	72.94	69.89
la2s	52.17	61.67	75.54	72.69
ohscal	66.17	72.38	51.01	47.80
new3s	50.54	65.77	69.00	65.66
reuters-transcribed	68.10	72.34	38.55	36.81
dmoz	69.92	75.37	40.68	38.24
mini-newsgroups	66.22	70.86	49.39	47.16
20-newsgroups1	55.18	63.59	63.89	61.27
20-newsgroups2	57.48	65.09	64.22	61.50

Table 7.2: Retrieval "badness" of 5% of the strongest "bad" hubs ($B^{5\%}$) and precision at 10 ($P@10$), with (columns labeled by $_a$) and without similarity adjustment (in %)
Tabela 7.2: "*Badness*" rezultata ograničen na 5% najjačih "loših" habova ($B^{5\%}$) i *preciznost za 10* ($P@10$), sa (kolone obeležene znakom $_a$) i bez podešavanja sličnosti (u %)

where normalization with N_k^{test} is done to keep the measure in the $[0, 1]$ range. The $B^{p\%}$ measure focuses on the contribution of "bad" hubs to erroneous retrieval of documents which represent false positives.

Table 7.2 shows $B^{p\%}$ on the same $p = 5\%$ of "bad" hubs before and after applying similarity adjustment. It can be seen that for the majority of data sets, the adjustment scheme greatly reduces the amount of erroneous retrieval caused by "bad" hubs.

To illustrate the improving effect of the adjustment scheme on the precision of retrieval, Figure 7.7 plots, for several data sets from Table 7.1, the precision of 10-fold cross-validation against the varying number (m) of documents retrieved as results.

Moreover, Table 7.2 also shows 10-fold cross-validation precision at 10 retrieved results, demonstrating the improvement of precision introduced by similarity adjustment

on all data sets. We verified the statistical significance of improvement of precision using the t-test at significance level 0.05, on all data sets (except tr11 and tr23). The motivation for selecting $m = 10$ results for reporting precision is the common use of this number by information-retrieval systems. We obtained analogous results with various other values of m.

7.4.3 Advanced Representations

The issues examined in previous sections relate to characteristics of VSM that are existing in most of its variations, particularly the high dimensionality. To examine the generality of our findings, we consider the Okapi BM25 weighting scheme [178], which consists of separate weightings for terms in documents and terms in queries. The comparison between document and query can then be viewed as taking the (unnormalized) dot-product of the two vectors. We examine the following basic variant of the BM25 weighting. Providing that n is the total number of documents in the collection, df the term's document frequency, tf the term frequency, dl the document length (the total number of terms), and $avdl$ the average document length, term weights of documents are given by

$$\log \frac{n - df + 0.5}{df + 0.5} \cdot \frac{(k_1 + 1)tf}{k_1((1 - b) + b\frac{dl}{avdl}) + tf},$$

while the term weights of queries are

$$(k_3 + 1)tf/(k_3 + tf),$$

where k_1, b, and k_3 are parameters for which we take the default values $k_1 = 1.2$, $b = 0.75$, and $k_3 = 7$ [178].

The existence of hubness within the BM25 scheme is illustrated in Figure 7.8, which plots the distribution of N_k ($k = 10$) for several real text data sets from Table 7.1 represented with BM25. Figure 7.9 demonstrates the improvement of precision obtained through the similarity adjustment scheme described in Section 7.4.2, when BM25 representation is considered.

7.5 Summary and Future Work

We have described the tendency, called hubness, of VSM-based models to produce some documents that are retrieved surprisingly more often than other documents in a collection. We have shown that the major factor for hubness is the high (intrinsic) dimensionality of vector spaces used by such models. We described the mechanisms from which the phenomenon originates, investigated its interaction with dimensionality reduction, and demonstrated its impact on IR by exploring its relationship with the cluster hypothesis.

In order to simplify analysis by allowing quantification of the degree of violation of the cluster hypothesis, in this research we focused on data containing category labels.

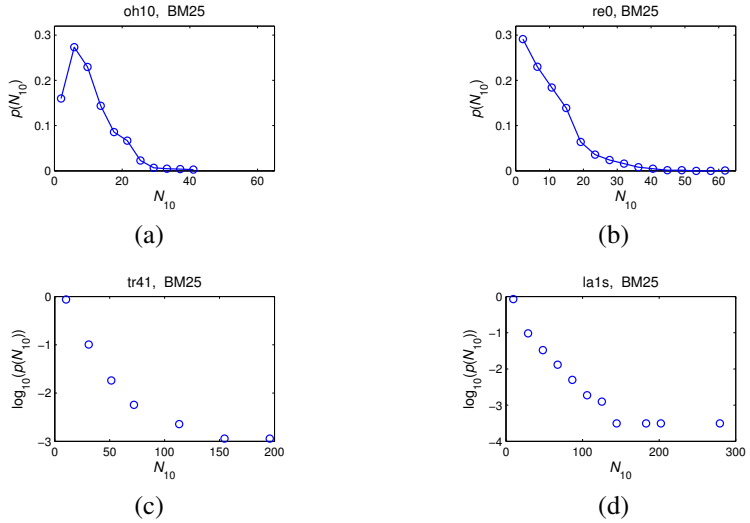


Figure 7.8: Distribution of N_{10} for real text data sets in the BM25 representation
Slika 7.8: Distribucija N_{10} za stvarne tekstualne skupove podataka u BM25 reprezentaciji

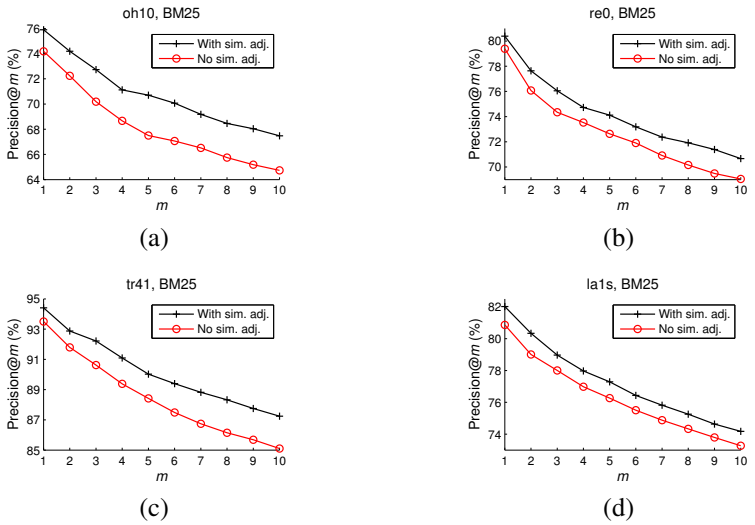


Figure 7.9: Precision at the number of retrieved results m, measured by 10-fold cross-validation, for the BM25 representation
Slika 7.9: Preciznost po broju vraćenih rezultata m, izmerena 10-*fold* kros-validacijom, za BM25 reprezentaciju

In future work we plan to extend our evaluation to larger data collections where relevance judgements are provided in a non-categorical fashion. Also, we will consider in more detail advanced models like BM25 [178] and pivoted cosine [196]. Finally, the similarity adjustment scheme described in this chapter was proposed primarily with the intent of demonstrating that hubness should be considered for the purposes of IR. In future research we intend to explore other strategies for assessing and mitigating the influence of ("bad") hubness in IR.

Part III

Document Representation and Feature Selection

Chapter 8

Term Weighting for Text Categorization

The initial motivation for the work presented in this chapter lays in the development a meta-search engine which uses text categorization (TC) to enhance the presentation of search results [162, 163]. From the context of this system, we intended to answer the three questions posed in [138]: (1) what representation to use in documents, (2) how to deal with the high number of features, and (3) which learning algorithm to use. This chapter focuses on question one and its interaction with question three, trying (but not completely succeeding) to avoid question two.

To provide answers to the questions above, we present an extensive experimental study of bag-of-words document representations, and their impact on the performance on five classifiers commonly used for text categorization: naïve Bayes, support vector machines, voted perceptron, k-nearest neighbor, and the C4.5 decision-tree learner. An unorthodox evaluation methodology is used to measure and compare the effects of different transformations of input data on each classifier, and to determine their mutual relationships with regards to classification performance. Our initial aim was to use the results as a guideline for the implementation of the meta-search system described in [163, 161, 167]. However, many of them ought to be applicable to the general case, revealing hidden relationships between transformations with respect to each other, and with respect to feature selection.

Following the discussion of related work in the next section, Section 8.2 outlines the experimental setup – how data sets were collected, which document representations were considered, and which classifiers. Section 8.3 presents the results – the representations that were found best, and the effects of, and relationships between different transformations: stemming, normalization, logtf, and idf, together with a discussion on the observed robustness of some classifiers, as well as data sets, with regards to transforming document representations. The final section summarizes the results presented in the chapter, and gives guidelines for future work.

8.1 Related Work

Although the absolute majority of works in TC employ the simple bag-of-words approach to document representation [68], studies of the impact of its variations on classification started appearing relatively recently. Leopold and Kindermann [119] experimented with the support vector machine (SVM) classifier with different kernels, term-frequency transformations, and lemmatization of German. They found that lemmatization usually degraded classification performance, and had the additional downside of great computational complexity, making SVMs capable of avoiding it altogether. Similar results were reported for neural networks on French [202]. Another study on the impact of document representation on one-class SVM [225] showed that, with a careful choice of representation, classification performance can reach 95% of the performance of SVM trained on both positive and negative examples. Kibriya et al. [106] compared the performance of SVM and a variant of the naïve Bayes classifier [176], emphasizing the importance of term-frequency and inverse-document-frequency transformations (see Section 2.1.1) for naïve Bayes. A comprehensive experimental study of term weighing schemes for text categorization with SVMs was outlined in [116], proposing a new transformation based on relevance frequencies. Debole and Sebastiani [43] investigated supervised learning of feature weights, and found that their replacement of the idf transformation can in some cases lead to significant improvement of classification performance. Another approach based on statistical confidence intervals is presented in [199], providing empirical evidence of superiority over the classical tfidf representation, and the method by Debole and Sebastiani. A bidimensional representation of documents which uses supervised term weighing was explored in [148]. The impact of word n-grams on text categorization was studied in [154] and [230]. Fuzzy approaches to document representation have also been explored, for example, in [177] and [233].

8.2 The Experimental Setup

The Weka machine-learning environment [224] was used to perform all experiments described in this chapter. The classical measures – accuracy, precision, recall, F_1, and F_2 (see Section 2.3.3) – were chosen to evaluate the performance of classifiers on many variants of the bag-of-words representation of documents (that is, short Web-page descriptions) taken from the dmoz Open Directory. The F_2 measure, which gives emphasis to recall over precision, is included for reasons similar to those in [137], where false positives are preferred to false negatives. What this means for categorization of search results is that it is preferred to overpopulate categories to a certain extent, over leaving results unclassified. Classification time and training time are also important factors, since the system needs to be running on a Web server classifying hundreds, possibly thousands of documents in real-time, and needs to be trained beforehand with as many examples as possible from the huge dmoz taxonomy.

Data set /	Features		Examples		
Category	Not stem.	Stemmed	Total	Pos.	Neg.
Arts	3811	3142	626	300	326
Business	4248	3444	655	317	338
Computers	4293	3479	700	336	364
Games	4276	3551	764	382	382
Health	4460	3617	766	380	386
Home	4425	3583	765	374	391
Recreation	4389	3564	735	365	370
Science	4695	3792	754	379	375
Shopping	4470	3633	729	361	368
Society	4164	3402	675	344	331
Sports	4094	3391	753	380	373

Table 8.1: Extracted dmoz data sets
Tabela 8.1: Skupovi podataka izdvojeni iz dmoz-a

8.2.1 Data Sets

A total of eleven data sets, one for each chosen top-level category, were extracted from the dmoz collection dated July 2, 2005. The examples are either positive – taken from the corresponding category, or negative – distributed over all other categories, making this a binary classification problem. As initial tests showed that many classifiers implemented in Weka had difficulties dealing with imbalanced class distributions, we kept the positive-negative ratio around 50–50. This is also justified by our preference of false positives to false negatives.

Since the dmoz hierarchy is large (the content occupies over two gigabytes of RDF data), we wrote a custom tool dmoz2arff which extracts examples from the dmoz RDF data in one pass, and offers basic facilities for selection of categories and examples, moving examples to higher levels, stop-word elimination (with the standard stop-word list from [184]), and stemming [158]. Table 8.1 summarizes the extracted data sets, showing the number of features (including the class feature) before and after stemming, the total number of examples, and the number of positive and negative ones.

When constructing the data sets and choosing the number of examples, care was taken to keep the number of features below 5000, for two reasons. The first reason was to give all classifiers an equal chance, because some of them are known not to be able to handle more than several thousand features, and to do this without using some explicit form of feature selection (basically, to avoid question two from the beginning of the chapter). The second reason was the feasibility of running the experiments with the C4.5 classifier, due to its long training time. However, results from Section 8.3.4 (regarding the idf transformation) prompted us to use the simple dimensionality-reduction method based on term frequencies (TFDR), eliminating features representing the least frequent terms, at the same time keeping the number of features at around 1000. Therefore, two bundles of data sets were generated, one with and one without TFDR.

Not stemmed		Stemmed	
Not normalized	Normalized	Not normalized	Normalized
01		m-01	
idf		m-idf	
tf	norm-tf	m-tf	m-norm-tf
logtf	norm-logtf	m-logtf	m-norm-logtf
tfidf	norm-tfidf	m-tfidf	m-norm-tfidf
logtfidf	norm-logtfidf	m-logtfidf	m-norm-logtfidf

Table 8.2: Document representations
Tabela 8.2: Reprezentacije dokumenata

8.2.2 Document Representations

The bag-of-words representation, together with all its transformations described in Section 2.1.1, was used in the experiments. For notational convenience, the abbreviations denoting each transformation were appended to the names of data sets, for example, Arts-norm-tf refers to the normalized term-frequency representation of the *Arts* data set.

All meaningful combinations of the transformations, along with stemming (m), add up to 20 different variations of document representations, summarized in Table 8.2. This accounts for a total of $11 \cdot 20 \cdot 2 = 440$ different data sets for the experiments.

8.2.3 Classifiers

Five classifiers implemented in Weka are used in this study: ComplementNaiveBayes (CNB), SMO, VotedPerceptron (VP), IBk, and J48.

CNB [176, 106] (page 38) is an improved version of the NaiveBayesMultinomial classifier [134], optimized for application to text. Initial tests showed that CNB consistently outperforms its predecessor on our data sets, although not at a statistically significant level (0.05). SMO [157, 101] (page 37) is an implementation of Platt's Sequential Minimal Optimization algorithm for training SVMs. VP was first introduced by Freund and Schapire [63] (page 35), and shown to be a simple, yet effective classifier for high-dimensional data. IBk is a variation of the classical k-nearest neighbor algorithm [4] (page 40), and J48 is based on revision 8 of the C4.5 decision-tree learner [159] (page 41).

All classifiers were run using their default parameters, with the exception of SMO, where the option not to normalize training data was chosen. IBk performed rather erratically during initial testing, with performance varying greatly with different data sets, choices of k and distance weighing, so in the end we kept $k = 1$ as it proved most stable. We were unable to reproduce the state-of-the-art performance achieved elsewhere [191], but report the results anyway, as some of them may still prove valuable. Only in the late phases of experimentation we realized that the Euclidean distance

measure is generally not suitable to high-dimensional text data sets, like the ones used in this chapter. Therefore, the bad performance of IBk may be treated as an experimental verification of this fact in the context of classification.

Although SVMs are generally considered the best classifier for text (especially when accuracy is concerned), much may depend on the properties of the data set (as was effectively demonstrated by Gabrilovich and Markovitch [68]), the evaluation measure that is considered important (we are placing an emphasis on the less commonly used F_2), and the final application of the classifier. For these reasons, all mentioned classifiers were included and equally treated in the experiments.

8.3 Results

A separate Weka experiment was run for every classifier with the 20 document representation data sets, for each of the 11 major categories. Results of all evaluation measures were averaged over five runs of 4-fold cross-validation, following [57, 68]. Measures were compared between *data sets* using the corrected resampled t-test [139], at significance level 0.05, and the number of statistically significant wins and losses of each document representation added up for every classifier over the 11 categories.

For the sake of future experiments and the implementation of the meta-search system, best representations for each classifier were chosen, based on wins–losses values summed-up over all data sets. The declared best representations were not winners for all 11 categories, but showed best performance overall. For VP and J48 the choice was simple: m-logtf appeared among the winners both without and with TFDR. Since TFDR broke the performance of IBk, m-norm-logtf was declared best in that department, since it was the winner without TFDR. As for CNB, m-norm-tf was best without TFDR, while idf was best with, but by a much smaller margin, therefore m-norm-tf was chosen. For SMO, the situation was opposite with regards to the idf transformation: m-norm-tfidf was the winner without TFDR, and m-norm-logtf with, by a bigger margin, so m-norm-logtf was considered best. Section 8.3.4 explains in more detail the reasons we decided to stay away from the idf transformation, and Chapter 9 presents a thorough study on its impact on feature selection and classification.

Table 8.3 shows the wins–losses values of the declared best document representations for each classifier, without and with TFDR. Binary representations were practically never among the best, for all data sets, confirming the widespread agreement on the need for tf-based document representations.

For illustrating the impact of document representations on classification, Tables 8.4 and 8.5 summarize the performance of classifiers on the best representations, and the improvements over the worst ones, on the *Home* data set, without and with TFDR, respectively. Note that the emphasis of the work described in this chapter is not on fine-tuning the performance of classifiers, even using document representations, as much as it is on determining the impacts and relationships between different transformations

	CNB		SMO		VP		IBk	J48	
	m-norm-tf		m-norm-logtf		m-logtf		m-norm-logtf	m-logtf	
Accuracy	41	1	1	37	15	2	119	40	40
Precision	45	1	20	6	29	12	11	-6	-5
Recall	4	1	-4	68	0	0	67	56	57
F_1	28	1	0	47	7	0	120	59	52
F_2	9	0	-3	71	0	0	78	63	57
Total	127	4	14	229	51	14	395	212	201

Table 8.3: Wins–losses values of best document representations for each classifier, on data sets without (left columns) and with dimensionality reduction
Tabela 8.3: *Pobede–porazi* najboljih reprezentacija dokumenata za svaki klasifikator, na podacima bez (leve kolone) i sa redukcijom broja dimenzija

	CNB	SMO	VP	IBk	J48
Accuracy	82.56 (**5.26**)	83.19 (1.67)	78.38 (**5.12**)	74.93 (**21.96**)	71.77 (**3.64**)
Precision	81.24 (**8.66**)	85.67 (**3.86**)	80.45 (**7.85**)	71.32 (**14.32**)	90.24 (1.60)
Recall	83.91 (1.81)	78.93 (3.80)	74.06 (0.96)	81.66 (**45.20**)	47.59 (**10.59**)
F_1	82.48 (**3.64**)	82.07 (2.17)	77.02 (**4.23**)	76.07 (**33.90**)	62.12 (**9.09**)
F_2	83.31 (2.19)	80.14 (3.30)	75.20 (2.16)	79.31 (**39.72**)	52.48 (**10.41**)

Table 8.4: Performance of classification (in %) using the best document representations on the *Home* data set, *without* dimensionality reduction, together with improvements over the worst representations (statistically significant ones are in **boldface**)
Tabela 8.4: Performanse klasifikacije (u procentima) za najbolje reprezentacije dokumenata na skupu podataka *Home*, *bez* redukcije broja dimenzija, zajedno sa poboljšanjima u odnosu na najgore reprezentacije (statistički značajna poboljšanja označena su **boldom**)

(stemming, normalization, logtf, idf) and dimensionality reduction, with regards to each classifier. This is the prevailing subject of the remainder of this section.

8.3.1 Effects of Stemming

The effects of stemming on classification performance were measured by adding-up the wins–losses values for stemmed and nonstemmed data sets, and examining their difference, depicted graphically in Figure 8.1. It can be seen that stemming improves almost all evaluation measures, both without and with TFDR. With TFDR, the effect of stemming is generally not as strong, which is understandable because its impact as a dimensionality-reduction method is reduced. CNB is then practically unaffected, only SMO exhibits an increased tendency towards being improved. Overall, J48 is especially sensitive to stemming, which can be explained by its merging of words into

	CNB	SMO	VP	J48
Accuracy	85.86 (1.12)	82.80 (**4.60**)	79.29 (4.18)	71.49 (**3.15**)
Precision	86.48 (1.78)	83.77 (4.79)	81.10 (**6.40**)	90.21 (1.48)
Recall	84.39 (1.07)	80.64 (**5.72**)	75.45 (2.24)	47.43 (**9.84**)
F_1	85.35 (1.04)	82.07 (**4.88**)	78.08 (3.57)	61.98 (**8.40**)
F_2	84.75 (0.67)	81.19 (**4.96**)	76.46 (2.24)	52.33 (**9.66**)

Table 8.5: Performance of classification (in %) using the best document representations on the *Home* data set, *with* dimensionality reduction, together with improvements over the worst representations (statistically significant ones are in **boldface**)
Tabela 8.5: Performanse klasifikacije (u procentima) za najbolje reprezentacije dokumenatana skupu podataka *Home*, *sa* redukcijom broja dimenzija, zajedno sa poboljšanjima u odnosu na najgore reprezentacije (statistički značajna poboljšanja označena su **boldom**)

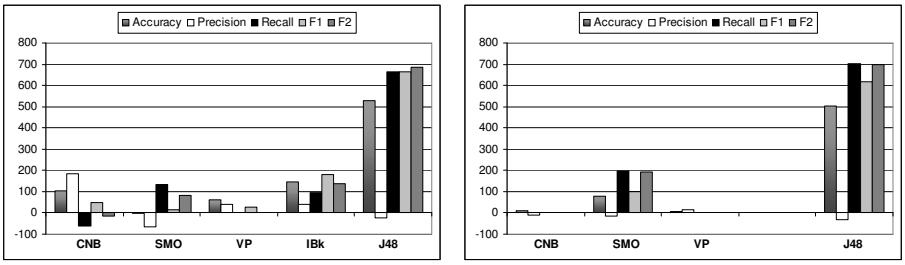


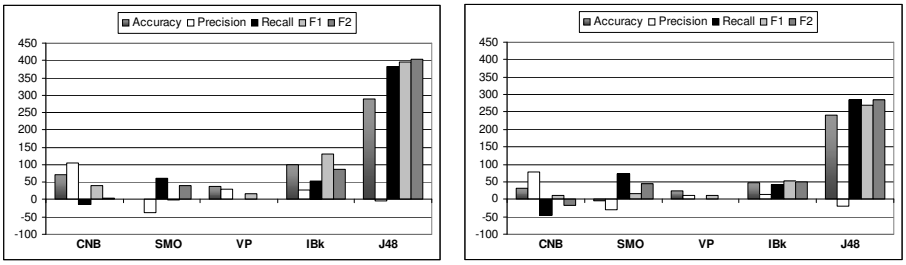
Figure 8.1: Effects of *stemming* without (a) and with dimensionality reduction (b)
Slika 8.1: Efekti *stemming*-a bez (a) i sa redukcijom broja dimenzija (b)

more discriminative features, suiting the algorithm's feature-selection method when constructing the decision tree.

To investigate the relationships between stemming and other transformations, a chart was generated for each transformation, measuring the effect of stemming on representations with and without the transformation applied. Figure 8.2 shows the effect of stemming on non-normalized and normalized data, without TFDR. It can be noted that normalized representations are affected by stemming more strongly (for the better). The same holds with TFDR applied (charts not shown).

The logtf transformation exhibited no influence on the impact of stemming, regardless of TFDR. The corresponding charts are only scaled-down versions of Figure 8.1, and are not shown.

Applying the idf transformation to tf, without TFDR, made no difference in stemming performance, except for greater improvements on IBk. With TFDR the situation



(a) (b)

Figure 8.2: Effects of *stemming* on non-normalized (a) and normalized data (b), *without* dimensionality reduction
Slika 8.2: Efekti *stemming*-a na ne-normalizovanim (a) i normalizovanim podacima (b), *bez* redukcije broja dimenzija

was a little different: application of idf led to a drop in the effect on accuracy and precision of ComplementNaiveBayes, and to a rise of the accuracy of the SMO classifier, as can be seen in Figure 8.3.

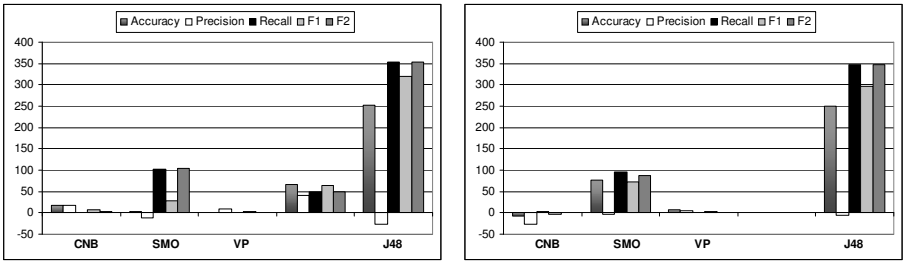
The above analysis confirms the common view of stemming as a method for improving classification performance for English. However, this may not be the case for other languages, for instance German [119] and French [202].

8.3.2 Effects of Normalization

The chart in Figure 8.4 shows that normalization tends to improve classification performance in a majority of cases. Without TFDR, VP was virtually unaffected, CNB and SMO were improved on all counts but recall (and consequently F_2), while the biggest improvement was on IBk, which was anticipated since normalization assisted the comparison of document vectors. J48 was the only classifier whose performance worsened with normalization. Apparently, J48 found it tougher to find appropriate numeric intervals within the normalized weights, for branching the decision tree. With TFDR, CNB joined VP in its insensitivity, while SMO witnessed a big boost in performance when data was normalized.

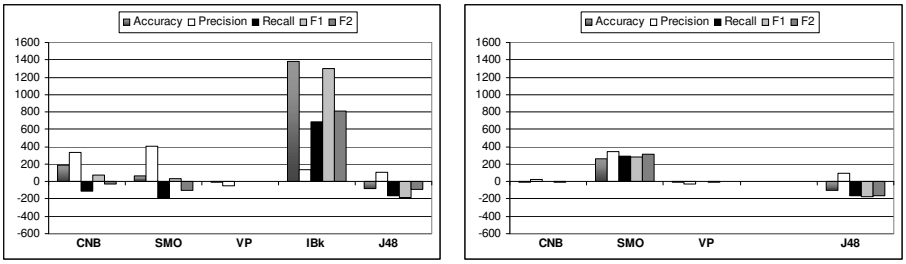
No significant interaction between normalization and stemming was revealed, only that stemmed J48 was more strongly worsened by normalization. It appears that normalization misleads J48 from the discriminative features introduced by stemming.

Normalization and the logtf transformation exhibited no notable relationship, while with idf transformed data, normalization had stronger influence on classification. After dimensionality reduction, this tendency was especially noticeable with the improvement of the precision of SMO (Figure 8.5). This can be explained by the fact that idf severely worsens the performance of SMO after TFDR (see Section 8.3.4), and normalization compensated somewhat for this. This compensating effect of one transformation on



(a) (b)

Figure 8.3: Effects of *stemming* on data without (a) and with the idf transformation applied to tf (b), *with* dimensionality reduction
Slika 8.3: Efekti *stemming*-a na podacima bez (a) i sa idf transformacijom primenjenom na tf (b), *sa* redukcijom broja dimenzija



(a) (b)

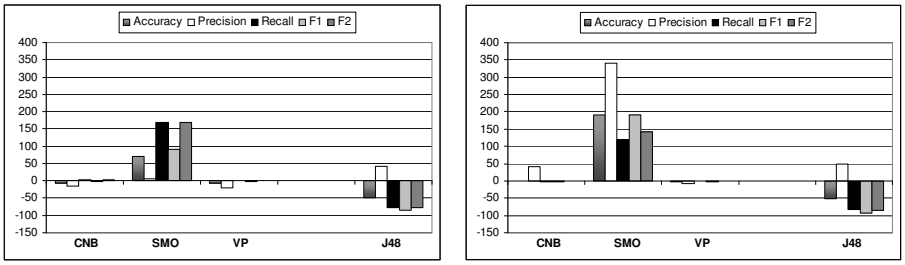
Figure 8.4: Effects of *normalization* without (a) and with dimensionality reduction (b)
Slika 8.4: Efekti *normalizacije* bez (a) i sa redukcijom broja dimenzija (b)

the performance-degrading influences of another was found to be quite common in all experiments conducted in this chapter.

It is important to emphasize that the data sets used in these experiments consist of short documents, and therefore normalization does not have as strong an impact as it would have if the differences in document lengths were more drastic. For this reason, the conclusions above may not hold for the general case, for which a further, more comprehensive study is needed.

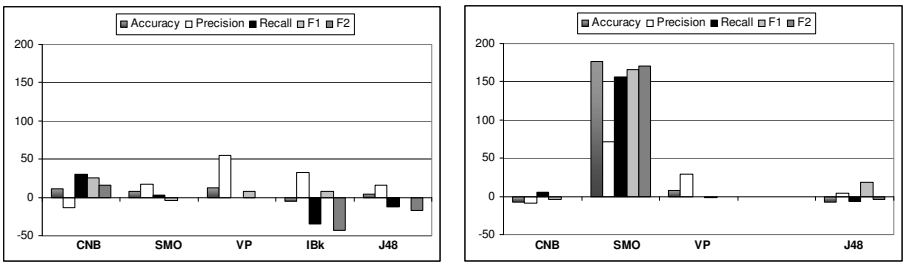
8.3.3 Effects of the logtf Transformation

As can be seen in Figure 8.6, the logtf transformation causes mostly mild improvements of classification. With TFDR, improvements are greater on SMO, while the impact on other classifiers is weaker.



(a) (b)

Figure 8.5: Effects of *normalization* on data without (a) and with the idf transformation applied to tf (b), *with* dimensionality reduction
Slika 8.5: Efekti *normalizacije* na podacima bez (a) i sa idf transformacijom primenjenom na tf (b), *sa* redukcijom broja dimenzija



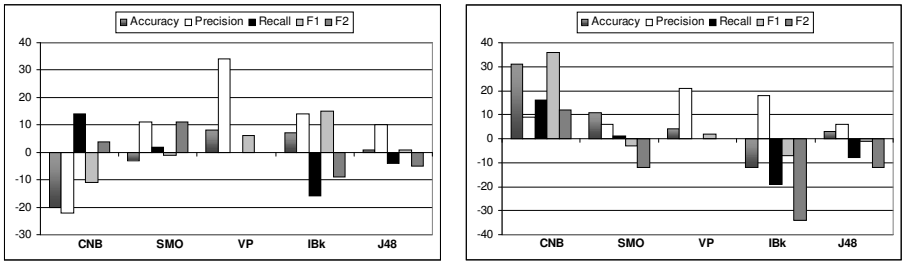
(a) (b)

Figure 8.6: Effects of the logtf transformation without (a) and with dimensionality reduction (b)
Slika 8.6: Efekti logtf transformacije bez (a) i sa redukcijom broja dimenzija (b)

Figure 8.7 shows that logtf has a much better impact on CNB when idf is also applied, without TFDR. This is similar to the compensating effect of normalization on idf with the SMO classifier from the previous section. Relations change quite dramatically when TFDR is applied (Figure 8.8), but the effect on SMO is again analogous to the previous section. The improvements on CNB are especially significant, meaning that logtf and idf work together on improving classification performance.

Without TFDR, the interaction of logtf and normalization varied across classifiers: logtf improved CNB and IBk on normalized data, while the others were improved without normalization. With TFDR, the chart looks very much like the left-right reverse of Figure 8.8 – with logtf having a weaker positive effect on normalized data, especially for CNB and SMO, which were already improved by normalization.

Understandably, the logtf transformation has a stronger positive impact on non-stemmed data, regardless of dimensionality reduction, with the exception of VP which



(a) (b)

Figure 8.7: Effects of the logtf transformation on data without (a) and with the idf transformation applied to tf (b), *without* dimensionality reduction
Slika 8.7: Efekti logtf transformacije na podacima bez (a) i sa idf transformacijom primenjenom na tf (b), *bez* redukcije broja dimenzija

exhibited no variations. This is in line with the witnessed improvements that stemming introduces on its own, and the already noted compensation phenomenon.

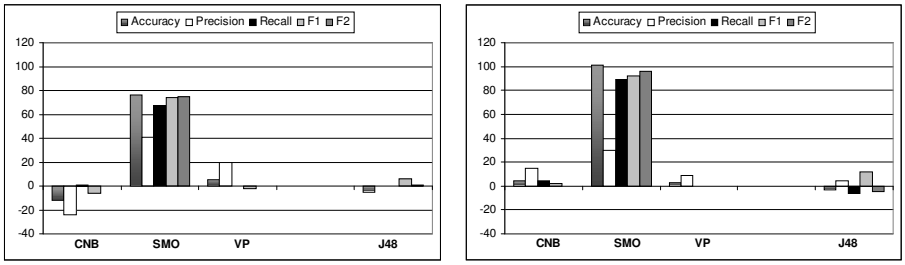
8.3.4 Effects of the idf Transformation

Applying the idf transformation to data turned out to have the richest repertoire of effects, from significant improvement, to severe degradation of classification performance. Figure 8.9(a) illustrates how idf drags down the performance of all classifiers except SMO, without TFDR. It was for this reason we introduced TFDR in the first place, being aware that our data had many features which were present in only a few documents. We expected idf to improve, or at least degrade to a lesser extent the performance of classification. That did happen, as Figure 8.9(b) shows, for all classifiers *except* SMO, whose performance was drastically degraded! The simple idf document representation rose from being one of the worst, to one of the best representations of documents, for all classifiers but SMO.

No significant correlation was detected by applying idf to stemmed and nonstemmed data, however plenty of different effects were noticeable with regards to normalization. Without TFDR (Figure 8.10), a stronger worsening effect on non-normalized data was exhibited with CNB, VP, and IBk, while for SMO normalization dampened idf's improvement of recall, but overturned the degradation of accuracy and precision. With TFDR, the picture is quite different (Figure 8.11): normalization improved the effects on CNB and VP, with SMO witnessing a partial improvement on precision, while J48 remained virtually intact.

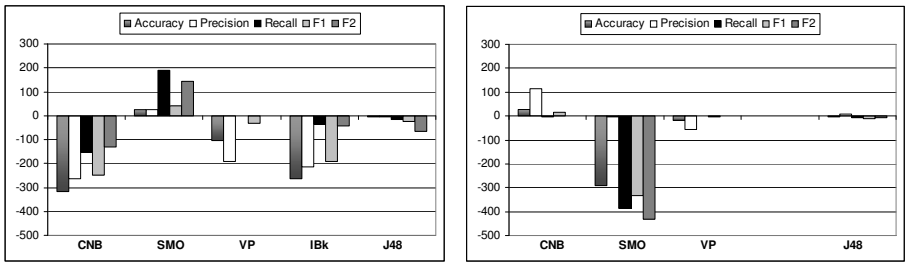
The impact of idf on (non-)logtfed data sets showed no big differences – trends remained the same as in Figure 8.9, perhaps a little stronger with logtf applied.

The above analysis shows that one needs to be careful when including the idf transformation in the representation of documents. Removing infrequent features is an im-



(a) (b)

Figure 8.8: Effects of the logtf transformation on data without (a) and with the idf transformation applied to tf (b), *with* dimensionality reduction
Slika 8.8: Efekti logtf transformacije na podacima bez (a) i sa idf transformacijom primenjenom na tf (b), *sa* redukcijom broja dimenzija



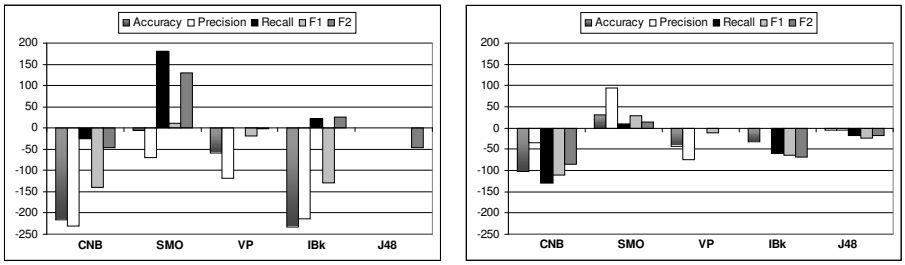
(a) (b)

Figure 8.9: Effects of idf applied to tf without (a) and with dimensionality reduction (b)
Slika 8.9: Efekti idf-a primenjenog na tf bez (a) i sa redukcijom broja dimenzija (b)

portant prerequisite to its application, since idf assigns them often unrealistic importance, but that may not be enough, as was proved by the severe degradation of SMO's performance.

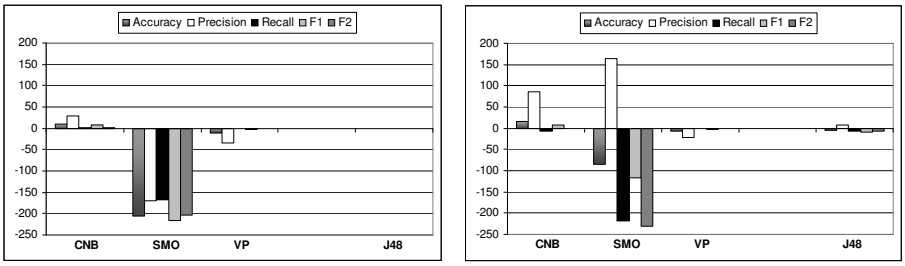
8.3.5 Robustness

An interesting phenomenon observed in the experiments is the apparent insensitivity of some classifiers to document representations, which we will refer to as *robustness*. The "Total" row of Table 8.3 provides a hint, with the winning representation for VP showing the lowest number of wins–losses overall, especially with regards to recall and F_2, as did CNB with, and SMO without TFDR (although the representations were not exactly optimal for those cases). A better indicator is the summed-up number of wins (the number of losses is the same) for each classifier, over all data sets, document representations and evaluation measures, shown in the Total *row* of Table 8.6, which



(a) (b)

Figure 8.10: Effects of idf applied to tf on non-normalized (a) and normalized data (b), *without* dimensionality reduction
Slika 8.10: Efekti idf-a primenjenog na tf, na ne-normalizovanim (a) i normalizovanim podacima (b), *bez* redukcije broja dimenzija



(a) (b)

Figure 8.11: Effects of idf applied to tf on non-normalized (a) and normalized data (b), *with* dimensionality reduction
Slika 8.11: Efekti idf-a primenjenog na tf, na ne-normalizovanim (a) i normalizovanim podacima (b), *sa* redukcijom broja dimenzija

	CNB		SMO		VP		IBk	J48		Total	
Accuracy	265	23	46	277	67	9	1155	336	321	1869	1785
Precision	355	79	262	465	124	34	494	64	56	1299	1128
Recall	121	5	234	626	0	0	929	468	486	1752	2046
F_1	178	14	35	297	21	2	1140	477	437	1851	1890
F_2	83	2	152	535	2	0	837	568	489	1642	1863
Total	1002	123	729	2200	214	45	4555	1913	1789	8413	8712

Table 8.6: Total number of wins (=losses) of all document representations, for each classifier and evaluation measure, on data sets without (left columns) and with TFDR. The right Total column includes the left IBk column in the sum, to enable comparison of the number of wins without and with TFDR

Tabela 8.6: Ukupan broj pobeda (=poraza) svih reprezentacija dokumenata, za svaki klasifikator i meru, na skupovima podataka bez (leve kolone) i sa TFDR. Desna kolona „Total“ uključuje levu kolonu „IBk“ u sumi, da bi se omogućilo poređenje broja pobeda bez i sa TFDR

confirms the above observation. When examining the partial sums for each evaluation measure (the Total *column*), precision shows the lowest variation with regards to document representation. However, every classifier exhibits its lowest sensitivity at different measures: CNB and VP at recall and F_2, SMO at accuracy and F_1, IBk and J48 at precision. This correlates with the lowest improvement rates exemplified on the *Home* data set in Tables 8.4 and 8.5.

Robustness regarding document representations was also observed on *data sets*. For example, the total number of wins for all document representations, over all classifiers and measures, for the *Computers* data set is 167, while for *Games* the number is 1101 (without TFDR; with TFDR the numbers are almost the same). This may be due to a presence of more discriminating features in the *Computers* data set (that is, features that are better correlated with the class feature), or a combination of that and other factors, and calls for a further investigation towards developing a simple, theoretical criteria for determining the robustness and, going further, a best document representation for a particular data set.

8.3.6 Training and Classification Speed

Since the experiments were performed on different computers and under different circumstances within one computer (that is, processes running in the background), no exact empirical quantification of the speed of different classifiers can be given. Nevertheless, our general impressions of both training and classification speeds agree with wide-spread opinions. In the following characterization the operator $>$ will be used to denote the relation “faster than,” and $\gg$ for “an order of magnitude faster than.”

For *training speed*, the observed relations are:

$$\text{CNB}, \text{IBk} \gg \text{VP} > \text{SMO} \gg \text{J48}\,.$$

Classification speed yields somewhat different relations:

$$\text{CNB}, \text{J48} \gg \text{SMO} > \text{VP} \gg \text{IBk}\,.$$

It can be seen that CNB is among the best at both speeds, while IBk and J48 occupy opposite extremes, which is understandable considering the principles of their functioning described in Section 2.3.1.

8.4 Summary and Future Work

By using transformations in bag-of-words document representations there is, essentially, no new information added to a data set which is not already there (except for the transition from 01 to tf representations). The general-purpose classification algorithms, however, are unable to derive such information without assistance, which is understandable because they are not aware of the exact nature of the data being processed. It is therefore expected of transformations to have a significant effect on classification performance, which was experimentally demonstrated at the beginning of Section 8.3. The fact that this issue is simply ignored by many studies and applications of text classification is somewhat striking.

Besides helping to determine a best representation for each classifier, the experiments revealed the individual effects of transformations on different evaluation measures of classification performance, and some of their relationships. Stemming generally improved classification, partly because of its role as a dimensionality-reduction method. It had an exceptionally strong improving impact on J48, which can be explained by its merging of words into more discriminative features, suiting the algorithm's feature-selection method when constructing the decision tree. Normalization enhanced CNB, SMO, and especially IBk, leaving VP practically unaffected and worsening the performance of J48. Although dmoz data consists of short documents, normalization did have a significant impact, but no definite conclusions can be drawn for the general case. The logtf transformation had mostly a mild improving impact, except on SMO with TFDR, which exhibited stronger improvement. SMO is known to work well with small numeric values, which explains its sensitivity to normalization and logtf. The situation with idf was trickier, with the effects depending strongly on dimensionality reduction for CNB and SMO, but in opposite directions: CNB was degraded by idf without TFDR, and improved with TFDR; for SMO it was vice versa.

The most common form of relationship between transformations that was noticed were the compensating effects of one transformation on the performance degrading impact of another (for example, normalization with idf on SMO, logtf with idf on CNB and SMO). The logtf and idf transformations seemed to work together on improving

CNB after TFDR. The impact of idf on normalization was most complex, with great variation in the effects on different evaluation measures. Note that the method for determining relations between transformations appeared not to be commutative – for instance, the effects of normalization on idf transformed data and of idf on normalized data are not the same. Some relationships can be missed when looking only one way.

The comments above refer to the general case of performance measuring. Some transformations (especially idf) may improve one measure, at the same time degrading another. Often, the preferred evaluation measure, chosen with the application of the classifier in mind, will need to be monitored when applying the presented results. In our case, for applying classification to search results, the F_2 measure is most important.

Robustness regarding document representations, which applies both to classifiers and data sets, is an interesting area for further theoretical investigations – exploring possibilities for developing simple tests for determining the robustness, interactions with particular transformations and, ultimately, a best document representation for a particular classifier and/or data set, without extensive experimentation. Such tests may be useful in situations where detailed fine-tuning of term weights is not feasible.

The main difficulty with comprehensive text-categorization experiments is sheer size. Roughly speaking, factors such as data sets, document representations, dimensionality-reduction methods, reduction rates, classifiers, and evaluation measures, all have their counts multiplied, leading to a combinatorial explosion which is hard to handle. We tackled this problem by excluding detailed experimentation with dimensionality reduction, and using dmoz as the only source of data. Therefore, no definite truths, but only pointers can be derived from the described experience. A more comprehensive experiment, featuring other common corpora (see page 48), longer documents, and dimensionality-reduction methods is called for to shed more light on the impacts and relationships of all the above mentioned factors.

In the next phase, however, we have conducted experiments with feature-selection methods on dmoz data, with the document representations that were determined best for each classifier, before applying the winning combination to categorization of search results. Experiences with this batch of experiments are the subject of the next chapter.

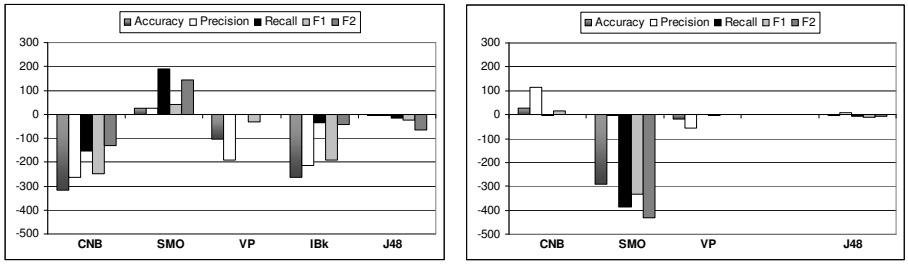
Chapter 9

Term Weighting and Feature Selection

Many studies in text categorization (TC) analyzed the interactions between feature-selection (FS) methods, reduction rates, and classifiers, on a great variety of corpora. However, many of them completely neglected the issue of document representation, fixing one particular representation and deriving general conclusions from performed experiments. In this study, different document representations will be used – each classifier will be trained and tested employing the document representation that was found most suitable in the previous chapter.

Chapter 8 demonstrated that there can be statistically significant differences between document representations for every considered classifier with at least one of the standard performance measures, and that feature selection can alter those differences, sometimes beyond recognition. The emphasis of that study, however, was on determining the relationships between different transformations of the bag-of-words representation, including stemming, normalization of weights, tf, idf, and logtf, that is, on determining their behavior when used in meaningful combinations.

Figure 9.1 (reproducing Figure 8.9 for convenience) depicts the experimentally measured effect of the idf transformation, when included in the document representation. The charts show the difference between the added up wins–losses values of representations including and not including idf. Figure 9.1(a) depicts the wins–losses differences without any kind of dimensionality reduction, while Figure 9.1(b) shows the values when only around 1000 most frequent terms are retained. There is a clear discrepancy between the effects of idf on CNB and SMO: while it degrades the performance of CNB and improves SMO without dimensionality reduction, with dimensionality reduction the result was completely opposite! This struck us as counterintuitive – discarding least frequent terms meant discarding terms with a high idf score, which should either have improved or degraded classification performance, but not both. Improvement would have happened in case the less frequent terms were not discriminative



(a) (b)

Figure 9.1: Effects of idf applied to tf without (a) and with dimensionality reduction (b)
Slika 9.1: Efekti idf-a primenjenog na tf bez (a) i sa redukcijom broja dimenzija (b)

(correlated with the class feature) and idf was giving them unrealistically high weight, while degradation would have taken place if such terms were highly discriminative (this depending on the actual data sets). Thus an interesting question was raised: do CNB and SMO function at such different levels that they were able to capture completely different notions of term frequencies and weights?

The initial motivation for the work that will be presented in this chapter was to determine the best feature-selection method, reduction rate, and classifier, in the context of dmoz Open Directory Web-page descriptions. The winning combination was used for classification of search results by the system presented in [163, 161, 167]. For this reason, Section 9.2.1 describes experimentally obtained rankings of feature-selection methods and reduction rates. Section 9.2.2 then presents a study motivated by the questions raised above, concentrating on the effects that transformations of the bag-of-words representation produce on several commonly used feature-selection methods. Besides considering more methods, both studies will take into account a much wider array of reduction rates than the study described in the previous chapter. Since the data sets used in that study were rather small in order to *avoid* issues of dimensionality reduction, this study will use bigger, more realistic data sets also extracted from the dmoz taxonomy.

The next section introduces the experimental setup: used data sets, document representations, feature-selection methods and classifiers. Section 9.2 describes the most interesting findings about the rankings of FS methods, and the interactions between the idf transformation, feature-selection methods, and reduction rates. The last section gives some concluding remarks and guidelines for possible future work.

9.1 The Experimental Setup

As in Chapter 8, the Weka machine-learning environment [224] was used as the platform for performing all experiments in this chapter. Classification performance was measured by the same standard metrics: accuracy, precision, recall, F_1, and F_2.

Data set	Features	Examples		
		Total	Pos.	Neg.
Arts	14144	6261	3002	3259
Computers	15152	7064	3390	3674
Sports	14784	7694	3881	3813
Arts/Music	13968	8038	4069	3969
Games/Roleplaying	12530	8948	4574	4374
Science/Physics	10558	4973	2519	2454

Table 9.1: Extracted dmoz data sets
Tabela 9.1: Skupovi podataka izdvojeni iz dmoz-a

9.1.1 Data Sets

Initially in Chapter 8, we restricted the domain of experimentation to 11 top-level categories of dmoz which were considered suitable for the task of sorting search results, namely *Arts*, *Business*, *Computers*, *Games*, *Health*, *Home*, *Recreation*, *Science*, *Shopping*, *Society*, and *Sports*. For the purposes of this study, six two-class data sets, summarized in Table 9.1, were extracted from the dmoz data. The table shows the number of features (not including the class feature) of each data set, the total number of examples (documents), and the number of positive and negative ones. Every data set corresponds to one dmoz topic from which positive examples are taken, while the negative ones are chosen in a stratified fashion from all other topics at the same level of the hierarchy, within a common parent topic. Thus, for each of the chosen first-level categories (*Arts*, *Computers*, *Sports*) the negative examples are extracted from all leftover dmoz data, while for the second-level topics (*Music*, *Roleplaying*, *Physics*), negative examples are restricted to their first-level parents (*Arts*, *Games*, and *Science*, respectively). As before, all texts were preprocessed by eliminating stop words using the standard stop-word list from [184]. Since best document representations that were determined in Chapter 8 all included stemming, the Porter Stemmer [158] was applied to every data set.

9.1.2 Document Representations

For each data set, the variations of the bag-of-words representation which were considered best for at least one classifier were generated – m-norm-tf, m-norm-logtf, and m-logtf. Also, in order to study the interaction between the idf transformation and feature selection (Section 9.2.2), m-norm-tfidf was also computed.

9.1.3 Feature Selection

The examined feature-selection methods are more-less well known and widely used in TC: chi-square (CHI), information gain (IG), gain ratio (GR), ReliefF (RF) and symmetrical uncertainty (SU), and were all described in Section 2.8.

In the experiments, the performance of classification was measured on data sets consisting of the top 100, 500, 1000, 3000, 5000, 7000, and 10000 features selected by each method, and on data sets with all features. The simple feature-selection method from our previous study in Chapter 8, TFDR, which discards least frequent features, was not used. The reason was the difficulty to follow the same reduction rates on different data sets without randomly discarding features with same frequencies of appearance, which would have compromised the validity of observations.

9.1.4 Classifiers

The same classifiers implemented in Weka that were used in Chapter 8 were employed in this study: ComplementNaiveBayes (CNB), SMO, VotedPerceptron (VP), IBk, and J48 (see Section 8.2.3).

As in Chapter 8, experiments were carried out on each data set with a particular document representation, FS method and number of features, and classifier, in five runs of 4-fold cross-validation. Due to the slow training time of J48, its results were generated only for data sets consisting of 100, 500, and 1000 features. Values of evaluation measures were again compared using the corrected resampled t-test [139] implemented in Weka, at significance level 0.05, and averaged over runs and folds for reporting.

9.2 Results

9.2.1 Rankings of Feature-Selection Methods and Reduction Rates

Table 9.2 shows the top five combinations of feature-selection methods and reduction rates measured by accuracy, precision, recall, F_1, and F_2, respectively. The wins–losses (WL) values are summed-up over all data sets, while the actual values of performance measures are averaged. Note that the tables are sorted in order of wins–losses, which does not necessarily correspond to the averaged measure values. We chose wins–losses as the primary indicator of performance because it already dominates all discussion of the experimental results presented in this part of the dissertation.

It can be seen from the tables that different classifiers are best performers when different measures are considered. CNB and SMO are best at accuracy and the F_1 measure, VP and IBk are ahead of the field in precision, and CNB alone is the clear winner at F_2, and especially recall, with 98.2% for GR at 100 selected features.

Also, it is evident that some measures produce rankings that are not very different from one another. Tables for F_1 and F_2 give quite similar rankings, which may have been expected, but so do accuracy and F_1, suggesting that these measures have some common properties.

Considering feature-selection methods, the tables indicate that CHI, IG, and SU tend to "stick together," with similar performance at same reduction rates, while GR sometimes "breaks away" from the pack, although it is based on the same theoretical

CNB m-norm-tf			**SMO** m-norm-logtf			**VP** m-logtf			**IBk** m-norm-logtf			**J48** m-logtf		
FS	WL	acc.	FS	WL	acc.	FS	WL	acc.	FS	WL	acc.	FS	WL	acc.
all	133	**89.4**	chi1000	169	**90.1**	chi1000	144	88.6	gr500	203	84.4	chi500	34	83.3
chi10000	113	89.2	ig1000	168	**90.1**	su1000	140	88.6	su100	172	80.5	su500	33	83.4
ig10000	113	89.1	su1000	167	**90.1**	ig1000	134	88.5	gr100	171	81.2	ig500	33	83.3
gr10000	112	89.1	gr1000	161	90.0	su500	117	88.2	ig100	166	79.8	ig1000	31	83.3
ig3000	93	88.9	gr3000	129	89.6	su3000	116	88.2	chi100	153	79.1	su1000	31	83.3
FS	WL	pr.	FS	WL	pr.	FS	WL	pr.	FS	WL	pr.	FS	WL	pr.
ig7000	158	88.9	ig1000	122	92.2	gr500	203	**93.6**	ig7000	146	**94.2**	gr500	50	87.7
su7000	158	88.9	su1000	120	92.2	gr1000	179	91.1	su7000	146	**94.2**	gr1000	48	89.0
chi7000	158	88.9	gr1000	120	92.2	gr100	130	89.9	gr7000	146	**94.2**	gr100	47	87.8
gr7000	157	88.9	chi1000	118	92.2	chi1000	70	89.4	chi7000	146	**94.2**	su100	7	87.1
ig5000	148	88.7	gr500	90	90.4	ig1000	68	89.4	gr100	93	89.4	ig100	2	86.9
FS	WL	re.	FS	WL	re.	FS	WL	re.	FS	WL	re.	FS	WL	re.
gr100	209	**98.2**	all	167	88.0	su1000	98	87.4	gr500	191	80.9	su500	33	78.8
gr500	187	95.9	chi1000	133	87.3	chi1000	97	87.4	su100	155	74.4	ig1000	32	78.8
all	134	92.7	ig1000	131	87.3	ig1000	92	87.3	ig100	153	74.1	su1000	32	78.8
gr1000	117	92.6	gr1000	131	87.2	su3000	86	87.0	chi100	138	73.6	chi500	32	78.8
su1000	117	92.6	su1000	130	87.3	ig3000	86	86.9	su500	126	72.9	ig500	31	78.8
FS	WL	F_1	FS	WL	F_1	FS	WL	F_1	FS	WL	F_1	FS	WL	F_1
all	162	**89.6**	chi1000	169	**89.7**	su1000	145	88.4	gr500	204	83.6	su500	38	82.4
chi10000	121	89.2	gr1000	166	**89.6**	chi1000	144	88.4	su100	175	79.1	ig500	37	82.4
gr10000	121	89.2	ig1000	164	**89.7**	ig1000	135	88.3	ig100	168	78.7	chi500	35	82.4
ig10000	120	89.2	su1000	163	**89.7**	su500	117	88.0	gr100	163	79.0	su1000	35	82.3
su10000	120	89.2	gr3000	129	89.2	su3000	114	88.0	chi100	156	78.3	ig1000	34	82.3
FS	WL	F_2	FS	WL	F_2	FS	WL	F_2	FS	WL	F_2	FS	WL	F_2
all	171	**91.4**	chi1000	153	88.2	chi1000	138	87.8	gr500	194	81.9	su500	35	80.2
gr1000	157	91.2	all	147	88.2	su1000	122	87.8	su100	162	76.2	ig500	34	80.2
su1000	148	91.1	ig1000	143	88.2	ig1000	120	87.7	ig100	154	75.9	chi500	34	80.2
ig1000	148	91.1	su1000	143	88.2	su500	95	87.3	chi100	140	75.4	ig1000	33	80.2
chi1000	148	91.1	gr1000	141	88.2	su3000	93	87.4	su500	131	74.6	su1000	33	80.2

Table 9.2: Top five feature-selection methods and reduction rates, by accuracy, precision, recall, F_1, and F_2 wins–losses (WL), for each classifier with its "best" document representation

Tabela 9.2: Pet najboljih metoda za odabir atributa, i stepeni redukcije, za pobede–poraze (WL) pri merama tačnost, preciznost, pokrivenost, F_1 i F_2, za svaki klasifikator sa njegovom „najboljom" reprezentacijom dokumenata

grounds as IG and SU. RF proved to be a complete failure, in all probability not because of any shortcoming as a feature-selection algorithm, or unsuitability for use in textual domains. It was a consequence of Weka's implementation using the Euclidean measure when calculating document vector distances for determining nearest neighbors. Therefore, the bad performance of RF may again be treated as an experimental verification of the fact that the Euclidean distance measure is generally not suitable for application to high-dimensional text data sets, this time in the context of feature selection.

It is interesting to note that CNB performs exceptionally well with no feature selection at F_1 and F_2, although the performance with "all" features was not near the best at precision, and was also trailing slightly at recall. This verifies that the F-measures do indeed reward balanced performance.

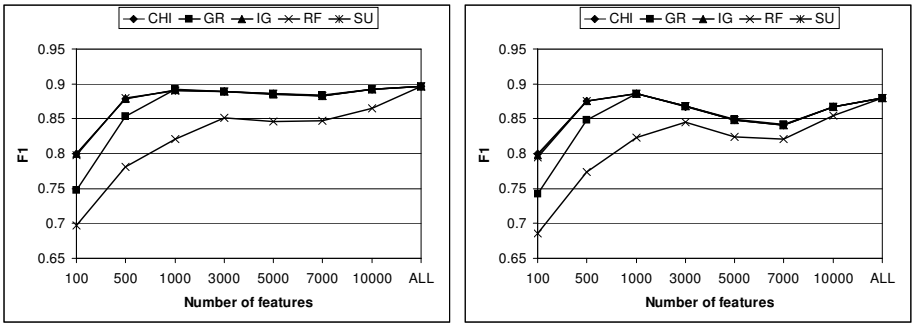
The high recall for CNB at only 100 features can be explained by our later observation that CNB tends to classify very sparse (and empty) vectors into the positive class. However, the fact that recall with all features is ranked very high signifies that sparsity is not the primary cause for CNB's good recall scores, because if it were the performance with all features would not have been among the best.

In summary, considering its good all-round performance, dominance at recall and F_2, the lack of need for feature selection, and superior speed (see Section 8.3.6), we can safely conclude that the CNB classifier, out of those considered, is best suited for the task of classifying search results [163, 161, 167]. SMO can be regarded a close second, because of its inferior performance at F_2, and longer training times.

9.2.2 Interaction Between Bag-of-Words Transformations and Feature Selection

This section will investigate how does the addition of normalization, logtf, and idf transformations to a baseline BOW document representation affect classification performance, from the viewpoint of several feature-selection methods. We will concentrate on two best-performing classifiers determined in the previous section: CNB and SMO. The baseline representation for normalization will be m-tf since stemming was beneficial to classification in all our previous experiments, while the baseline for logtf and idf will be the m-norm-tf representation because normalization is included in the determined best representations for the two classifiers. Since idf provided the motivation for this investigation, transformations will be presented in reverse order compared to Chapter 8: idf, logtf, and then norm.

The idf transformation. Standard-style charts which show the performance of CNB, measured by F_1, for various feature-selection algorithms and reduction rates, are given in Figure 9.2. The measurements are averaged over the six data sets, and the used representations are m-norm-tf in Figure 9.2(a), and m-norm-tfidf in Figure 9.2(b). It can be seen, more clearly than in Section 9.2.1, that CHI, IG, and SU feature-selection methods exhibit almost identical behavior. GR follows the three down to the smaller numbers of features (500 and 100), where its performance decays. Since the described



(a) (b)

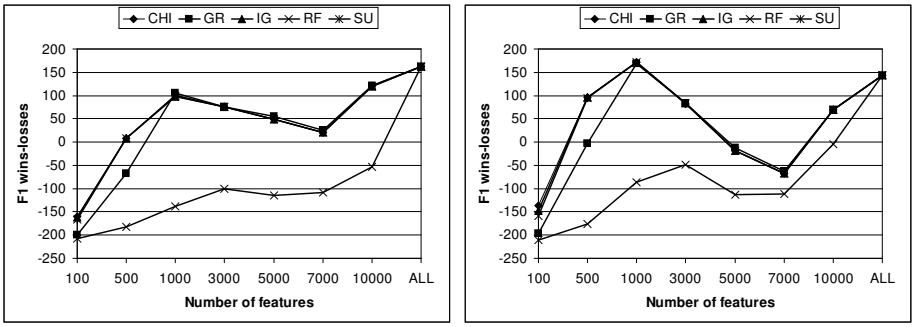
Figure 9.2: Performance of CNB measured by F_1, with tf (a) and tfidf representation (b)
Slika 9.2: Performanse CNB-a izražene merom F_1, sa tf (a) i tfidf reprezentacijom (b)

trends in the performance of FS methods were consistent over all evaluation measures with both classifiers, the following comments will generally refer to the CHI–IG–SU trio, unless otherwise stated.

Noticeable differences in the behavior of CNB in the two charts of Figure 9.2 are the more obvious dent between 3000 and 10000 features for tfidf, and the fact that CNB performance with tfidf is scaled down several percent from tf. But, when instead looking at the summed-up statistically significant wins–losses values, shown in Figure 9.3, the differences between reduction rates become more pronounced.[1] What these charts depict is independent of absolute classifier performance at given reduction rates, but rather express how FS methods and reduction rates fare against one another within the realms of a particular classifier and document representation. Peaks at 1000 selected features and no feature selection are more clearly pronounced than in Figure 9.2, as well as the dents between 3000 and 10000.

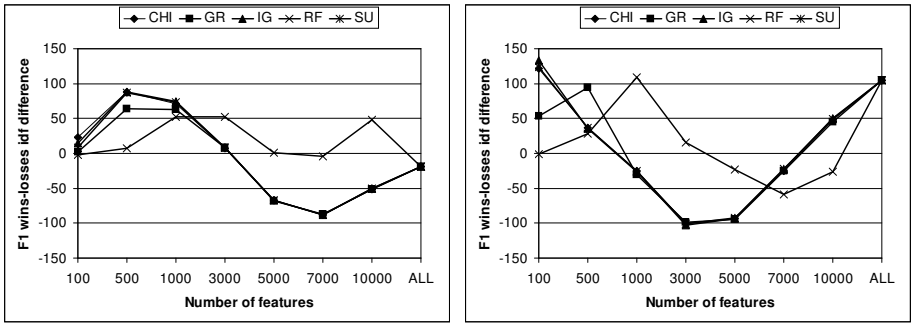
Subtracting the wins–losses of the baseline m-norm-tf representation (Figure 9.3(a)) from the wins–losses of m-norm-tfidf (Figure 9.3(b)), we obtained the chart in Figure 9.4(a). Effectively, it expresses the impact of the idf transformation on the performance of the CNB classifier (as measured by F_1) throughout the range of feature-selection methods and reduction rates. According to the chart, idf improves CNB between 100 and 3000 chosen features, while degrading it for higher feature counts. It should be made clear that the indicated improvement or degradation is relative in nature, since using wins–losses limited the comparisons to the boundaries of the same document representation and classifier. Using wins–losses instead of actual perfor-

[1]The wins–losses axes ranges from -210 to 210: this is six data sets times 35 – the maximum number of wins (and losses) for a FS method at a particular number of features, out of a total of 5 methods $\cdot$ 7 reduction rates $+ 1 = 36$.



(a) (b)

Figure 9.3: Wins–losses for F_1 and CNB, with tf (a) and tfidf representation (b)
Slika 9.3: Pobede–porazi za F_1 i CNB, sa tf (a) i tfidf reprezentacijom (b)

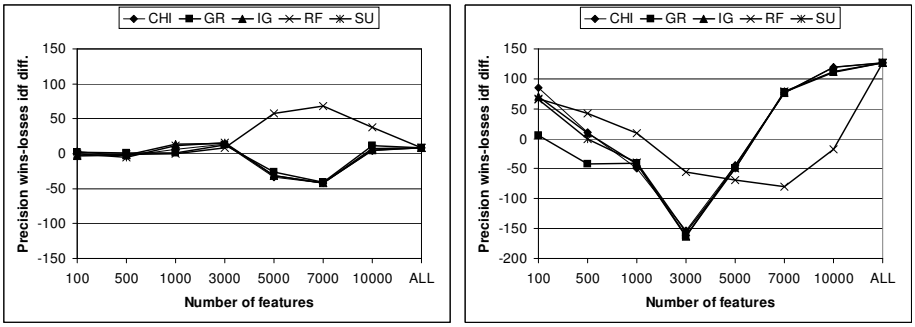


(a) (b)

Figure 9.4: Effect of idf on F_1 wins–losses for CNB (a) and SMO (b)
Slika 9.4: Efekat idf-a na pobede–poraze u meri F_1, za CNB (a) i SMO (b)

mance measurements permitted the subtraction of values by avoiding the issue of scale when comparing the measurements on different document representations. Information about absolute performance was sacrificed in order to express the relationship between the idf transformation and feature selection. Therefore, the 100–3000 range in Figure 9.4(a) can only be viewed as the place to *expect* improvement when introducing the idf transformation to the document representation. Whether there will be actual improvement is determined by the properties of classifiers and data. Our experience showed that tfidf representations are usually inferior for text categorization, which is certainly not a general rule [106, 119].

Figure 9.4(b) shows the corresponding graph of wins–losses differences introduced by idf for the SMO classifier. Contrary to the chart for CNB, this chart points to *two* possible areas of idf performance improvement: one at higher numbers of features,



(a) (b)

Figure 9.5: Effect of idf on *precision* wins–losses for CNB (a) and SMO (b)
Slika 9.5: Efekat idf-a na pobede–poraze u preciznosti, za CNB (a) i SMO (b)

approximately above the 8000 mark, and one in the lower feature counts below 800. This shows that the idf transformation affects the two classifiers differently regarding FS reduction rates, and explains the discrepancy noticed at the beginning of the chapter. With no feature selection idf degrades CNB and improves SMO, while at 2000–3000 selected features[2] the effect is opposite. What makes the correspondence with our previous study from Chapter 8 even more remarkable is the fact that different data sets, and even feature-selection methods were used.

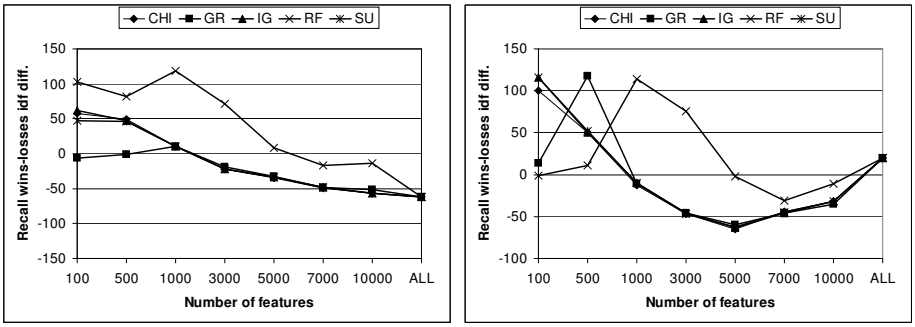
The general shape of the graphs for CNB and SMO in Figure 9.4 (regarding the CHI–IG–SU trio) is quite similar, except for the drop of the CNB graph below 500 features. A corresponding drop for SMO may exist at a lower number of features which was not measured. This indicates that the CNB and SMO do not behave in such opposing fashion with regards to the idf transformation as was suggested, since the graphs are not totally contrary to one another.

However, observing the charts of wins–losses differences introduced by the idf transformation with CNB (Figure 9.5(a)) and SMO (Figure 9.5(b)) using *precision* as the evaluation measure reveals quite different effects of idf on the two classifiers.

Since precision is one of the building blocks of the F_1 measure together with recall, this may suggest that the above correspondence may have been accidental, especially when additionally taking into consideration the charts for recall (Figure 9.6). But, we argue that the correspondence is not accidental, since the wins–losses differences charts for *accuracy*, shown in Figure 9.7, are almost identical to those of F_1 (Figure 9.4).

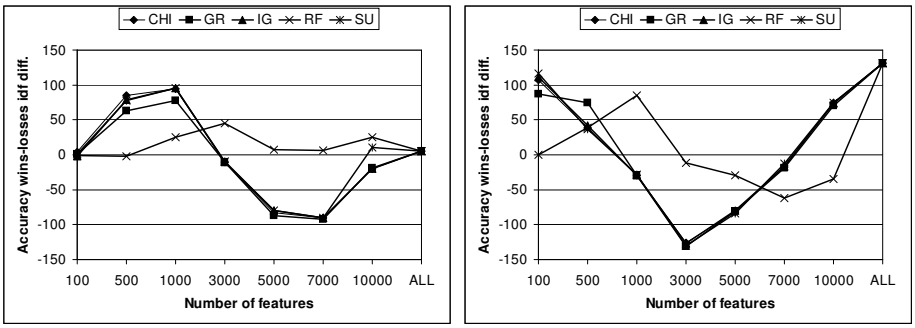
The logtf transformation. The effects of the logtf transformation when measured by F_1, shown in Figure 9.8, are almost a complete contrast to those of idf. Generally, in the ranges where idf caused improvement of performance, logtf causes degradation,

[2]This roughly corresponds to 1000 features from the previous batch of experiments since those data sets had a considerably lower number of features.



(a) (b)

Figure 9.6: Effect of idf on *recall* wins–losses for CNB (a) and SMO (b)
Slika 9.6: Efekat idf-a na pobede–poraze u pokrivenosti, za CNB (a) i SMO (b)



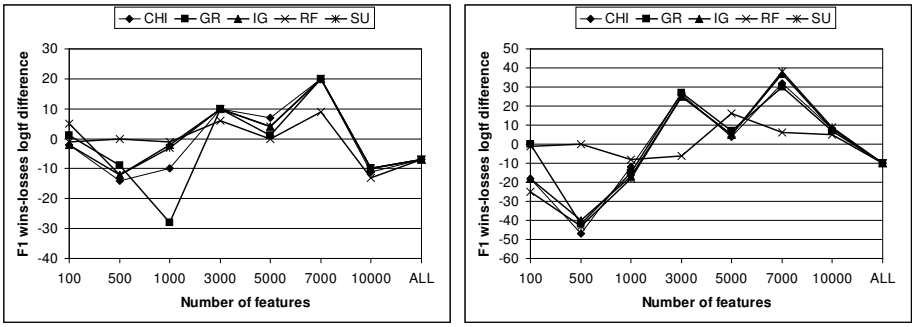
(a) (b)

Figure 9.7: Effect of idf on *accuracy* wins–losses for CNB (a) and SMO (b)
Slika 9.7: Efekat idf-a na pobede–poraze u tačnosti, za CNB (a) i SMO (b)

and vice versa. However, judging by the scale, logtf has a much milder impact than idf, especially on the CNB classifier. Again as with idf, the graphs for the two classifiers exhibit a certain degree of similarity.

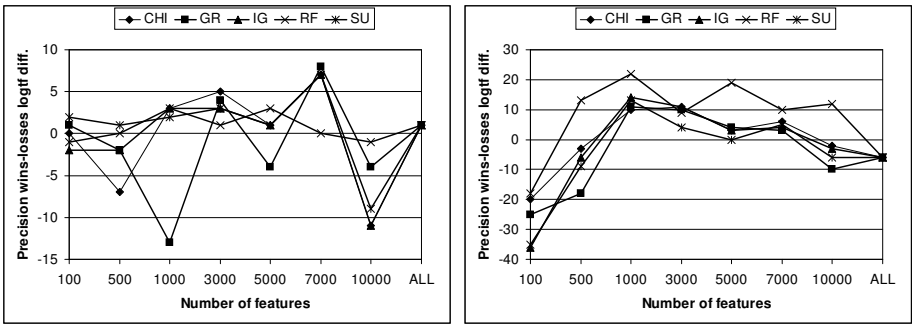
Looking at the charts showing the wins–losses differences with the precision measure it can be seen that logtf has a more clearly pronounced degrading effect on SMO in the feature range below 1000, which accounts for the sharper drop in the same area on the F_1 chart. The scale in Figure 9.9 indicates that the effect of the logtf transformation on CNB is quite weak.

When observing recall in Figure 9.10, no notable differences between the effects of logtf on CNB and SMO are visible, except the generally weaker impact on CNB. Unlike idf, the logtf transformation provided no surprises and differences between its effects on different classifiers. The stronger impact on SMO, similar to the one already



(a) (b)

Figure 9.8: Effect of logtf on F_1 wins–losses for CNB (a) and SMO (b)
Slika 9.8: Efekat logtf-a na pobede–poraze u meri F_1, za CNB (a) i SMO (b)

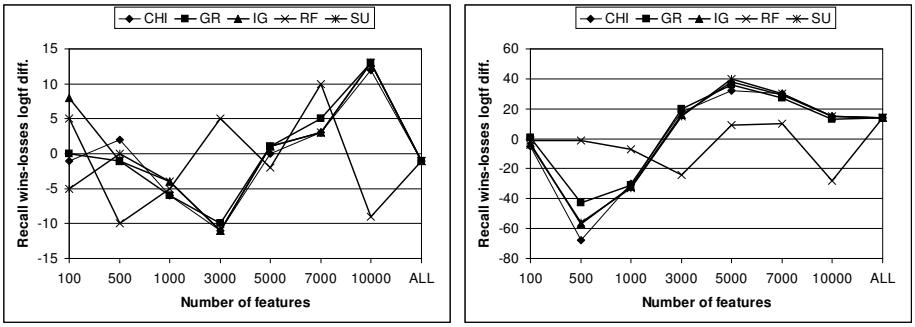


(a) (b)

Figure 9.9: Effect of logtf on *precision* wins–losses for CNB (a) and SMO (b)
Slika 9.9: Efekat logtf-a na pobede–poraze u preciznosti, za CNB (a) i SMO (b)

observed in Chapter 8, can be interpreted by the logarithm's smoothing effect on feature weights, and the scaling down to small numeric values. We find the logtf transformation more predictable and safer to use in the text-categorization setting.

The norm transformation. Figure 9.11 depicts the impact of normalization in the context of CNB and SMO classifiers. Somewhat counterintuitively, the shapes of the graphs are more similar to those of idf (Figure 9.4) than logtf (Figure 9.8). Compared to idf, the effects of normalization are much milder. Also, the behavior of the graphs for CNB and SMO classifiers in Figure 9.11 is quite similar. What is especially notable in the charts is the radical departure of the GR feature-selection method from the behavior of CHI, IG, and SU on feature counts below 3000, in a similar fashion for both classifiers. However, the charts in Figures 9.12 and 9.13 reveal that the improving effect



(a) (b)

Figure 9.10: Effect of logtf on *recall* wins–losses for CNB (a) and SMO (b)
Slika 9.10: Efekat logtf-a na pobede–poraze u pokrivenosti, za CNB (a) i SMO (b)

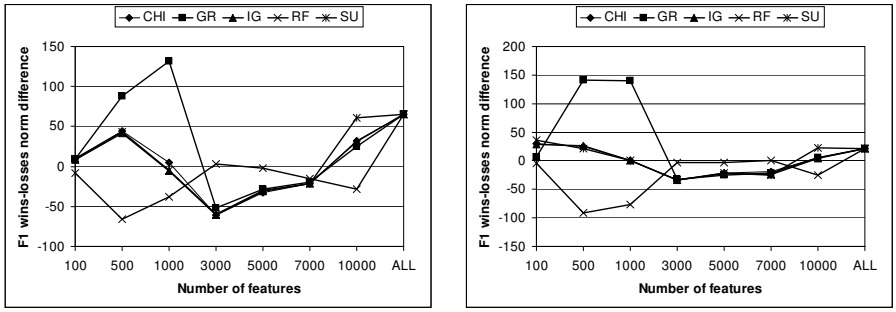
of GR measured by F_1 with the CNB classifier is caused by improvement of precision, while with SMO the main cause is recall. Overall, gain-ratio feature selection exhibited erratic behavior at lower feature counts with all investigated transformations.

9.3 Summary and Future Work

The round of experiments described in this chapter concentrated its first part on determining the best combinations of feature-selection methods, reduction rates, and classifiers on the chosen data sets. Results from Section 9.2.1 made it clear that CNB and SMO are the best performing classifiers, with CNB eventually being selected for the task of classifying search results in the CatS meta-search engine [163, 167] because of its speed, good performance with the F_2 measure, and no need for feature selection.

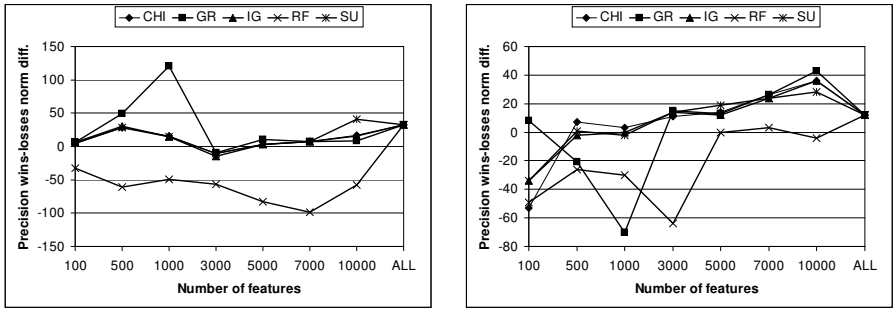
The intuition introduced at the beginning of the chapter, that there may be significant interactions between the idf transformation in the bag-of-words document representation, and feature selection, has been verified by the subsequent study presented in Section 9.2.2. This interaction was quantified in the context of two best-performing classifiers, CNB and SMO, using charts of wins–losses values and their differences. The study concluded that the two examined classifiers behaved in different, but not entirely opposing ways with respect to the interaction between idf and feature selection. Similar treatment was given to two other transformations, logtf and norm, also revealing interesting effects, but less radical and erratic than those of idf.

Another possibility opened by the quantification of interaction between document representation and feature selection is the comparison of the behavior of *data sets*. One interesting direction of future work would certainly be to extend the experiments to corpora more commonly used in TC research, such as Reuters, OHSUMED, WebKB,



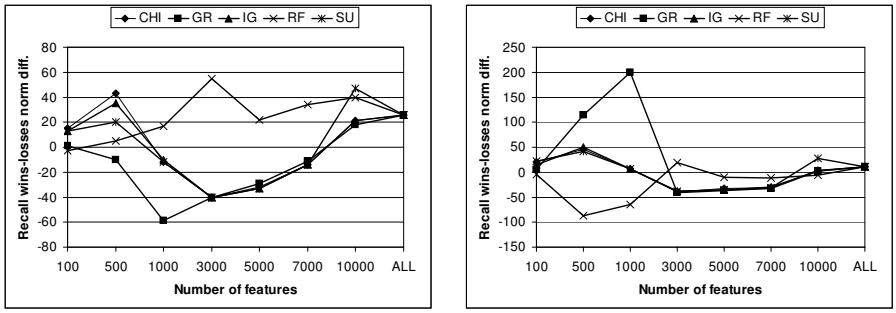
(a) (b)

Figure 9.11: Effect of norm on F_1 wins–losses for CNB (a) and SMO (b)
Slika 9.11: Efekat norm-a na pobede–poraze u meri F_1, za CNB (a) i SMO (b)



(a) (b)

Figure 9.12: Effect of norm on *precision* wins–losses for CNB (a) and SMO (b)
Slika 9.12: Efekat norm-a na pobede–poraze u preciznosti, za CNB (a) i SMO (b)



(a) (b)

Figure 9.13: Effect of norm on *recall* wins–losses for CNB (a) and SMO (b)
Slika 9.13: Efekat norm-a na pobede–poraze u pokrivenosti, za CNB (a) i SMO (b)

or 20-newsgroups, which would enable drawing some additional parallels with existing experimental and theoretical results.

Besides the basic BOW transformations studied in this chapter, it would be interesting to examine the behavior of supervised feature-weighing schemes such as those introduced by Debole and Sebastiani [43], which, unlike idf, do attempt to weigh text features in correspondence to the class feature. Intuitively, such transformations should generate wins–losses difference charts that are much more placid and closer to 0 than the ones in Figure 9.4. Finally, the proposed method for quantification of interaction between document representation and feature selection can be used in the context of more advanced document representation schemes, which may include n-grams [137], information derived from hyperlinks, and elements of document structure [28].

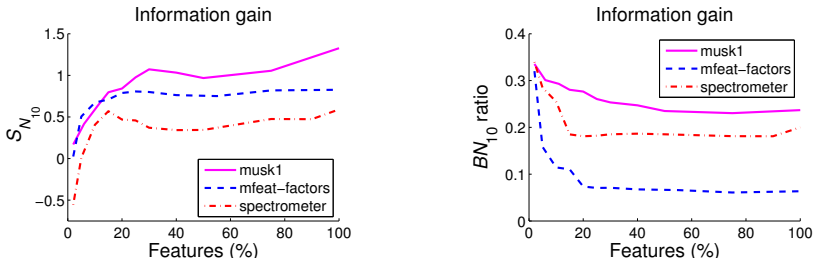
With the rapid expanse of research and application of classification algorithms in the 1990s, the field of document representations was left somewhat under-studied. Recent papers focusing on issues of document representation [119, 202, 225, 106, 43] showed that some of the "old truths" (like, "tfidf is the best variant of the bag-of-words representation") may not always hold for new and improved algorithms. Further understanding of the relationships between document representation, feature-selection methods, and classification algorithms can provide useful insights to both researchers and practitioners, assisting them in choosing the right tools and methods for their classification-related tasks.

9.4 A Note on Hubness in the Context of Feature Selection and Generation

In the research described in Part III of this dissertation, when examining methods for feature selection, we have primarily focused on their direct impact on classification performance and the interaction with term-weighting schemes in the bag-of-words representation of textual documents, without consideration of the hubness phenomenon, which was the predominant topic of Part II. To place hubness into the perspective of feature-selection methods, as well as issues regarding feature *generation* (that is, possible concerns behind the process of adding new features into the data representation), in this section we will briefly discuss their relationships.

We have concentrated our research efforts so far on explaining the origins of hubness and its effects on different tasks, assuming a given set of features defined for a particular data set. Although our work considers the interaction of hubness with dimensionality reduction in the sense of feature *extraction* (Sections 5.5, 6.4, and 7.3), the implications of hubness on feature selection, and especially generation, are still open research questions.

As in Part II, besides hubness, our discussion will consider the notion of the *cluster assumption* which, assuming that data contains labels, roughly states that two points in the same cluster should in most cases be of the same class. This assumption is one of the pillars of semi-supervised machine-learning methods [33], and is also known in



(a) (b)

Figure 9.14: Skewness (a) and "badness" ratio (b) with respect to the percentage of the original number of features selected by information gain
Slika 9.14: Koeficijent asimetrije (a) i „odnos neslaganja" (b) u odnosu na procenat originalnog broja atributa odabranih *information gain*-om

information-retrieval circles as the *cluster hypothesis* [212], which is formulated in an analogous manner using the notion of relevance. The cluster assumption, that is, the degree of its violation, effectively represents the degree to which the featural representation of the data fails to correspond with some notion of "ground truth" about the data given, for example, by class labels. A high degree of cluster assumption violation indicates that models will be difficult to build from the data, and may suggest a reconsideration of the featural representation.

For a given data set and distance measure, let $N_k(\mathbf{x})$ denote the number of times point $\mathbf{x}$ occurs in the k-NN lists of other points in the data set. We express hubness using the skewness of the distribution of $N_k(\mathbf{x})$, as its standardized third moment, denoted S_{N_k} (see Section 5.2). Also, let $BN_k(\mathbf{x})$ be the number of times $\mathbf{x}$ occurs in the k-NN lists of other points, where the labels of $\mathbf{x}$ and the points in question do not match (making $BN_k(\mathbf{x})$ a measure of "badness" of $\mathbf{x}$). The normalized sum of $BN_k(\mathbf{x})$ for a given data set, BN_k ratio, represents one way to express the degree of violation of the cluster assumption, which was demonstrated in Section 5.6.1.

Figure 9.14 illustrates how S_{N_k} and BN_k ratio change when features are selected using the classical information-gain method (see Section 2.8.1), on three data sets from the UCI repository (Table 5.1). Regarding Figure 9.14(a), looking from right to left, skewness of N_k stays relatively constant until a small percentage of the original number of features is left, when it abruptly drops. This is the point where the intrinsic dimensionality is reached, with further selection incurring loss of information. This loss is also visible in Figure 9.14(b), where at similar points there is an increase in BN_k ratio, suggesting that the reduced representation ceases to reflect the information provided by labels very well.

Observing the two charts in the opposite direction, from left to right, offers a glimpse into the benefits and drawbacks of feature *generation*. Adding features that bring new information to the data representation will ultimately increase S_{N_k} and produce

hubs. Furthermore, for the chosen examples, the reduction of BN_k ratio "flattens out" fairly quickly, limiting the usefulness of adding new features in the sense of being able to express the "ground truth." Depending on the application, instead of BN_k ratio some other criterion could have been used in Figure 9.14(b), like classifier error rate, producing similarly shaped curves. While the majority of research in feature selection/generation, including our own work described in Part III of this dissertation, has focused on optimizing criteria reminiscent to those in Figure 9.14(b), little attention has been paid to the fact that in intrinsically high-dimensional data hubness will result in an uneven distribution of the cluster assumption violation (in our case, hubs will generally attract more label mismatches with neighboring points), and with it an uneven distribution of responsibility for classification or retrieval error among data points. We believe that investigating the interaction between hubness and different notions and analogues of cluster assumption violation can result in important new insights relevant to the tasks of feature selection and generation. We plan to address this investigation as a point of future work.

Chapter 10

Conclusion

The research presented in this dissertation studied the properties and effects of high dimensionality in data representations from two angles: (1) the behavior of distance (and similarity) measures with increasing dimensionality of data, and (2) feature-selection methods, primarily through their interaction with high-dimensional document representation schemes for text data. The main results of the dissertation can therefore be summarized as follows.

Regarding the first angle, addressed in Part II of the dissertation, Chapter 3 began the study of the behavior of (dis)similarity measures by exploring the concentration property of cosine similarity, providing theoretical results applicable to a wide range of data distributions. The following chapters discussed the novel phenomenon of hubness, manifested in the tendency of some points in a high-dimensional data set to be included in unexpectedly many k-NN lists of other points. Chapter 4 provided a comprehensive characterization of the phenomenon predominantly for Euclidean distance, explained the mechanisms of hub formation, their location in the data space, and presented theoretical results which provided insight into the behavior of distance measures in high-dimensional data spaces that lead to the emergence of hubness. Next, Chapter 5 generalized the explanations to real data from various application domains, establishing the connection between hubness and the intrinsic dimensionality of data, providing thorough empirical evidence for the proposed observations, linking hubness with various notions of outliers, discussing the interaction of hubness with dimensionality reduction, and demonstrating the impact of the phenomenon on key (distance-based) algorithms belonging to major families of ML techniques: supervised, semi-supervised, and unsupervised.

Chapters 6 and 7 focused their attention on the role of the previously established phenomenon of hubness in the tasks of time-series classification and information retrieval, respectively. Considering the state-of-the-art dynamic time warping (DTW) distance, Chapter 6 confirmed the existence of hubness in time-series data, examined its influence on time-series classification with the k-NN classifier, and presented a

framework for categorizing time-series data sets based on properties of hubness and the distribution of class labels among nearest neighbors, which allows one to assess whether hubness can be used to improve the performance of the k-NN classifier, and provides insight into the mechanisms underlying nearest-neighbor classification of time series. Chapter 7 investigated hubness in the context of the popular vector space model (VSM) of information retrieval, explaining the origins of the phenomenon, with respect to cosine similarity, as a consequence of high intrinsic dimensionality of text data, as opposed to other factors, such as sparsity and skewness of the distribution of term frequencies (caused, for example, by differences in document lengths). Hubness was first considered within the classical VSM based on tfidf term weighting and cosine similarity, and the conclusions generalized to the more advanced variation Okapi BM25 [178]. Results of experiments, involving a similarity adjustment scheme that considers hubness, indicated that the phenomenon can considerably affect the performance of IR by including persistent irrelevant documents in the search result lists.

Regarding the second angle of research into high dimensionality within data representations, addressed in Part III of the dissertation, Chapter 8 experimentally demonstrated that there may be statistically significant differences in classification performance of five major classifiers when using different transformations of the bag-of-words document representation. The chapter also gave a detailed description of the effects of individual transformations on five commonly used performance evaluation measures (accuracy, precision, recall, F_1, and F_2), indicating that the effects on different measures can be quite opposite. Also, relationships between different transformations, when used together in the document representation, were captured, showing some interesting correlations. This was achieved by using wins–losses instead of absolute performance measure values, permitting manipulation such as addition and subtraction which would not otherwise have been possible. Furthermore, it was demonstrated that the simple dimensionality-reduction method that was used can significantly alter these effects and relationships. A feature of both data sets and classifiers, named "robustness," was observed, meaning that some data sets and classifiers showed less sensitivity to changes in document representations than others. Chapter 9 built on the conclusions of Chapter 8 by considering a wider array of feature-selection methods and reduction rates, but using only the document representations that were found most suitable to each classifier. Also, the chapter focused its attention on idf – the transformation that exhibited greatest variation of behavior with regards to feature selection (and not just feature selection) in the previous chapter – in the context of two best performing classifiers. The intuition that there may be significant interactions between the idf transformation and feature selection has been verified, and this interaction was quantified using charts of wins–losses and their differences. Stemming and normalization transformations were also examined using the same methodology, revealing lower degrees of interaction with feature selection than was observed with idf.

Since the evidence presented in this dissertation suggests that hubness is a ubiquitous phenomenon, an important aspect of future work will be identifying hubness within data and methods from various application fields, as well as designing application-

specific methods to mitigate or take advantage of the phenomenon. We already established the existence of hubness in collaborative filtering data with commonly used variants of cosine distance [141], time-series data sets in the context of k-NN classification involving dynamic time warping (DTW) distance (Chapter 6), text data within several variations of the classical vector space model for information retrieval (Chapter 7), and audio data for music information retrieval using spectral similarity measures [100]. In the immediate future we plan to perform a more detailed investigation of hubness in the fields of outlier detection and image mining. Another application area that could directly benefit from an investigation into hubness are reverse k-NN queries.

Possible directions for future work within different aspects of machine learning include a more formal and theoretical study of the interplay between hubness and various distance-based machine-learning models, possibly leading to approaches that account for the phenomenon at a deeper level. Further directions of research may involve determining whether the phenomenon is applicable to probabilistic models, (unboosted) decision trees, and other techniques not explicitly based on distances between points; and also to algorithms that operate within general metric spaces. Since we determined that for K-means clustering of high-dimensional data hubs tend to be close to cluster centers, it would be interesting to explore whether this can be used to improve iterative clustering algorithms, like K-means or self-organizing maps [108]. Nearest-neighbor clustering [22] of high-dimensional data may also directly benefit from hubness information. Topics that could also be worth further study are the interplay of hubness with learned metrics [222] and dimensionality reduction. Finally, it would be interesting to see whether some measure of hubness can be used for estimation of the intrinsic dimensionality of a data set.

In the domain of time-series classification, we plan to expand the set of considered distance measures beyond dynamic time warping, and explore other state-of-the-art distances such as those which exhibited good performance in recent experiments with the 1-NN classifier [47]: longest common subsequence (LCSS) [218], edit distance on real sequence (EDR) [37], and edit distance with real penalty (ERP) [36]. Furthermore, time-series classification by methods other than k-NN may benefit from an investigation into the influence of hubness. Another possible direction for future work on time series is a more detailed exploration of hubness in the context of different representation techniques. Besides the briefly considered DFT, DWT, and SVD methods, a more detailed study could include, for example, piecewise aggregate approximation (PAA) [102], piecewise constant approximation (APCA) [26], and symbolic aggregate approximation (SAX) [127]. Finally, in the time-series domain hubness may be relevant to tasks other than classification. Interesting avenues for future research include assessing the influence of hubness on time-series clustering, indexing, and prediction.

For practical reasons, the research on the impact of hubness on IR focused on data containing category labels. In future work we plan to extend the evaluation to larger data collections where relevance judgements are provided in a non-categorical fashion. Also, we will consider in more detail advanced models like BM25 [178] and pivoted cosine [196]. Finally, in future research we intend to explore other strategies for as-

sessing and mitigating the influence of hubness in IR, besides the proposed similarity adjustment scheme.

Finally, future experiments concerning the interactions between document representations and feature selection can be performed on other corpora, in order to draw more accurate parallels with existing research, and verify that the conclusions are not relevant only for data sets used in the presented research. Robustness regarding document representations, which applies both to classifiers and data sets, is also an interesting area for further theoretical investigations. Giving transformations other than idf the treatment of Chapter 9 may also reveal insightful relationships, especially for transformations derived with supervised learning which have emerged recently.

Appendix A

Term Weighting in the BOW Representation

This appendix will illustrate the bag-of-words (BOW) representation and some of the commonly used term-weighting schemes (also referred to as transformations in this dissertation). We will consider a short hypothetical text data set consisting of five one-sentence documents, as follows.

Document 1: "I do not like apples, and I do not like peaches."
Document 2: "I like clocks, and I like parking meters."
Document 3: "Smurfs are blue, I hate Smurfs!"
Document 4: "Parks are not blue, I like parks."
Document 5: "Peaches are likely not like apples."

First, we will discuss the simple BOW representation where words are taken as terms, without any modification (Section A.1). Then, we will show the representation with stemming applied (Section A.2).

A.1 Term Weighting Without Stemming

After removal of punctuation marks, 16 distinct words can be identified in the document set, making up the dictionary W (all words are given in lowercase): and, apples, are, blue, clocks, do, hate, i, like, likely, meters, not, parking, parks, peaches, smurfs.

Table A.1 shows the basic *binary representation* of the text data set (denoted 01), where rows represent documents, and columns correspond to the words from the dictionary. For document i (denoted d_i), element w_{ij} from the table ($i \in \{1, 2, \ldots, 5\}$, $j \in \{1, 2, \ldots, 16\}$) signifies whether the jth word from the dictionary is present in the ith document (value 1), or not (value 0).

In the *term-frequency representation* (denoted tf), $w_{ij} = tf_{ij}$, the frequency of appearance of the jth word in the ith document. This representation of the example data set is shown in Table A.2.

A common transformation of term weights is *normalization* (denoted norm), which is typically performed by dividing the weights with the Euclidean norm of the appropriate document vector, producing document vectors of unit length in the multidimensional vector space. If we denote the vector of term weights corresponding to document d_i by $\mathbf{w}_i$, the normalized representation of d_i is given by

$$\mathrm{norm}(d_i) = (w_{i1}/\|\mathbf{w}_i\|, \ldots, w_{i|W|}/\|\mathbf{w}_i\|).$$

This particular normalization scheme is often referred to as *cosine normalization*, since computing the cosine of the angle between two vectors normalized in this manner involves only the computation of the dot product. Table A.3 gives the norm representation of the example document set. A variant of this normalization scheme involves additionally multiplying every weight by the average of all document vector norms, that is, the average document length. In our example, the average document length is 3.22.

Taking the logarithm of every term weight (after adding 1 to avoid $\log(0)$) results in the logtf transformation, which produces:

$$\mathrm{logtf}(d_i) = (\log(1 + w_{i1}), \ldots, \log(1 + w_{i|W|})).$$

Table A.4 shows the logtf representation of the example text data set.

The *inverse document frequency* (idf) transformation yields the representation of document d_i:

$$\mathrm{idf}(d_i) = (w_{i1} \log(|D|/\mathrm{docfreq}(D, 1)), \ldots, w_{i|W|} \log(|D|/\mathrm{docfreq}(D, |W|))),$$

where $\mathrm{docfreq}(D, j)$ is the number of documents from D the ith term occurs in. It can be used by itself (multiplied by binary weights w_{ij}), or multiplied with term frequencies to form the popular tfidf representation, illustrated in Table A.5.

A.2 Term Weighting With Stemming

Applying stemming (using the Porter stemmer [158]) transforms the dictionary W obtained from the example document set into: and, appl, ar, blue, clock, do, hate, i, like, meter, not, park, peach, smurf. The new dictionary contains 14 words (that is, word stems) in contrast to the original 16, since words “like” and “likely” were fused to the stem “like,” and “parking” and “parks” to the stem “park.”

Tables A.6–A.10 show all term-weighting schemes demonstrated in Section A.1 in the word space obtained by stemming.

and	apples	are	blue	clocks	do	hate	i	like	likely	meters	not	parking	parks	peaches	smurfs
1	1	0	0	0	1	0	1	1	0	0	1	0	0	1	0
1	0	0	0	1	0	0	1	1	0	1	0	1	0	0	0
0	0	1	1	0	0	1	1	0	0	0	0	0	0	0	1
0	0	1	1	0	0	0	1	1	0	0	1	0	1	0	0
0	1	1	0	0	0	0	0	1	1	0	1	0	0	1	0

Table A.1: Example text data set in the binary (01) representation
Tabela A.1: Primer tekstualnog skupa podataka u binarnoj (01) reprezentaciji

and	apples	are	blue	clocks	do	hate	i	like	likely	meters	not	parking	parks	peaches	smurfs
1	1	0	0	0	2	0	2	2	0	0	2	0	0	1	0
1	0	0	0	1	0	0	2	2	0	1	0	1	0	0	0
0	0	1	1	0	0	1	1	0	0	0	0	0	0	0	2
0	0	1	1	0	0	0	1	1	0	0	1	0	2	0	0
0	1	1	0	0	0	0	0	1	1	0	1	0	0	1	0

Table A.2: Example text data set in the term-frequency (tf) representation
Tabela A.2: Primer tekstualnog skupa podataka reprezentovanog preko frekvencija termova (tf)

and	apples	are	blue	clocks	do	hate	i	like	likely	meters	not	parking	parks	peaches	smurfs
0.23	0.23	0	0	0	0.46	0	0.46	0.46	0	0	0.46	0	0	0.23	0
0.29	0	0	0	0.29	0	0	0.58	0.58	0	0.29	0	0.29	0	0	0
0	0	0.35	0.35	0	0	0.35	0.35	0	0	0	0	0	0	0	0.71
0	0	0.33	0.33	0	0	0	0.33	0.33	0	0	0.33	0	0.67	0	0
0	0.41	0.41	0	0	0	0	0	0.41	0.41	0	0.41	0	0	0.41	0

Table A.3: Example text data set in the normalized term-frequency (norm) representation
Tabela A.3: Primer tekstualnog skupa podataka reprezentovanog preko normalizovanih frekvencija termova (norm)

and	apples	are	blue	clocks	do	hate	i	like	likely	meters	not	parking	parks	peaches	smurfs
0.69	0.69	0	0	0	1.1	0	1.1	1.1	0	0	1.1	0	0	0.69	0
0.69	0	0	0	0.69	0	0	1.1	1.1	0	0.69	0	0.69	0	0	0
0	0	0.69	0.69	0	0	0.69	0.69	0	0	0	0	0	0	0	1.1
0	0	0.69	0.69	0	0	0	0.69	0.69	0	0	0.69	0	1.1	0	0
0	0.69	0.69	0	0	0	0	0	0.69	0.69	0	0.69	0	0	0.69	0

Table A.4: Example text data set in the log term-frequency (logtf) representation
Tabela A.4: Primer tekstualnog skupa podataka reprezentovanog preko logaritma frekvencija termova (logtf)

and	apples	are	blue	clocks	do	hate	i	like	likely	meters	not	parking	parks	peaches	smurfs
0.92	0.92	0	0	0	3.22	0	0.45	0.45	0	0	1.02	0	0	0.92	0
0.92	0	0	0	1.6	0	0	0.45	0.45	0	1.6	0	1.6	0	0	0
0	0	0.51	0.92	0	0	1.6	0.22	0	0	0	0	0	0	0	3.22
0	0	0.51	0.92	0	0	0	0.22	0.22	0	0	0.51	0	3.22	0	0
0	0.92	0.51	0	0	0	0	0	0.22	1.6	0	0.51	0	0	0.92	0

Table A.5: Example text data set in the term frequency – inverse document frequency (tfidf) representation
Tabela A.5: Primer tekstualnog skupa podataka reprezentovanog preko proizvoda frekvencija termova i inverznih frekvencija u dokumentima (tfidf)

and	appl	ar	blue	clock	do	hate	i	like	meter	not	park	peach	smurf
1	1	0	0	0	1	0	1	1	0	1	0	1	0
1	0	0	0	1	0	0	1	1	1	0	1	0	0
0	0	1	1	0	0	1	1	0	0	0	0	0	1
0	0	1	1	0	0	0	1	1	0	1	1	0	0
0	1	1	0	0	0	0	0	1	0	1	0	1	0

Table A.6: Example text data set in the stemmed binary representation (m-01)
Tabela A.6: Primer tekstualnog skupa podataka u stemovanoj binarnoj reprezentaciji (m-01)

and	appl	ar	blue	clock	do	hate	i	like	meter	not	park	peach	smurf
1	1	0	0	0	2	0	2	2	0	2	0	1	0
1	0	0	0	1	0	0	2	2	1	0	1	0	0
0	0	1	1	0	0	1	1	0	0	0	0	0	2
0	0	1	1	0	0	0	1	1	0	1	2	0	0
0	1	1	0	0	0	0	0	2	0	1	0	1	0

Table A.7: Example text data set in the stemmed term-frequency representation (m-tf)
Tabela A.7: Primer tekstualnog skupa podataka reprezentovanog preko frekvencija stemovanih termova (m-tf)

and	appl	ar	blue	clock	do	hate	i	like	meter	not	park	peach	smurf
0.23	0.23	0	0	0	0.46	0	0.46	0.46	0	0.46	0	0.23	0
0.29	0	0	0	0.29	0	0	0.58	0.58	0.29	0	0.29	0	0
0	0	0.35	0.35	0	0	0.35	0.35	0	0	0	0	0	0.71
0	0	0.33	0.33	0	0	0	0.33	0.33	0	0.33	0.67	0	0
0	0.35	0.35	0	0	0	0	0	0.71	0	0.35	0	0.35	0

Table A.8: Example text data set in the stemmed normalized term-frequency representation (m-norm)
Tabela A.8: Primer tekstualnog skupa podataka reprezentovanog preko normalizovanih frekvencija stemovanih termova (m-norm)

and	appl	ar	blue	clock	do	hate	i	like	meter	not	park	peach	smurf
0.69	0.69	0	0	0	1.1	0	1.1	1.1	0	1.1	0	0.69	0
0.69	0	0	0	0.69	0	0	1.1	1.1	0.69	0	0.69	0	0
0	0	0.69	0.69	0	0	0.69	0.69	0	0	0	0	0	1.1
0	0	0.69	0.69	0	0	0	0.69	0.69	0	0.69	1.1	0	0
0	0.69	0.69	0	0	0	0	0	1.1	0	0.69	0	0.69	0

Table A.9: Example text data set in the stemmed log term-frequency representation (m-logtf)
Tabela A.9: Primer tekstualnog skupa podataka reprezentovanog preko logaritma frekvencija stemovanih termova (m-logtf)

and	appl	ar	blue	clock	do	hate	i	like	meter	not	park	peach	smurf
0.92	0.92	0	0	0	3.22	0	0.45	0.45	0	1.02	0	0.92	0
0.92	0	0	0	1.61	0	0	0.45	0.45	1.61	0	0.92	0	0
0	0	0.51	0.92	0	0	1.61	0.22	0	0	0	0	0	3.22
0	0	0.51	0.92	0	0	0	0.22	0.22	0	0.51	1.83	0	0
0	0.92	0.51	0	0	0	0	0	0.45	0	0.51	0	0.92	0

Table A.10: Example text data set in the stemmed term frequency – inverse document frequency representation (m-tfidf)
Tabela A.10: Primer tekstualnog skupa podataka reprezentovanog preko proizvoda frekvencija stemovanih termova i inverznih frekvencija u dokumentima (m-tfidf)

Bibliography

[1] M. Abramowitz and I. A. Stegun, editors. *Handbook of Mathematical Functions With Formulas, Graphs and Mathematical Tables*. National Bureau of Standards, USA, 1964.

[2] C. C. Aggarwal, A. Hinneburg, and D. A. Keim. On the surprising behavior of distance metrics in high dimensional spaces. In *Proceedings of the 8th International Conference on Database Theory (ICDT)*, volume 1973 of *Lecture Notes in Computer Science*, pages 420–434. Springer, 2001.

[3] C. C. Aggarwal and P. S. Yu. Outlier detection for high dimensional data. In *Proceedings of the 27th ACM SIGMOD International Conference on Management of Data*, pages 37–46, 2001.

[4] D. W. Aha, D. Kibler, and M. K. Albert. Instance-based learning algorithms. *Machine Learning*, 6(1):37–66, 1991.

[5] R. Albert and A.-L. Barabási. Statistical mechanics of complex networks. *Reviews of Modern Physics*, 74(1):47–97, 2002.

[6] E. Alpaydin. *Introduction to Machine Learning*. MIT Press, 2nd edition, 2010.

[7] A. Asuncion and D. J. Newman. UCI machine learning repository. `http://www.ics.uci.edu/~mlearn/MLRepository.html`, 2010. University of California, Irvine, School of Information and Computer Sciences.

[8] G. Attardi, A. Gullí, and F. Sebastiani. Automatic Web page categorization by link and context analysis. In *Proceedings of the 1st European Symposium on Telematics, Hypermedia and Artificial Intelligence (THAI)*, pages 105–119, 1999.

[9] J.-J. Aucouturier and F. Pachet. A scale-free distribution of false positives for a large class of audio similarity measures. *Pattern Recognition*, 41(1):272–284, 2007.

[10] R. E. Bellman. *Adaptive Control Processes: A Guided Tour*. Princeton University Press, 1961.

[11] V. R. Benjamins, J. Contreras, O. Corcho, and A. Gómez-Pérez. Six challenges for the Semantic Web. In *Proceedings of the KR-2002 Semantic Web Workshop*, Toulouse, France, 2002.

[12] A. Berenzweig. *Anchors and Hubs in Audio-based Music Similarity*. PhD thesis, Columbia University, New York, USA, 2007.

[13] P. Berkhin. Survey of clustering data mining techniques. In J. Kogan and C. N. M. Teboulle, editors, *Grouping Multidimensional Data – Recent Advances in Clustering*, pages 25–71. Springer, 2006.

[14] D. J. Berndt and J. Clifford. Using dynamic time warping to find patterns in time series. In *Proceedings of the AAAI Workshop on Knowledge Discovery in Databases (KDD)*, pages 359–370, 1994.

[15] M. Berthold and D. J. Hand, editors. *Intelligent Data Analysis*. Springer, 2nd edition, 2003.

[16] K. S. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft. When is "nearest neighbor" meaningful? In *Proceedings of the 7th International Conference on Database Theory (ICDT)*, volume 1540 of *Lecture Notes in Computer Science*, pages 217–235. Springer, 1999.

[17] C. M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1996.

[18] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[19] R. R. Bouckaert. Choosing between two learning algorithms based on calibrated tests. In *Proceedings of the 20th International Conference on Machine Learning (ICML)*, pages 51–58, 2003.

[20] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. Chapman & Hall, 1984.

[21] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander. LOF: Identifying density-based local outliers. *ACM SIGMOD Record*, 29(2):93–104, 2000.

[22] S. Bubeck and U. von Luxburg. Nearest neighbor clustering: A baseline method for consistent clustering with arbitrary objective functions. *Journal of Machine Learning Research*, 10:657–698, 2009.

[23] R. Caruana, N. Karampatziakis, and A. Yessenalina. An empirical evaluation of supervised learning in high dimensions. In *Proceedings of the 25th International Conference on Machine Learning (ICML)*, pages 96–103, 2008.

[24] G. Casella and R. L. Berger. *Statistical Inference*. Duxbury, 2nd edition, 2002.

[25] W. B. Cavnar and J. M. Trenkle. N-gram-based text categorization. In *Proceedings of the 3rd Annual Symposium on Document Analysis and Information Retrieval (SDAIR)*, pages 161–175, 1994.

[26] K. Chakrabarti, E. Keogh, S. Mehrotra, and M. Pazzani. Locally adaptive dimensionality reduction for indexing large time series databases. *ACM Transactions on Database Systems*, 27(2):188–228, 2002.

[27] S. Chakrabarti. Data mining for hypertext: A tutorial survey. *ACM SIGKDD Explorations*, 1(2):1–11, 2000.

[28] S. Chakrabarti. *Mining the Web: Discovering Knowledge from Hypertext Data*. Morgan Kaufmann Publishers, 2003.

[29] S. Chakrabarti, B. E. Dom, and P. Indyk. Enhanced hypertext categorization using hyperlinks. In *Proceedings of the 24th ACM SIGMOD International Conference on Management of Data*, pages 307–318, 1998.

[30] S. Chakrabarti, S. Roy, and M. Soundalgekar. Fast and accurate text classification via multiple linear discriminant projections. In *Proceedings of the 28th International Conference on Very Large Data Bases (VLDB)*, pages 658–669, 2002.

[31] K.-P. Chan and W.-C. Fu. Efficient time series matching by wavelets. In *Proceedings of the 15th IEEE International Conference on Data Engineering (ICDE)*, pages 126–133, 1999.

[32] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. *ACM Computing Surveys*, 41(3):15, 2009.

[33] O. Chapelle, B. Schölkopf, and A. Zien, editors. *Semi-Supervised Learning*. MIT Press, 2006.

[34] S. Chapman. String similarity metrics for information integration. `http://www.dcs.shef.ac.uk/~sam/stringmetrics.html`, 2010.

[35] J. Chen, H. ren Fang, and Y. Saad. Fast approximate kNN graph construction for high dimensional data via recursive Lanczos bisection. *Journal of Machine Learning Research*, 10:1989–2012, 2009.

[36] L. Chen and R. Ng. On the marriage of Lp-norms and edit distance. In *Proceedings of the 30th International Conference on Very Large Data Bases (VLDB)*, pages 792–803, 2004.

[37] L. Chen, M. T. Özsu, and V. Oria. Robust and fast similarity search for moving object trajectories. In *Proceedings of the 31st ACM SIGMOD International Conference on Management of Data*, pages 491–502, 2005.

[38] P. R. Christopher D. Manning and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[39] W. W. Cohen. Learning to classify English text with ILP methods. In L. De Raedt, editor, *Advances in Inductive Logic Programming*, pages 124–143. IOS Press, 1995.

[40] W. W. Cohen, P. D. Ravikumar, and S. E. Fienberg. A comparison of string distance metrics for name-matching tasks. In *Proceedings of IJCAI Workshop on Information Integration on the Web (IIWeb)*, pages 73–78, 2003.

[41] W. W. Cohen and Y. Singer. Context-sensitive learning methods for text categorization. *ACM Transactions on Information Systems*, 17(2):141–173, 1999.

[42] M. Craven, D. DiPasquo, D. Freitag, A. McCallum, T. Mitchell, K. Nigam, and S. Slattery. Learning to extract symbolic knowledge from the World Wide Web. In *Proceedings of the 15th National Conference on Artificial Intelligence (AAAI)*, pages 509–516, 1998.

[43] F. Debole and F. Sebastiani. Supervised term weighting for automated text categorization. In S. Sirmakessis, editor, *Text Mining and its Applications*, volume 138 of *Studies in Fuzziness and Soft Computing*, pages 81–98. Springer, 2004.

[44] P. Demartines. *Analyse de Données par Réseaux de Neurones Auto-Organisés*. PhD thesis, Institut national polytechnique de Grenoble, Grenoble, France, 1994.

[45] I. S. Dhillon. Co-clustering documents and words using bipartite spectral graph partitioning. In *Proceedings of 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 269–274, 2001.

[46] I. S. Dhillon, S. Mallela, and R. Kumar. Enhanced word clustering for hierarchical text classification. In *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 191–200, 2002.

[47] H. Ding, G. Trajcevski, P. Scheuermann, X. Wang, and E. J. Keogh. Querying and mining of time series data: Experimental comparison of representations and distance measures. In *Proceedings of the 34th International Conference on Very Large Data Bases (VLDB)*, pages 1542–1552, 2008.

[48] D. Djorić, V. Jevremović, J. Mališić, and E. Nikolić-Djorić. *Atlas of Distributions*. Faculty of Civil Engineering, University of Belgrade, Belgrade, Serbia, 2007. In Serbian.

[49] G. Doddington, W. Liggett, A. Martin, M. Przybocki, and D. Reynolds. SHEEP, GOATS, LAMBS and WOLVES: A statistical analysis of speaker performance in the NIST 1998 speaker recognition evaluation. In *Proceedings of the 5th International Conference on Spoken Language Processing (ICSLP)*, 1998. Paper 0608.

[50] R. O. Duda, P. E. Hart, and D. G. Storck. *Pattern Classification*. Wiley, 2nd edition, 2001.

[51] R. J. Durrant and A. Kabán. When is 'nearest neighbour' meaningful: A converse theorem and implications. *Journal of Complexity*, 25(4):385–397, 2009.

[52] D. Eads, D. Hill, S. Davis, S. Perkins, J. Ma, R. Porter, and J. Theiler. Genetic algorithms and support vector machines for time series classification. In *Proceedings of the International Society for Optical Engineering (SPIE)*, volume 4787, pages 74–85, 2002.

[53] L. Egghe. New relations between similarity measures for vectors based on vector norms. *Journal of the American Society for Information Science and Technology*, 60(2):232–239, 2009.

[54] P. Erdős and A. Rényi. On random graphs. *Publicationes Mathematicae Debrecen*, 6:290–297, 1959.

[55] P. F. Evangelista, M. J. Embrechts, and B. K. Szymanski. Taming the curse of dimensionality in kernels and novelty detection. In A. Abraham, B. Baets, M. Koppen, and B. Nickolay, editors, *Applied Soft Computing Technologies: The Challenge of Complexity*, volume 34 of *Advances in Soft Computing*, pages 425–438. Springer, 2006.

[56] C. Faloutsos, M. Ranganathan, and Y. Manolopoulos. Fast subsequence matching in time-series databases. In *Proceedings of the 20th ACM SIGMOD International Conference on Management of Data*, pages 419–429, 1994.

[57] G. Forman. An extensive empirical study of feature selection metrics for text classification. *Journal of Machine Learning Research*, 3:1289–1305, 2003.

[58] G. Forman. BNS feature scaling: An improved representation over TF-IDF for SVM text classification. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management (CIKM)*, pages 263–270, 2008.

[59] D. François. *High-dimensional Data Analysis: Optimal Metrics and Feature Selection*. PhD thesis, Université catholique de Louvain, Louvain, Belgium, 2007.

[60] D. François, V. Wertz, and M. Verleysen. Non-euclidean metrics for similarity search in noisy datasets. In *Proceedings of the 13th European Symposium on Artificial Neural Networks (ESANN)*, pages 339–344, 2005.

[61] D. François, V. Wertz, and M. Verleysen. The concentration of fractional distances. *IEEE Transactions on Knowledge and Data Engineering*, 19(7):873–886, 2007.

[62] Y. Freund and R. E. Schapire. Experiments with a new boosting algorithm. In *Proceedings of the 13th International Conference on Machine Learning (ICML)*, pages 148–156, 1996.

[63] Y. Freund and R. E. Schapire. Large margin classification using the perceptron algorithm. *Machine Learning*, 37(3):277–296, 1999.

[64] J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: A statistical view of boosting. *The Annals of Statistics*, 28(2):337–374, 2000.

[65] A. W.-C. Fu, E. J. Keogh, L. Y. H. Lau, C. A. Ratanamahatana, and R. C.-W. Wong. Scaling and time warping in time series querying. *VLDB Journal*, 17(4):899–921, 2008.

[66] Y. Fujikoshi. Computable error bounds for asymptotic expansions of the hypergeometric function ${}_1F_1$ of matrix argument and their applications. *Hiroshima Mathematical Journal*, 37(1):13–23, 2007.

[67] J. Fürnkranz. Round robin classification. *Journal of Machine Learning Research*, 2:721–747, 2002.

[68] E. Gabrilovich and S. Markovitch. Text categorization with many redundant features: Using aggressive feature selection to make SVMs competitive with C4.5. In *Proceedings of the 21st International Conference on Machine Learning (ICML)*, pages 321–328, 2004.

[69] G. Gan, C. Ma, and J. Wu. *Data Clustering: Theory, Algorithms, and Applications*. ASA-SIAM Series on Statistics and Applied Probability. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA; American Statistical Association (ASA), Alexandria, VA, USA, 2007.

[70] H. Gävert, J. Hurri, J. Särelä, and A. Hyvärinen. The FastICA package for Matlab. `http://www.cis.hut.fi/projects/ica/fastica/`, 2005.

[71] X. Geng, D.-C. Zhan, and Z.-H. Zhou. Supervised nonlinear dimensionality reduction for visualization and classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 35(6):1098–1107, 2005.

[72] D. Gleich. MatlabBGL: A Matlab graph library. `http://www.stanford.edu/~dgleich/programs/matlab_bgl/`, 2008.

[73] L. A. Goodman. On the exact variance of products. *J. Am. Stat. Assoc.*, 55(292):708–713, 1960.

[74] D. A. Grossman and O. Frieder. *Information Retrieval: Algorithms and Heuristics*. Springer, 2nd edition, 2004.

[75] L. Gupta, D. L. Molfese, R. Tammana, and P. Simos. Nonlinear alignment and averaging for estimating the evoked potential. *IEEE Transactions on Biomedical Engineering*, 43(4):348–356, 1996.

[76] U. Haagerup. The best constants in the Khintchine inequality. *Studia Mathematica*, 70:231–283, 1982.

[77] M. A. Hall and G. Holmes. Benchmarking attribute selection techniques for discrete class data mining. *IEEE Transactions on Knowledge and Data Engineering*, 15(3):1437–1447, 2003.

[78] M. A. Hall and L. A. Smith. Practical feature subset selection for machine learning. In *Proceedings of the 21st Australasian Computer Science Conference*, pages 181–191, 1998.

[79] E.-H. Han and G. Karypis. Centroid-based document classification: Analysis and experimental results. In *Proceedings of the 4th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD)*, volume 1910 of *Lecture Notes in Artificial Intelligence*, pages 424–431. Springer, 2000.

[80] E.-H. Han, G. Karypis, and V. Kumar. Text categorization using weight-adjusted k-nearest neighbor classification. In *Proceedings of the 5th Pacific-Asia Conferenece on Knowledge Discovery and Data Mining (PAKDD)*, volume 2035 of *Lecture Notes in Artificial Intelligence*, pages 53–65. Springer, 2001.

[81] J. Han and M. Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers, 2006.

[82] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2nd edition, 2009.

[83] W. Hersh, C. Buckley, T. J. Leone, and D. Hickam. OHSUMED: An interactive retrieval evaluation and new large test collection for research. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 192–201, 1994.

[84] A. Hicklin, C. Watson, and B. Ulery. The myth of goats: How many people have fingerprints that are hard to match? Internal Report 7271, National Institute of Standards and Technology (NIST), USA, 2005.

[85] A. Hinneburg, C. C. Aggarwal, and D. A. Keim. What is the nearest neighbor in high dimensional spaces? In *Proceedings of the 26th International Conference on Very Large Data Bases (VLDB)*, pages 506–515, 2000.

[86] G. Hinton and S. Roweis. Stochastic neighbor embedding. In *Advances in Neural Information Processing Systems 15*, pages 833–840, 2003.

[87] C.-M. Hsu and M.-S. Chen. On the design and applicability of distance functions in high-dimensional data space. *IEEE Transactions on Knowledge and Data Engineering*, 21(4):523–536, 2009.

[88] A. Hyvärinen and E. Oja. Independent component analysis: Algorithms and applications. *Neural Networks*, 13(4–5):411–430, 2000.

[89] K. T.-U. Islam, K. Hasan, Y.-K. Lee, and S. Lee. Enhanced 1-NN time series classification using badness of records. In *Proceedings of the 2nd International Conference on Ubiquitous Information Management and Communication*, pages 108–113, 2008.

[90] K. Itô, editor. *Encyclopedic Dictionary of Mathematics*. MIT Press, 2nd edition, 1993.

[91] M. Iwayama and T. Tokunaga. Hierarchical Bayesian clustering for automatic text classification. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1322–1327, 1995.

[92] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: A review. *ACM Computing Surveys*, 31(3):264–323, 1999.

[93] T. Jebara, J. Wang, and S.-F. Chang. Graph construction and b-matching for semi-supervised learning. In *Proceedings of the 26th International Conference on Machine Learning (ICML)*, pages 441–448, 2009.

[94] F. V. Jensen. *Bayesian Networks and Decision Graphs*. Springer, 2nd edition, 2007.

[95] T. Joachims. A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization. In *Proceedings of the 14th International Conference on Machine Learning (ICML)*, pages 143–151, 1997.

[96] T. Joachims. Text categorization with support vector machines: Learning with many relevant features. In *Proceedings of the 10th European Conference on Machine Learning (ECML)*, volume 1398 of *Lecture Notes in Artificial Intelligence*, pages 137–142. Springer, 1998.

[97] T. Joachims. Making large-scale SVM learning practical. In B. Schölkopf, C. J. C. Burges, and A. J. Smola, editors, *Advances in Kernel Methods – Support Vector Learning*, pages 169–184. MIT Press, 1999.

[98] N. L. Johnson, S. Kotz, and N. Balakrishnan. *Continuous Univariate Distributions*, volume 1. Wiley, 2nd edition, 1994.

[99] I. T. Jolliffe. *Principal Component Analysis*. Springer, 2nd edition, 2002.

[100] I. Karydis, M. Radovanović, A. Nanopoulos, and M. Ivanović. Looking through the "glass ceiling": A conceptual framework for the problems of spectral similarity. In *Proceedings of the 11th International Society for Music Information Retrieval Conference (ISMIR)*, 2010.

[101] S. S. Keerthi, S. K. Shevade, C. Bhattacharyya, and K. R. K. Murthy. Improvements to Platt's SMO algorithm for SVM classifier design. *Neural Computation*, 13(3):637–649, 2001.

[102] E. Keogh, K. Chakrabarti, M. Pazzani, and S. Mehrotra. Dimensionality reduction for fast similarity search in large time series databases. *Knowledge and Information Systems*, 3(3):263–286, 2001.

[103] E. Keogh, C. Shelton, and F. Moerchen. Workshop and challenge on time series classification. International Conference on Knowledge Discovery and Data Mining (KDD), 2007.
`http://www.cs.ucr.edu/~eamonn/SIGKDD2007TimeSeries.html`.

[104] E. Keogh, X. Xi, L. Wei, and C. A. Ratanamahatana. The UCR time series classification/clustering homepage, 2006.
`http://www.cs.ucr.edu/~eamonn/time_series_data/`.

[105] A. M. Kibriya and E. Frank. An empirical comparison of exact nearest neighbour algorithms. In *Proceedings of the 11th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD)*, volume 4702 of *Lecture Notes in Artificial Intelligence*, pages 140–151. Springer, 2007.

[106] A. M. Kibriya, E. Frank, B. Pfahringer, and G. Holmes. Multinomial naive Bayes for text categorization revisited. In *Proceedings of the 17th Australian Joint Conference on Artificial Intelligence (AI)*, volume 3339 of *Lecture Notes in Artificial Intelligence*, pages 488–499. Springer, 2004.

[107] K. Kira and L. Rendell. A practical approach to feature selection. In *Proceedings of the 9th International Conference on Machine Learning (ICML)*, pages 249–256, 1992.

[108] T. Kohonen. *Self-Organizing Maps*. Springer, 3rd edition, 2001.

[109] T. Kohonen, S. Kaski, K. Lagus, J. Salojärvi, V. Paatero, and A. Saarela. Self organization of a massive document collection. *IEEE Transactions on Neural Networks*, 11(3):574–585, 2000.

[110] I. Kononenko. Estimating attributes: Analysis and extensions of RELIEF. In *Proceedings of the 7th European Conference on Machine Learning (ECML)*, volume 1224 of *Lecture Notes in Artificial Intelligence*, pages 412–420. Springer, 1997.

[111] F. Korn, B.-U. Pagel, and C. Faloutsos. On the "dimensionality curse" and the "self-similariy blessing". *IEEE Transactions on Knowledge and Data Engineering*, 13(1):96–111, 2001.

[112] R. Kosala and H. Blockeel. Web mining research: A survey. *SIGKDD Explorations Newsletter*, 2(1):1–15, 2000.

[113] C. A. Kumar. Analysis of unsupervised dimensionality reduction techniques. *Computer Science and Information Systems*, 6(2):217–227, 2009.

[114] V. Kurbalija, M. Ivanović, and Z. Budimac. Case-based curve behaviour prediction. *Software: Practice and Experience*, 39(1):81–103, 2009.

[115] S. Lafon and A. B. Lee. Diffusion maps and coarse-graining: A unified framework for dimensionality reduction, graph partitioning, and data set parameterization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(9):1393–1403, 2006.

[116] M. Lan, C.-L. Tan, H.-B. Low, and S.-Y. Sung. A comprehensive comparative study on term weighting schemes for text categorization with support vector machines. In *Proceedings of the 14th International World Wide Web Conference (WWW)*, pages 1032–1033, 2005.

[117] J. A. Lee and M. Verleysen. *Nonlinear Dimensionality Reduction*. Springer, 2007.

[118] P. Legendre and L. Legendre. *Numerical Ecology*. Elsevier, 2nd edition, 1998.

[119] E. Leopold and J. Kindermann. Text categorization with support vector machines. How to represent texts in input space? *Machine Learning*, 46(1–3):423–444, 2002.

[120] E. Levina and P. J. Bickel. Maximum likelihood estimation of intrinsic dimension. In *Advances in Neural Information Processing Systems 17*, pages 777–784, 2005.

[121] D. D. Lewis. An evaluation of phrasal and clustered representations on a text categorization task. In *Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 37–50, 1992.

[122] D. D. Lewis. Naive (Bayes) at forty: The independence assumption in information retrieval. In *Proceedings of the 10th European Conference on Machine Learning (ECML)*, volume 1398 of *Lecture Notes in Artificial Intelligence*, pages 4–15. Springer, 1998.

[123] D. D. Lewis, Y. Yang, T. G. Rose, and F. Li. Rcv1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research*, 5:361–397, 2004.

[124] J. Li, H. Liu, and L. Wong. Mean-entropy discretized features are effective for classifying high-dimensional biomedical data. In *Proceedings of the 3rd ACM SIGKDD Workshop on Data Mining in Bioinformatics*, pages 17–24, 2003.

[125] L. Li, D. Alderson, J. C. Doyle, and W. Willinger. Towards a theory of scale-free graphs: Definition, properties, and implications. *Internet Mathematics*, 2(4):431–523, 2005.

[126] T. W. Liao. Clustering of time series data—a survey. *Pattern Recognition*, 38(11):1857–1874, 2005.

[127] J. Lin, E. Keogh, L. Wei, and S. Lonardi. Experiencing SAX: A novel symbolic representation of time series. *Data Mining and Knowledge Discovery*, 15(2):107–144, 2007.

[128] B. Liu. *Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data*. Springer, 2007.

[129] T. Liu, S. Liu, Z. Chen, and W.-Y. Ma. An evaluation on feature selection for text clustering. In *Proceedings of the 20th International Conference on Machine Learning (ICML)*, pages 488–495, 2003.

[130] H. Lodhi, C. Saunders, J. Shawe-Taylor, N. Cristianini, and C. Watkins. Text classification using string kernels. *Journal of Machine Learning Research*, 2:419–444, 2002.

[131] L. T. Maloney. Nearest neighbor analysis of point processes: Simulations and evaluations. *Journal of Mathematical Psychology*, 27(3):251–260, 1983.

[132] C. D. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, 1999.

[133] J. Matthews. The perceptron. `http://www.generation5.org/content/1999/perceptron.asp`, 2010.

[134] A. McCallum and K. Nigam. A comparison of event models for naive Bayes text classification. In *Proceedings of the AAAI Workshop on Learning for Text Categorization*, pages 41–48, 1998.

[135] M. Meilă and J. Shi. Learning segmentation by random walks. In *Advances in Neural Information Processing Systems 13*, pages 873–879, 2001.

[136] T. M. Mitchell. *Machine Learning*. McGraw-Hill, 1997.

[137] D. Mladenić. *Machine Learning on non-homogenous, distributed text data*. PhD thesis, University of Ljubljana, Ljubljana, Slovenia, 1998.

[138] D. Mladenić. Text-learning and related intelligent agents. *IEEE Intelligent Systems, Special Issue on Applications of Intelligent Information Retrieval*, 14(4):44–54, 1999.

[139] C. Nadeau and Y. Bengio. Inference for the generalization error. *Machine Learning*, 52(3):239–281, 2003.

[140] B. Nadler, S. Lafon, R. R. Coifman, and I. G. Kevrekidis. Diffusion maps, spectral clustering and reaction coordinates of dynamical systems. *Applied and Computational Harmonic Analysis*, 21(1):113–127, 2006.

[141] A. Nanopoulos, M. Radovanović, and M. Ivanović. How does high dimensionality affect collaborative filtering? In *Proceedings of the 3rd ACM Conference on Recommender Systems (RecSys)*, pages 293–296, 2009.

[142] R. E. Neapolitan. *Learning Bayesian Networks*. Prentice Hall, 2003.

[143] C. M. Newman and Y. Rinott. Nearest neighbors and Voronoi volumes in high-dimensional point processes with various distance functions. *Advances in Applied Probability*, 17(4):794–809, 1985.

[144] C. M. Newman, Y. Rinott, and A. Tversky. Nearest neighbors and Voronoi regions in certain point processes. *Advances in Applied Probability*, 15(4):726–751, 1983.

[145] A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems 14*, pages 849–856, 2002.

[146] V. Niennattrakul and C. A. Ratanamahatana. Inaccuracies of shape averaging method using dynamic time warping for time series data. In *Proceedings of the 7th International Conference on Computational Science (ICCS)*, volume 4487 of *Lecture Notes in Computer Science*, pages 513–520, 2007.

[147] V. Niennattrakul and C. A. Ratanamahatana. Shape averaging under time warping. In *Proceedings of the 6th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications, and Information Technology (ECTI-CON)*, 2009.

[148] D. M. Nunzio. A bidimensional view of documents for text categorization. In *Proceedings of the 26th European Conference on IR Research (ECIR)*, volume 2997 of *Lecture Notes in Computer Science*, pages 112–126. Springer, 2004.

[149] L. Oberto and F. Pennecchi. Estimation of the modulus of a complex-valued quantity. *Metrologia*, 43(6):531–538, 2006.

[150] A. M. Odlyzko and N. J. A. Sloane. New bounds on the number of unit spheres that can touch a unit sphere in n dimensions. *Journal of Combinatorial Theory, Series A*, 26(2):210–214, 1979.

[151] E. Osuna, R. Freund, and F. Girosi. An improved training algorithm for support vector machines. In *Proceedings of the 7th IEEE Neural Networks for Signal Processing Workshop*, pages 276–285, 1997.

[152] B. Pang and L. Lee. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 271–278, 2004.

[153] V. Pavlovic, B. J. Frey, and T. S. Huang. Time-series classification using mixed-state dynamic Bayesian networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 2609–2615, 1999.

[154] F. Peng and D. Schuurmans. Combining naive Bayes and n-gram language models for text classification. In *Proceedings of the 25th European Conference on IR Research (ECIR)*, volume 2633 of *Lecture Notes in Computer Science*, pages 335–350. Springer, 2003.

[155] M. Penrose. *Random Geometric Graphs*. Oxford University Press, 2003.

[156] V. Petridis and A. Kehagias. Predictive modular neural networks for time series classification. *Neural Networks*, 10(1):31–49, 1997.

[157] J. C. Platt. Fast training of support vector machines using sequential minimal optimization. In B. Schölkopf, C. J. C. Burges, and A. J. Smola, editors, *Advances in Kernel Methods – Support Vector Learning*, pages 185–208. MIT Press, 1999.

[158] M. F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.

[159] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, 1993.

[160] R. Quinlan. Learning logical definitions from relations. *Machine Learning*, 5:239–266, 1990.

[161] M. Radovanović. Machine learning in Web mining. Master's thesis, Department of Mathematics and Informatics, University of Novi Sad, Novi Sad, Serbia, 2006.

[162] M. Radovanović and M. Ivanović. Search based on ontologies. In *Proceedings of the 16th Conference on Applied Mathematics (PRIM)*, pages 255–266, Budva, Serbia and Montenegro, 2004.

[163] M. Radovanović and M. Ivanović. CatS: A classification-powered meta-search engine. In M. Last, P. S. Szczepaniak, Z. Volkovich, and A. Kandel, editors, *Advances in Web Intelligence and Data Mining*, volume 23 of *Studies in Computational Intelligence*, pages 191–200. Springer, 2006.

[164] M. Radovanović and M. Ivanović. Document representations for classification of short Web-page descriptions. In *Proceedings of the 8th International Conference on Data Warehousing and Knowledge Discovery (DaWaK)*, volume 4081 of *Lecture Notes in Computer Science*, pages 544–553. Springer, 2006.

[165] M. Radovanović and M. Ivanović. Interactions between document representation and feature selection in text categorization. In *Proceedings of the 17th International Conference on Database and Expert Systems Applications (DEXA)*, volume 4080 of *Lecture Notes in Computer Science*, pages 489–498. Springer, 2006.

[166] M. Radovanović and M. Ivanović. Document representations for classification of short Web-page descriptions. *Yugoslav Journal of Operations Research*, 18(1):123–138, 2008.

[167] M. Radovanović, M. Ivanović, and Z. Budimac. Text categorization and sorting of Web search results. *Computing and Informatics*, 28(6):861–893, 2009.

[168] M. Radovanović, A. Nanopoulos, and M. Ivanović. Hubs in space: Popular nearest neighbors in high-dimensional data. *Journal of Machine Learning Research*. Forthcoming.

[169] M. Radovanović, A. Nanopoulos, and M. Ivanović. Nearest neighbors in high-dimensional data: The emergence and influence of hubs. In *Proceedings of the 26th International Conference on Machine Learning (ICML)*, pages 865–872, 2009.

[170] M. Radovanović, A. Nanopoulos, and M. Ivanović. Hubness in the context of feature selection and generation (extended abstract). In *Proceedings of the SIGIR Feature Generation and Selection for Information Retrieval Workshop (FGSIR)*, page 9, 2010.

[171] M. Radovanović, A. Nanopoulos, and M. Ivanović. On the existence of obstinate results in vector space models. In *Proceedings of the 33rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 186–193, 2010.

[172] M. Radovanović, A. Nanopoulos, and M. Ivanović. Time-series classification in many intrinsic dimensions. In *Proceedings of the 10th SIAM International Conference on Data Mining (SDM)*, pages 677–688, 2010.

[173] W. M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the Americal Statistical Association*, 66(336):846–850, 1971.

[174] C. A. Ratanamahatana and E. Keogh. Three myths about dynamic time warping. In *Proceedings of the 5th SIAM International Conference on Data Mining (SDM)*, pages 506–510, 2005.

[175] G. Rätsch, T. Onoda, and K.-R. Müller. Soft margins for AdaBoost. *Machine Learning*, 42(3):287–320, 2001.

[176] J. D. M. Rennie, L. Shih, J. Teevan, and D. R. Karger. Tackling the poor assumptions of naive Bayes text classifiers. In *Proceedings of the 20th International Conference on Machine Learning (ICML)*, pages 616–623, 2003.

[177] A. Ribeiro, V. Fresno, M. C. Garcia-Alegre, and D. Guinea. Web page classification: A soft computing approach. In *Proceedings of the 1st Atlantic Web Intelligence Conference (AWIC)*, volume 2663 of *Lecture Notes in Artificial Intelligence*, pages 103–112, 2003.

[178] S. Robertson. Threshold setting and performance optimization in adaptive filtering. *Information Retrieval*, 5(2–3):239–256, 2002.

[179] S. E. Robertson, S. Walker, and M. Beaulieu. Okapi at TREC-7: Automatic ad hoc, filtering, vlc and interactive track. In *Proceedings of the 7th Text Retrieval Conference (TREC-7)*, pages 253–264, 1998.

[180] S. E. Robertson, S. Walker, S. Jones, M. M. H. Beaulieu, and M. Gatford. Okapi at TREC-3. In *Proceedings of the 3rd Text Retrieval Conference (TREC-3)*, pages 109–126, 1995.

[181] F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65:386–408, 1959.

[182] M. E. Ruiz and P. Srinivasan. Automatic text categorization using neural networks. In *Proceedings of the 8th ASIS/SIGCR Workshop on Classification Research*, pages 59–72, Washington, USA, 1997.

[183] H. Sakoe and S. Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-26(1):43–49, 1978.

[184] G. Salton, editor. *The SMART Retrieval System: Experiments in Automatic Document Processing*. Prentice-Hall, 1971.

[185] G. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620, 1975.

[186] S. Salvador and P. Chan. Determining the number of clusters/segments in hierarchical clustering/segmentation algorithms. In *Proceedings of the 16th IEEE International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 576–584, 2004.

[187] S. L. Salzberg. On comparing classifiers: Pitfalls to avoid and a recommended approach. *Data Mining and Knowledge Discovery*, 1(3):317–328, 1997.

[188] R. E. Schapire and Y. Singer. Improved boosting algorithms using confidence-rated predictions. *Machine Learning*, 37(3):297–336, 1999.

[189] J. P. Scott. *Social Network Analysis: A Handbook*. Sage Publications, 2nd edition, 2000.

[190] F. Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47, 2002.

[191] F. Sebastiani. Text categorization. In A. Zanasi, editor, *Text Mining and its Applications to Intelligence, CRM and Knowledge Management*, pages 109–129. WIT Press, Southampton, UK, 2005.

[192] N. Sebe, I. Cohen, A. Garg, and T. S. Huang. *Machine Learning in Computer Vision*. Springer, 2005.

[193] G. A. F. Seber. *Multivariate Observations*. Wiley, 1984.

[194] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

[195] A. Singh, H. Ferhatosmanoğlu, and A. Şaman Tosun. High dimensional reverse nearest neighbor queries. In *Proceedings of the 12th International Conference on Information and Knowledge Management (CIKM)*, pages 91–98, 2003.

[196] A. Singhal. *Term Weighting Revisited*. PhD thesis, Cornell University, Ithaca, New York, USA, 1997.

[197] A. Singhal, C. Buckley, and M. Mitra. Pivoted document length normalization. In *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 21–29, 1996.

[198] A. Singhal, J. Choi, D. Hindle, D. D. Lewis, and F. Pereira. AT&T at TREC-7. In *Proceedings of the 7th Text Retrieval Conference (TREC-7)*, pages 239–252, 1998.

[199] P. Soucy and G. W. Mineau. Beyond TFIDF weighting for text categorization in the vector space model. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1130–1135, 2005.

[200] K. Spärck Jones, S. Walker, and S. E. Robertson. A probabilistic model of information retrieval: Development and comparative experiments. *Information Processing and Management*, 36(6):779–808, 809–840, 2000.

[201] M. Steinbach, G. Karypis, and V. Kumar. A comparison of document clustering techniques. In *Proceedings of the ACM SIGKDD Workshop on Text Mining*, 2000.

[202] M. Stricker, F. Vichot, G. Dreyfus, and F. Wolinski. Vers la conception automatique de filtres d'informations efficaces. In *Proceedings of Reconnaissance des Formes et Intelligence Artificielle (RFIA)*, pages 129–137, 2000.

[203] P.-N. Tan, M. Steinbach, and V. Kumar. *Introduction to Data Mining*. Addison Wesley, 2005.

[204] Y. Tao, D. Papadias, X. Lian, and X. Xiao. Multidimensional reverse *k*NN search. *VLDB Journal*, 16(3):293–316, 2007.

[205] B. Taskar. CIS520: Machine learning. `http://alliance.seas.upenn.edu/~cis520/wiki/`, 2010.

[206] K. Teknomo. Similarity measurement. `http://people.revoledu.com/kardi/tutorial/Similarity/`, 2010.

[207] J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

[208] A. Tversky and J. W. Hutchinson. Nearest neighbor analysis of psychological spaces. *Psychological Review*, 93(1):3–22, 1986.

[209] A. Tversky, Y. Rinott, and C. M. Newman. Nearest neighbor analysis of point processes: Applications to multidimensional scaling. *Journal of Mathematical Psychology*, 27(3):235–250, 1983.

[210] L. van der Maaten. An introduction to dimensionality reduction using Matlab. Technical Report MICC 07-07, Maastricht University, Maastricht, The Netherlands, 2007.

[211] L. van der Maaten, E. Postma, and J. van den Herik. Dimensionality reduction: A comparative review. Technical Report TiCC-TR 2009-005, Tilburg University, Tilburg, The Netherlands, 2009.

[212] C. J. van Rijsbergen. *Information Retrieval*. Butterworths, 2nd edition, 1979.

[213] V. Vapnik. Estimation of dependencies based on empirical data (in Russian). *Nauka*, 1979. English translation Springer-Verlag, Berlin, 1982.

[214] V. Vapnik and A. Chervonenkis. Theory of pattern recognition (in Russian). *Nauka*, 1974. German translation Akademie-Verlag, Berlin, 1979.

[215] M. Verleysen and D. François. The curse of dimensionality in data mining and time series prediction. In *Proceedings of the 8th International Workshop on Artificial Neural Networks (IWANN)*, volume 3512 of *Lecture Notes in Computer Science*, pages 758–770, 2005.

[216] D. Verma. SpectraLIB – package for symmetric spectral clustering. `http://www.stat.washington.edu/spectral/`, 2003.

[217] A. Vezhnevets. GML AdaBoost Matlab Toolbox. MSU Graphics & Media Lab, Computer Vision Group, `http://graphics.cs.msu.ru/`, 2006.

[218] M. Vlachos, D. Gunopoulos, and G. Kollios. Discovering similar multidimensional trajectories. In *Proceedings of the 18th International Conference on Data Engineering (ICDE)*, paper 0673, 2002.

[219] N. Vlassis, Y. Motomura, and B. Kröse. Supervised dimension reduction of intrinsically low-dimensional data. *Neural Computation*, 14(1):191–215, 2002.

[220] U. von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007.

[221] A. S. Weigend, E. D. Wiener, and J. O. Pedersen. Exploiting hierarchy in text categorization. *Information Retrieval*, 1(3):193–216, 1999.

[222] K. Q. Weinberger and L. K. Saul. Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research*, 10:207–244, 2009.

[223] Wikipedia. Expectation-maximization algorithm. `http://en.wikipedia.org/wiki/Expectation-maximization_algorithm`, 2010.

[224] I. H. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann Publishers, 2nd edition, 2005.

[225] X. Wu, R. Srihari, and Z. Zheng. Document representation for one-class SVM. In *Proceedings of the 15th European Conference on Machine Learning (ECML)*, volume 3201 of *Lecture Notes in Artificial Intelligence*, pages 489–500. Springer, 2004.

[226] Y. Yang. Expert network: Effective and efficient learning from human decisions in text categorization and retrieval. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 13–22, 1994.

[227] Y. Yang and J. O. Pedersen. A comparative study on feature selection in text categorization. In *Proceedings of the 14th International Conference on Machine Learning (ICML)*, pages 412–420, 1997.

[228] Y. Yang, S. Slattery, and R. Ghani. A study of approaches to hypertext categorization. *Journal of Intelligent Information Systems, Special Issue on Automated Text Categorization*, 18(2):219–241, 2002.

[229] Y.-C. Yao and G. Simons. A large-dimensional independent and identically distributed property for nearest neighbor counts in Poisson processes. *Annals of Applied Probability*, 6(2):561–571, 1996.

[230] M. Yetisgen-Yildiz and W. Pratt. The effect of feature representation on MEDLINE document classification. In *Proceedings of the American Medical Informatics Association Annual Symposium (AMIA)*, pages 849–853, 2005.

[231] P. N. Yianilos. Normalized forms for two common metrics. Technical report, NEC Research Institute, 1991, 2002.

[232] F. W. Young. Multidimensional scaling. In S. Kotz and N. L. Johnson, editors, *Encyclopedia of Statistical Sciences*, volume 5, pages 649–658. Wiley, 1985.

[233] S. Zadrozny and J. Kacprzyk. Computing with words for text processing: An approach to the text categorization. *Information Sciences*, 176(4):415–437, 2006.

[234] K. Zeger and A. Gersho. Number of nearest neighbors in a Euclidean code. *IEEE Transactions on Information Theory*, 40(5):1647–1649, 1994.

[235] D. Zhang, Z.-H. Zhou, and S. Chen. Semi-supervised dimensionality reduction. In *Proceedings of the 7th SIAM International Conference on Data Mining (SDM)*, pages 629–634, 2007.

[236] Y. Zhao and G. Karypis. Empirical and theoretical comparisons of selected criterion functions for document clustering. *Machine Learning*, 55(3):311–331, 2004.

[237] X. Zhu. Semi-supervised learning literature survey. Technical Report 1530, Computer Sciences, University of Wisconsin-Madison, Madison, Wisconsin, USA, 2005.

[238] X. Zhu, Z. Ghahramani, and J. Lafferty. Semi-supervised learning using Gaussian fields and harmonic functions. In *Proceedings of the 20th International Conference on Machine Learning (ICML)*, pages 912–919, 2003.

[239] V. Zoonekynd. Time series. `http://zoonek2.free.fr/UNIX/48_R/15.html`, 2010.

Sažetak

„Informatičko doba“ u kom živimo donosi brojne pogodnosti u mnogim aspektima ljudskog delovanja. Automatizacija i kompjuterizacija aktivnosti koje su se u prošlosti obavljale manuelno samo je jedan primer uticaja računara na živote ljudi, kako na profesionalnom, tako i na privatnom planu. Međutim, uz pogodnosti dolaze i mnogi izazovi. Ova disertacija izučava probleme koji proizilaze iz rastuće količine informacija generisanih, čuvanih, i korišćenih na današnjim računarskim sistemima. Brzina kojom informacije nastaju obično premašuje brzinu kojom one mogu biti obrađene, strukturirane, i efektivno korišćene kao *znanje*, što je dovelo do pojave fenomena *zasićenja informacijama*, čiji je uticaj može osetiti, na primer, na *World Wide Web*-u [112, 11, 161]. Pored velikih količina i slabe strukture, informacije prikupljene u formi podataka često sadrže šum, u smislu pogrešnih ili irelevantnih podataka, ili prosto podataka koji su suvišni u kontekstu određene aktivnosti.

Gore pomenute osobine podataka – velika količina, slaba ili neodgovarajuća struktura, i šum – čine ih pogodnim za primenu tehnika mašinskog učenja (*machine learning*, ML), *data mining*-a (DM) i *information retrieval*-a (IR). Oblasti mašinskog učenja i *data mining*-a pružaju mnoge korisne metode za otkrivanje pravilnosti i izvođenje znanja iz sirovih ili slabo obrađenih podataka, npr. iz (hiper)teksta tipičnog za *Web*. Mašinsko učenje i *data mining* imaju sposobnost da obavljaju pomenute zadatke *automatski*. *Information retrieval*, s druge strane, pruža korisniku mogućnost da pronađe jedinice informacija (npr. dokumente) koji mogu zadovoljiti njegovu protrebu za informacijom izraženu u formi upita.

Osnovna reprezentacija u kojoj se informacije sakupljaju i čuvaju jeste tabela (često nazivana *data set*, skup podataka), gde vrste odgovaraju objektima (instancama, primerima, tačkama) opisanim pomoću jednog ili više atributa (osobina, promenljivih) koji formiraju kolone tabele. Povećanje količine dostupnih informacija može se manifestovati u jednoj ili obe sledeće osobine: (1) velikom broju objekata u tabeli, i (2) velikom broju atributa. Fokus ove disertacije je na drugoj osobini, često nazivanoj *velika dimenzionalnost*, za koju je poznato da može da prouzrokuje probleme u mnogim oblastima, uključujući mašinsko učenje, *data mining* i *information retrieval*. Ovi problemi su poznati pod nazivom „*prokletstvo dimenzionalnosti*“.

Disertacija izučava implikacije „prokletstva dimenzionalnosti“ u višedimenzionalnim reprezentacijama podataka, u dva pravca:

1. ponašanje mera udaljenosti (i sličnosti) u uslovima povećanja dimenzionalnosti podataka, i

2. metode odabira atributa, prvenstveno kroz njihovu interakciju sa višedimenzionalnim reprezentacijama tekstualnih dokumenata.

Disertacija je organizovana u tri dela. Deo I posvećen uvodu, koji uključuje prva dva poglavlja disertacije, daje uvid u motivaciju i probleme proučavane u prikazanom istraživanju, i pruža pregled tehnika mašinskog učenja, *data mining*-a i *information retrieval*-a, kao ispomoć razumevanju materijala koji sledi. Pregled, dat u poglavlju 2, uključuje reprezentacije podataka, mere udaljenosti i sličnosti, klasifikaciju, klastering, semi-supervizirano učenje (*semi-supervised learning*), detekciju *outlier*-a, i redukciju dimenzionalnosti.

Deo II posvećen metrikama, koji uključuje poglavlja 3–7, predstavlja pravac istraživanja posvećen ponašanju mera udaljenosti i sličnosti u prostoru podataka rastuće dimenzionalnosti. Poglavlje 3 proučava fenomen koncentracije u okviru kosinusne mere sličnosti. Preostala poglavlja dela II istražuju nov fenomen *habovitosti* (postojanja habova), koji označava tendenciju grafova suseda u višedimenzionalnim skupovima podataka da sadrže čvorove (habove) koji su češće uključeni u liste najbližih suseda drugih tačaka. Poglavlje 4 proučava fenomen sam po sebi na sintetičkim distribucijama tačaka, iz teorijske i empirijske perspektive. Poglavlje 5 stavlja fenomen habova u kontekst stvarnih podataka iz različitih domena primene, generalizuje zaključke iz poglavlja 4, i istražuje efekte fenomena na tehnike mašinskog učenja i *data mining*-a za klasifikaciju, semi-supervizirano učenje, klastering i detekciju *outlier*-a. Poglavlje 6 proučava habovitost u domenu vremenskih serija i predstavlja implikacije fenomena na klasifikaciju vremenskih serija. Na kraju, poglavlje 7 izučava fenomen habova u kontekstu *information retrieval*-a, gde habovi predstavljaju dokumente koji se uporno pojavljuju u rezultatima pretraživanja za mnoge različite upite.

Deo III posvećen reprezentacijama dokumenata i odabiru atibuta, koji se sastoji iz poglavlja 8 i 9, okreće se drugom pravcu istraživanja implikacija „prokletstva dimenzionalnosti“: metodama odabira atributa, i njihovom interakcijom sa višedimenzionalnim reprezentacijama tekstualnih podataka. Poglavlje 8 opisuje eksperimentalnu studiju o uticaju višedimenzionalnih reprezentacija dokumenata na performanse pet klasifikatora. Utvrđeno je da različite transformacije ulaznih podataka: stemovanje, normalizacija, logtf i idf, zajedno sa redukcijom dimenzionalnosti, imaju statistički značajan efekat na performanse klasifikacije, u smislu i poboljšanja i pogoršanja. Pored utvrđivanja najbolje reprezentacije dokumenata za svaki klasifikator, studija opisuje efekte svake transformacije na klasifikaciju, kao i njihove uzajamne odnose. Poglavlje 9 proučava odnos između različitih transformacija reprezentacija tekstualnih podataka i nekoliko široko korišćenih metoda odabira atributa, u kontekstu klasifikacije, pokazujući da idf transformacija značajno utiče na distribuciju performansi klasifikacije u odnosu na stepen redukcije dimenzionalnosti, i predstavljajući metodu evaluacije koja dozvoljava otkrivanje odnosa između različitih reprezentacija dokumenata i metoda odabira atributa nezavisno od apsolutnih razlika u performansama klasifikacije.

Na kraju, poglavlje 10 zaključuje disertaciju, sumirajući glavne rezultate i predstavljajući mogućnosti za dalji rad.

Naučni doprinosi disertacije mogu se sagledati u okvirima pravaca istraživanja datim u delovima II i III.

U delu II, originalni doprinosi počinju sa teorijskim rezultatima koji se tiču koncentracije kosinusne mere sličnosti (poglavlje 3). U preostalim poglavljima dela II pažnja ML, DM i IR istraživača se skreće ka fenomenu habova, koji predstavlja fundamentalnu osobinu distribucija podataka u višedimenzionalnim prostorima, a privukao je iznenađujuće malo ili čak nimalo pažnje do sad. Habovitost i njeni uzroci objašnjeni su sa teorijskog i empirijskog stanovišta na veštačkim distribucijama podataka (poglavlje 4), i stavljeni u kontekst stvarnih podataka iz različitih domena (poglavlje 5), opisujući veze između habovitosti, latentne dimenzionalnosti, i strukture klastera u podacima. Poglavlje 5 takođe proučava interakciju između habovitosti i informacija koje nose oznake klasa, i pokazuje da je fenomen relevantan za različite metode klasifikacije zasnovane na udaljenostima. Poglavlje pruža empirijske rezultate koji opisuju efekat habovitosti na grafovske metode semi-superviziranog učenja, kao i metode za klastering i detekciju *outlier*-a zasnovane na udaljenostima. Značaj fenomena demonstriran je i na podacima iz domena vremenskih serija (poglavlje 6), u kontekstu metoda za klasifikaciju vremenskih serija koje se trenutno smatraju vodećim, pružajući okvir za kategorizaciju skupova podataka zasnovan na osobinama habovitosti i distribucije klasa, a radi razumevanja procesa klasifikacije vremenskih serija. Na kraju, značaj fenomena habovitosti demonstriran je na klasičnim metodama IR-a koje uključuju modele vektorskih prostora. Objašnjen je uzrok fenomena u odnosu na ponašanje kosinusne mere sličnosti na podacima velikih dimenzija, a zaključci su generalizovani na naprednije tehnike reprezentacije i poređenja dokumenata.

U delu III, glavni doprinosi sadržani su u eksperimentlnim rezultatima i metodologijama za klasifikaciju višedimenzionalnih tekstualnih podataka. Opisani su uticaji i odnosi između različitih transformacija *bag-of-words* reprezentacije dokumenata u kontekstu klasifikatora često korišćenih na tekstu (poglavlje 8), i kvantifikovana interakcija između transformacija *bag-of-words* reprezentacije dokumenata i metoda za selekciju atributa (poglavlje 9).

Kratka biografija

Miloš Radovanović je rođen 2. decembra 1977. godine u Novom Sadu. Na Univerzitetu u Novom Sadu 1996. godine upisao je Prirodno-matematički fakultet (Odsek za matematiku, smer „Diplomirani informatičar“), na kojem je diplomirao 2001. godine sa prosečnom ocenom 9,24 odbranivši diplomski rad pod nazivom „Implementacija programskog jezika LispKit LISP u Javi“ sa ocenom 10. Odmah zatim upisao je magistarske studije informatike na istom fakultetu, gde je od jeseni 2002. godine zaposlen kao asistent. Decembra 2006. odbranio je magistarsku tezu „Machine Learning in Web Mining“. Držao je vežbe, za studente informatike, iz više predmeta: Uvod u programiranje, Programski jezici, Strukture podataka i algoritmi, Konstrukcija kompajlera, Izborni seminar (iz oblasti *data mining*-a), Evolucija softvera. Aktivno radi na naučnim projektima koje finansira Ministarstvo za nauku i tehnološki razvoj Republike Srbije. Učestvovao je u realizaciji četiri međunarodna projekta iz programa DAAD, TEMPUS, i iz bilateralne saradnje sa Republikom Slovenijom (dva projekta). Bio je član organizacionih odbora nekoliko konferencija u zemlji, kao i „14th East-European Conference on Advances in Databases and Information Systems“ (ADBIS 2010), koja je održana u Novom Sadu. Takođe, od 2009. član je uredništva časopisa Computer Science and Information Systems (ComSIS). Autor je jedne zbirke zadataka iz programiranja, kao i preko 20 radova iz oblasti mašinskog učenja, *data mining*-a, *text mining*-a, *Web mining*-a i konstrukcije kompajlera.

A Short Biography

Miloš Radovanović was born on December 2, 1977 in Novi Sad, Serbia. In 1996 he began studies of Computer Science at the Department of Mathematics and Informatics, Faculty of Science, University of Novi Sad, where he graduated in 2001 with a 9.24/10.00 grade point average, defending his BSc thesis entitled “An Implementation of Programming Language LispKit LISP in Java,” (in Serbian) with the highest grade. He immediately enrolled in master’s studies of Computer Science at the same department, where he has been employed as a teaching assistant from Fall, 2002. In December 2006 he defended his master’s thesis “Machine Learning in Web Mining.” He teaches, or used to teach, in the following Computer Science courses: Introduction to Programming, Programming Languages, Data Structures and Algorithms, Compiler Construction, Elective Seminar (in the field of data mining), Software Evolution. He actively participates in scientific projects financed by the Ministry of Science and Technological Development of the Republic of Serbia. He was a member of four international projects supported by DAAD, TEMPUS, and bilateral programs with the Republic of Slovenia. He was an organizing committee member of several national conferences, as well as the 14th East-European Conference on Advances in Databases and Information Systems (ADBIS 2010), which was held in Novi Sad. Also, from 2009 he is Managing Editor of the Computer Science and Information Systems (ComSIS) journal. He coauthored one programming textbook, and over 20 papers in the fields of machine learning, data mining, text mining, Web mining, and compiler construction.

Univerzitet u Novom Sadu
Prirodno-matematički fakultet
Ključna dokumentacijska informacija

Redni broj: RBR	
Identifikacioni broj: IBR	
Tip dokumentacije: TD	Monografska dokumentacija
Tip zapisa: TZ	Tekstualni štampani materijal
Vrsta rada: VR	Doktorska disertacija
Autor: AU	Miloš Radovanović
Mentor: MN	dr Mirjana Ivanović
Naslov rada: NR	High-Dimensional Data Representations and Metrics for Machine Learning and Data Mining
Jezik publikacije: JP	engleski
Jezik izvoda: JI	srpski/engleski
Zemlja publikovanja: ZP	Srbija
Uže geografsko područje: UGP	Vojvodina
Godina: GO	2010
Izdavač: IZ	autorski reprint
Mesto i adresa: MA	Novi Sad, Trg D. Obradovića 4
Fizički opis rada: (broj poglavlja/strana/lit. citata/tabela/slika/grafika/priloga) FO	10/244/239/29/71/0/1

Naučna oblast: Računarske nauke
NO
Naučna disciplina: Veštačka inteligencija
ND
Predmetna odrednica/ Ključne reči: Machine learning, data mining, information retrieval, text categorization, curse of dimensionality, concentration, nearest neighbors, classification, semi-supervised learning, clustering, time series, vector space model
PO
UDK
Čuva se:
ČU
Važna napomena:
VN
Izvod: U tekućem „informatičkom dobu“, masivne količine podataka se sakupljaju brzinom koja ne dozvoljava njihovo efektivno strukturiranje, analizu, i pretvaranje u korisno znanje. Ovo zasićenje informacijama se manifestuje kako kroz veliki broj objekata uključenih u skupove podataka, tako i kroz veliki broj atributa, takođe poznat kao velika dimenzionalnost. Disertacija se bavi problemima koji proizilaze iz velike dimenzionalnosti reprezentacije podataka, često nazivanim „prokletstvom dimenzionalnosti“, u kontekstu mašinskog učenja, *data mining*-a i *information retrieval*-a. Opisana istraživanja prate dva pravca: izučavanje ponašanja metrika (ne)sličnosti u odnosu na rastuću dimenzionalnost, i proučavanje metoda odabira atributa, prvenstveno u interakciji sa tehnikama reprezentacije dokumenata za klasifikaciju teksta. Centralni rezultati disertacije, relevantni za prvi pravac istraživanja, uključuju teorijske uvide u fenomen koncentracije kosinusne mere sličnosti, i detaljnu analizu fenomena habovitosti koji se odnosi na tendenciju nekih tačaka u skupu podataka da postanu habovi tako što bivaju uvrštene u neočekivano mnogo lista k najbližih suseda ostalih tačaka. Mehanizmi koji pokreću fenomen detaljno su proučeni, kako iz teorijske tako i iz empirijske perspektive. Habovitost je povezana sa (latentnom) dimenzionalnošću podataka, opisana je njena interakcija sa strukturom klastera u podacima i informacijama koje pružaju oznake klasa, i demonstriran je njen efekat na poznate algoritme za klasifikaciju, semi-supervizirano učenje, klastering i detekciju *outlier*-a, sa posebnim osvrtom na klasifikaciju vremenskih serija i *information retrieval*. Rezultati koji se odnose na drugi pravac istraživanja uključuju kvantifikaciju interakcije između različitih transformacija višedimenzionalnih reprezentacija dokumenata i odabira atributa, u kontekstu klasifikacije teksta.
IZ

Datum prihvatanja teme od strane
NN veća: 15. oktobar 2009.
DP
Datum odbrane:
DO
Članovi komisije:
(Naučni stepen/ime i prezime/zvanje/fakultet)
KO

Predsednik:	dr Zoran Budimac, redovni profesor, Prirodno-matematički fakultet, Univerzitet u Novom Sadu
Mentor:	dr Mirjana Ivanović, redovni profesor, Prirodno-matematički fakultet, Univerzitet u Novom Sadu
Član:	dr Aleksandros Nanopulos, docent, Institut za računarske nauke, Univerzitet u Hildeshajmu, Nemačka
Član:	dr Branimir Todoroović, docent, Prirodno-matematički fakultet, Univerzitet u Nišu

University of Novi Sad
Faculty of Science
Key Words Documentation

Accession number: NO	
Identification number: INO	
Document type: DT	Monograph documentation
Type of record: TR	Textual printed material
Contents code: CC	Doctoral dissertation
Author: AU	Miloš Radovanović
Advisor: MN	Dr. Mirjana Ivanović
Title: TI	High-Dimensional Data Representations and Metrics for Machine Learning and Data Mining
Language of text: LT	English
Language of abstract LA	Serbian/English
Country of publication: CP	Serbia
Locality of publication: LP	Vojvodina
Publication year: PY	2010
Publisher: PU	Author's reprint
Publ. place: PP	Novi Sad, Trg D. Obradovića 4
Physical description: (no. of chapters/pages/bib. refs/tables/figures/graphs/appendices) PO	10/244/239/29/71/0/1

Scientific field: Computer Science
SF
Scientific discipline: Artificial Intelligence
SD
Subject/Key words: Machine learning, data mining, information retrieval, text categorization, curse of dimensionality, concentration, nearest neighbors, classification, semi-supervised learning, clustering, time series, vector space model
SKW
UC
Holding data:
HD
Note:
N
Abstract: In the current information age, massive amounts of data are gathered, at a rate prohibiting their effective structuring, analysis, and conversion into useful knowledge. This information overload is manifested both in large numbers of data objects recorded in data sets, and large numbers of attributes, also known as high dimensionality. This dissertation deals with problems originating from high dimensionality of data representation, referred to as the "curse of dimensionality," in the context of machine learning, data mining, and information retrieval. The described research follows two angles: studying the behavior of (dis)similarity metrics with increasing dimensionality, and exploring feature-selection methods, primarily with regard to document representation schemes for text classification. The main results of the dissertation, relevant to the first research angle, include theoretical insights into the concentration behavior of cosine similarity, and a detailed analysis of the phenomenon of hubness, which refers to the tendency of some points in a data set to become hubs by being included in unexpectedly many k-nearest neighbor lists of other points. The mechanisms behind the phenomenon are studied in detail, both from a theoretical and empirical perspective, linking hubness with the (intrinsic) dimensionality of data, describing its interaction with the cluster structure of data and the information provided by class labels, and demonstrating the interplay of the phenomenon and well known algorithms for classification, semi-supervised learning, clustering, and outlier detection, with special consideration being given to time-series classification and information retrieval. Results pertaining to the second research angle include quantification of the interaction between various transformations of high-dimensional document representations, and feature selection, in the context of text classification.
AB

Accepted by Scientific Board on:	October 15, 2009
AS	
Defended:	
DE	
Dissertation Defense Board:	
(Degree/first and last name/title/faculty)	
DB	
President:	Dr. Zoran Budimac, full professor, Faculty of Science, University of Novi Sad
Advisor:	Dr. Mirjana Ivanović, full professor, Faculty of Science, University of Novi Sad
Member:	Dr. Alexandros Nanopoulos, assistant professor, Institute of Computer Science, University of Hildesheim, Germany
Member:	Dr. Branimir Todorović, assistant professor, Faculty of Science, University of Niš