REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 074-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATE COVERED
	4 May 2007	

4. TITLE AND SUBTITLE
Exploring Dimensionality Reduction for Text Mining

Underhill, David G.

5. FUNDING NUMBERS

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)		8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)		10. SPONSORING/MONITORING AGENCY REPORT NUMBER
US Naval Academy Annapolis, MD 21402		**Trident Scholar project report no. 362 (2007)**

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT	12b. DISTRIBUTION CODE
This document has been approved for public release; its distribution is UNLIMITED.	

13. ABSTRACT Text mining is the extraction of important information from a collection of textual data sources. For instance, text mining can be used to discover related concepts or to categorize previously unseen documents. In this age of information overload, text mining applications can potentially yield tremendous benefits to both individuals and organizations. However, the effectiveness of text mining is limited by the large volume of textual data, as well as its complex and noisy characteristics. Both of these challenges can be addressed with “dimensionality reduction” (DR).DR is the process of transforming a large amount of data into a much smaller, less noisy representation that preserves important relationships from the original data. DR techniques have been shown to effectively simplify large geometric datasets, but have yet to be adequately evaluated for textual data. This project evaluated five DR techniques (Principal Components Analysis, Multidimensional Scaling, Isomap, Locally Linear Embedding, and Laplace-Beltrami Diffusion Maps) from two distinct perspectives. First, the impact of each DR technique on the ability to automatically perform document classification on corpuses of scientific abstracts or news articles was measured. For each technique, the dataset was reduced, then a standard linear, quadratic, or nearest neighbor classifier was used to assign categories to a test set of documents based upon a labeled training set. Results showed that, for any fixed number of dimensions used by the classifier, performing any kind of DR almost always improved classification accuracy compared to using the non-reduced data. Amongst different DR techniques, Isomap and Multi-dimensional Scaling were best able to reduce the data and eliminate noise, yielding improved accuracy. This suggests that these textual data sets lie primarily on a linear manifold for which the more complex non-linear techniques do not have an advantage.

14. SUBJECT TERMS	15. NUMBER OF PAGES
LISA, satellite, formation, insertion error	**119**
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 074-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATE COVERED
	4 May 2007	

4. TITLE AND SUBTITLE
Exploring Dimensionality Reduction for Text Mining

Underhill, David G.

5. FUNDING NUMBERS

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)		8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)		10. SPONSORING/MONITORING AGENCY REPORT NUMBER
US Naval Academy Annapolis, MD 21402		**Trident Scholar project report no. 362 (2007)**

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT	12b. DISTRIBUTION CODE
This document has been approved for public release; its distribution is UNLIMITED.	

13. ABSTRACT Second, this project examined the impact of each DR technique on the ability to automatically discover interesting, previously unknown relationships between scientific articles. This process, known as “Literature-Based Discovery” (LBD), has been previously performed by hand to uncover significant new information such as novel medical treatments. This project first reduced the data with some DR technique, and then used computed similarities between documents from disparate fields such as Chemistry and Physics to identify candidate discoveries. Keywords were extracted from these candidates and used to formulate web search engine queries for articles which contained these disparate ideas together. Effectiveness was measured by a novel numerical method which evaluated the quality of candidates based on the number of search results retrieved. Results suggest that applying DR to a corpus before performing LBD may be an effective way to improve the quality of results for this text mining application.

14. SUBJECT TERMS	15. NUMBER OF PAGES **119**
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT

U.S.N.A. — Trident Scholar project report; no. 362 (2007)

EXPLORING DIMENSIONALITY REDUCTION FOR TEXT MINING

by

Midshipman David G. Underhill, Class of 2007
United States Naval Academy
Annapolis, Maryland

Certification of Adviser Approval

Assistant Professor Lucas K. McDowell
Computer Science Department

(signature)

(date)

Acceptance for the Trident Scholar Committee

Professor Joyce E. Shade
Deputy Director of Research & Scholarship

(signature)

(date)

USNA-1531-2

Abstract

Text mining is the extraction of important information from a collection of textual data sources. For instance, text mining can be used to discover related concepts or to categorize previously unseen documents. In this age of information overload, text mining applications can potentially yield tremendous benefits to both individuals and organizations. However, the effectiveness of text mining is limited by the large volume of textual data, as well as its complex and noisy characteristics. Both of these challenges can be addressed with "dimensionality reduction" (DR).

DR is the process of transforming a large amount of data into a much smaller, less noisy representation that preserves important relationships from the original data. DR techniques have been shown to effectively simplify large geometric datasets, but have yet to be adequately evaluated for textual data. This project evaluated five DR techniques (Principal Components Analysis, Multi-dimensional Scaling, Isomap, Locally Linear Embedding, and Laplace-Beltrami Diffusion Maps) from two distinct perspectives.

First, the impact of each DR technique on the ability to automatically perform document classification on corpuses of scientific abstracts or news articles was measured. For each technique, the dataset was reduced, then a standard linear, quadratic, or nearest neighbor classifier was used to assign categories to a test set of documents based upon a labeled training set. Results showed that, for any fixed number of dimensions used by the classifier, performing any kind of DR almost always improved classification accuracy compared to using the non-reduced data. Amongst different DR techniques, Isomap and Multi-dimensional Scaling were best able to reduce the data and eliminate noise, yielding improved accuracy. This suggests that these textual data sets lie primarily on a linear manifold for which the more complex non-linear techniques do not have an advantage.

Second, this project examined the impact of each DR technique on the ability to automatically discover interesting, previously unknown relationships between scientific articles. This process, known as "Literature-Based Discovery" (LBD), has been previously performed by hand to uncover significant new information such as novel medical treatments. This project first reduced the data with some DR technique, and then used computed similarities between documents from disparate fields such as Chemistry and Physics to identify candidate discoveries. Keywords were extracted from these candidates and used to formulate web search engine queries for articles which contained these disparate ideas together. Effectiveness was measured by a novel numerical method which evaluated the quality of candidates based on the number of search results retrieved. Results suggest that applying DR to a corpus before performing LBD may be an effective way to improve the quality of results for this text mining application.

Acknowledgements

Thanks to my advisor Dr. Luke McDowell for the countless hours he generously spent providing me with guidance which helped make this research a success.

Thanks are also due to collaborators Dr. David Marchette and Dr. Jeff Solka from Naval Surface Warfare Center Dahlgren. They not only provided me with an excellent introduction to the complexities of the material this past summer but also took the time to routinely meet with my advisor and myself throughout the year and provided critical feedback and suggestions numerous times.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Individuals, companies, and governments are surrounded by millions of database records, web pages, communications, and other documents that are potentially relevant, yet in danger of being overlooked amongst all the other data. This information overload is made all the more difficult by the great variety of information being considered and by the need to perform analysis very quickly. For instance, a number of modern Naval missions require the prompt fusion and accurate interpretation of information from disparate sources. These include the Maritime Domain Awareness (MDA) mission and the Global War on Terror (GWOT) mission. These missions are challenging because the information that needs to be fused is not simply mathematical, like target track information, but information about changing groups of loosely organized individuals. In this situation, as with many others, the information that needs to be fused exists primarily as free form text.

The free form nature of this information necessitates the use of text mining methodologies for analysis. Text mining is the extraction of important information from a collection of textual data sources. The primary challenge with performing text mining is the complexity of a document and the process by which this complexity is simplified into a more manageable representation. This general process, known as dimensionality reduction (DR), has been studied in many contexts besides that of text mining. Recently, several promising new techniques for performing this reduction have been proposed, including Isomap [18] and LDM's spectral technique [24]. These techniques offer great promise for effectively reducing vast amounts of information while maintaining the most interesting details, but have yet to

be adequately evaluated for text mining.

This work evaluates the effectiveness of several methods for dimensionality reduction as they relate to two distinct text mining applications. First, how dimensionality reduction impacts the ability of standard algorithms to effectively classify documents among known categories has been studied. It was theorized that some newer dimensionality reduction methods which stress local relationships would perform best. Results from classification, however, contradict this hypothesis. Nonetheless, results did show that many DR techniques are able to reduce the data such that classification accuracy is improved when comparing against a classifier that performs no DR but uses the same number of dimensions. In addition, results showed that many of the DR techniques could produce strong accuracies when using only a small number of dimensions.

Second, the impact of dimensionality reduction on the ability to automatically uncover interesting relationships between seemingly unrelated documents was explored. It was theorized that the DR methods which worked best on classification may not be the same techniques which excelled at this new task of "literature-based discovery" (LBD). Results suggested that applying DR techniques could result in relationships of higher quality being uncovered. One of the most effective DR techniques for classification also performed well on LBD, but it was in turn dominated by a DR method which only performed well on the less difficult data sets with regards to classification. Together, these results provide some insight into the best ways to use dimensionality reduction for classification and for LBD.

The next chapter provides brief background information on text mining, DR, and keyword extraction. In Chapter 3, the approach used to conduct these experiments is outlined. The algorithm for translating documents into a machine-usable form is detailed, as well as algorithms for performing dimensionality reduction, classification, and LBD. The data sets tested and the tools that were implemented to carry out the experiments are detailed in Chapter 4. Chapter 5 analyzes the results obtained with each classifier and presents conclusions drawn from those results. Chapter 6 does the same for LBD results. Chapter 7 presents related work. Finally, Chapter 8 describes the overall conclusions of this work as well as ways that this work could be built upon in the future.

Chapter 2

Background

Section 2.1 introduces the concept of text mining and explains some of the inherent difficulties associated with it. Section 2.2 introduces dimensionality reduction as a possible means for improving such analysis. Finally, Section 2.3 introduces keyword extraction, another text-based tool that is used by this work for further processing.

2.1 Text Mining and Literature Based Discovery

Text mining is the process of automatically extracting information from unstructured text. One particular research focus for text mining has been on the development of algorithms to automatically uncover interesting relationships between documents. Such relationships are difficult to uncover by hand because of the sheer amount of information that is available. This task has been referred to by Swanson as *literature-based discovery* (LBD) [28] and more recently by Priebe and Solka as *automated serendipity* [22].

LBD was most famously illustrated by Dr. Don Swanson in 1986 when he discovered that fish oil could be used to treat Reynaud's disease [28]. Swanson made the connection through the process of LBD. First, he started with a disease - Reynaud's disease. He then searched the Medline database, a large set of papers focused on the medical field, and found terms associated with Reynaud's disease, namely high blood viscosity and platelet aggregation. Finally, using only the terms associated with Reynaud's disease, he again searched the Medline database. This second search showed that fish oil could be used to help

control these symptoms, though it was never mentioned as a treatment for Reynaud's disease. Since the symptoms were related to both fish oil and the disease, Swanson hypothesized that fish oil could be used to treat the disease. After further research by medical professionals and clinical trials, Swanson's theory was confirmed [14]. Figure 2.1 demonstrates this *A-B-C approach* used by Swanson. Swanson termed the publications which bridged the connection between Reynaud's disease and fish oil as *complementary but disjoint structures* (CBD). Though each focused on a unique thesis, together they led to something interesting that neither found alone [29].

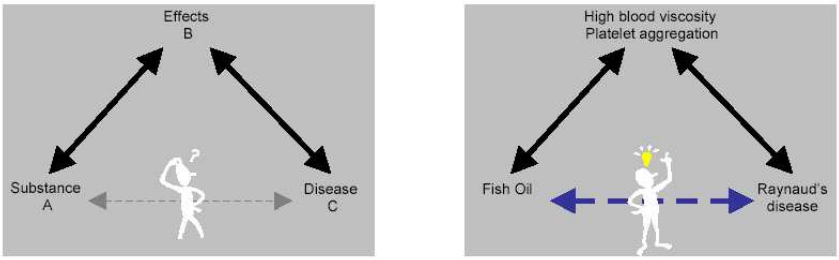

Figure 2.1: The A-B-C approach which Swanson used for his first successful LBD attempts [14].

The difficulty in finding these connections is a result of a large amount of information coupled with researchers whose specialties are so focused that it is difficult to port knowledge from one area of expertise to another [14]. Furthermore, it is an intense undertaking to keep up with just a single field, let alone the great diversity of new information which appears every day. For example, the Medline database contains over 13,000,000 publications and over 3,500 new publications are added each day. Unfortunately, automating the extraction of data from publications is difficult because of their unstructured, natural language representation. Furthermore, the context of an article can be difficult to analyze. The next section describes one approach for addressing these problems.

2.2 Dimensionality Reduction

One of the major hurdles in decoding and comparing documents lies in the complexity of the data [27]. Even a partial representation of a document is described by a space which is composed of many thousands of dimensions. Even when using modern computers, manipulating

such large amounts of data can be computationally expensive. Furthermore, in high dimensional representations, meaningful associations can be lost and important relationships can be obscured by unimportant information. The goal of dimensionality reduction is to reduce a complex set of data in a way which preserves the meanings and associations inherent in the original data. "Reduction" in this sense is reducing the number of dimensions in a sample (document).

A number of methods currently exist for accomplishing this reduction. These methods are broadly grouped into linear and non-linear approaches. These approaches include principal component analysis (PCA), multi-dimensional scaling (MDS), Isomap, locally linear embedding (LLE), and most recently LDM's method. Each of these methods seeks to find a mapping which can represent the important features of the original data in a smaller space with substantially fewer dimensions. This can be expressed mathematically as mapping the original space R^D to a new space R^d where $d << D$. Section 3.2 describes these algorithms, as used in this research, in more detail.

2.3 Keyword Extraction

Keyword extraction is the process of determining which words in a document best describe its content [20]. Keywords are important to the discovery process utilized by this research because they can be used to quickly search extremely large collections of data. In particular, combinations of keywords from potentially related documents can be used to search online databases like *Google*™. The process of computing a word's importance to a particular document is discussed in Section 3.1. The keyword extraction process is presented in Section 3.4.

Chapter 3

Approach

This work's approach to text mining consists of a three-step process illustrated by Figure 3.1. First, a collection of d documents documents are encoded into a form which the computer can manipulate as described in Section 3.1. The second (optional) step, explained in Section 3.2, is to perform dimensionality reduction on the information captured in the encoding process. Finally, Section 3.3 describes analysis via document classification. Sections 3.5 and 3.4 describe analysis with LBD.

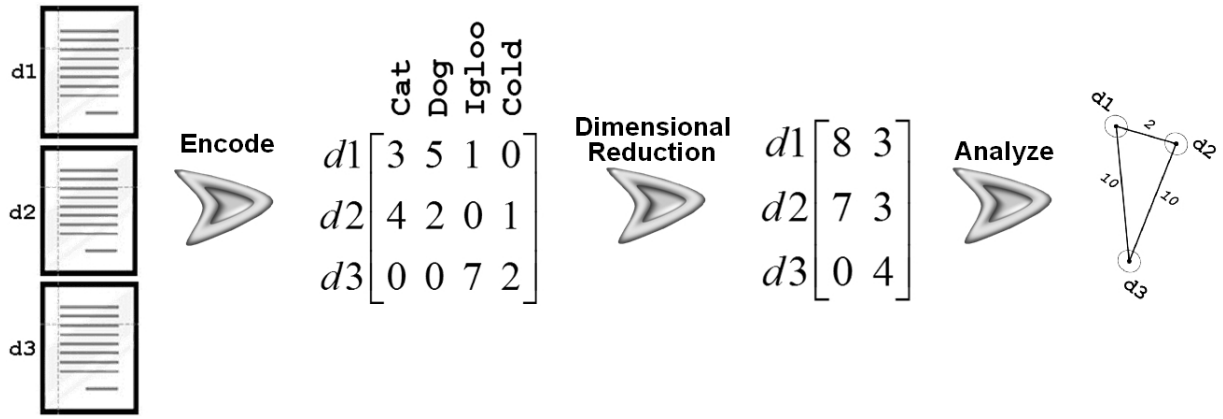

Figure 3.1: Our approach to text mining consists of a three-step process explained in Section 3. For simplicity, this figure shows just 3 documents and 4 words; normally both of these numbers would be much larger.

3.1 Encoding

In order to work with documents and find relationships between them, they are first encoded in some format which a machine is able to process. These encodings are formulated as matrices where each document is described by a row in a matrix. There are a number of possible matrices and ways to compute them, but based on previous work, a relatively simple variation on word counting is used to encode a corpus of documents [26, 21].

Each column in the aforementioned matrix is referred to as a *feature* which describes each document. These features are computed based on weighted word counts. The resulting matrix is known as a (weighted) Term-Document Matrix (TDM) and can be computed as follows:

1) Build a list of all stemmed words, also known as terms, in the corpus. Stemming is the process of finding the root of a word. For example, the stemmed version of "love," "loves," and "loved" is "lov." This allows the encoding process to be largely indifferent to tense and pluarity of words. More complicated approaches of encoding such as analyzing words in context is known as Natural Language Processing (NLP). However, NLP is extremely difficult and cannot efficiently scale up to corpuses containing hundreds or thousands of documents like the data sets used in this research. As a result, no NLP is performed. This is consistent with other research in the text mining field.

When terms are collected individually, they are called unigrams. To try to gain some contextual information, terms may be collected in pairs or triplets (bigrams or trigrams). Unigrams are often used because the additional complexity added by bigrams or trigrams contributes little to the quality of the encoding.

2) Count the number of times each stemmed term appears in each document.

3) Count the total number of times each stemmed term appears in every document.

4) Prune the term list. Several techniques are used to prune words from the term list.

First, terms which do not contribute to the overall meaning of the document, known as stopper or noise terms, can be removed during this stage. Stopper terms are domain-

specific, but some such as "the" and "of" are removed from all corpora. The list of stopper words removed is provided in Appendix C.1.

Second, terms are restricted to the domain of alphabetic or alphanumeric characters. Therefore numbers are pruned out. Terms are also required to be at least three characters long.

Finally, the terms can be pruned based on the frequency at which they appear which allows for the removal of terms which appear too often or too infrequently.

The terms remaining on the pruned term list are the columns in the TDM. Specifically, the term-document matrix M would contain the number of times word j appeared in document i at $M_{i,j}$.

5) The weighted term-document matrix T can be computed as the weight of each term for each document. Weight each term of each document using the Term-Frequency Inverse-Document Frequency formula. Given the number of times t term j appears in document i, the total number of times T all terms appeared in document i, the total number of documents in the corpus D and the number of documents d which contain the term j, the TF-IDF weight $T_{i,j}$ for term j in document i is as follows:

$$T_{i,j} = (t/T) * ln(D/d) \tag{3.1}$$

The term-document matrix is the input to some dimensionality reduction and classifications techniques. Others require an interpoint distance matrix (IPDM), also known as a dissimilarity or distance matrix. The IPDM is defined as an $n \times n$ matrix which defines the strength of the relationships between each document and every other document. These relationship strengths may also be referred to as distances. A small distance between two documents indicates that the two are very similar while a large distance represents documents which are weakly connected or far apart. In other words, the IPDM contains pairwise distances between documents given some function δ which defines the distance between two document doc_i and doc_j.

The IPDM is defined as:

$$Dist_{i,j} = Dist_{j,i} = \delta(doc_i, doc_j) \tag{3.2}$$

3.2 Dimensionality Reduction

Each row in the encoded matrix can be viewed as the "signature vector" of its corresponding document and is represented by some large dimensional space such as R^{10000} (10000 dimensional space). This space is too large and noisy to work with as it is, so some form of dimensionality reduction is applied. The goal of dimensionality reduction would be to reduce the entire matrix so that each signature vector is stored in a much small number of dimensions, e.g., R^5 space.

The input of a dimensionality reduction technique is a term-document matrix T (though some techniques just compute the IPDM from T and use that). The output of a dimensionality reduction technique is a matrix P which consists of points in the new space. Each row contains the new representation of the corresponding document in the input TDM or IPDM.

There are two broad types of dimensionality reduction - *linear* and *non-linear*.

Linear Methods

Linear approaches look at the relationships between all of the documents, regardless of whether the distance between a pair of documents is small or great (i.e. whether the pair is very similar or not). Since relationships between documents which are "far" away are also influential, these approaches are known as *global* approaches.

3.2.1 Principal Components Analysis (PCA)

PCA, also known as the discrete Karhunen-Loeve transform, is a correlation-based dimensionality reduction technique. In this approach, a set of representative dimensions called the principal components (PCs) are chosen based on the degree of variation that they capture from the original set of dimensions [16]. For instance, Figure 3.2 shows how the dimensions which vary most are identified on a simple two-dimensional plot. These new dimensions do not necessarily correspond to the original dimensions. In general, dimensions are ordered

based on how much variance they capture. Since the first ones maximize variance, the last few indicate less variable dimensions. Furthermore, the dimensions of lower importance may also usually represent noise in a data set. This means PCA is also capable of distinguishing between noise and interesting details [18]. The relative significance of a dimension can be inferred from a scree plot like the one presented in Figure 3.3.

A *scree plot* is a plot of the eigenvalues found by an eigenvalue decomposition such as SVD. Typically, the plot bends as the relative importance of the eigenvalues drops off. The bend is referred to as the *elbow* of the graph and the eigenvalue at which this elbow occurs is usually a good point to cut off all less significant eigenvalues [22]. This choice of how many eigenvalues to use to create the output matrix determines how many dimensions are in the output matrix. See Figure 3.3 for an example of a scree plot.

One of the biggest advantages of PCA is its lack of a tuning parameter [23]. Furthermore, it is a non-iterative algorithm. This means the process by which the dimensionality of the input is reduced is reached by taking a specific number of non-looping steps which can provide a significant performance boost over iterative algorithms.

The embedding found by PCA is suboptimal with regards to the ability to accurately map differences between categories because it does not adequately separate different categories of observations [13].

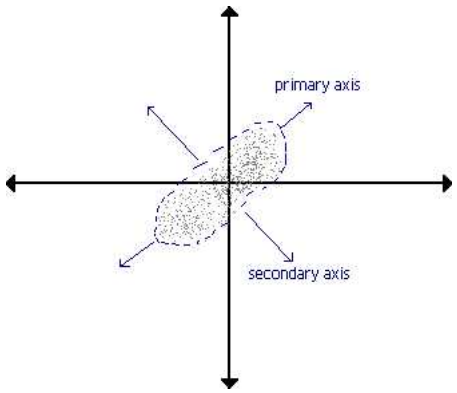

Figure 3.2: PCA determines the directions in which the greatest unique variances occur. For instance, in this picture the first principal component found (the "primary axis") captures the most variance in the original data, followed by the "secondary axis."

The principal components are computed as follows:

Input: Term-Document Matrix, T ($d \times N$ where d = number of documents and N = number of terms)

Algorithm:

1) *(Optional)* Normalize the data in the term-document matrix T. Let x be the scaled and column-centered values computed for T by Equation B.9 (see Appendix).

2) Compute Σ, the $r \times c$ covariance matrix of x:

Covariance is the average spread, or how far a variable is from the mean. The covariance between two documents k and l in x is:

$$\Sigma_{k,l} = \Sigma_{l,k} = \frac{1}{N-1} \sum_{i=1}^{N} ((x_{k,i} - \overline{x_{*,i}})(x_{l,i} - \overline{x_{*,i}})) \tag{3.3}$$

The $d \times d$ covariance matrix of x is:

$$\Sigma = \begin{bmatrix} \Sigma_{1,1} & \cdots & \Sigma_{1,d} \\ \vdots & \ddots & \vdots \\ \Sigma_{d,1} & \cdots & \Sigma_{d,d} \end{bmatrix} \tag{3.4}$$

3) Compute the singular value decomposition (SVD) of Σ:

SVD decomposes a real rectangular matrix Σ into its factorization as follows:

$$SVD(\Sigma) = U \cdot D \cdot V \tag{3.5}$$

D is a diagonal matrix whose off-diagonal values are zeros and whose diagonal is the square root of each eigenvalue in decreasing order of magnitude:

$$D = \begin{bmatrix} \sqrt{\lambda_1} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sqrt{\lambda_r} \end{bmatrix} \tag{3.6}$$

4) Determine n, the dimensionality of the reduced matrix.

The output matrix M still has d rows but the new number of columns n may be chosen to be a small number such $n << N$. The maximum value of n may be as large as N, though it may be smaller depending on the input matrix x. n may not be larger than N or larger than the number of positive, real eigenvalues found by SVD.

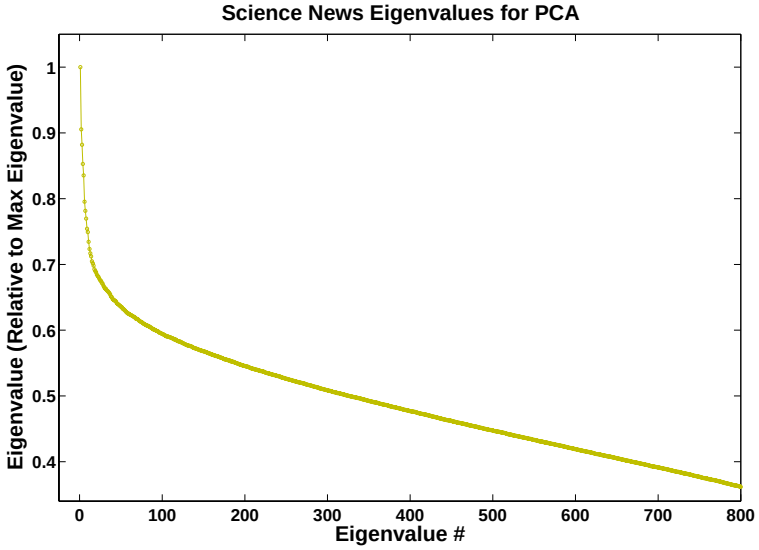

Figure 3.3: This scree plots shows the relative magnitude of eigenvalues found by PCA while performing a singular value decomposition on the encoding of the Science News data set. The elbow of this graph occurs at approximately the 15th eigenvalue.

5) Construct the new matrix P from the column vectors resulting from the multiplication of the square root of the most significant (greatest magnitude) n eigenvalues λ by their corresponding eigenvectors u:

$$P_{PCA} = \begin{bmatrix} u_{1,1}\sqrt{\lambda_1} & \cdots & u_{n,1}\sqrt{\lambda_n} \\ \vdots & \ddots & \vdots \\ u_{1,n}\sqrt{\lambda_1} & \cdots & u_{n,n}\sqrt{\lambda_n} \end{bmatrix} \tag{3.7}$$

3.2.2 Multi-dimensional Scaling (MDS)

MDS is an alternative dimensionality reduction technique that focuses on preserving pairwise distances, or distances between pairs of points [33]. The input consists of a set of proximities or distances between elements in the data, usually in the form of an IPDM, which MDS attempts to embed into a smaller space without losing the relationships between each.

The dimensionally reduced embedding is computed by performing an eigenvalue decomposition on the input. Specifically, MDS is traditionally performed on a corpus as follows [3]:

Input :

A. Dissimilarity Matrix, M ($d \times d$ where d = number of documents) (see Equation 3.2)

B. Number of dimensions to output, r (this is the number of dimensions which is used to represent each dimensionally reduced observation)

Algorithm:

Preprocessing the input: *(Optional)* Embed the input M into a Euclidean space. MDS assumes distances are Euclidean, but text mining often uses cosine distances. Though cosine distances usually work well without doing this step, it is is possible that MDS may fail to find a low-dimensional embedding if the dissimilarities are not a valid Euclidean embedding. If MDS does fail to find a low-dimensional embedding, there is a standard procedure for embedding a non-Euclidean dissimilarities into a Euclidean space. This procedure is detailed in Section B.2.

1) Let S hold the squares of all dissimilarities in M:

$$S_{i,j} = M_{i,j}^2$$

2) Normalize the squared dissimilarities in S (see Equation B.1). Let C be the double-centered values computed for S with Equation B.6 and then multiplied by $-1/2$.

3) Compute the eigenvalues and eigenvectors for C. Keep only the r biggest eigenvalues and associated eigenvectors.

Let D be the diagonal matrix of positive eigenvalues in descending order such that the biggest eigenvalue is $D_{1,1}$ and the second biggest eigenvalue is $D_{2,2}$ and so on.

Let Q be the matrix of eigenvectors associated with the eigenvalues in D such that the first eigenvalue, or $D_{1,1}$, corresponds to the first eigenvector $Q_*, 1$. Therefore, each eigenvector is a column vector in Q.

4) Create the new, lower-dimensional representation by multiplying eigenvectors by the diagonal matrix of the square roots of their associated eigenvalues in descending order. This means the output points P (a $d \times r$ matrix) is computed as follows:

$$P_{MDS} = QD^{\frac{1}{2}} \tag{3.8}$$

Non-Linear Methods

Unfortunately, the global view resulting from the linear approaches can obscure local interactions, possibly obscuring overall patterns in collections of information. PCA and MDS value relationships between items even if the two are far away. This behavior is detrimental to determining the importance of local structures [25]. In contrast, the non-linear approaches focus on local relationships, more accurately preserving the patterns which are blurred by PCA and MDS. The distinction between linear and non-linear methods can be clearly demonstrated when they are applied to determining the structure of images. Non-linear methods work well with geometric figures resembling circles and spirals, where the linear approaches perform comparatively poorly [27]. Isomap and LDM's approach have

been used quite a bit with images, but how they work with regards to text mining was not understood very well.

Dimension reduction is often thought of in terms of geometric transformations of points in high-dimensional spaces. For text mining, each of these points represents a single document. Non-linear methods separate points into a number of sets, often represented by graphs, which contain nodes which are in the same local area. The members of these sets are usually determined by placing the k "nearest neighbors" in each set. An alternative to the nearest neighbor approach is to include all points within a specified radius [24].

Graphs are used to represent points and the relationships between them (modeled as relationships or similarity ratings) because much of graph theory lends itself to operations which are required by current dimensionality reduction techniques. Many existing algorithms have been optimized for graphs including shortest path determination, matrix diagonalization, minimum spanning tree determination (Kruskal's algorithm), and many others [24]. These tools make graphs the best way to represent sparse, related data [18].

3.2.3 Locally Linear Embedding (LLE)

This non-linear approach is a non-iterative process founded on geometric intuition [25]. This unusual foundation contrasts with other graph-based, non-linear techniques. It relies on an assumption that high-dimensional data actually resides on some low-dimensional manifold within the large input space. If this is true, then uncovering a low-dimensional embedding is only a matter of translating, rotating, and scaling the existing data based on weights which maintains "intrinsic geometric properties" present on the low-dimensional manifolds.

LLE has been successfully used to determine the relationships of highly non-linear surfaces with success by a number of researchers. Results for several interesting experiments that benefit from LLE's application are presented in the related work discussion in Section 7.

LLE can be computed as follows [25]:

Input :

A. Term-Document Matrix, T ($d \times N$ where d = number of documents and N = number of terms)

B. Dissimilarity Matrix, M ($d \times d$) (see Equation 3.2)

C. Number of dimensions to output, r

D. *Tuning Parameter*: Number of neighbors to consider for each document, k

Algorithm:

1) Construct a neighborhood graph based on the distances between observations specified by M. Place the k neighbors of observation i in the matrix ν_i ($k \times N$). The weight of term b in neighbor a is $T_{a,b}$. Store this weight for observation i in $\nu_{i_{a,b}}$.

2) Solve for the $k \times d$ reconstruction weights matrix W as follows:

A. Construct the local observation matrix O_i for each observation i. The local observation matrix has k rows - one for each of observation i's k neighbors in ν such that:

$$O_i = \begin{bmatrix} \nu_{i_{1,1}} & \cdots & \nu_{i_{1,N}} \\ \vdots & \ddots & \vdots \\ \nu_{i_{k,1}} & \cdots & \nu_{i_{k,N}} \end{bmatrix} \tag{3.9}$$

B. Subtract observation i's weights from its neighbors' weights. Store the result the $k \times N$ matrix D_i such that:

$$D_i = \begin{bmatrix} O_{i_{1,1}} - T_{i,1} & \cdots & O_{i_{1,N}} - T_{i,N} \\ \vdots & \ddots & \vdots \\ O_{i_{k,1}} - T_{i,1} & \cdots & O_{i_{k,N}} - T_{i,N} \end{bmatrix} \tag{3.10}$$

C. For each observation i, create the $k \times k$ matrix X_i which is the product of multiplying D_i with its transpose such that:

$$X_i = D_i D_i^T \tag{3.11}$$

D. For each observation i, solve the linear system $X_i w_i = 1$ for the weights w_i. Let 1 one be a column vector ones. w_i is a $k \times 1$ column vector of which is being solved for.

E. For each observation i and each term weight j, scale its weights w_{i_j} by the largest weight in w_i and store the scaled weights in a column vector in the $k \times d$ weights matrix W as follows:

$$W_{j,i} = \frac{w_{i_j}}{max(w_i)} \tag{3.12}$$

3) Compute the $d \times d$ cost matrix C as follows:

$$C = (I - W)^T (I - W) \tag{3.13}$$

4) Compute the eigenvalues and eigenvectors for C. Keep only the r smallest eigenvalues and associated eigenvectors.

Let D be the diagonal matrix of positive eigenvalues in ascending order such that the smallest eigenvalue is $D_{1,1}$ and the second smallest eigenvalue is $D_{2,2}$ and so on.

Let Q be the matrix of eigenvectors associated with the eigenvalues in D such that the first eigenvalue, or $D_{1,1}$, corresponds to the first eigenvector $Q_*, 1$. Therefore, each eigenvector is a column vector in Q.

5) Create the new, lower-dimensional representation by multiplying eigenvectors by the square root of the number of observations. This means the output points P (a $d \times r$ matrix) is computed as follows:

$$P_{LLE} = Q \cdot \sqrt{d} \tag{3.14}$$

3.2.4 Isomap

This technique, though similar to LLE, approaches the problem of efficiently maintaining local patterns by utilizing graphs. Like LLE, it is a non-linear, non-iterative algorithm. It has been shown to work well in a variety of situations, but it is not flawless. Its biggest weakness, much like LLE, is the tuning parameter. If the tuning parameter k (number of neighbors to consider) is picked to be too large, then the local relationships may extend to points which are not truly relevant to a set's local structure, skewing the output. On the other hand, if k is too small then the data appears disconnected and show up as isolated islands of data [2]. Unlike PCA which was able to effectively deal with and identify dimensions which contributed to noise, Isomap is more vulnerable to noise. It does not directly handle noise, though if enough data is present the noise's impact is likely to be inconsequential [9].

Experiments by Tobias Friedrich which compared the effectiveness of Isomap and LLE found Isomap to be more efficient and better able to accurately reduce the dimensionality of the original data than LLE. The accuracy of Isomap's dimensional reduction on Swiss rolls and s-curves were one to two orders of magnitude higher than the LLE. It was also found to work better on sparse data sets. LLE's worst-case runtime is higher than Isomap's as well. However, the results focused on somewhat limited data and both performed far better than linear techniques (namely PCA and MDS) did on the same data sets [12].

Other research on Isomap shows that it is quite strong when it comes to geometric and image data. It can determine the relationships of these highly non-linear surfaces. Results for several interesting experiments that benefit from Isomap's application are presented in the related work discussion in Section 7.

Isomap can be computed as follows [30]:

Input :

A. Dissimilarity Matrix, M $(d \times d)$ (see Equation 3.2)

B. Number of dimensions to output, r

C. *Tuning Parameter*: Number of neighbors to consider for each document, k

Algorithm:

1) Construct a neighborhood graph G based on the distances between observations specified by M. Each node in graph G is connected to its k nearest neighbors.

2) Construct a new dissimilarity matrix D from the shortest distance from every observation to every other observation through the neighborhood graph. In other words, distance is defined as the length of the shortest path through the graph G constructed in step 1. This means that the paths connecting nearest neighbors in G are used to find distances between all points. G is typically very sparse.

With this knowledge, there is an efficient means to compute these paths. To efficiently find the distances between all pairs, a special version of Dijkstra's algorithm can be used which conducts a priority first search.

3) Run MDS on the new dissimilarity matrix D to construct the low-dimensional embedding. The result is dimensionally reduced observtions:

$$P_{Isomap} = MDS(D) \tag{3.15}$$

3.2.5 Laplace-Beltrami Diffusion Maps (LDM)

LDM's method, like other non-linear methods, is aimed at preserving the local features when reducing the number of dimensions from the original data set. The fundamental idea behind LDM is to perform non-linear transformations on the initial interpoint distance matrix in a way that helps accentuate local relationships [6, 7]. In particular, LDM tries to preserve relationships based on path aggregations, or how many paths exist between two observations. A path describes how two documents are connected.

One example to help visualize how LDM works is shown in Figure 3.4. In this example, the input points are in the shape of a barbell. This means that for points in one side of the barbell to get to points on the other side of the barbell, they first travel through the single line of points connecting the two halves. This means there are many fewer paths between a point in the left half and a point in the right half than between two points on the left half.

As a result, LDM clumps each half relatively close to each other, but spreads out the left and right halves and the points which connect them because there are relatively few paths between them.

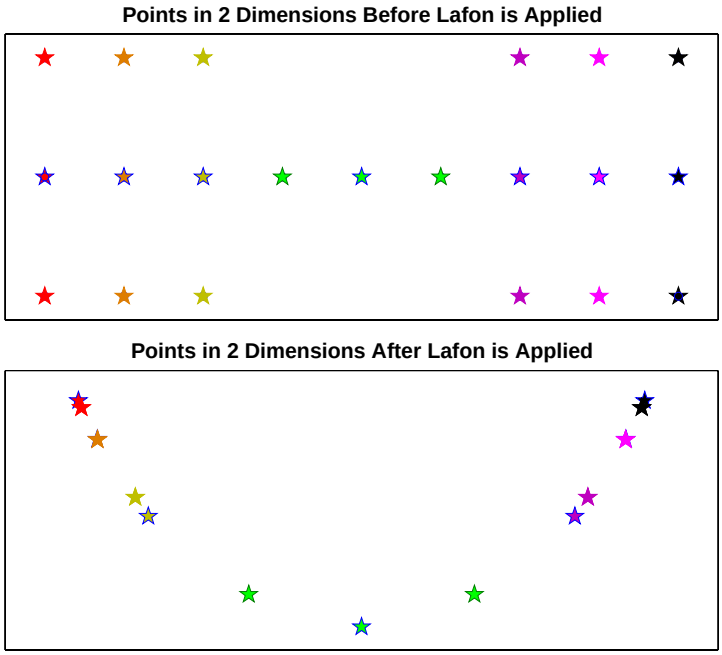

Figure 3.4: The top graph shows a barbell-shaped collection of points. The lower graph shows the points constructed in 2-dimensions by applying LDM to the barbell-shaped points [31].

LDM has the advantage of being an unsupervised dimensionality reduction algorithm. Like PCA and MDS, it has no tuning parameter that is not generally computable. Other non-linear techniques like Isomap and LLE both require a tuning parameter, the number of neighbors, to be set. Setting such parameters is difficult and they can in some cases overfit the model [15]. LDM instead has a tuning parameter ϵ that is defined based on the data.

LDM has been successfully used to analyze various biological data such as CATSCAN images and microarray data from fibroblast cells [15]. Results for these experiments are presented in the related work discussion in Section 7.

LDM can be computed as follows [19]:

Input :

A. Dissimilarity Matrix, M $(d \times d)$ (see Equation 3.2)

B. Number of dimensions to output, r

Algorithm:

1) Let ϵ be the minimum distance between any two observations in the corpus.

$$\epsilon = min(M_{i,j}) \tag{3.16}$$

2) Compute the $d \times d$ matrix $K1$ such that:

$$K1_{i,j} = e^{-\frac{M_{i,j}^2}{\epsilon}} \tag{3.17}$$

3) Let p be the row sums for each row in $K1$ (let 1 be a column vector of 1s):

$$p = K1 \cdot 1 \tag{3.18}$$

4) Define $K2$ to be the element-by-element division of $K1$ by the product of p and its transpose:

$$K2_{i,j} = K1_{i,j}/(p \cdot p^T) \tag{3.19}$$

5) Let v be the square root of the row sums for each row in $K2$ (let 1 be a column vector of 1s). The product of $K2$ and the column vector of 1s produces a column vector which contains the row sums for each row in $K2$. The square root of each element in that vector is then taken. The result is v:

$$v = \sqrt{K2 \cdot 1} \tag{3.20}$$

6) Define $K3$ to be the element-by-element division of $K2$ by the product of v and its transpose:

$$K3_{i,j} = K2_{i,j}/(v \cdot v^T) \tag{3.21}$$

7) Compute the singular value decomposition of $K3$ to get U, D, and V as specified in Equation 3.5.

8) The output points can then be computed as in Equation 3.7 except each eigenvector is scaled by the first eigenvector and then the first eigenvector is discarded [19]. After discarding the first eigenvector, only keep the first r dimensions for the output points as specified in Equation 3.7.

3.2.6 Discussion

There are a variety of techniques for dimensionality reduction. Non-linear techniques are more adept at handling complex data, at least for images and related visual data. However, LDM and Isomap had not been adequately tested with regards to text mining applications prior to this research.

Some of the techniques choose a value for the "free parameter" k. There is no direct way to find the best value for k. As a result, different values of k are empirically tested in order to determine which value is most appropriate. The significance of k varies from one algorithm to another. Isomap, for example, is less sensitive to changes in k than locally linear embedding [12]. LDM's method has the significant advantage of not needing to define a k value.

3.3 Analysis via Classification

In order to gauge the effectiveness of the dimensionality reduction techniques, their output is analyzed. Classification is one way to analyze the effectiveness of the algorithms. Classification uses a set of "training data" with known categories to decide how to assign categories to observations in a "test set." In effect, the "training set" is used to gain an understanding what features distinguish observations from each class.

There are numerous methods for classifying data. This project uses three methods for classifying documents including the k-nearest neighbor (kNN), linear, and quadratic classifiers. The input for these classifiers and some constants based on that input is as follows:

Input: The features which describe each document and the class of each document.

- Feature-Document Matrix, x ($d \times N$ where d = number of documents and N = number of features).
- Class vectors $\hat{c}$, where $\hat{c}_i$ is the vector of document indices in class i such that $x_{\hat{c}_i,k}$ is the k^{th} document in class i.

Only documents being used as training data are assigned to a class vector. Documents which are not being used as training data do not yet have categories and so they are not assigned to any class vector.

Constants: z is defined as the number of classes. n_i is defined as the the number of documents in class i. n is defined as the number of all documents in all classes:

$$n = \sum_{i=1}^{z} \hat{n}_i \tag{3.22}$$

3.3.1 k-Nearest Neighbor Classification

The k-nearest neighbor classifier assigns classes to untrained observations based on the class(es) of the closest observations in the training data [10]. These "closest observations" are also known as *nearest neighbors*. The number of nearest neighbors used to classify each piece of untrained data is k. If k is not one, then this algorithm is called the k-nearest neighbors algorithm.

To classify untrained data using the k-nearest neighbor classifier:

Algorithm:

1) Compute a distance matrix D from the input matrix x using the Equation 3.2.

2) For each unclassified document d_i in D:

A. Compute the vector $\hat{v}$ of the k nearest classified neighbors of d_i.

B. Assign d_i's class as the class of document which appears most often in vector $\hat{v}$.

C. If there is a tie between which class appears most often, then randomly select the class to assign from those which tied with the most occurrences in $\hat{v}$.

D. Sometimes, instead of choosing the class based on which was most frequent among one's neighbors, neighbors could be weighted based on their distance from the point. However, this research used the simpler voting system without weighting nearest neighbors.

3.3.2 Linear Classification

Linear classification is characterized by linear functions which separates classes [10]. In two dimensions with two classes, this can be visualized as a line represented by some linear equation in the form $y = m * a + b$ which separates the two classes. In three dimensions, two classes would be separated by a plane.

Applying a linear classifier is done as follows:

Algorithm:

Part 1: Learning the Classifier

1) Create the training set by constructing a vector $\hat{C}$ of class-specific $d \times N$ feature-document matrices, where $\hat{C}_i$ is a feature-document matrix containing only the documents from x which are in class i. This means $\hat{C}_{i_{j,k}}$ is the value of the k^{th} feature of the j^{th} document in class i.

2) Compute the matrix μ of class-specific feature means, where μ_i is the row vector of feature means from class i. This means μ_{i_j} is the mean of the j^{th} feature in class i:

$$\mu_{i_j} = avg(\hat{C}_{i_{1,j}} : \hat{C}_{i_{n_i,j}}) = \frac{\sum_{q=1}^{n_i} \hat{C}_{i_{q,j}}}{n_i} \tag{3.23}$$

$$\mu = \begin{bmatrix} \mu_{1,1} & \cdots & \mu_{1,N} \\ \vdots & \ddots & \vdots \\ \mu_{z,1} & \cdots & \mu_{z,N} \end{bmatrix} \tag{3.24}$$

3) Compute the class-specific $z \times z$ covariance matrices, where $\hat{\Phi}_i$ is a covariance matrix computed from transpose of $\hat{C}_i$ as defined by Equation 3.4.

4) Compute the pooled covariance matrix Σ from the weighted class-specific covariance matrices in $\hat{\Phi}$:

$$\Sigma = \frac{\sum_{i=1}^{z}(n_i - 1)(\hat{\Phi}_i)}{n - z} \tag{3.25}$$

5) Create the function for each class which predicts the likelihood that some observation belongs to its class.

The likelihood of document x_i being in class $\hat{c}_j$ is defined by the likelihood function $f(x_i, \hat{c}_j)$:

$$f(x_i, \hat{c}_j) = \frac{1}{(2\pi)^{\frac{N}{2}} \cdot |\Sigma|^{\frac{1}{2}}} \cdot e^{-\frac{1}{2}(x_i - \mu_j)\Sigma^{-1}(x_i - \mu_j)^T} \tag{3.26}$$

Since the class of x_i can be determined by comparing the likelihood functions, only the relative order of the function's values are important, not the values themselves. As a result, a simpler likelihood function $f'(x_i, c_j)$ is used:

$$f'(x_i, \hat{c}_j) = (-1) \cdot (x_i - \hat{\mu_j})\Sigma^{-1}(x_i - \hat{\mu_j})^T \tag{3.27}$$

Note: $\hat{\mu_j}$ is the vector of feature means for the class $\hat{c}_j$.

Part 2: Apply the Classifier

Classify each point based on the simplified likelihood function values. For each category j and each unclassified document x_i, compute $f'(x_i, \hat{c}_j)$. Whichever class j yields the highest $f'(x_i, \hat{c}_j)$ is assigned to x_i.

3.3.3 Quadratic Classification

Quadratic classification is characterized by non-linear quadratic functions which separate classes [10]. This provides a more flexible separation scheme than linear classifiers and can better distinguish classes lying on non-linear manifolds. Specifically, quadratic classifiers use the same general process as linear classifiers but use the covariance matrix for class i instead of the pooled covariance matrix for all classes. In particular, the steps are identical except for the following changes below:

Algorithm:

4) This step is not necessary for quadratic classifiers – skip computing the pooled covariance matrix Σ.

5) As with linear classification, create the function for each class which predicts the likelihood that some observation belongs to its class. However, when computing the likelihood use the appropriate class-specific covariance matrix $\hat{\Phi}$ instead of the pooled covariance matrix Σ:

- The likelihood of document x_i being in class $\hat{c}_j$ is defined by the likelihood function $f(x_i, \hat{c}_j)$:

$$f(x_i, \hat{c}_j) = \frac{1}{(2\pi)^{\frac{N}{2}} \cdot |\hat{\Phi}_j|^{\frac{1}{2}}} \cdot e^{-\frac{1}{2}(x_i - \mu_j)\hat{\Phi}_j^{-1}(x_i - \mu_j)^T} \tag{3.28}$$

- Equation 3.28 can be simplified by dropping the $(2\pi)^{\frac{d}{2}}$ because ordering is still the important factor:

$$f(x_i, \hat{c}_j) = \frac{1}{|\hat{\Phi}_j|^{\frac{1}{2}}} \cdot e^{-\frac{1}{2}(x_i - \mu_j)\hat{\Phi}_j^{-1}(x_i - \mu_j)^T} \tag{3.29}$$

Apply the Classifier

Classify each point based on the likelihood function values. For each category j and each unclassified document x_i, compute $f(x_i, \hat{c}_j)$. Whichever class j yields the highest $f(x_i, \hat{c}_j)$ is assigned to x_i.

3.3.4 Confusion Matrices

Confusion matrices (CMs) help identify the source of some classification errors. Each row in the matrix identifies how well a particular category was classified and where it was classified. Table 3.1 shows an example with these categories: Astronomy, History, and Physics. The first row specifies what percentage of astronomy articles were classified as astronomy, as history, and as physics. This means that the diagonal contains the percentage correct for each category. Off-diagonals identify where errors occurred.

Table 3.1: Confusion Matrix (CM) Example

	Predicted Category		
Category	Astronomy	History	Physics
Astronomy	**85.0**	2.0	13.0
History	1.0	**95.0**	4.0
Physics	8.0	2.0	**90.0**

These matrices allow one to quickly see if sources of error seem reasonable. For example, it is understandable to have errors due to confusion between astronomy and physics articles

because those two fields are related. However, if there was a substantial amount of error between history and physics, more rigorous analysis would need to be done in order to understand why unexpected misclassifications of this sort were occurring.

3.4 Keyword Extraction Process

Keyword extraction is driven by the weighted term-document matrix. The most straightforward method is to find keywords for a particular document, is to find keywords for a particular document [5]. This relatively simple method produces relevant keywords, but many of the top keywords are often proper nouns. Proper nouns, however, are not particularly good keywords for finding related documents because they may limit the search to a particular entity's work. Therefore, to improve the quality of the searches built from the extracted keywords, proper nouns had to be removed.

The proper noun removal process used by this research utilizes a number of ideas from past keyword extraction work referenced in Section 7.3. Words were flagged as proper nouns if they met the following criteria:

- A word was considered to be proper noun if it was capitalized and was not the first word in a sentence or in the title.
- A word was considered to be proper noun if it was (or followed) a predefined title such as "Mr." or "Prof."
- If a word appeared multiple times in a single document, all occurrences would be considered proper nouns if at least half of the occurrences were considered to be proper nouns.
- If a word was considered to be a proper noun in more than half of the documents in which it appeared, all occurrences of that proper noun would be considered to be proper nouns in all documents.

Once proper nouns have been removed, keywords can be extracted. Keywords may be individual words (unigrams) or phrases of consecutive words. This research considered all unigrams and bigrams when choosing keywords.

3.5 Analysis via Literature-Based Discovery

LBD is another method which can be used to analyze the effectiveness of dimensionality reduction. The LBD process utilized by this research is composed of several steps. First, pairs of documents which may have an interesting association are uncovered. Second, keywords are extracted from each document in the pair. Third, the keywords are used to search for related documents. Finally, the search results are used to compute a score which estimates the discovery's merit. The process of keyword and search-based LBD evaluation, and the specific scoring technique used, are new contributions of this project.

3.5.1 Step 1: Identify Candidate Discoveries

First, documents which may be associated in a meaningful and interesting way are paired. A meaningful association is defined as a pair of documents which are closely related. The distance matrix defined in Equation 3.2 (computed after DR is performed) is used to determine which pairs of documents are closest. Not all closely related documents, however, are interesting associations. Since documents in the same category are expected to be closely related, only associations between documents in disparate categories are considered interesting.

3.5.2 Step 2: Extract Relevant Keywords

Second, keywords are extracted from each document. This process is described in Section 3.4.

3.5.3 Step 3: Query for Related Documents

The third step uses the keywords to search the web for scientific documents which combine the main ideas from both documents in a pair. Results from this search are interesting because they reveal that the main ideas from the two documents may indeed have an interesting scientific connection, even though they come from different categories. Finding such a match on the web indicates an interesting connection that has been previously described by some scientific articles. This project uses such prior discoveries to assess which DR tech-

niques best facilitate LBD. Closely related documents (according to step 1) for which no appropriate web documents are found may represent previously undiscovered associations. Discovering such associations is the ultimate goal of LBD.

The web searches are formulated so that at least one keyword from each document is present in resulting documents. Three different types of searches were considered. These correspond to searches that use:

1) Two unigram keywords and two bigram keywords from each document.

2) Zero unigram keywords and three bigram keywords from each document.

3) Up to two unigram keywords (minimum weight required) and two bigram keywords from each document. This minimum weight requirement for unigrams was implemented because unigrams were frequently less discriminate than bigrams.

3.5.4 Step 4: Assess Discoveries Based on Query Results

Finally, the results of these searches are used to compute a score which indicates the relevance and novelty of the association. These scores are analyzed in Section 6. To determine a pair's score, searches are executed on both Google™ and Google Scholar™.

First, searches are done that require at least one keyword from *both* documents to be present in resulting documents. The number of documents, $GS_{\cap}$, found by the Google Scholar™ search is an indication of how well known and how valid an association is within the academic community. The number of documents, $G_{\cap}$, found by the Google™ search is an indication of whether or not an association is well known on the web in general.

Second, searches are done that only require one keyword from *either* document to be present in resulting documents. The number of documents, $GS_{\cup}$, found by the Google Scholar™ search is an indication of how common the general concepts are within the academic community. The number of documents, $G_{\cup}$, found by the Google™ search is an indication of how common the general concepts are on the web in general.

The results of the searches are then combined to construct two scores as follows:

$$score_{GS} = \frac{GS_{\cap}}{GS_{\cup}} \tag{3.30}$$

$$score_G = \frac{G_\cap - GS_\cap}{G_\cup - GS_\cup} \tag{3.31}$$

The first score computes how relevant the two documents were to each other in the scientific community, using $GS_\cup$ to normalize by the overall frequency of the documents' keywords. The second score computes a similar relevance score for the general web community. Results from Google Scholar are subtracted out of this score in order to measure the relevance *excluding* the academic community (Google Scholar is roughly a subset of Google).

Finally, these two scores are combined:

$$score_{novelty} = \frac{score_{GS}}{score_G} \tag{3.32}$$

Dividing the Google Scholar score by the Google score returns an overall measure of *novelty* for a candidate association. This score is high for associations that are highly relevant in the academic community ($score_{GS}$), but less high in the general web community ($score_G$).

Chapter 4

Experimental Method

4.1 Data Sets

4.1.1 Science News Corpus

This corpus was comprised of 1,160 previous hand-categorized articles. 113 articles were assigned to multiple categories, so these were removed from the corpus as done by other researchers. The remaining 1,047 articles exist in eight categories: Anthropology (54 articles), Astronomy (121), Behavior (72), Earth Sciences (137), Life Sciences (205), Mathematics (60), Medical (280), and Physics (118).

These categories provided several interesting opportunities. First, some of these categories are very distinct - in other words, they have little in common. This kind of data is easy for dimensionality reduction techniques to take advantage of since there is a relatively clear boundary between such categories. This led to the creation of a special 2-category version of this corpus which is referred to as Science News-2 (the full version is referred to as Science News-8). Science News-2 contains all Astronomy and Medical articles which are very well-separated categories.

Another interesting case was to introduce two other categories to this mix such that they would probably overlap a little with the existing categories and thereby make the data set harder than the simple 2-category, well-separated Science News-2 data set. This data set,

called Science News-4 Separated, also included Earth and Life Sciences. Life Sciences had some overlap with Medical, and Earth was originally thought to have been fairly isolated, but in reality it caused some confusion for both Astronomy and Life Sciences articles. Another version containing four categories is termed Science News-4 Overlapping. It contains the categories from Science News which overlap most - Anthropology, Behavior, Life Sciences, and Medicine. This data set was created as a challenge for the classification analysis. Finally, the full data set was condensed into four "meta-categories" which grouped similar categories together into larger categories. The first meta-category included Behavior, Life Sciences, and Medicine. The second meta-category contained Anthropology and Earth Sciences. The third meta-category contained Astronomy and Physics. The fourth meta-category included Math. This version of Science News, termed Science News-4M, was created for the discovery process. Since cross-category pairs are required, it was theorized that if similar categories were combined, then the candidate discoveries would be more interesting since they would span broader areas.

In the full data set, Science News-8, the categories are still relatively distinct. Also helpful for a classifier, the average length of each article was long compared to articles in other data sets. The average Science News-8 article contains 7.80KB of text.

4.1.2 Google News Corpus

This is a large corpus consisting of 3,028 articles. These articles are distributed among five reasonably well-defined and fairly evenly distributed categories including Business (566 articles), Health (518), Science & Technology (599), US News (672), and World News (673). The average document length is 4.0KB of text, or about half as long as Science News articles. This means there is less information for the classifier to help distinguish documents, but the sheer numbers of documents offsets this less information per document by still providing a lot of information about each category.

4.1.3 Science & Technology Corpus

This corpus is relatively small and is comprised of 658 articles scattered across seven categories. These categories vary greatly in size and are not as clearly separable as categories in the previously mentioned corpora. The seven categories break down as follows: Information (85), Weapons (66), Sensors (162), Ground (96), Biomedical (40), Anti-Submarine (47), and Materials and Processes (162). The articles in this corpus are noisier as they contain less relevant information including copyrights, where something was published, etc. Furthermore, the amount of information per article is much less than Science News - the average article size is only 2.7KB of text. Some articles only contain 2-3 sentences beyond headings and other unimportant information. This makes the Science & Technology corpus the most difficult corpus to obtain accurate classification results for.

4.2 Parameters

Text mining is a complicated process with many parameters. This Section details the parameters relevant to the experiments carried out by this research and explains choices made regarding these parameters. Figures 4.1, 4.2, and 4.3 show the parameters which could be set for each step in the experimental process. Values which are underlined are the typical values used.

4.2.1 Encoding Process Parameters

The process for encoding a corpus is highly configurable. Figure 4.1 illustrates the parameters involved in the encoding process. Based on previous research by Martinez and Wegman, collaborators' experience, and experimentation a "standard" set of encoding parameters was developed for use with this research.

The output of the encoding process is either a TDM or an IPDM depending on the next step in the overall process. The weighted TDM is always computed because it is needed to compute the IPDM. The distance metric and dissimilarity computer parameters only apply if the IPDM is being computed, however. The IPDM is defined by Equation 3.2. However,

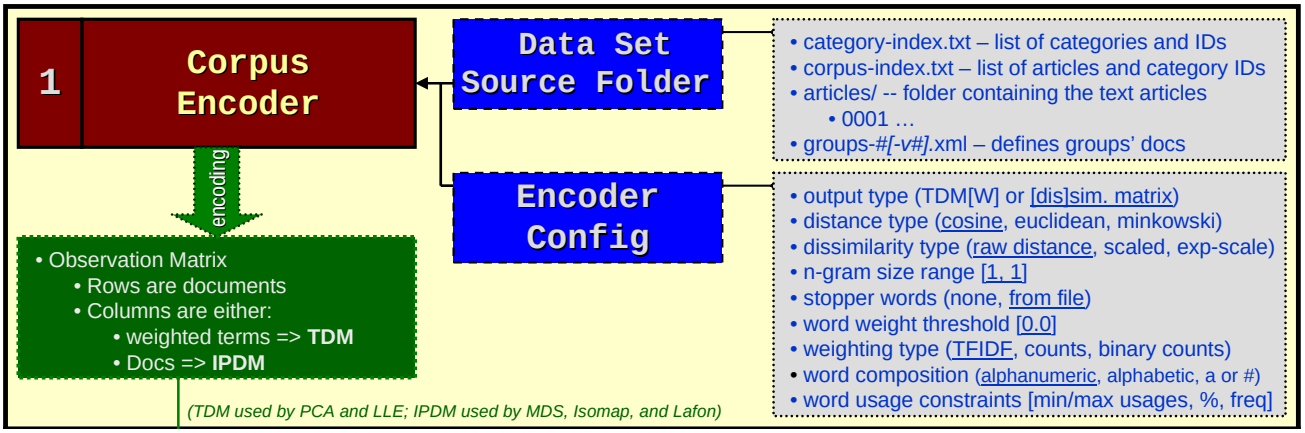

Figure 4.1: Illustration of the parameters that configure the corpus encoding process

there is more than one way to define the distance between two documents. The cosine distance metric is used to create IPDMs for all dimensionality reduction techniques except LDM. The cosine distance d between two documents A and B is defined as the following (recall, documents are represented as a vectors of weights in the TDM):

$$d_{A,B} = \frac{A \bullet B}{\sqrt{(A \bullet A) \cdot (B \bullet B)}} \tag{4.1}$$

LDM performs best with scaled Euclidean distances. The Euclidean distance metric used to produce the Euclidean IPDM for LDM is the usual Euclidean distance, except it is scaled so that the (scaled) distance $\delta_{A,B}$ between two documents A and B is defined as the following (given that $d_{A,B}$ is the standard Euclidean distance between A and B):

$$\delta_{A,B} = (1 - \frac{d_{A,B}}{max(d_{i,j})}) \tag{4.2}$$

Though any range of n-grams could be considered, this research only considered unigrams. In this case, the simpler unigram seemed to better represent the data than more complicated representations based on bigrams, trigrams, or larger n-grams. A standard list of stopper words provided by the collaborators is always used in the encoding process (see Appendix C.1). Weights (as defined by Equation 3.1) are thresholded at $2 \cdot 10^{-32}$ so that weights below that value are set to zero. Weighting is always done using the TF-IDF defined by Equation 3.1 because plain counts are much less informative and they drastically reduced classification accuracy. Words containing alphanumeric compositions are allowed, but not numbers by

themselves since taken out of context a number usually means very little. Words are also only considered if they occur at least three times in the corpus – below that quantity they are so insignificant that it is better to not consider them since each term adds a dimension to the TDM.

These parameters, along with the data set to encode, are passed to the Corpus Encoder tool which produces the TDM or IPDM as specified. That output is then directly passed to the dimensionality reduction tool.

4.2.2 Dimensionality Reduction Parameters

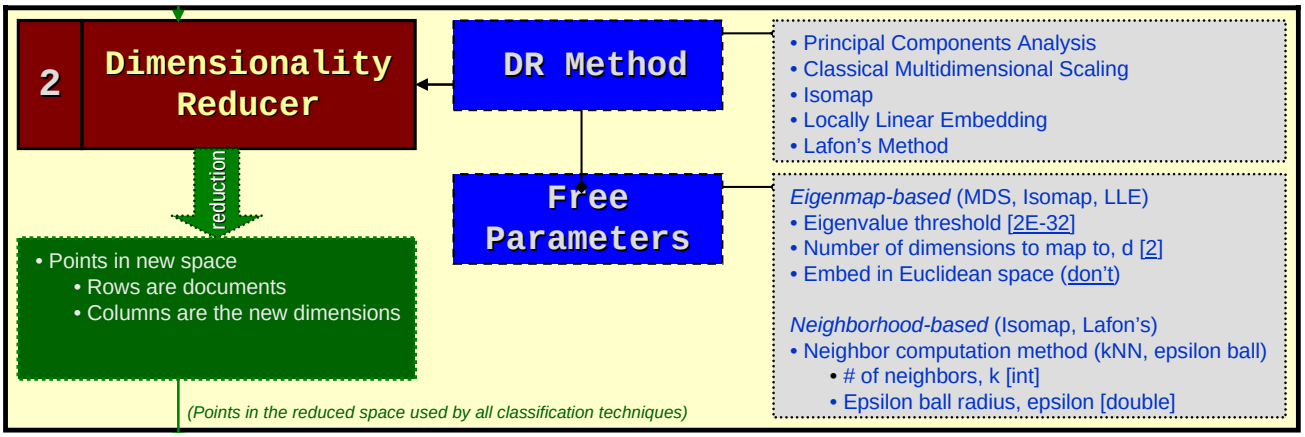

Figure 4.2: Illustration of the parameters that configure the dimensionality reduction process

Though dimensionality reduction is a fairly complicated process, there are relatively few parameters to consider, and only two techniques, Isomap and LLE, require users to specify a tuning parameter. Figure 4.2 illustrates the parameters involved in the dimensionality reduction process.

Every dimensionality reduction technique analyzed here does some sort of eigenvalue decomposition. Eigenvalues which have a value of 0 or less are thrown out because only positive eigenvalues convey useful information. Instead of comparing exactly for 0, eigenvalues are thresholded at $2 \cdot 10^{-32}$ and eigenvalues less than that are treated is if they were 0.

The number of possible output dimensions is equal to the number of positive eigenvalues found. However, since the object of dimensionality reduction is to minimize the number of output dimensions, that parameter may be specified as something other than all pos-

sible dimensions. Specifying a lower parameter value just truncates information in higher dimensions.

Some TDMs and IPDMs are formed in such a way that eigenvalue decompositions may not be possible. This problem can be corrected for any IPDM because it is a symmetric matrix. The fix is to embed the IPDM in a Euclidean space. This process and more details about the motivation for it are discussed in Section B.2.

Finally, Isomap and LLE require a tuning parameter k which affects how many neighbors are considered when constructing "neighborhoods" of documents. Alternatively, a neighborhood may be formed with the ϵ-ball method which makes other observations neighbors if they are within a radius ϵ. In this research, the k-nearest neighbor approach is used with a fixed value of 10. This was experimentally chosen and is further explained in Section 5.4.3.

4.2.3 Classification Parameters

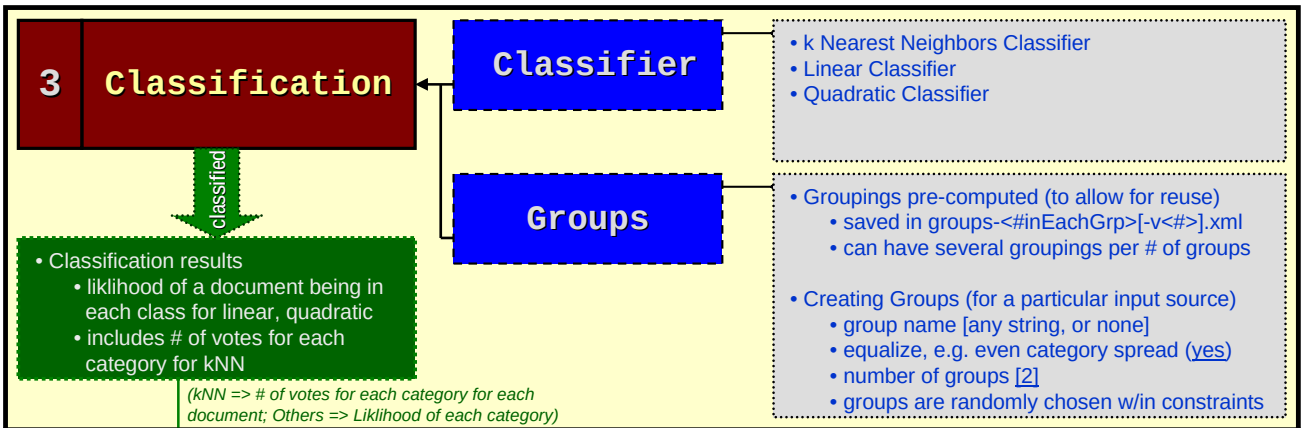

Figure 4.3: Illustration of the parameters that configure the classification process

Like dimensionality reduction, classification is a fairly straightforward process. The only tuning parameter belongs to the kNN classifier which has to specify how many neighbors it looks at to determine the category of an observation. Figure 4.3 illustrates the parameters involved in the classification process.

The number of neighbors (k) which are used to effectively classify documents was extensively tested and a reasonable value for all data sets considered was 9. Classification performance is analyzed in Section 5.2 and the range of most effective values for k is pre-

sented in Section 5.4.2.

All classifiers operate with training and test sets. The training sets are given to the classifier so it has some data which it can use to classify observations whose categories are unknown. Training sets can be specified using "groups" which are simply predefined lists of documents which can be created, stored, and then re-used for the sake of consistency. Groups are only used for cross-validation classification which is when some groups are used to train the classifier and then other groups are classified. For leave-one-out validation, all observations are classified. Each observation is classified using every observation except itself as training data for the classifier. Performance was slightly higher when this kind of validation was done, presumably because more training data results in a more effective classifier. As a result, leave-one-out-validation was used throughout all experiments detailed in Section 5.

The number and types of keywords to choose for each selected document was another parameter (see Section 3.4 for details).

4.3 Tools

The tools described in this section are Java libraries created to perform this research. However, despite the focus of this research of text mining, these tools have been created to be as much of a general-purpose tool as possible. For example, the dimensionality reduction techniques readily work on image, geometric, and other kinds of data.

The *Corpus Encoder* is a tool which takes a collection of articles organized in separate files and encodes them according to user specifications. All parameters discussed in Section 4.2.1 are supported. This tool can be run as a standalone program. Its user manual is included in Section C.5.

The *Groups Encoder* is a tool which takes a list of filenames and their categories and creates groups as specified by the user. The contents groups are output as files so that the groups can be reused (since they can be randomly generated within given constraints). These group files are used by the Experiment Runner tool to split observations between training and test data sets when cross-validation is being performed. The user manual for this tool

is included in Section C.4.

The *Experiment Runner* is a tool which runs a set of classification experiments from start to finish. It allows the user to specify what parameters to vary, etc. Results and associated data can be stored in files where it can be later referenced. The user manual for this tool is included in Section C.3.

Dimensionality reduction and classification tools libraries have been created and are used in the experiment runner. They are stand-alone libraries but no user interface which can be run independently of another program has been created for these yet. However, they have been created so that they may operate on any numerical data so they are not limited to the domain of text mining.

The *Keyword Generator* is a tool which generates keywords for all documents in a corpus. It is able to generate keywords in the form of unigrams, bigrams, and larger n-grams. It also performs proper noun removal as discussed in Section 3.4.

The *Pair Finder* is a tool which finds the closest pairs from disparate categories for a given dimensionality reduction technique and data set. It uses the *Corpus Encoder* to encode a data set. It saves pairs and all relevant data to files which can be later used for LBD.

The *Query Preparer* is a tool which generates searches from pairs of documents and their keywords. In particular, in generates querries which can be easily run against the Google ™ database.

The *Query Runner* is a tool which actually runs querries against the Google ™ databases. It caches results so that identical querries are not run more than once. It is responsible for parsing results and identifying the number of results which were returned.

4.4 Performance Measure

For classification experiments, all performance results are reported in terms of "classification accuracy," which is the percentage of test documents that were assigned to the correct category by the classifier. For LBD experiments, performance results are reported in terms of the relevance and novelty "score," which is defined in Section 3.5.

Chapter 5

Classification Results

This chapter details how the output from each dimensionality reduction technique affected the ability of several classifiers to successfully categorize documents in the corpuses discussed in Section 4.1. Results are presented in order from the simplest classifier, the kNN classifier, to the more complicated Linear and Quadratic classifiers. This section concludes with visualizations of the dimensionally reduced data in addition to some final supporting evidence.

Throughout this chapter, results for each of the five dimensionality reduction techniques are presented and discussed. The term "None-Rand" is used to refer to cases when no dimensionality reduction is applied and instead d random features are chosen from the original features. The term "None-Sort" is used to refer to cases when no dimensionality reduction is applied and the d best features (based on their average TF-IDF score as defined in Equation 3.1) are chosen from the original features.

5.1 k-Nearest Neighbor Classification

5.1.1 Varying Number of Dimensions

(fixed $k = 9$)

Figure 5.1 shows how performance is affected by altering the number of dimensions. The number of nearest neighbors has been fixed at 9 (Figure 5.6 later shows that all five DR techniques performed within a small deviation of their peak performance on these data sets

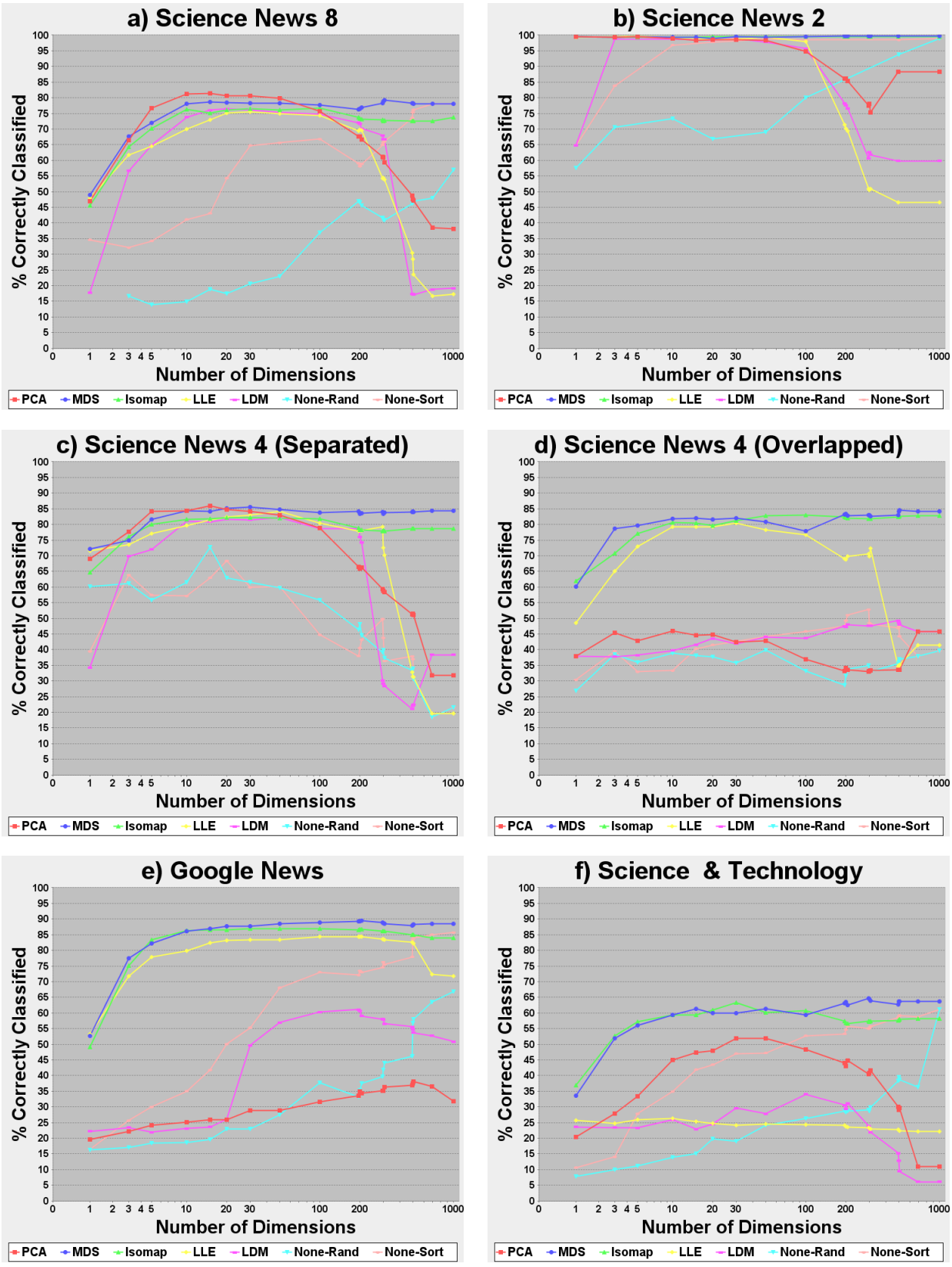

Figure 5.1: kNN: % Correctly Classified vs. Number of Dimensions (log scale), k=9

when 9 nearest neighbors were used).

Though the techniques do comparatively poorly when the number of dimensions is small, they quickly improve so that in most cases they plateau at approximately peak performance when the number of dimensions being reduced to is 10. This performance is maintained until the data is projected into over 100 dimensions. At this point, adding new dimensions adds more noise than useful information which causes the classifier to quickly become confused. It is interesting to note that MDS and Isomap, however, are relatively unaffected by the increase in the projection space beyond 100 dimensions. This ability to recognize dimensions from which no benefit can be derived is a property of MDS. Instead of producing noisy extra dimensions, higher dimensions are essentially constant and hence do not affect classification. Since Isomap makes use of MDS as a final step, it inherits this benefit.

On Science News-8 (5.1a), all five dimensionality reduction techniques perform similarly. This relationship is present in the smaller Science News-2 and Science News-4 Separated variations as well. As the number of number of categories decreases from 8 to 4 (5.1c) to 2 (5.1b), peak accuracy rises from 80% to nearly 100%. This dramatic difference in accuracy is a product of confusion between similar categories in the data sets with more categories. Evidence of confusion occurring between such categories under these parameters is presented later in Table 5.1. The two categories in Science News-2 are very well separated which enables dimensionality reduction techniques to produce embeddings which lead to extremely high classification accuracy in very few dimensions.

For the most part, the Science & Technology data set (5.1f) is consistent with the Science News data sets. Overall performance is lower across the board, likely due to the more difficult nature of the data set with regards to classification. However, it is interesting to see that PCA and in particular LDM, the SVD-based techniques, perform especially poorly. LLE's performance is also poor. It is likely that these techniques are more sensitive to the relatively small amounts of data per observation - both the Google News (5.1e) and the Science & Technology data set have relatively short articles in comparison with Science News. Despite this common property, significantly different performance occurs on each of the two data sets. On Google News, peak performance reaches almost 90%. This can be attributed to the data set having just five reasonably well-separated categories fairly

evenly split among over 3,000 articles. Science & Technology, on the other hand, has seven categories and is unevenly distributed over 658 articles, less than a quarter of the size of Google News - which means its training set for classification is smaller and noisier. These properties make Science & Technology's categories much more difficult to distinguish. These two disadvantages result in a significant performance reduction due to increased confusion. The difficulty with overlapping categories is not limited to Science & Technology. When a subset of Science News which contains only portions of closely related categories is tested (5.1d), PCA and LDM's performance are substantially lower than on the complete Science News corpus.

When dimensionality reduction is not applied and all dimensions are used, performance is quite high across all data sets. When given all of the dimensions (not shown on the graphs), the classification performance for kNN on Science News-8 is 82%, Science News-4 Separated is 87%, Science News-4 Overlapped is 86%, Google News is 91%, and Science & Technology is 65%. All of these classification accuracies exceed the performance achieved when dimensionality reduction is performed, but only slightly. This is exciting because it shows that these dimensionality reduction techniques are able to reduce the encoded data while maintaining much of the important information that was present in the original data's high-dimensional space. Generally peak performance occurs in the neighborhood of just 30 dimensions which is tiny in comparison to the 8,000-20,000 dimensions, depending on the corpus, that are fed to the classifier when no dimensionality reduction is performed.

The None-Rand and None-Sort techniques, for which dimensionality reduction is not applied but where the number of dimensions is limited to the number of dimensions the DR techniques are outputting, perform worse than most DR techniques on average. They generally perform at or below the worst DR performance when the number of dimensions is low, though these both typically start to catch up to MDS and Isomap by 1,000 dimensions.

MDS and Isomap provide consistently high performance across all six data sets. They are also relatively flexible with regards to the number of dimensions that are chosen in order to obtain high classification accuracy with the kNN classifier. In all cases, they approach the performance of doing classification using all features without performing any dimensionality reduction.

Table 5.1: CM: Google News reduced w/MDS to 1 dimension (k=9) – 51.9% Accurate

	Predicted Category				
Category	Business	Health	S&T	US News	World News
Business	**44.3**	19.1	28.8	6.0	1.8
Health	12.9	**54.8**	6.0	19.1	7.1
Science & Technology	27.9	16.5	**50.3**	4.7	0.7
US News	2.2	15.6	1.8	**51.6**	28.7
World News	0.4	6.1	0.6	35.2	**57.7**

In contrast, Table 5.1 shows a confusion matrix for MDS with only one dimension which yielded a much lower accuracy (51.9%). Every category does mediocre with accuracies for each category close to 50%. This is understandable considering the extreme reduction of dimensionality. Furthermore, it is interesting to note that categories are indeed confused with closely related categories. For example, US News and World News have substantial overlap - almost a third of the mistakes in these categories are due to each other. This may be a result of similar language used to describe both scopes of news. Another interesting overlap occurs between Business, Health, and Science & Technology. This indicates a believable connection between these categories.

Table 5.2: CM: Google News reduced w/MDS to 30 dimensions (k=9) – 87.7% Accurate

	Predicted Category				
Category	Business	Health	S&T	US News	World News
Business	**81.1**	3.0	8.1	4.1	3.7
Health	0.2	**91.9**	2.1	3.1	2.7
Science & Technology	5.3	2.3	**89.0**	2.3	1.0
US News	2.5	0.3	0.4	**87.8**	8.9
World News	2.4	0.7	0.6	7.6	**88.7**

As the number of dimensions increases, confusion is steadily reduced. In particular, Table 5.2 shows a confusion matrix for MDS with Google News where the number of dimensions has increased to 30. Here, classification accuracy improved from the 51.9% of Table 5.1 to almost 90%. Adding the extra dimensions allowed the classifier to almost completely separate the Health category from both Business and US News categories. Confusion between those two pairs improved by an order of magnitude over other improvements. However, despite these improvements in performance, sources of confusion have not been completely eliminated.

Business and Science & Technology are still somewhat confused, as are US News and World News.

Table 5.3: CM: Google News reduced w/LDM to 20 dimensions (k=9) – 26.2% Accurate

	Predicted Category				
Category	Business	Health	S&T	US News	World News
Business	**21.9**	15.9	17.8	30.7	13.6
Health	15.1	**23.9**	22.8	25.1	13.1
Science & Technology	16.7	22.0	**22.0**	25.4	13.9
US News	17.1	13.1	14.9	**38.1**	16.8
World News	13.5	16.8	16.2	30.2	**23.3**

Table 5.3 is a confusion matrix that explores LDM's poor performance on Google News. As Figure 5.1e showed, LDM was still performing little better than chance (20%, or one in five for this data set) when embedding into 20 dimensions, even though MDS, Isomap, and LLE obtained accuracies between 75% and 85% with just five dimensions. Table 5.3 shows a near random distribution of the predicted categories for each row, suggesting that LDM has not correctly found an embedding for this large data set of relatively short articles.

This section showed that while varying the number of dimensions has a significant impact on performance, there is a reasonable window in which near-peak performance is achieved. Furthermore, each technique is, for the most part, consistent across each data set and the best performers, with respect to the kNN classifier and varying the number of dimensions, were MDS and Isomap.

5.1.2 Varying Number of Nearest Neighbors

(fixed d = 30)

Figure 5.2 shows how performance is affected by altering the number of nearest neighbors being considered by the kNN classifier. The number of dimensions the input is reduced to has been fixed at 30 (Figure 5.6 shows that all five techniques performed within a small deviation of their peak performance when the data was embedded into a 30-dimensional space). In addition to the DR techniques, None-All is also tested. This approach uses all of the 10,000+ features as they are in the encoded term document matrix.

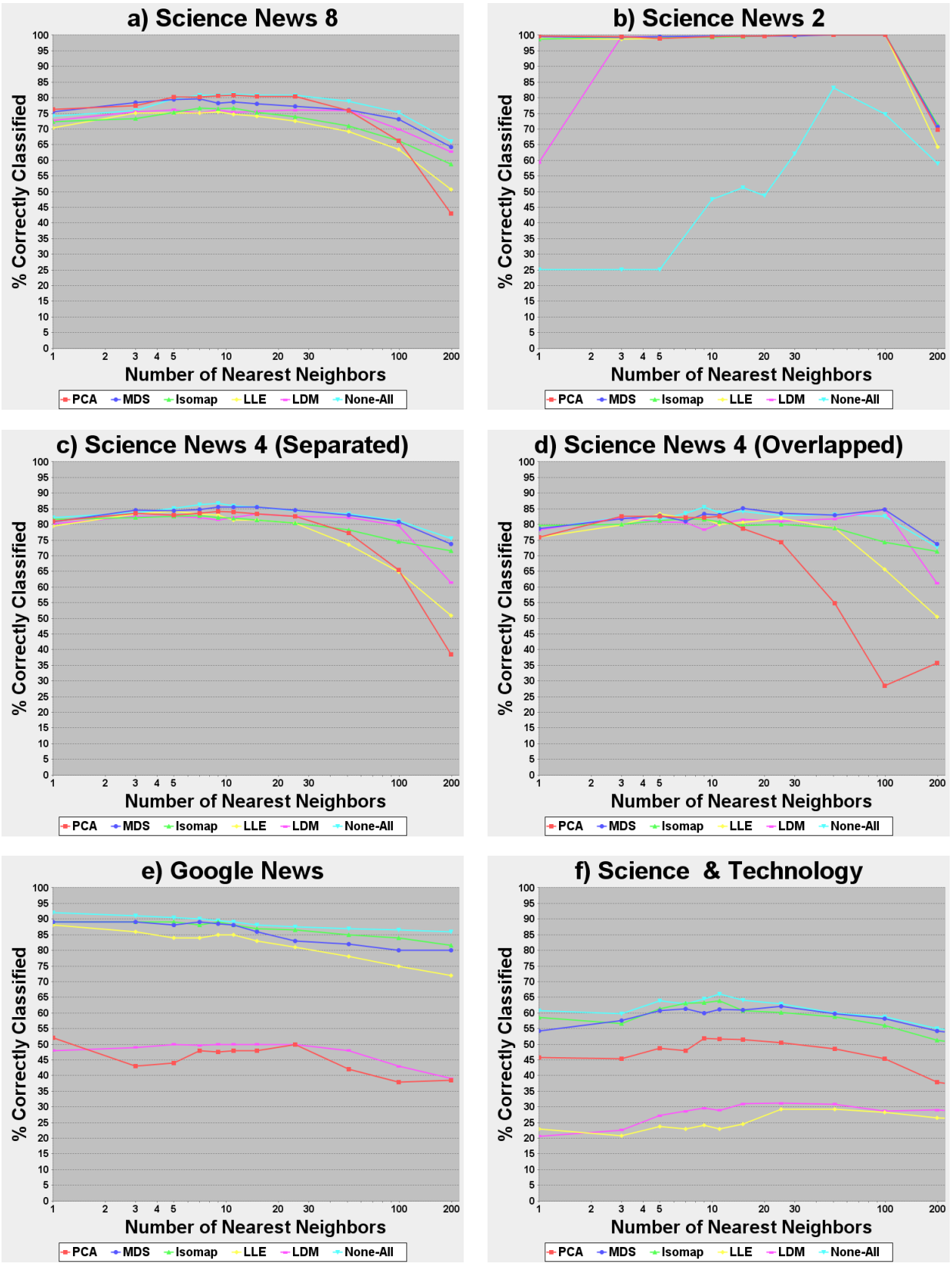

Figure 5.2: kNN: % Correctly Classified vs. Number of Nearest Neighbors (log scale), d=30

Even with just a single neighbor ($k = 1$), performance is strong. This demonstrates that the dimensionality reduction methods are closely mapping documents to other documents in their class. This performance is maintained until the number of neighbors being considered crosses some data set-dependent threshold. Beyond this threshold, adding more neighbors confuses the classifier. This occurs because so many neighbors are being considered that documents in nearby classes are being is chosen in large numbers. Performance degrades slowly as the classification accuracy regarding documents towards the "edge" of their classes erodes. As the number of neighbors being considered grows extremely large, classification accuracy drops very rapidly as classes with many documents start to be the dominant prediction. At the extreme, if the number of nearest neighbors being considered were equal to the number of documents in the training set, then the largest category would be chosen every time.

Each data set has fairly consistent results. The Science News variants (5.2a, 5.2b, 5.2c, 5.2d) start out near peak performance and improve slightly as more neighbors are considered before falling off when the number being considered grows too high. The Science & Technology data set (5.1f) follows a similar pattern, but its performance does not fall off as sharply when too many neighbors are considered. This suggests that the data set's class boundaries are not as clearly defined as those for categories in the Science News data sets. In other words, classes are already confused when k is low, and increasing k thus has less of a negative effect. PCA's and LDM's performance continues to lag as seen earlier in Figure 5.1. Finally, Google News (5.2e) somewhat maintains its plateau of good performance throughout all numbers of nearest neighbors shown on this figure. This is a result of the comparatively large categories in the data set. With an average of over 600 documents per category, it is much more tolerant of high numbers of nearest neighbors than data sets like Science News-8 (5.2a) which has only a fifth as many documents per category as Google News.

MDS and Isomap provide consistently high performance across all six data sets. They are also relatively flexible with regards to the number of nearest neighbors which are chosen in order for the kNN classifier to effectively distinguish between documents in each category. In all cases, they approach the classification accuracy achieved if dimensionality reduction is not performed. PCA performs slightly better on Science News-8, but its relatively poor

performance on the Science & Technology data set makes it less attractive than MDS or Isomap.

This section showed that varying the number of nearest neighbors only has a significant impact on performance if a particularly high number of nearest neighbors is chosen such that it is approaching the size of the average category in the corpus. Up through that point, all values work reasonably well. Furthermore, each technique is, for the most part, consistent across each data set and the best performers, with respect to the kNN classifier and varying the number of nearest neighbors, were MDS and Isomap.

5.2 Linear Classification

Figure 5.3 shows how performance of the linear classifier is affected by altering the number of dimensions. The linear classifier performs quite well once some relatively small dimension threshold is surpassed. Representations using less than 15 dimensions perform poorly, but by 30 dimensions every DR technique achieves 75% or higher accuracy on the simpler Science News variants (5.3a, 5.3b, 5.3c) and Google News (5.3e) with the exception of LDM. As the number of dimensions increases, all techniques perform increasingly better and approach perfect accuracy. On Science News-2, 99% accuracy is achieved by all techniques except LDM with just 3 dimensions, although no technique reaches 100% accuracy until the number of dimensions is in the range of 50 to 300. Similarly a number of techniques reach approximately 90% effectiveness by 50 dimensions on Science News-4 Separated, but are never able to achieve perfect accuracy, though 99% of documents are correctly classified by 500 dimensions.

The linear classifier's performance consistently improves as new dimensions are added because it is very effective at minimizing the impact of noise on its results. Furthermore, it is still able to gain a little from each new dimension. Essentially, the classifier automatically recognizes, using the training data, when a dimension does not helpfully discriminate between categories. Though the linear classier can do quite well with relatively few dimensions, its performance can almost always be improved or maintained by adding additional dimensions. However, there may be limits on exactly how far the linear classifier can expand this behavior [32]. In particular, results with None-All yielded extremely poor classification performance

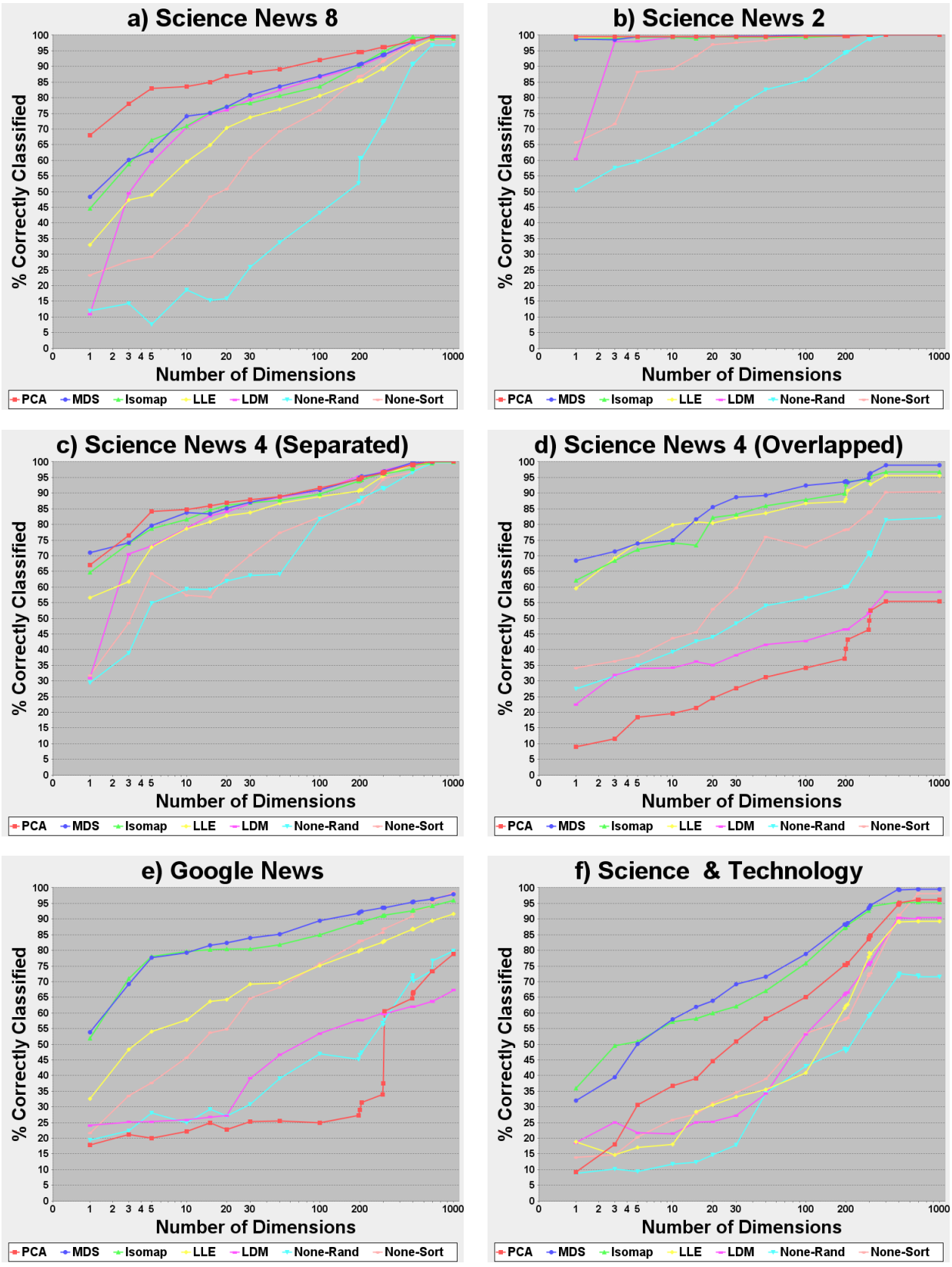

Figure 5.3: Linear: % Correctly Classified vs. Number of Dimensions (log scale)

beyond the 1,000 dimensions (on the order of 20% accuracy). There was so much noise in the greater than 10,000-dimensional space in which the unreduced data lies that the classifier was unable to obtain any useful information. If the number of dimensions is limited with the None-Rand and None-Sort techniques, performance improves. They generally perform at or below the worst DR performance when the number of dimensions is low, though on harder data sets like Science News-4 Overlapped and Science & Technology, these techniques outperformed PCA and LDM.

On Science News, no single dimensionality reduction technique stands out. PCA does perform slightly better on Science News-8, but only by about 5%, and it is approximately equivalent on the smaller Science News data sets (except Science News-4 Overlapped). On the more difficult Science & Technology and Google News data sets, MDS and Isomap outperform other techniques, though on Science & Technology even these techniques require 100 dimensions to reach 80% accuracy, and 200 dimensions to reach 90% accuracy.

Table 5.4: CM: Science News-8 reduced w/Isomap to 1 dimension – 44.6% Accurate

	Predicted Category							
Category	Anthro.	Astro.	Behavior	Earth	Life	Math	Medical	Physics
Anthropology	**33.3**	3.7	1.9	11.1	29.6	11.1	7.4	1.9
Astronomy	0.0	**84.3**	0.0	0.0	0.0	0.8	0.0	14.9
Behavior	11.1	0.0	**18.1**	2.8	15.3	2.8	50.0	0.0
Earth Sciences	10.2	21.2	8.0	**13.9**	9.5	9.5	3.6	24.1
Life Sciences	17.6	1.0	17.6	4.9	**22.4**	11.7	21.0	3.9
Math	13.3	3.3	3.3	26.7	6.7	**31.7**	0.0	15.0
Medical	1.8	0.4	17.9	0.0	8.2	0.7	**71.1**	0.0
Physics	1.7	31.4	0.0	10.2	6.8	5.9	0.8	**43.2**

Table 5.4 shows a confusion matrix for Isomap embedding Science News-8 into 1 dimension and classifying with a linear classifier. This embedding yielded a relatively low accuracy because the 1-dimensional embedding did not contain enough information for the linear classifier to separate all eight categories. This confusion matrix shows a lot of general confusion resulting from the extreme reduction of the data to just one dimension. However, some interesting relationships are still evident in the midst of this general confusion. The Astronomy and Medical categories both do relatively well and are the only categories to achieve 50% accuracy. Their success can be attributed to the relative isolation of the former,

and the sheer size of the latter (Medical contains over a quarter of the total articles in the corpus and so there is more information about what documents in its category looks like).

Even more interesting, several relatively focused areas of confusion present themselves. Medicine and Behavior are particularly confused (respectively 17.9% and 50.0%), as well as Physics and Astronomy (31.4%, 14.9%). These two confusions are the only significant mistakes made for the two categories which did particularly well overall and can be attributed to significant overlap with these fields. Overlap is also the probable cause for other notable confusions occurring between Anthropology and Life Sciences (29.6%, 17.6%), Physics and Earth Sciences (10.2%, 24.1%), and Behavior and Life Sciences (15.3%, 17.6%). On the other hand, categories for which overlap is improbable have very little confusion between them. Examples of this include Astronomy and Behavior (1.9%, 0.0%), Astronomy and Life Sciences (0.0%, 1.0%), and Medicine and Physics (0.0%, 0.8%).

Table 5.5: CM: Science News-8 reduced w/Isomap to 200 dimensions – 90.5% Accurate

	Predicted Category							
Category	Anthro.	Astro.	Behavior	Earth	Life	Math	Medical	Physics
Anthropology	**87.0**	1.9	5.6	1.9	1.9	1.9	0.0	0.0
Astronomy	0.0	**95.9**	0.0	0.8	0.0	0.0	0.0	3.3
Behavior	1.4	0.0	**93.1**	0.0	2.8	0.0	1.4	1.4
Earth Sciences	1.5	0.7	0.0	**89.1**	7.3	0.0	0.7	0.7
Life Sciences	2.0	2.0	1.5	2.9	**82.9**	0.0	7.3	1.5
Math	1.7	0.0	0.0	3.3	3.3	**91.7**	0.0	0.0
Medical	0.4	0.4	1.4	0.4	4.3	0.0	**93.2**	0.0
Physics	0.0	0.8	0.8	2.5	0.8	0.0	1.7	**93.2**

Table 5.5 shows a confusion matrix for Isomap embedding Science News-8 into 200 dimensions and classifying with a linear classifier. This embedding yielded a much higher accuracy than with the 1-dimensional embedding shown in Table 5.4. This confusion matrix shows very similar confusions, albeit in much smaller quantities. The Astronomy and Medical categories are still the most accurately classified categories, though the other categories have made more significant improvements.

Categories which were being confused before are still being confused, but all of the ones which used to have the worst confusion have been addressed. This is consistent with the linear classifier's performance trend in which the biggest difference is made early on, and

smaller and smaller gains are made by adding additional dimensions. Confusions which used to be significant includes Anthropology and Life Sciences (now just 1.9%, 2.0%), Physics and Earth Sciences (2.5%, 0.7%), and Behavior and Life Sciences (2.8%, 1.5%). Furthermore, categories for which overlap is improbable still have very little confusion between them, as expected. These pairs included Astronomy and Behavior (0.0%, 0.0%), Astronomy and Life Sciences (0.0%, 2.0%), and Medicine and Physics (0.0%, 1.7%). Despite this enormous reduction in confusion and successful maintainence of well-separated categories, some other confusion persists. Though not the most significant source of confusion in the 1-dimensional reduction, Earth Sciences and Life Sciences (10.2%, 6.8%) and Life Sciences and Medical (11.2%, 8.2%) now exhibit the most significant sources of confusion. Further increasing the number of dimensions to five to six-hundred greatly reduces the remaining confusion so that the overall accuracy goes to 99%, as seen in Figure 5.3.

This section showed that the linear classifier consistently achieves high accuracy on all data sets without being overly sensitive to the number of dimensions used. Beyond just a few dimensions, increasing the number of dimensions in small quantities has a small impact on performance. All techniques follow this behavior and, for the most part, are consistent across each data set. It was also shown that dimensionality reduction is a prerequisite before the linear classifier is capable of accurately classifying this kind of data because it performs poorly when the input is extremely large. The best performers, with respect to the linear classifier, were MDS and Isomap.

5.3 Quadratic Classification

Figure 5.4 shows how performance of the quadratic classifier is affected by altering the number of dimensions. Similar to the linear classifier, the quadratic classifier performs quite well once some relatively small dimension threshold is surpassed. In theory it should perform better than the linear classifier, and in some ways it does - it reaches over 95% accuracy on Science News with approximately 100 dimensions while it takes nearly 5 to 10 times as many dimensions for the linear classifier to exhibit the same performance. Very small representations which use fewer than approximately 5 or 10 dimensions perform poorly, but

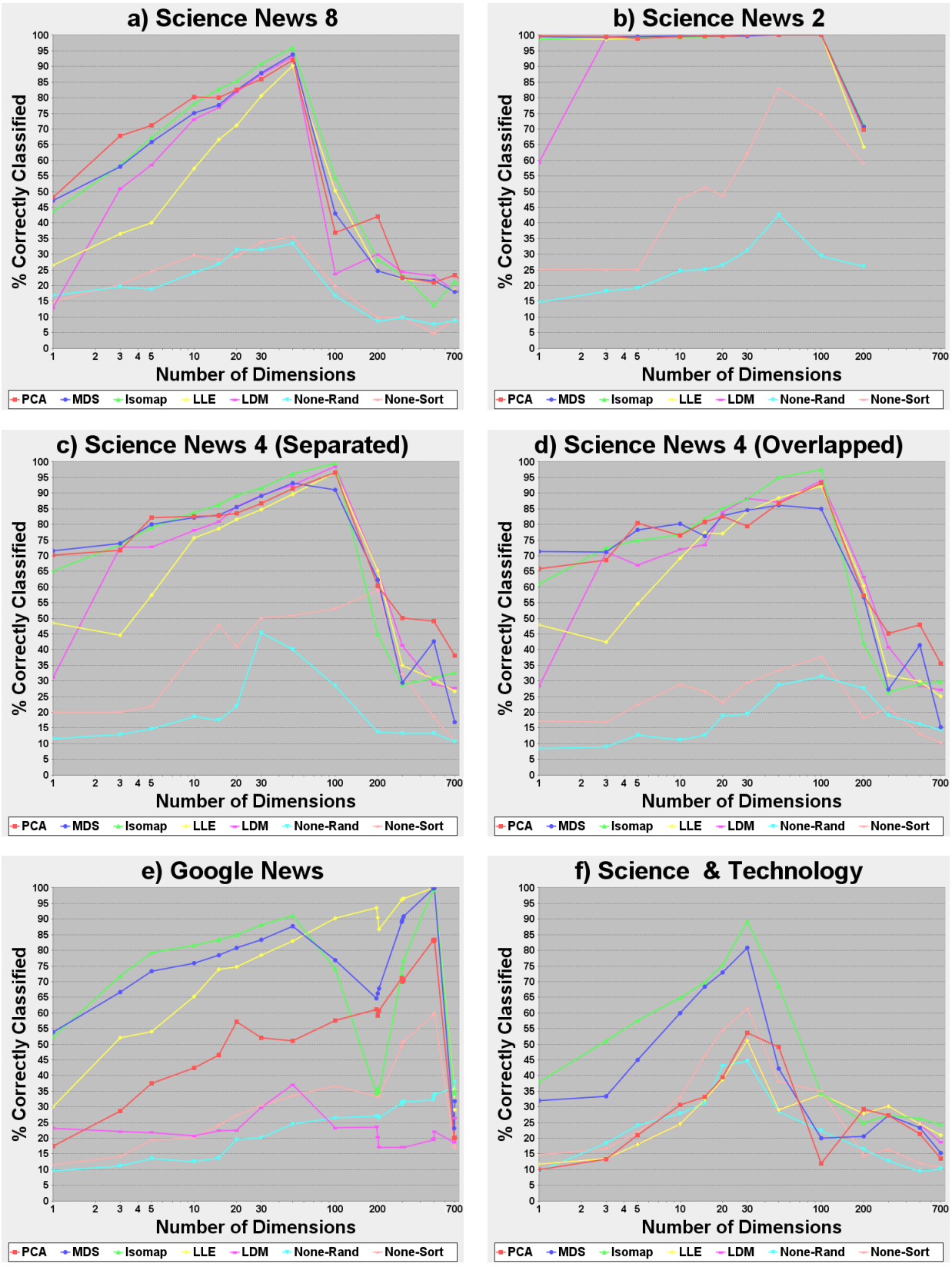

Figure 5.4: Quadratic: % Correctly Classified vs. Number of Dimensions (log scale)

in slightly higher dimensions every technique achieves 80% or higher accuracy on the simple Science News variants (5.4a, 5.4b, 5.4c) and Google News (5.4e). Performance peaks much earlier than the linear classifier, but performance degrades much faster too.

This degradation in performance is related to the complexity of the quadratic classifier. As noise increases in higher dimensions, the quadratic classifier performance suffers greatly. One potential source of this error is a computational issue. With many dimensions, the class-specific covariance matrices (see Section 3.3) have less and less variance which causes the quadratic classifier to encounter numerical issues when inverting determinants of those matrices. This leads to instability and poor class likelihood predictors. A second problem is that the quadratic classifier needs a larger training set to adequately learn its larger set of parameters. Larger corpuses address both of these problems. This is primarily evident in Google News which despite one significant dip, flows cleanly from one point to the next and even peaks at 500 dimensions.

Much like the linear classifier, no single dimensionality reduction technique stands out on Science News. Isomap is consistently the best performer on all data sets, though it separation from less effective techniques is extremely small. LLE stands out on Google News because it has a barely perceptible dip, when compared to the degradation in performance suffered by MDS and Isomap, as the number of dimensions increase until its peak performance. By approximately 500 dimensions, all techniques have reached peak performance, though performance drops quickly to below 40% as the next 200 dimensions are added.

The quadratic classifier achieves the strongest performance in the fewest number of dimensions, but it is extremely sensitive to the number of dimensions used, making it more difficult to use. Inside its optimal performance window, Isomap and MDS consistently outperform other techniques. On the tougher Science & Technology and Google News data sets, the classifier is more confused by reductions produced by SVD-based algorithms like PCA and LDM. In contrast, eigenvalue decomposition techniques like MDS, Isomap, and LLE all achieve comparable results on each data set except Science & Technology where LLE flounders some. Overall, Isomap has the best and most consistent performance across all data sets with the quadratic classifier.

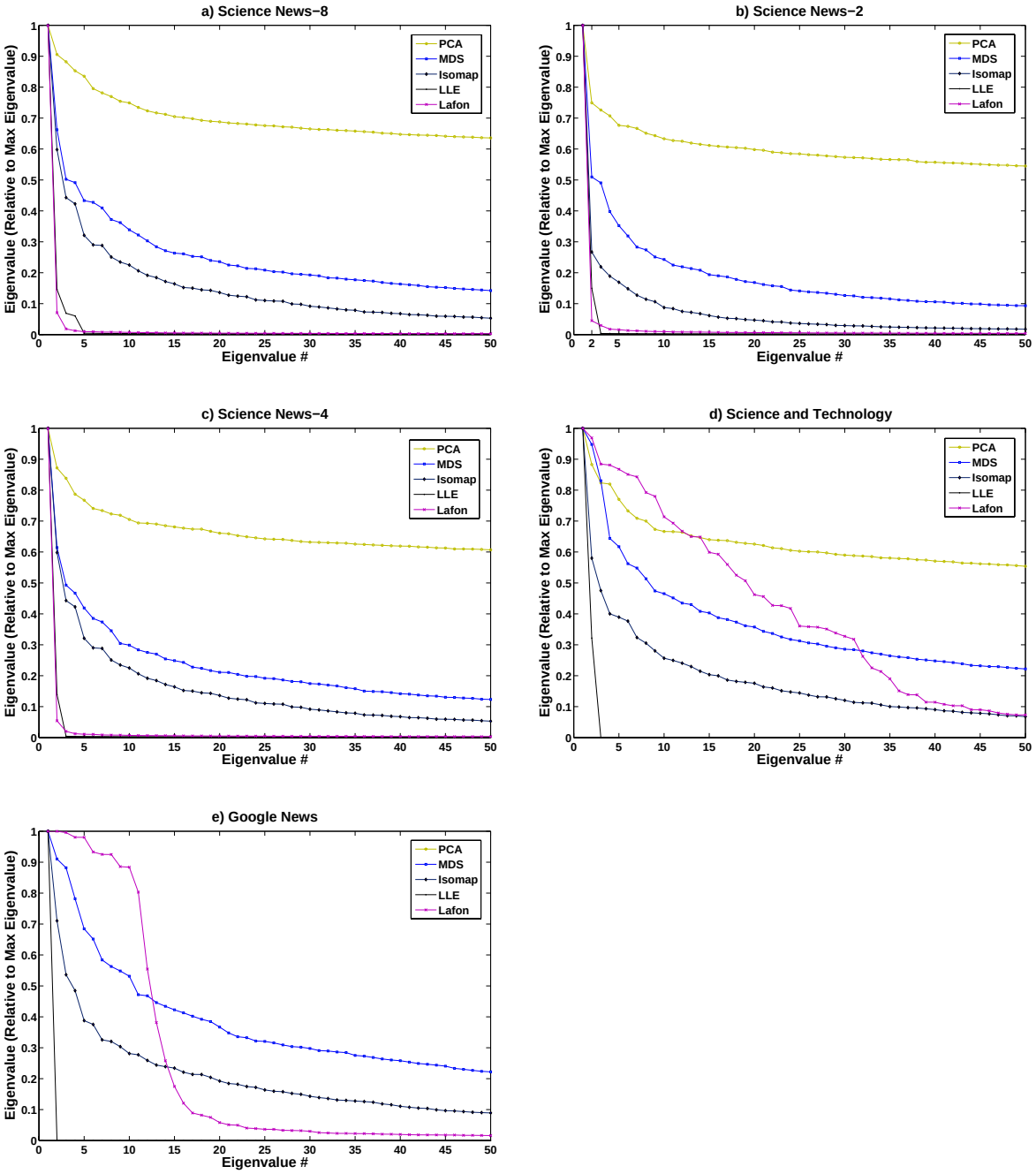

Figure 5.5: Eigenvalues Relative Value Plot

5.4 Supporting Evidence

5.4.1 Eigenvalues Plot

Figure 5.5 shows the relative values of the eigenvalues obtained by each dimensionality reduction technique on each data set. An eigenvalue plot shows the relative value of each

eigenvalue. The y-axis is the value of the eigenvalue relative to the biggest eigenvalue. The x-axis is the number of the eigenvalue and the y-axis is the value of the eigenvalue relative to the max eigenvalue (thus ranging from 1 to the last scaled eigenvalue). Therefore, the x-axis ranges from 1 to the last eigenvalue. The eigenvalues arranged in decreasing order of importance. An eigenvalue plot typically looks like an exponential decay. This is because the first eigenvalue yields the greatest amount of information and each successive eigenvalue yields less and less information. This exponential decay forms an "elbow" in the graph. Typically, the elbow is the location at which future eigenvalues do not hold much more useful information.

The first interesting thing to note is that the Science News-2's eigenvalue plots have their elbow pretty clearly at 2 dimensions while elbows on the eigenvalue plots for Science News-8 are somewhat less distinct. This is reasonable because Science News-8 has more than twice as many articles in four times as many categories. Therefore, it makes sense that it would probably take more dimensions to accurately represent the data. When compared to Google News, the elbow is even larger and less distinct, though in practice these experiments have found that a fairly aggressive (lower number of dimensions) dimensionality choice on the Google News eigenvalue plot elbow would be sufficient. Figures 5.1 and 5.3 show that Google News and Science News peak at approximately the same number of dimensions when reducing data for kNN and linear classification. However, the quadratic classifier shown in Figure 5.4 requires slightly more dimensions for peak performance, but using the same number of dimensions would yield only a small difference in overall accuracy.

Like Google News, Science & Technology also has fairly non-distinct elbows. However, in this case what is really interesting is the eigenvalue plot for LDM. LDM has a somewhat linear plot instead of the usual exponential decay. This indicates that LDM is unable to find a meaningful eigenvalue decomposition with which it can produce an effective embedding with. This is probably a major factor in why LDM has such low performance on the Science & Technology data set.

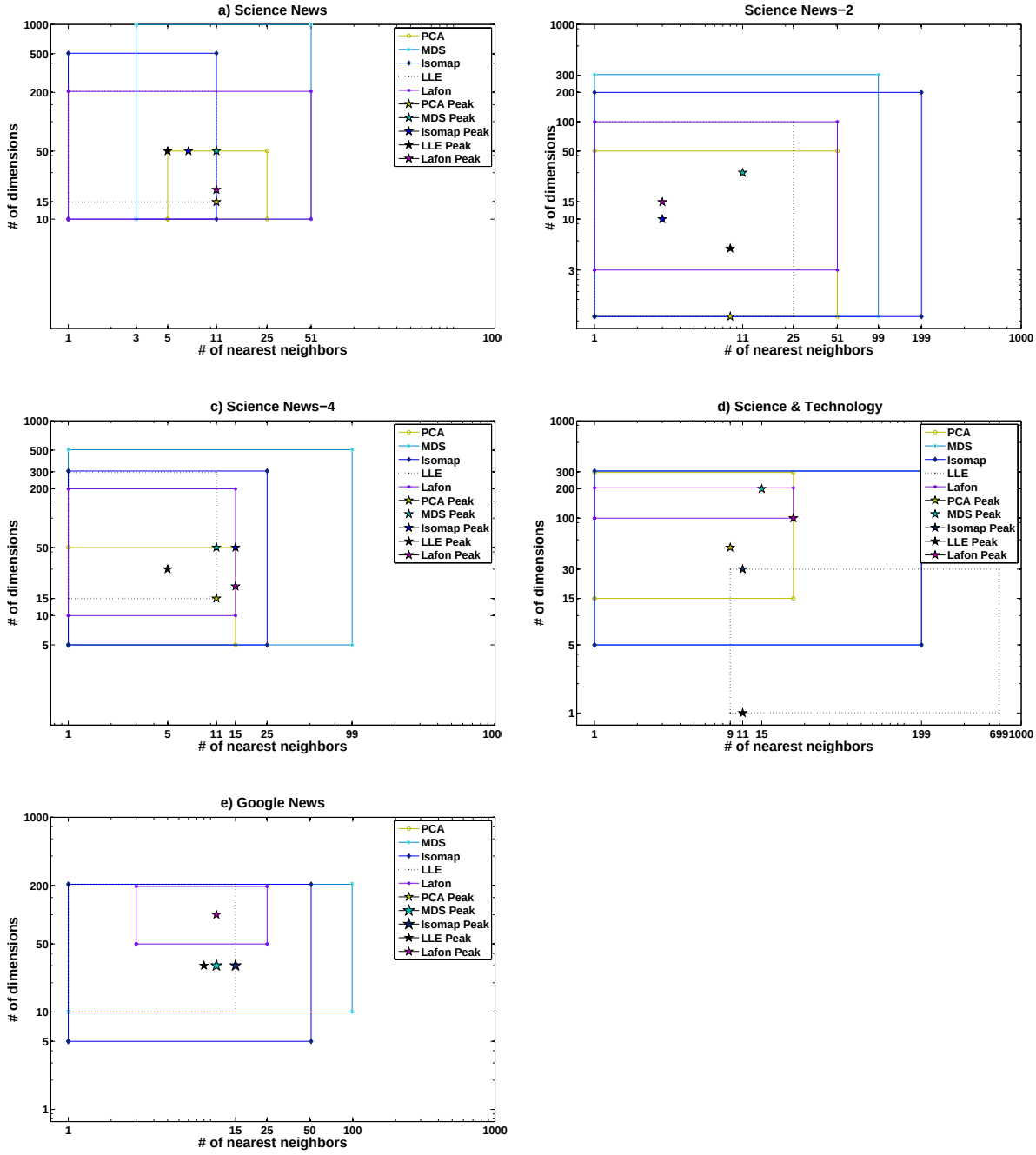

Figure 5.6: kNN: % Effective Boundaries

5.4.2 kNN Effective Boundaries

Figure 5.6 shows the regions of in which two key parameters for kNN (the number of nearest neighbors and the number of dimensions) can be varied without significantly deviating from the maximum performance. A significant deviation is considered to be more than 10% away from the peak performance achieved by each dimensionality reduction technique. The pairing of parameters at which peak performance occurred is labeled as well for each dimensionality reduction technique. This graph shows that these dimensionality reduction techniques are relatively insensitive to parameter tuning on these data sets when performing classification with kNN. It also shows that MDS, and to a lesser extent Isomap, perform well under an especially large range of parameter specifications.

5.4.3 Tuning Parameter Choice for Isomap and LLE

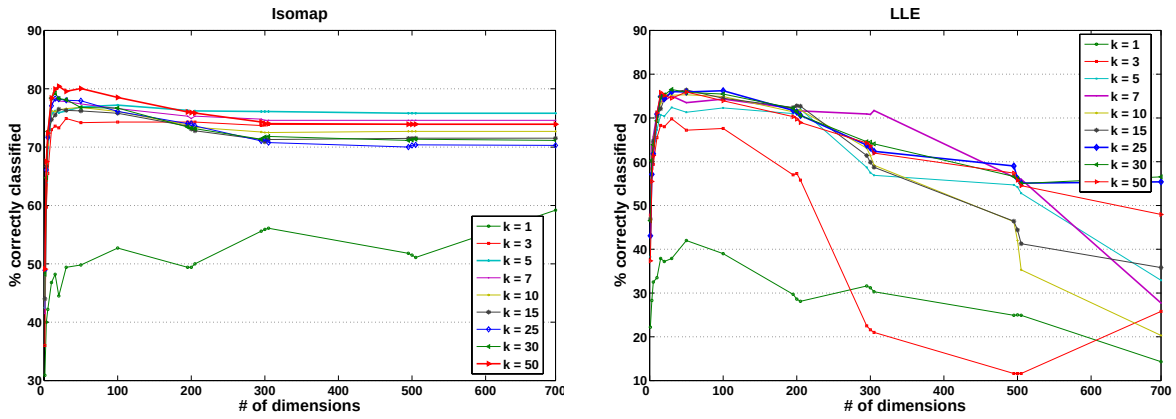

Figure 5.7: Varying DR Tuning Parameter (k=9, d=30)

Figure 5.7 shows how dimensionality reduction techniques which require a tuning parameter react to changes in that parameter. The two techniques being considered in this research are Isomap and LLE. These tuning parameters are explained in Section 3.2.2.

For both techniques, the tuning parameter had only a very small effect on the outcome, though generally the higher the number of neighbors considered (the tuning parameter), the more stable and slightly better performance was achieved. Choosing only a single neighbor had poor performance, but increasing to just three gained the vast majority of the perfor-

mance difference between the worst performing value of 1 and the best performing value. Based on these results, all experiments in this paper which use Isomap or LLE have their tuning parameter fixed at a value of 10 unless otherwise noted.

5.4.4 Data Set Visualization

Intuition helps one reason about which categories may or may not overlap with others. This section helps visually describe the relationships between the documents in different categories. This is done by performing dimensionality reduction on the corpus and projecting the data into a 2-dimensional space. The documents in this 2-dimensional space are then graphed in a scatterplot. Each document is marked with some marker which identifies what category it belongs to. These scatterplots show how the two most differentiating dimensions for each dimensionality reduction technique on each data set separate the data.

Figure 5.8 shows the 2-dimensional embeddings for each dimensionality reduction technique applied to Science News-2 which consists of two well-separated categories: astronomy and medical. Each technique is able to produce a 2-dimensional embedding which clearly distinguishes the majority of documents in each class. Most points are clustered near their own class, though the ones on the boundary and very near the other category's points is probably a minor source of error in classification experiments on this data set.

Figure 5.9 shows the 2-dimensional embeddings for each dimensionality reduction technique applied to Science News-4. Astronomy and Medical are still far apart, which is expected since Figure 5.8 showed that the could be well-separated. Life Sciences is sandwiched in between Medical and Earth. with results in the confusion matrix for MDS mapping Science News-4 to 3 dimensions for the kNN classifier.

The separation of Life Sciences from Astronomy is consistent with the results for that confusion matrix as well. Furthermore, the small number of mistakes between Astronomy and Earth are likely the result of the small number of Earth documents which can be seen mixing in with the bottom of the Astronomy section in the MDS portion of the figure. The number of mistakes between Medical and Life Sciences was greater, and this is validated by the more significant overlap of those categories on this scatterplot. Finally, Earth Sciences and Life Sciences had significant confusion and this confusion seems reasonable given that MDS

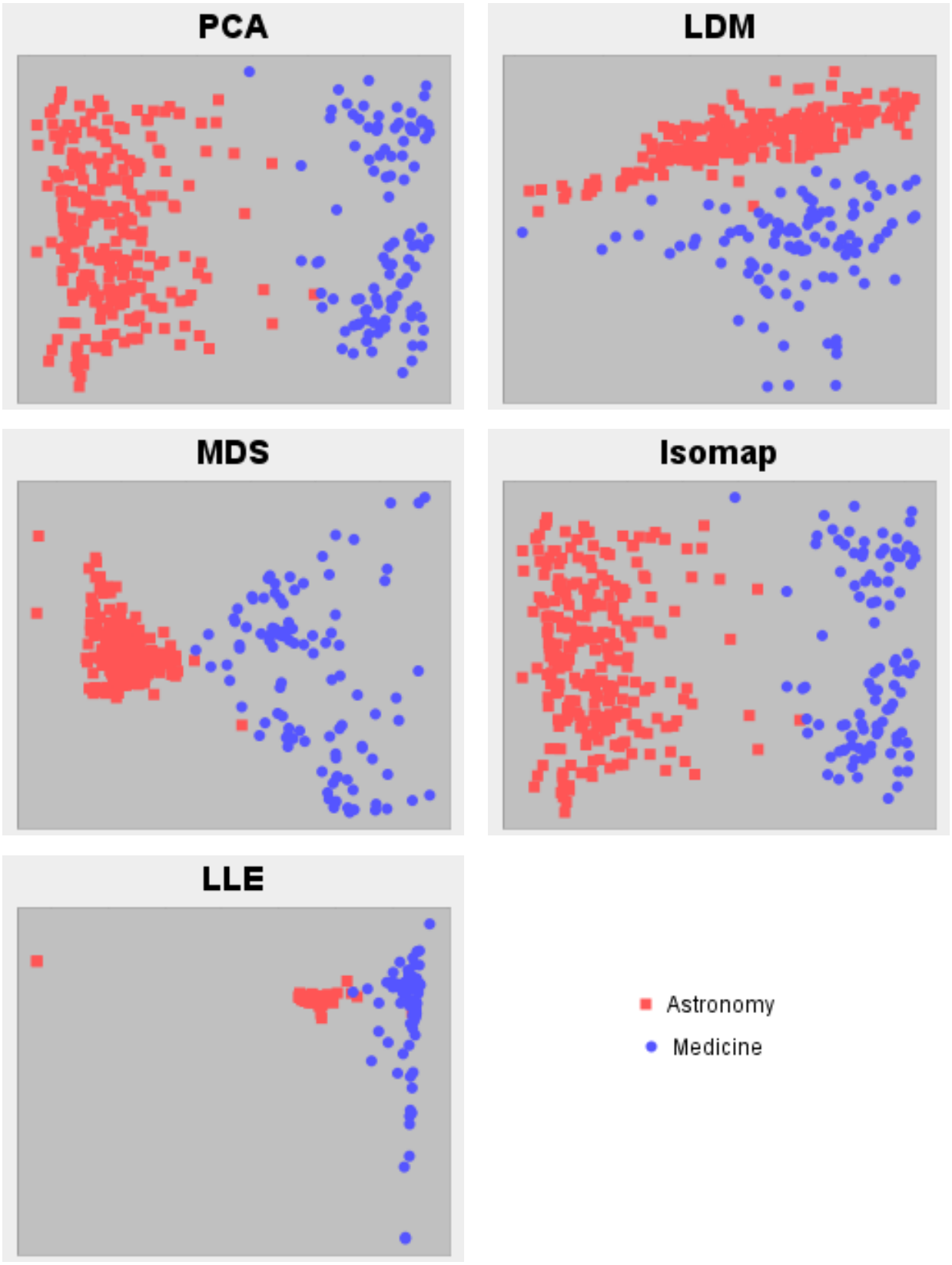

Figure 5.8: Science News-2: 2-D Embeddings

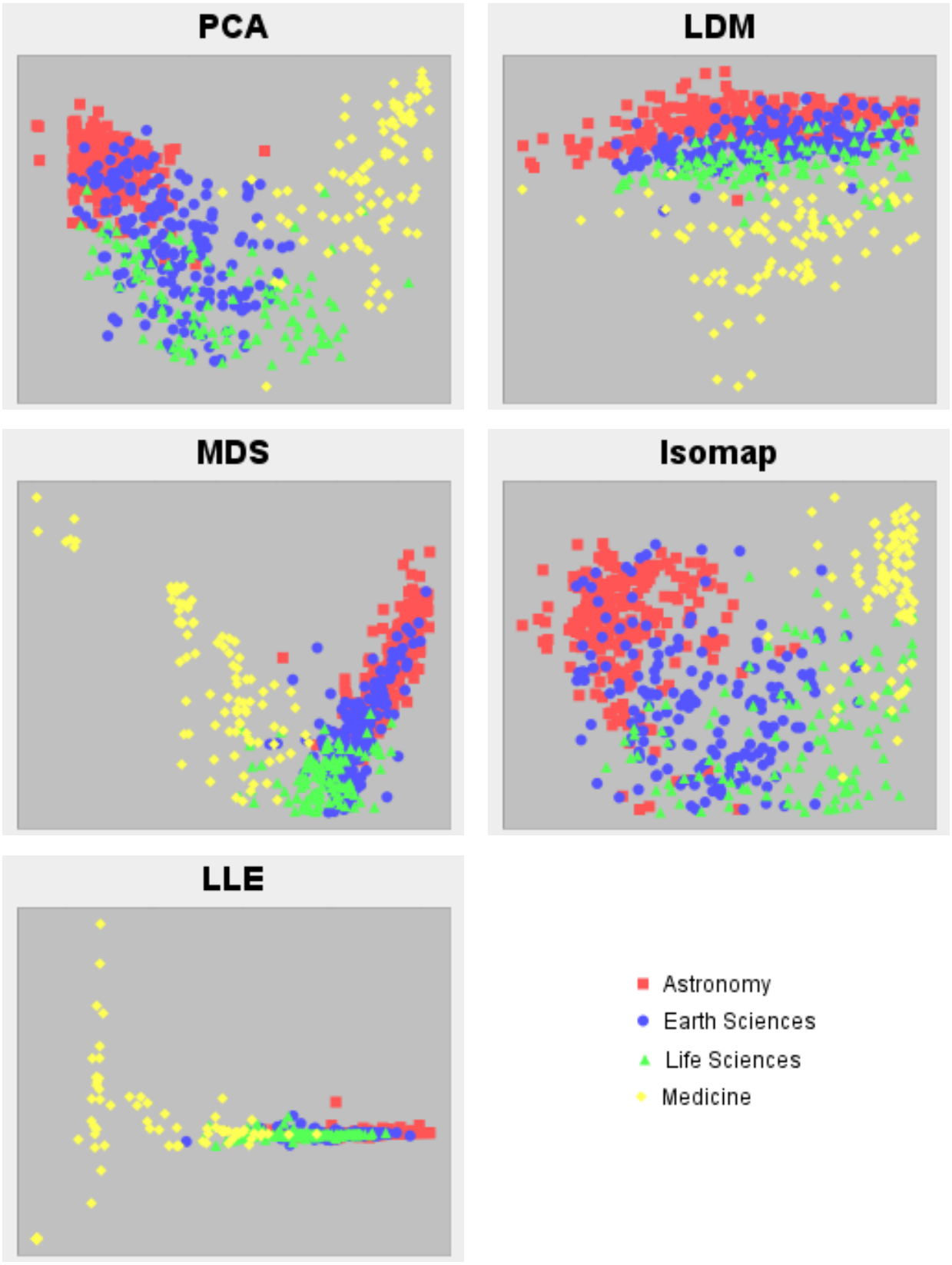

Figure 5.9: Science News-4 Separated: 2-D Embeddings

mapped the two categories practically on top of each over. MDS does do quite well as the number of dimensions increase because though these categories are not completely separated in two dimensions, the new dimensions introduce new ways to differentiate documents from different categories which generally improves performance as seen in Figure 5.1.

Figure 5.10 shows the 2-dimensional embeddings for each dimensionality reduction technique applied to Science News-8. Though only in 2-dimensions, the scatterplot is rather busy due to the relatively large number of categories in Science News-8. This representation of the data set helps explain the confusion encountered when performing linear classification on Isomap's reduction of this data set. This confusion is described in Table 5.4 for the case where Isomap embedded the data in a single dimension and in confusion matrix 5.5 for the case when Isomap embedded the data in 200 dimensions.

Astronomy and Medical are still very well separated, which is expected since figures 5.8 and 5.9 showed that this fact is true. Astronomy and Medical, the two best performing categories on this input, were confused over Physics and Behavior. This confusion is also present in the scatterplot - Behavior articles are present around various parts of the Medical articles, and the same with the Physics and Astronomy articles. The significant confusion between Anthropology and Life Sciences is backed by the scatterplot which shows the Anthropology to be right on top of many Life Sciences articles - this same phenomenon is occurring in other categories which experienced similar problems with the classification of the 1-dimensional Science News-8. This overlap in 2-dimensions indicates that 2 dimensions are not sufficient for representing this data set with the dimensionality reduction techniques utilized by this research.

Figure 5.11 shows the 2-dimensional embeddings for each dimensionality reduction technique applied to Google News. This scatterplot has very dense groups of points thanks to the sheer size of the Google News data set. This representation of the data set helps explain the confusion encountered when performing classification on MDS's reduction of this data set with a kNN classifier. This confusion is described in confusion matrix 5.1 for the case where MDS embedded the data in a single dimension.

Having just one dimension greatly complicated the classifier's job because many of the points overlap even more if just the x-axis is being used to distinguish categories. With this

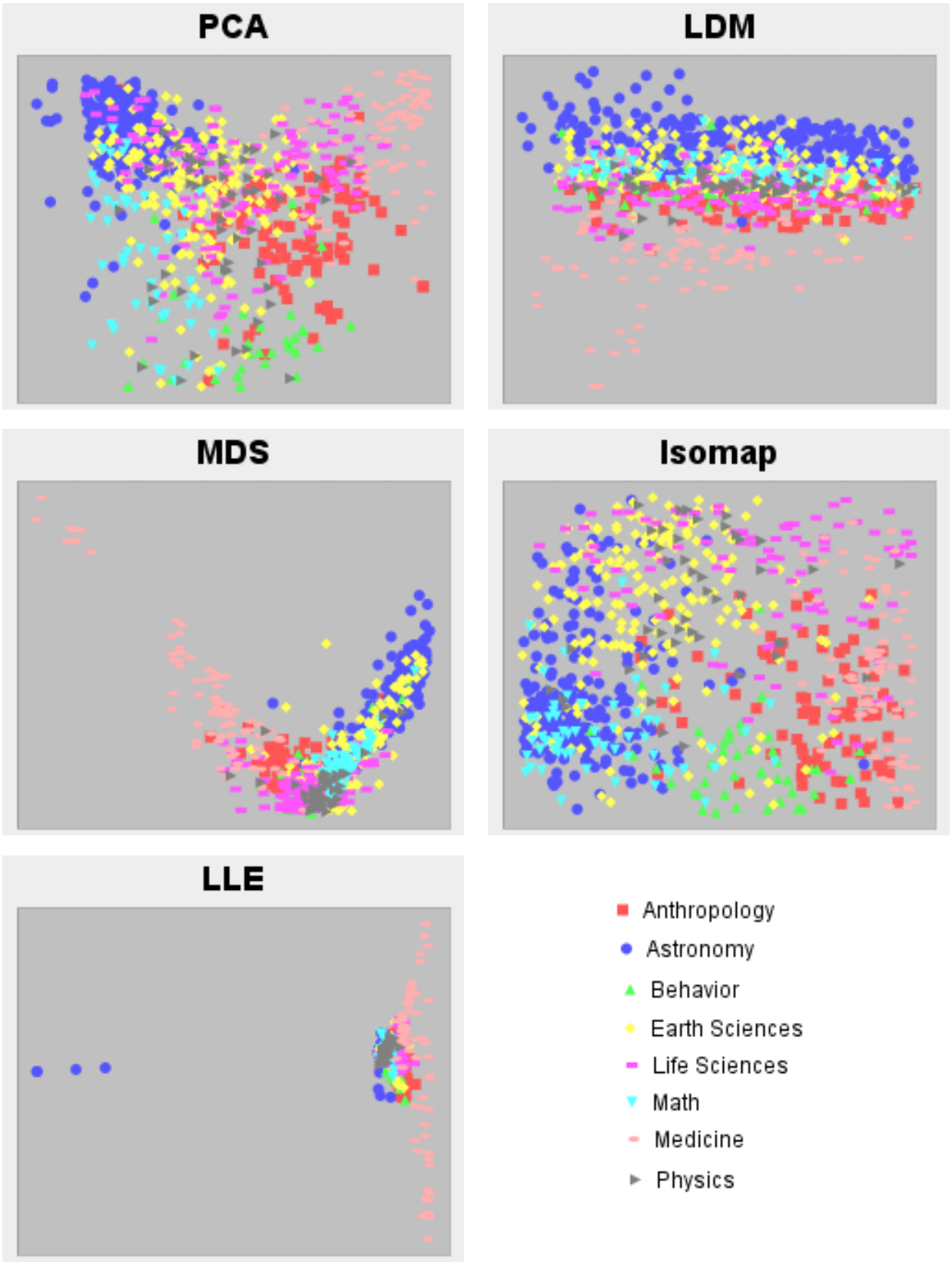

Figure 5.10: Science News-8: 2-D Embeddings

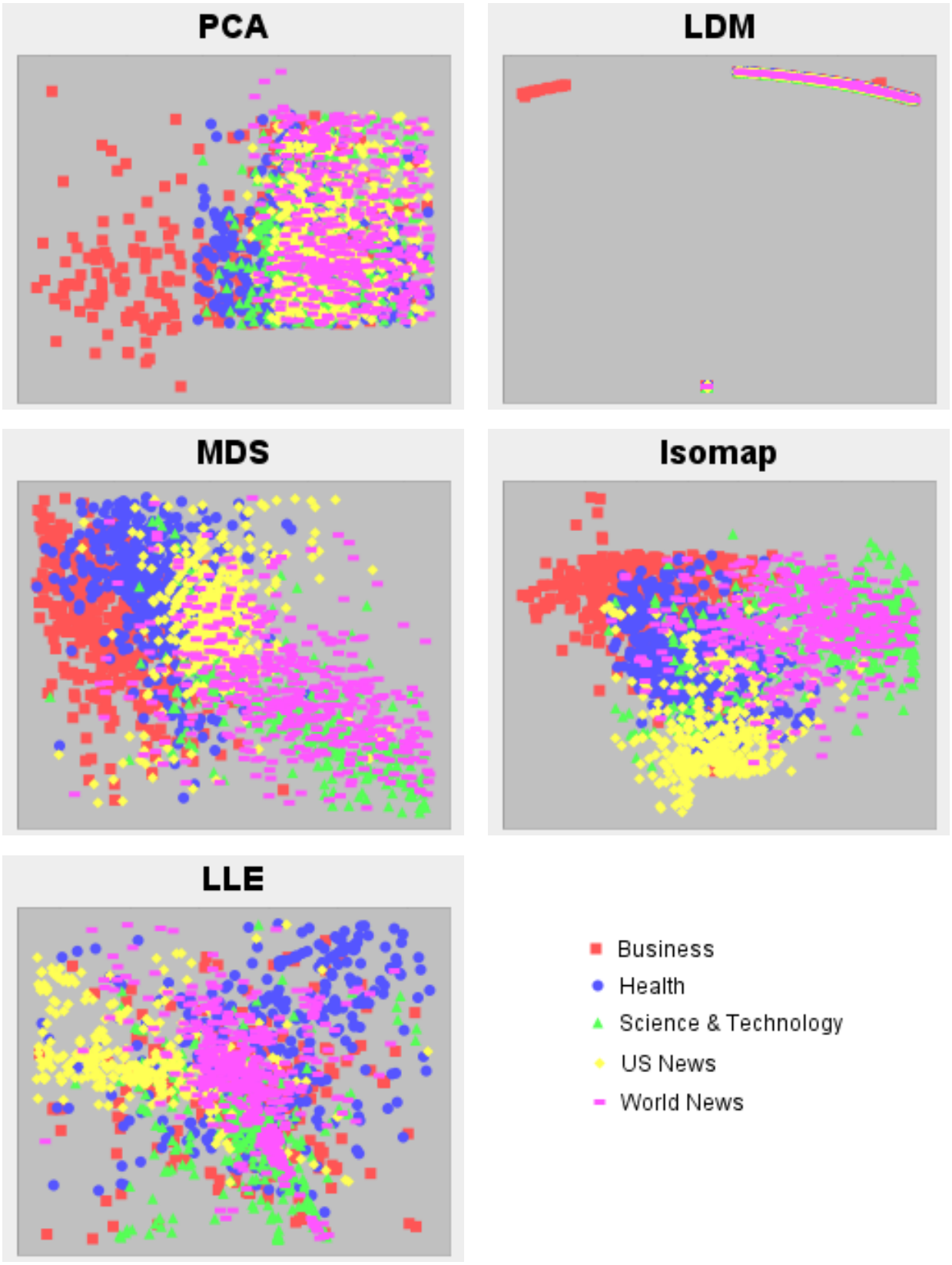

Figure 5.11: Google News: 2-D Embeddings

is mind, it seems quite reasonable that the classifier had a good deal of trouble distinguishing US and World News. Business and Science & Technology are practically on top of each other so this contributes to the confusion between these two. Furthermore, Health articles are also in the midst of the Business and News section which causes it to be miscategorized under all of those categories.

Figure 5.12 shows the 2-dimensional embeddings for each dimensionality reduction technique applied to Science & Technology. This scatterplot is interesting because it is much less clear where class boundaries lie than with any other data set. This difficulty to distinguish much in these dimensions hints that in this data set the categories are not as clearly separated as the others.

MDS and Isomap have some structure, but the points are still pretty loosely organized and very frequently mix with other groups. PCA also has some structure, and the Anti-Submarine class is pretty tightly packed. LLE may have some structure, but it is very difficult to discern exactly what that might be in these two dimensions. This is probably why LLE performs rather poorly with the kNN classifier as seen in figures 5.1 and 5.2. Finally, LDM spreads the documents out along a semi-circle shape, but unfortunately all of the documents seem to be spread without a clear distinction as far as their category. This may indicate that LDM is not able to effectively differentiate between documents in separate classes. This is apparent in results with each classifier on this data set because LDM's routinely performs the worse than other techniques on this data set.

This section provided visual evidence regarding the ability of each dimensionality reduction technique to separate classes. The visualizations reinforce reasoning about why Science News-2 was easy for the classifiers to achieve high classification accuracy on and why others like Science & Technology were much more difficult. These visualizations also helped validate which categories are more difficult to distinguish from one another.

5.5 Analysis

For a given number of dimensions, applying dimensionality reduction significantly improved accuracy versus not applying DR. Furthermore, the best DR techniques were able to achieve

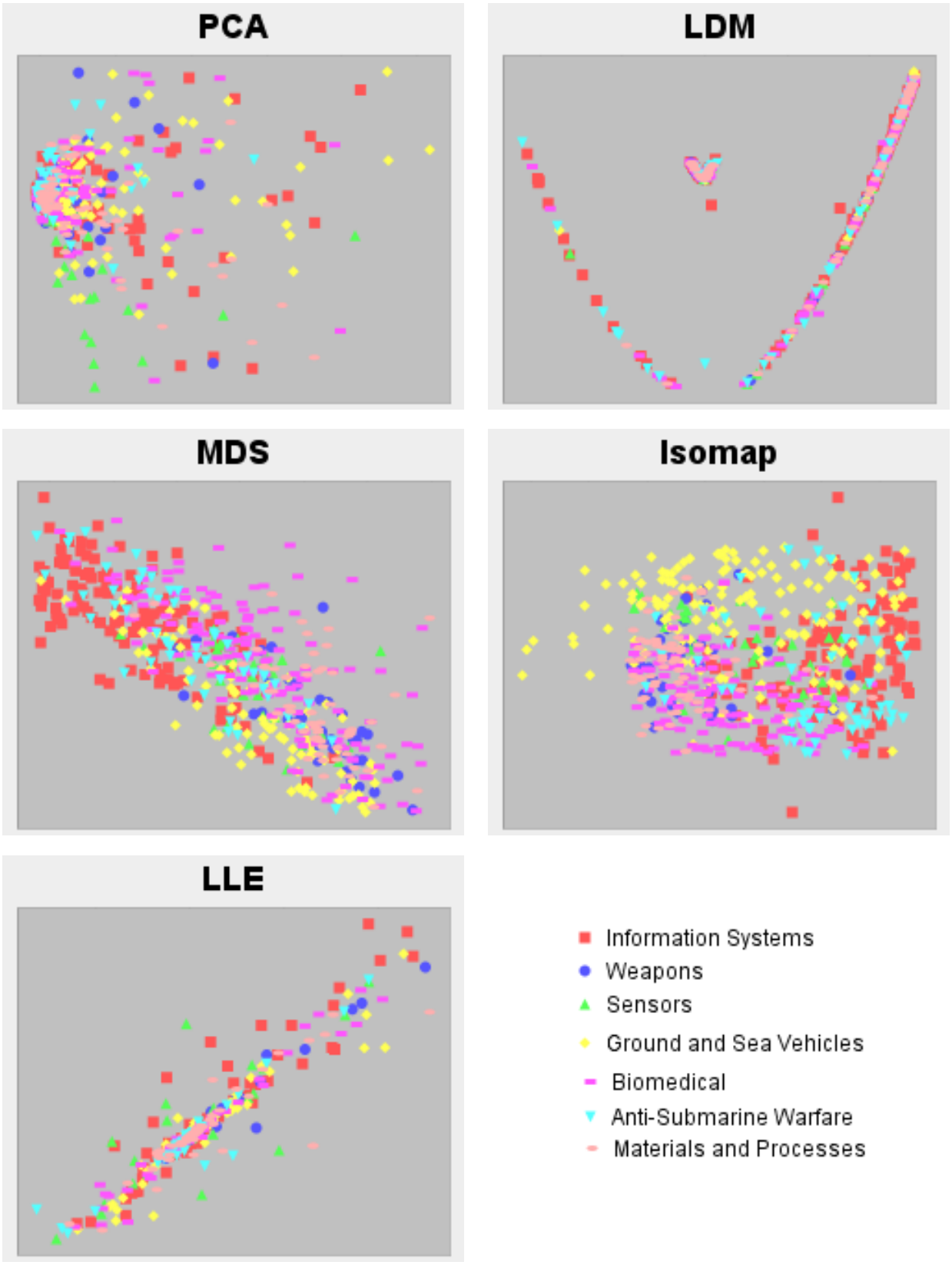

Figure 5.12: Science & Technology: 2-D Embeddings

a high degree of accuracy in just a few dimensions. MDS and Isomap were consistently the best and most reliable of all of the techniques. Interestingly, their advantage is even more pronounced on more difficult corpuses such as Science & Technology and Science News-4 Overlapping.

5.5.1 Impact of the Dimensionality Reduction Technique

Every dimensionality reduction technique gave good classification performance in most cases. MDS and Isomap were the most reliable dimensionality reduction techniques across all five data sets. They came very close to the performance of just keeping all dimensions with kNN (e.g. no dimensionality reduction) which is impressive since in most cases this level of performance is reached with less than 50 dimensions. In addition, for both the kNN and linear classifiers, the DR techniques typically outperformed the None-Rand and None-Sort techniques with the same number of dimensions. Performing no DR and keeping all dimensions with the linear classifier produced extremely poor performance as described in Section 5.2.

All dimensionality reduction techniques were also fairly insensitive to the number of dimensions being projected into as well as the number of nearest neighbors for the kNN classifier. This is illustrated by Figure 5.6 which shows that all techniques have strong classification performance across a large spread of parameter values. MDS and Isomap have the largest area in which they perform well.

MDS and Isomap also exhibit the best performance with a very low number of dimensions (in the one to five dimension range). The only exception to this is with the linear classifier. On Science News PCA does the best, especially on Science News-8, but overall it is less consistent than Isomap and MDS. Results with LDM were also interesting but for the opposite reason - it had been expected to perform quite well but basically failed on more difficult data sets.

5.5.2 Impact of the Classifier

All three classifiers were able to obtain good classification accuracies under the right conditions. The linear classifier had the best performance across all five data sets. However, to reach its peak performance on all data sets other than Science News-2, it had to use over approximately 500 dimensions. When combined with DR, it was very consistent: all data sets appeared to approach nearly 100% as the number of dimensions grew to the maximum number which the data could be projected into. Performance with the linear classifier usually matches the peak performance of the kNN classifier by approximately 20 to 30 dimensions, though on Google News it requires more dimensions to match kNN's peak performance than kNN needed (however, it still eventually meets that peak performance and exceeds it as more dimensions are added).

The quadratic classifier is also able to exceed most of kNN's peak performances in fewer dimensions than kNN needs to accomplish those peaks. It also reaches its peak performance around 100 dimensions - much fewer than the the 500 needed by the linear classifier for its peak performance. The peak performances between the linear and quadratic are fairly similar, though unfortunately the quadratic classifier is much more sensitive with regards to the number of dimensions. Furthermore, the quadratic classifier is less reliable for some DR techniques. Overall, amongst the three classifiers considered, the linear classifier appears to be the best all-around technique for classification in this domain.

Chapter 6

LBD Results

This chapter examines the quality of results obtained from performing LBD (as described in Section 3.5) after reducing the data using some dimensionality reduction technique. To do so, Section 6.1 first evaluates and calibrates the experimental procedure for automatically scoring a candidate association between two documents. Section 6.2 then uses this scoring metric to measure the impact of dimensionality reduction on the associations that are found. Finally, Section 6.3 summarizes and analyzes the results.

Each dimensionality reduction technique finds pairs of documents from different categories. These pairs are identified based on which documents are closest together in terms of Euclidean distance in the dimensionally reduced matrix. Each technique finds identifies between 10, 20, 50 and 100 closest pairs. The details of this process were previously presented in Section 3.5.

Once pairs are identified, they are scored using the keywords identified by the process explained in Section 3.5.4. A higher score indicates that there is a stronger connection between the documents. This process was previously presented in Section 3.5.4.

The median score from each pair count (10, 20, 50, and 100) is then averaged and used as a benchmark for analyzing each dimensionality reduction techniques effectiveness. To compare effectiveness across different levels of reduction, the number of dimensions which the data is reduced to is varied from 5 to 71.

6.1 Validating and Calibrating Automated Scoring

Ideally, a candidate association between two documents would be evaluated for relevance and novelty by human participants in a double blind study. Since such a study was not possible in the context of this work, LBD quality for one document pair is instead measured by extracting keywords from the two documents (Section 3.4) and then using those keywords to query Google and compute a novelty score from the results (Section 3.5). Both of these processes are error-prone, and mistakes in either may influence the overall LBD assessment. Consequently, the next two sections evaluate each of these two steps, identify initial problems, and develop solutions that are used in the final LBD analysis.

6.1.1 Keyword Extraction Assessment

Descriptive keywords are essential to the LBD process presented by this research. Keyword extraction is performed as specified in Section 3.4. The process is demonstrated with the following example.

This assessment focuses on the results from a document entitled *Olestra: Too good to be true? Researchers flush out health risks of fake fat* by Kathleen Fackelmann from the Science News corpus. This article has been included in the appendix (see Appendix C.2). Similar results were found with other articles.

Naive Version

The naive version of the keyword extraction process identified the most highly weighted terms, but a number of the top terms for each document were often proper nouns. Such proper nouns are not particularly helpful for finding related content since they were often names of people, companies, or places rather than general content descriptors. For instance, applying the naive keyword extractor to the Olestra example found many highly ranked proper nouns among the top 10 keywords including “Proctor,” “Gamble,” and “FDA,” all of which appeared in the top 5 keywords.

Capitalization-Based Proper Noun Removal

The first iteration of the proper noun removal algorithm removed all words which were capitalized in the middle of a sentence. In addition, if a word was considered to be a proper noun in any document in a corpus, it was considered to be a proper noun in every document. This caused problems because sometimes common words appeared as a person's name or in the title of a paper being cited, etc. Thus words like "gamble" (from Proctor and Gamble) would always be considered as a proper noun. Though the impact of this problem for a small set of documents would be rather trivial, the aggregate impact of thousands of articles in a data set caused many legitimate nouns to be considered as proper nouns.

Capitalization and Frequency-Based Proper Noun Removal

The improved version of the proper noun removal algorithm used frequency statistics to prevent reasonable words from being considered as proper nouns in all documents. To be considered a proper noun in all documents, a word has to first pass the capitalization test in at least half of its occurrences (excluding occurrences at the beginning of sentences, where a word is always capitalized). This final tweak led to a much more effective keyword identification methodology.

Table 6.1 compares the final improved keywords to the ones extracted by the naive method. Keywords containing words like FDA, Harvard, Proctor, Gamble, scientific (stemmed to science), and Olestra have been discarded because they appeared to be proper nouns (science was capitalized often enough that it was flagged as a proper noun). As these proper nouns were filtered out, the more descriptive keywords like beta carotene and carotenoid became the top unigram and bigram keywords.

6.1.2 Novelty Scoring Metric

Once keywords have been extracted from each document in a candidate discovery, they are used to determine the novelty of that candidate discovery. An informal inspection of the scores indicated that the computed novelty score is generally a good indicator of the potential a candidate discovery has. However, some documents are very short or have other

Table 6.1: Science News Article #25 Top 5 Keywords: Naive vs. Proper Noun Removal

Type	Naive	Capitalization and Frequency-Based
top bigrams	scientific evidence Harvard School beta carotene macular degeneration FDA Review	beta carotene macular degeneration potato chip prostate cancer fake fat
top unigrams	FDA carotenoid Proctor Gamble Olestra	carotenoid carotene approval chip fat

complications which make it difficult to extract meaningful keywords. Poor keywords (e.g., overly general terms) sometimes lead to skewed novelty scores because bad keywords may result in an unusually small denominator in Equation 3.32 which causes the novelty score to be inflated.

Table 6.2 compares several variants of novelty scores for each dimensionality reduction technique on the Science News-4M data set. Each value is the median score chosen from the top 20 candidate associations for each DR technique. At left, the column labeled "Raw" holds the original score as computed by Equation 3.32. This column shows values for PCA and LDM that are much higher than for the other techniques. Closer inspection revealed that these values were artificially high due to some documents having poor keyword selection. For instance, one pair's keywords included "alike observes" and "deeply colored" for one document, and "cross cultural" and "fall asleep" for another document. This led to a very high score because the union of these broad keywords produced a very large number of hits (and thus a very small value for $score_G$), yielding a small denominator and a very large result for $score_{novelty}$ via Equation 3.32.

In general, results showed that a very large fraction of the problematic keywords contained verb phrases (e.g., "fall asleep"), whereas the most appropriate keywords often seemed to be noun phrases only (e.g., "beta carotene"). Hence, future work to filter keywords based on their part of speech within a sentence may be useful.

To enable meaningful analysis of the scores without such part of speech processing, this project investigated several techniques for handling these problematic keywords, and the

Table 6.2: Science News-4M: A Comparison of Raw, Thresholded, and Filtered Results

	Score Type			
DR	Raw	Threshold = 100	Threshold = 50	Filtered
PCA	97.4	66.9	18.0	18.6
MDS	7.3	6.7	4.4	8.2
Isomap	12.6	11.3	10.8	11.2
LLE	13.1	6.4	6.4	7.5
LDM	64.2	7.7	5.2	5.4
None-All	30.8	24.0	17.25	19.8

results are also shown in Table 6.2. The column labeled "Threshold=100" shows the results where candidate discoveries with a computed score greater than 100 are automatically removed from consideration (since manual inspection indicated that these were almost always a result of poor keyword selection). The next column shows results computed with a threshold of 50. Finally, the last column, "Filtered" shows the results obtained when associations were manually removed if the keywords of either document in a pair appeared to be to be poor content summaries. This last technique is not practical for large scale experiments, since it involves manual inspection of the results, but can serve as a baseline for comparison. The similarity between the filtered results and the thresholded results indicates that thresholding performed reasonably well at automatically removing candidate discoveries with poor keywords. Thus, the remaining results in this chapter all apply a threshold of 50 to the candidate associations that are found.

6.1.3 Evaluating Novelty Scores for No DR

To further validate the novelty score computation, this section examines the novelty scores of None-Rand and None-Sort for varying numbers of dimensions. Table 6.3 compares the thresholded novelty scores for these methods on the Science News-4M data set, using twenty candidate discoveries for each technique. As expected, the results show that both None-Sort and None-Rand performed best when they had access to all of the original dimensions (over 10,000). When limited to only a few dimensions, None-Sort usually outperforms None-Rand by a small margin.

Table 6.3: Science News-4M: A Comparison of Variations of No DR

	Number of Dimensions					
No DR Type	5	8	11	41	71	All
None-Rand	2.5	2.4	3.6	3.8	3.8	24.2
None-Sort	3.4	2.1	3.7	4.1	4.2	24.2

6.2 Assessing DR's Impact on LBD

This section first presents results which shows the median quality of candidate discoveries for each DR technique. The number of dimensions and the corpus are varied. Section 6.2.2 then displays the relative effectiveness of each DR technique. Finally, Section 6.2.3 presents evidence which suggests that each DR technique may have something unique to contribute to the LBD process.

6.2.1 Evaluating Median Novelty Scores for DR

Figure 6.1 compares the thresholded novelty scores on various data sets. These results are based on twenty candidate discoveries for each technique.

On Science News-4M PCA performed above all other techniques, while Isomap did slightly better than other techniques on average. Surprisingly, the other techniques, as well as None-Sort and None-Rand, were mostly indistinguishable from each other.

Science News-8 was mostly consistent with the results for Science News-4M. However, Isomap was closer to achieving PCA's performance in a small number of dimensions. Furthermore, most of the other techniques were even closer to Isomap's peak performance, except None-Rand and None-Sort which had significantly worse performance in all dimensions than the other techniques.

On Google News, PCA and Isomap performed very similarly beyond 5 dimensions. Interestingly, None-Rand and None-Sort outperformed the other DR techniques on this corpus.

Science & Technology had slightly different results. PCA was still the top and most consistent performer, but LDM performed significantly better than usual and slightly better than Isomap and LLE which were just below it. MDS, None-Rand, and None-Sort did relatively poorly and performed substantially worse than the other DR techniques.

Overall, PCA had the best performance followed by Isomap. Other techniques did well at times, but were not consistent performers. Often, many of the other techniques were clumped together and their performance was essentially indistinguishable from each other. Surprisingly, results using very few dimensions were almost as good as those using 40 to 70 dimensions.

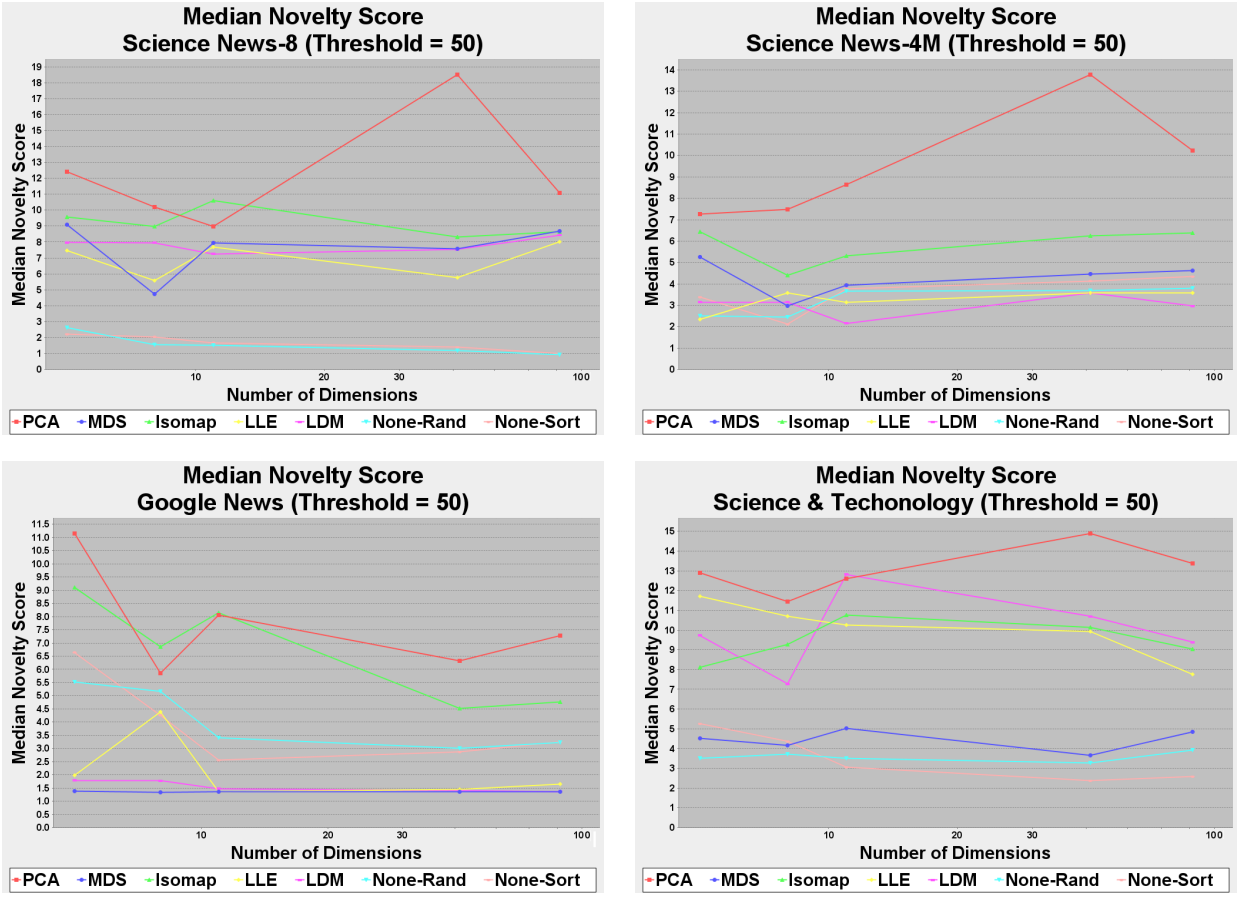

Figure 6.1: Median Novelty Score vs. Number of Dimensions (log scale)

6.2.2 Evaluating Relative DR Performance

The previous section presented results in terms of raw novelty scores. In a typical use case, however, a researcher may be more interested in the relative performance of the DR techniques than in their absolute scores. In addition, the researcher may be willing to experiment with more than one choice for the number of dimensions to find good candidate associations.

Figures 6.2, 6.3, 6.4, and 6.5 summarize results for this comparison. Each DR technique is evaluated based upon the best median score it obtained in the previous section using any number of dimensions. The value shown for each technique is the fraction that this best score comprises of the total best scores for that corpus. For example, PCA on Science News-8 has a best median score of 18.5 in Figure 6.1, compared to the total of the best scores of about 59.7. Thus, PCA is represented in Figure 6.3 by a value of 30% (18.5 divided by 59.7).

These figures contain one additional variant for comparison. The values labeled "None-All" are similar to None-Sort and None-Rand, but represent the case where *all* of the raw, unreduced features are used. Consistent with Table 6.3, the results show that None-All usually, but not always, out-performs None-Sort and None-Rand, which in these figures use at most 71 dimensions. Hence, when using unreduced data, more dimensions is better for LBD. However, comparing None-All against the DR techniques demonstrates that while None-All compares favorably with some of the DR techniques, PCA and to some extent Isomap typically do better than None-All.

6.2.3 Candidate Discovery Overlap

The previous section demonstrated that PCA and Isomap tended to produce the best novelty scores, but other techniques still produced some reasonable scores. Hence, it may be useful to perform LBD using more than one DR technique – if those techniques tended to find different associations.

Table 6.4 shows the overlap of one hundred candidate discoveries between different DR techniques on the Science News-4M corpus. The number indicates how many of the same discoveries were found by two techniques. For instance, 13% of the candidates found by

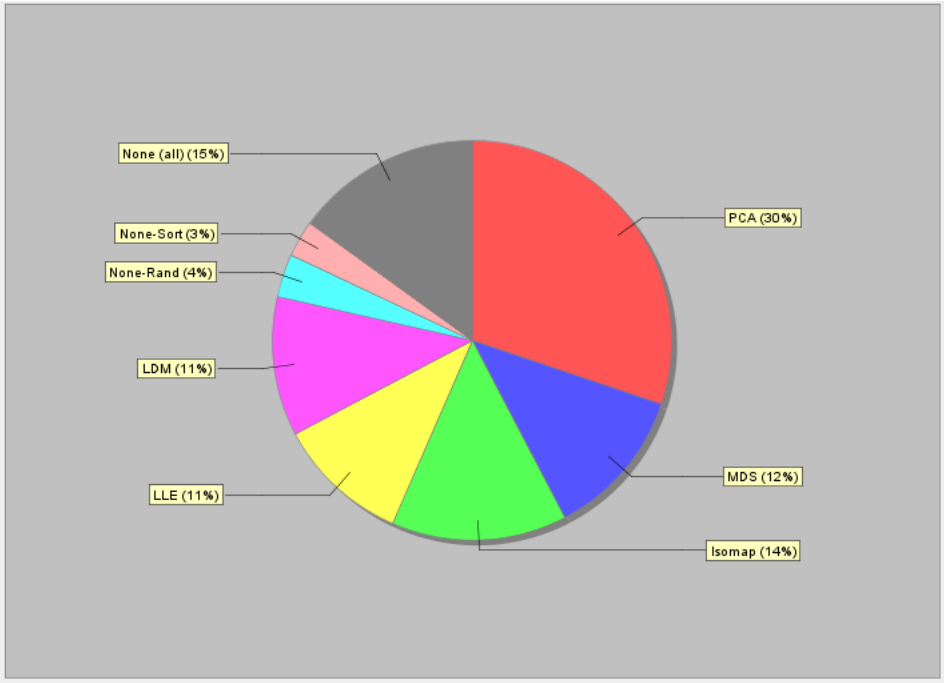

Figure 6.2: Best Median Novelty Score (as fraction of total) for Science News-8

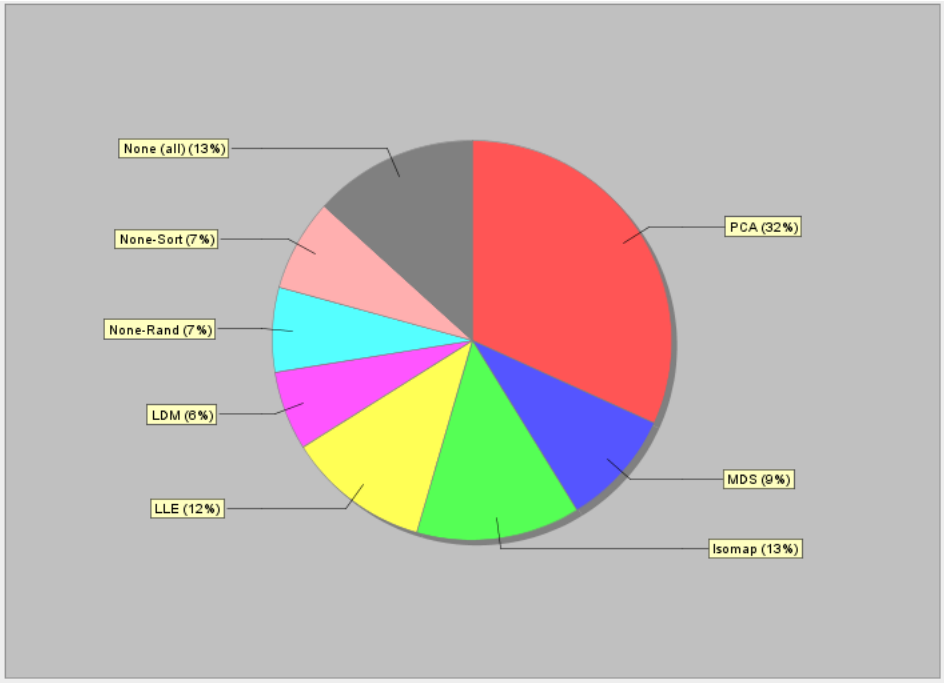

Figure 6.3: Best Median Novelty Score (as fraction of total) for Science News-4M

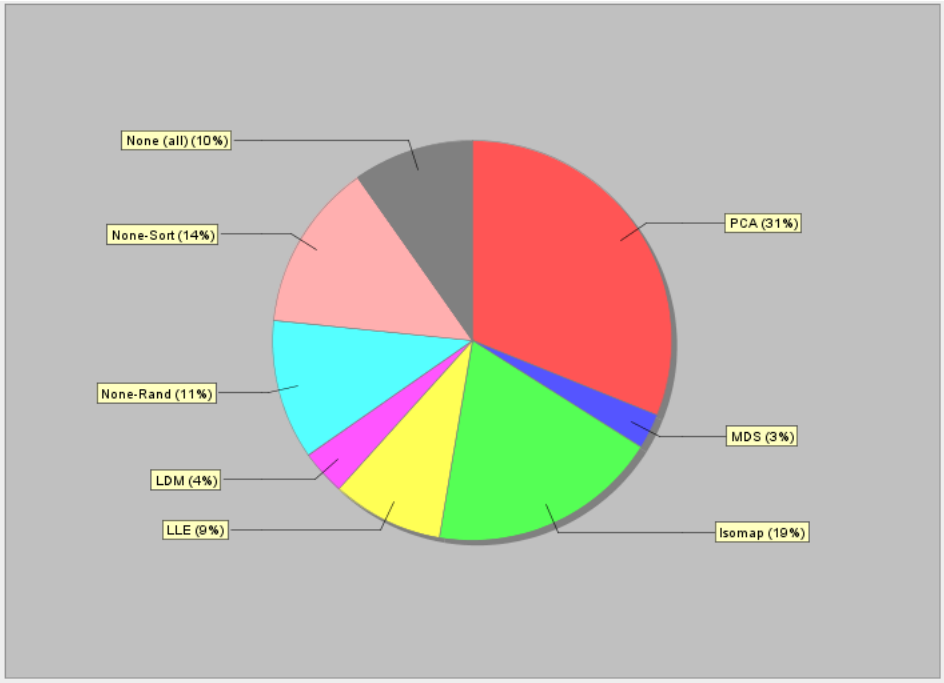

Figure 6.4: Best Median Novelty Score (as fraction of total) for Google News

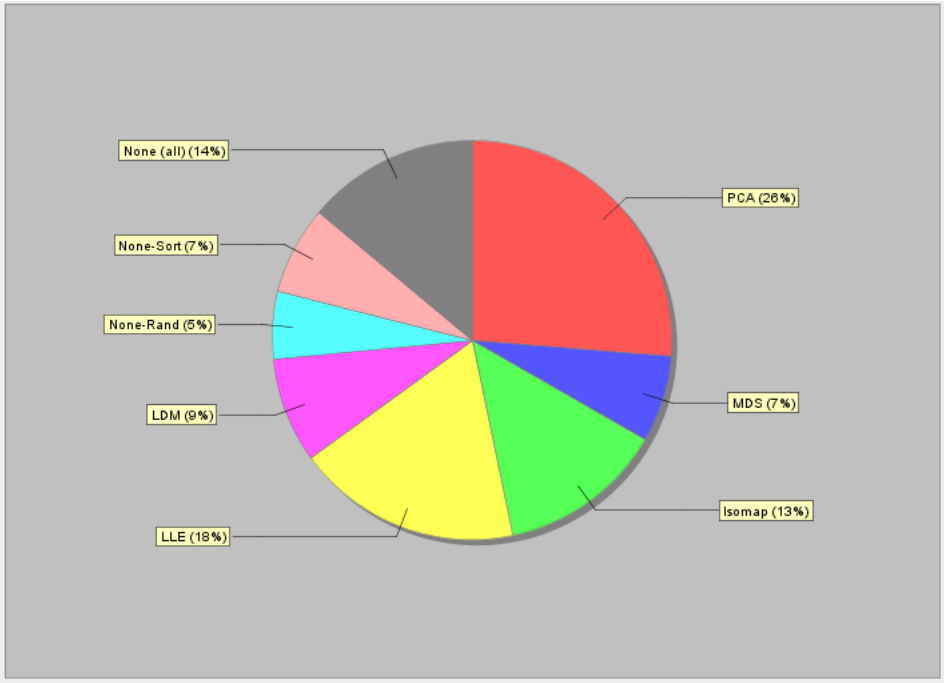

Figure 6.5: Best Median Novelty Score (as fraction of total) for Science & Technology

Isomap were also found by MDS. Along the diagonal, the number is always 100% because a technique always finds the exact same results as itself.

This table reveals several interesting results. First, it shows that each technique produces pairs which largely differ from the other techniques. This means that each technique may produce worthwhile – and unique – candidate discoveries. Second, Isomap is the only non-linear technique which overlaps with PCA, the technique which had the best overall LBD performance. It is interesting to note that Isomap also overlaps with MDS (which it uses as the last step in its own process). Third, each non-linear technique has significant overlap with other non-linear techniques but no significant overlap with linear techniques. The reverse is also true. Finally, no technique has significant overlap with the candidate discoveries found when DR is not used.

Table 6.4: Science News-4M, 11 dims: % Candidate Discovery Overlap between DR Techniques

	DR Type					
DR Type	PCA	MDS	Isomap	LLE	LDM	None-All
PCA	100	**10**	**27**	4	4	6
MDS	**10**	100	**13**	4	7	2
Isomap	**27**	**13**	100	**17**	**16**	4
LLE	4	4	**17**	100	**12**	1
LDM	4	7	**16**	**12**	100	1
None-All	6	2	4	1	1	100

6.3 Analysis

Dimensionality reduction was often able to improve the estimated quality of candidate discoveries when compared to not applying DR. However, PCA was the only technique that consistently did significantly better than None-All. Isomap also yielded consistent performance at the level of None-All and sometimes a little better. Both PCA and Isomap were effective with very few dimensions, while None-All used all of the original 10,000+ dimensions. The other DR techniques occasionally did well, but were unable to produce consistently good candidate discoveries. However, despite lower overall scores, different techniques did produce many unique candidate discoveries, so each technique may have some candidates which can

contribute to the overall effectiveness of the LBD process by identifying unique and possibly relevant candidate discoveries.

It is interesting to compare these results with those obtained for document classification. In both cases, Isomap did well. PCA did well on classification but was not the best. However, it was the most effective technique at uncovering quality candidate discoveries. This suggests that classification performance does not directly correlate with LBD performance, although they appear to be closely enough related to be able to infer possible winners for one task from the other.

With both classification and LBD, the best techniques needed only a few dimensions to equal or outperform None-All with all 10,000+ dimensions. These results support this project's conjecture that DR can be an effective facilitating technology for multiple text mining applications.

Chapter 7

Related Work

Section 7.1 discusses how dimensionality reduction has been previously applied to geometric and image data. Section 7.2 presents work in the text mining field by collaborators Dr. Dave Marchette and Dr. Jeff Solka as well as work by others in the field. Section 7.3 describes work which has used keyword extraction as an integral part of larger text mining and information retrieval applications.

7.1 DR Applications to Geometric and Image Data Visualization

In 2000, Tenenbaum, Silva, and Langford illustrated how Isomap could uncover the inherent dimensionality of various complex geometric data using a new dimensionality reduction technique they termed Isomap [30]. They illustrated the algorithm's ability to map 698 computer-generated faces in various poses from 4,096 dimensions (pixel count of each image) to a two-dimensional representation (three including the lighting direction). The logical ordering of faces shown demonstrates that Isomap has found a meaningful low-dimensional embedding [30].

Isomap has also been used to map a "Swiss roll." The linear techniques are confused by the straight-line Euclidean distances because as the roll twists, an inner loop's point may be relatively close in Euclidean space to an outer loop's point, even though the distance along the roll is quite large. Isomap is able to uncover the correct two-dimensional structure of

the roll [30].

In 2001, Roweis and Saul published a report explaining LLE and demonstrating its capability on two data sets [25]. Though they claimed it had similar capabilities to Isomap's ability to uncover nonlinear geometric structures, they also showed that LLE took advantage of local neighborhoods to construct a sparse matrix which could be more easily decomposed into corresponding eigenvalues and eigenvectors.

7.2 DR Applications to Text Mining

In 2004, Priebe and Solka explored an iterative method for the discovery of interesting relationships between documents in a corpus [22]. They termed the process *iterative denoising*. Iterative denoising recursively encodes and analyzes sub-corpora to focus on the content in specific areas without interference from other documents in the corpus. An example of the work was demonstrated on the same 1,047 documents from the Science News data set being used in this paper's experiments. The encoding of the data set consisted of 10,906 stemmed unigrams after stopper words were removed.

Iterative denoising starts by encoding a corpus into a similarity matrix by using weighted unigram counts. This similarity matrix is used to compute an interpoint distance matrix which is embedded into a Euclidean space with MDS which is set to keep all numerically stable dimensions. This Euclidean embedding is reduced with PCA to a smaller space. The dimensionality of the space is determined with a scree plot. Next, this reduced space is clustered, typically into halves, using hierarchical clustering. This process is iterated on new clusters until the user is satisfied with the cluster obtained. Each iteration produces successively more focused and less noisy sub-corpora. Each pair of documents from *different categories* which appear in the same final cluster are potentially interesting. Their association is further analyzed to determine why it is interesting.

In 2005, Solka, Bryant, and Wegman used minimal spanning trees (MSTs) to analyze two different corpora of science-related articles [26]. The goal of their research was to use literature-based discovery techniques to find subtle, unknown relationships between documents in different categories. They also showed how a minimal spanning tree could be used

to perform hierarchical clustering on the corpus such that documents in the same categories would be clustered together. They encoded their corpus by computing a similarity matrix from the bigram proximity matrix using the Ochai similarity measure. A bigram proximity matrix enumerates the number of times each pair of words, or bigram, appears in each document in the corpus. This encoding was chosen because it yielded good results when analyzed by Martinez with various classification techniques [21].

Computing the MST is done with Kruskal's algorithm. An MST helps isolate important information which may otherwise be obscured by the complexity of the complete graph of the relationships between all documents in the corpus. Once the minimal spanning tree was computed, several methods for finding interesting relationships between documents were experimented with. First, one could look for the most related articles between categories. Though this produced some interesting results, other, less direct methods produced some interesting relationships in less obvious ways. Another method was to look for documents near boundaries between categories. With a minimal spanning tree, finding these was relatively easy. Finally, articles with similar relationships to a boundary could be explored.

The above techniques were tested on the Science News data set and the Science & Technology data set. The former consisted of 1,117 articles (slightly more than the version of the data set used in this paper's experiment) and the latter consists of a subset of the data set used in this paper's experiment (only 343 articles). Stopper words were removed for the tests, but words were not stemmed.

In their 2005 paper entitled *Diffusion Wavelets*, Coifman and Maggioni discuss the application of wavelet analysis techniques from signal processing to text mining [8]. They describe a *diffusion operator* which is similar to the Laplace-Beltrami operator as described in Section 3. Both techniques relate distance between observations to the number of paths between the two in a graph based on the closest interpoint distances. They applied a variety of diffusion operators to the similarity matrix computed from the Science News data set of 1,047 articles and 10,906 unigrams. The resulting embedded space could be used to accurately cluster the documents into their respective categories with k-means and hierarchical clustering techniques. Detailed results are expected but have not yet been published. This differs from the research of this project in that they are only analyzing a single approach

where as this research compares results from five different techniques. Furthermore, the ultimate goal of this research was LBD rather than clustering.

7.3 Keyword Extraction

In 1994, Efthimiadis evaluated several keyword ranking algorithms [11], focusing particularly on their use in augmenting search queries with other relevant terms.

He proposed several simple criteria for choosing good keywords. First, words which are used too often are not very useful. Words which are not used very frequently are good because they describe a very focused subject well, but these words often restrict the search base too much since they are rarely found in documents. These rules make words which appear with about average frequency the best search terms in the eyes of the author. In particular, these criteria serve search needs well because searches require broad applicability. However, with regards to the research presented by this paper, infrequent terms may be relatively more useful since only the very few best matches are considered during the literature-based discovery process.

In 1999, Anick and Tipirneni developed a tool they called the Paraphrase Search Assistant [1]. As with Efthimiadis' work, this tool was designed to augment user's search queries with additional relevant terms to help improve the results of the search. Their approach hinges on the "Lexical Dispersion Hypothesis" which states that "new concepts are often expressed not as new single words, but as concatenations of existing nouns and adjectives" [1]. Therefore, if such combinations of existing words can be identified, then the focus of the search can be used to extract additional keywords which can augment the search. The authors found that by presenting various combinations of potential augmented searches to users, more effective searches could be constructed. They also found that using an iterative process to conduct multiple small searches allowed the user to concentrate the search on keywords which produced documents of interest.

In 1995, Church and Gale presented inverse-document frequency (IDF) as defined in Equation 3.1, a quantity which could assist the weighting of terms within a document that belonged to a particular corpus [5]. In particular, the authors assert that "not all equally

frequent words are equally meaningful." For instance, in some corpus the words "somewhat" and "boycott" may appear equally frequently, but clearly "boycott" is a more significant term. If the IDF of these terms is considered, the two terms are well separated because while "somewhat" appears rather evenly over all documents (making its IDF closer to one), "boycott" is concentrated in only a few documents (makes its IDF closer to zero).

In 1999, Budzik and Hammond proposed a keyword-driven system called an Information Management Assistant (IMA) [4]. The purpose of this system was to uncover helpful materials related to the user's current needs based on observations of the user's interactions with software like word processors and web browsers. Based on the content of the user's actions, keyword searches on search engines and other databases of information could be executed to retrieve relevant information. To generate the needed keywords, Budzik and Hammond considered a number of heuristics for evaluating the relative importance of individual terms from a single text document:

- The first heuristic suggests that stopper words be removed. As discussed in Section 3.1, this is also an important part of the encoding process used by this research.

- The second heuristic is to value frequently-used words, which are assumed to be representative of the document. This heuristic is helpful with information retrieval, but it ignores the context of a term's frequency in the overall corpus.

- Another heuristic suggests that words which appear earlier in the document are more important. Though this assertion may be true, the encoding process used by this research only considers word counts. This sort of context-sensitive processing is thus beyond the scope of this research, but would be an interesting factor to consider in conjunction with more sophisticated natural language processing techniques.

- Finally, the authors suggest adjusting the weight of words based on their emphasis in a document as determined by their font size and style. Such techniques were not considered because the plain text data sets used by this research did not contain any such formatting.

In 1999, Lagus and Kaski presented a keyword extraction methodology [20]. Unlike

the keyword extraction done by this research, their work focused on generating keywords for clusters of documents. These keywords are then used to label groups of documents in document maps. These maps can be used to generate a graphical layout of document clusters based on the Self-Organizing Map algorithm [17].

Chapter 8

Conclusions and Future Work

Section 8.1 synthesizes the results from the classification and LBD experiments described in Chapter 5 and Chapter 6. Section 8.2 describes the future work which could build on this research.

8.1 Conclusions

This research evaluated two distinct text mining processes with regards to dimensionality reduction techniques. The effects of dimensionality reduction on the effectiveness of various text mining tasks was not well understood prior to this research. The results showed that dimensionality reduction can be highly effective at improving performance for both classification and LBD processes. Surprisingly, non-linear techniques did not generally improve performance over their best linear counterparts.

Though PCA is one of the most commonly used dimensionality reduction techniques, its performance was inconsistent on text classification. While it did well on easier data sets like Science News, it struggled on harder data sets like Google News and Science & Technology. Instead, MDS and Isomap were the best overall techniques for classification. They consistently performed as well or better than the other techniques.

Literature-Based Discovery was previously explored by Dr. Swanson as a human-supervised algorithm. Since he first introduced this technique, others have explored semi-automated LBD approaches. This research developed a fully automated approach based upon a novel

methodology for scoring candidate LBD discoveries. A novel keyword extraction technique was also developed to facilitate the requirements of the LBD process. Results showed that PCA was frequently the best performer on LBD. However, though PCA had the highest median scores, each DR technique finds different pairs. Thus, performing LBD with multiple DR techniques uncover different, interesting candidate discoveries.

8.2 Future Work

LBD is a subjective process. Though this research was able to develop an unsupervised method for estimating the novelty of a particular candidate discovery, the process would also benefit from a human analysis. Human-based studies of the pairs found by each technique would help further validate and improve the automated approach.

It would also be interesting to investigate the pairs found by each DR technique for the LBD process and examine the scores of overlapping pairs. It may be the case that pairs found by multiple DR techniques are generally better than pairs found by only one technique, though future research is needed to confirm this hypothesis.

The keyword extraction process is the foundation of the automated scoring methodology. An improvement on this keyword extraction process would benefit the LBD process as a whole too. The current algorithm occasionally makes mistakes. In particular, it sometimes ascribes too much importance to verbs that in fact convey little meaning. As a result, the algorithm could be improved if each word's part of speech was considered.

Finally, the current dimensionality reduction pipeline could be improved in several ways. The algorithms, in particular PCA and LDM, are time-consuming and would benefit from algorithmic changes which improve efficiency. Also, this work only explored dimensionality reduction with respect to complete, unchanging data sets. It would be interesting to investigate ways to insert new documents into the dimensionally reduced space without requiring the dimensionality reduction process to be run on the entire set of data.

Bibliography

[1] Peter G. Anick and Suresh Tipirneni. The paraphrase search assistant: Terminological feedback for iterative information seeking. In *Proceedings of the 22nd annual international ACM SIGIR conference on research and development in information retrieval*. ACM Special Interest Group on Information Retrieval, ACM Press, 1999.

[2] Mukund Balasubramanian and Eric L. Schwartz. The Isomap algorithm and topological stability. *Science*, 295:7a, Jan. 2002.

[3] Ingwer Borg and Patrick Groenen. *Modern Multidimensional Scaling*. Springer-Verlag, New York, 1997.

[4] Jay Budzik and Kristian Hammond. Watson: Anticipating and contextualizing information needs. *Annual Meeting of the American Society for Information Science*, 62, 1999.

[5] Kenneth W. Church and William A. Gale. Inverse document frequencies (idf): A measure of deviations from Poisson. In *Annual ACM Conference on Research and Development in Information Retrieval*. ACM Press, 1995.

[6] R. R. Coifman, Stéphane Lafon, A. B. Lee, M. Maggioni, B. Nadler, F. Warner, and S. W. Zucker. Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps. *PNAS*, 102(21):7426–7431, May 2005.

[7] R. R. Coifman, Stéphane Lafon, A. B. Lee, M. Maggioni, B. Nadler, F. Warner, and S. W. Zucker. Geometric diffusions as a tool for harmonic analysis and structure definition of data: Multiscale methods. *PNAS*, 102(21):7432–7437, May 2005.

[8] Ronald R. Coifman and Mauro Maggioni. Diffusion wavelets. *submitted to Elsevier Science*, March 2005.

[9] Vin de Silva and Joshua B. Tenenbaum. Global versus local methods in nonlinear dimensionality reduction. *Proceedings in Neural Information Processing Systems*, 15:721–728, 2003.

[10] Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification*. Wiley-Interscience, 2nd edition, November 2000.

[11] Efthimis N. Efthimiadis. User choices: A new yardstick for the evaluation of ranking algorithms for interactive query expansion. *Information Processing and Management*, 31(4):605–620, 1995.

[12] Tobias Friedrich. Nonlinear dimensionality reduction - Locally Linear Embedding versus Isomap. Technical report, The University of Sheffield - Machine Learning Group, Sheffield S1 4DP, U.K., December 2004.

[13] Keinosuke Fukunaga. *Introduction to Statistical Pattern Recognition*. Elsevier, September 1990.

[14] Murat C. Ganiz, William M. Pottenger, and Christopher D. Jameck. Recent advances in Literature Based Discovery. Technical report, Lehigh University, 2005.

[15] Brandon W Higgs, Jennifer Weller, and Jeffrey L Solka. Spectral embedding finds meaningful (relevant) structure in image and microarray data. *BMC Bioinformatics*, 7:74, February 2006.

[16] I.T. Jolliffe. Principal component analysis. Technical report, Springer, October 2002.

[17] Teuvo Kohonen. *Self Organizing Maps*. Springer-Verlag New York, Inc., 1997.

[18] Dejan Kulpinski. LLE and Isomap analysis of spectra and colour images. Master's thesis, Simon Fraser University, March 2002.

[19] Stéphane Lafon. *Diffusion Maps and Geometric Harmonics*. PhD thesis, Yale University, Mathematics Department, 2004.

[20] Krista Lagus and Samuel Kaski. Keyword selection method for characterizing text document maps. *Artificial Neural Networks*, 470(470), September 1999.

[21] Angel R. Martinez and Edward J. Wegman. A text stream transformation for semantic-based clustering. *Computing Science and Statistics*, 34:184–203, 2002.

[22] C. E. Priebe, D. J. "Marchette", and al. Iterative denoising for cross-corpus discovery. *COMPSTAT 2004 Symposium*, 2004.

[23] Lawrence Saul. Spectral methods for dimensionality reduction. *NIPS 2002 Workshop*, July 2004.

[24] Lawrence K. Saul, Kilian Q Einberger, Fei Sha, Jihun Ham, and Daniel D. Lee. Spectral methods for dimensionality reduction. *UCLA Institute for Pure and Applied Mathematics*, page 18, 2005.

[25] Lawrence K. Saul and Sam T. Roweis. An introduction to locally linear embedding. Technical report, AT&T Labs Research and Gatsby Computational Neuroscience Unit, UCL, 2001.

[26] Dr. Jeffrey Solka, Avory C. Bryant, and Edward J. Wegman. Text data mining with minimal spanning trees. *Handbook of Statistics on Data Mining and Visualization*, 24, 2005.

[27] Jeffrey L. Solka and David J. Marchette. Lecture on dimensionality reduction, November 2005.

[28] D. R. Swanson. Fish oil, Raynaud's syndrome, and undiscovered public knowledge. *Perspectives in biology and medicine*, 30(1):7–18, 1986.

[29] D. R. Swanson. Complementary structures in disjoint science literatures. *Proceedings of the Fourteenth Annual Internation ACM SIGIR Conference*, 1991.

[30] Joshua B. Tenenbaum, Vin de Silva, and John C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290:2319–2323, December 2000.

[31] Michael W. Trosset. Classical Multidimensional Scaling and Laplacian Eigenmaps. Joint Statistics Meeting, 2006.

[32] G.V. Trunk. A problem of dimensionality: A simple example. *IEEE Transactions Pattern Analysis and Machine Intelligence*, 1(3):306–307, July 1979.

[33] Florian Wickelmaier. An introduction to MDS. Technical report, Aalborg University (Denmark), May 2003.

Appendix A

Glossary

- **Association** - See Candidate Discovery.
- **Candidate Discovery** - A pair of documents from disparate categories which are closely related.
- **Classification** - This process uses a set of "training data" with known categories to decide how to assign categories to observations in a "test set." In effect, the "training set" is used to gain an understanding what features distinguish observations from each class.
- **Confusion Matrix** - These are used to help identify the source of some classification errors. Each row in the matrix identifies how well a particular category was classified and where it was classified.
- **Corpus** - A collection of articles.
- **Dimensionality Reduction (DR)** - Process of transforming a large amount of data into a much smaller representation.
- **Disparate** - Indicates that two things are from different categories (e.g. Chemistry and Astronomy).
- **Dissimilarity Matrix** - See Interpoint Distance Matrix.
- **Distance Matrix** - See Interpoint Distance Matrix.

- **Distance** - A measure of how related two documents are. A distance of zero would indicate that the documents are exactly alike while a larger number would indicate that the documents are less alike.
- **DR** - See Dimensionality Reduction.
- **Encoding** - A matrix representation of a corpus of text documents.
- **Feature** - A single word in the encoded matrix. The word's count or weight is a column in the matrix (each cell defines the number of times the word appears in a particular document). After dimensionality reduction is applied, features no longer correspond to words, but are just arbitrary descriptors of each document.
- **Interpoint Distance Matrix (IPDM)** - A matrix which defines the distance between all documents in a corpus.
- **IPDM** - See Interpoint Distance Matrix.
- **Isomap** - A nonlinear type of Dimensionality Reduction based on geodesic distances (distances through a nearest neighbors graph).
- **Keyword Extraction** - The process of finding keywords in a document.
- **Keyword** - A descriptive term for a document.
- **Laplace-Beltrami Diffusion Maps (LDM)** - A nonlinear type of Dimensionality Reduction based on the number of paths between a pair of documents.
- **LBD** - See Literature-Based Discovery.
- **LDM** - See Laplace-Beltrami Diffusion Maps.
- **Literature-Based Discovery (LBD)** - Automatically discovering interesting, previously unknown relationships between two documents (usually from different categories).
- **LLE** - See Locally Linear Embedding.

- **Locally Linear Embedding (LLE)** - A nonlinear type of Dimensionality Reduction which works by translating, rotating, and scaling the data before performing an Eigenvalue Decomposition.
- **MDS** - See Multidimensional Scaling.
- **Multidimensional Scaling (MDS)** - A linear type of Dimensionality Reduction based on an Eigenvalue Decomposition.
- **Noise Words** - See Stopper Words.
- **PCA** - See Principal Components Analysis.
- **Principal Components Analysis (PCA)** - A linear type of Dimensionality Reduction based on SVD.
- **Stem** - The process of finding a word's root (e.g. "love," "loved," and "loving" all become "lov").
- **Stopper Words** - Words which are not encoded because they usually have little meaning (e.g. "the" or "it").
- **Supervised Method** - A method which requires manual tweaking in order to run (e.g. requires a tuning parameter).
- **TDM** - See Term-Document Matrix.
- **Term-Document Matrix (TDM)** - A matrix in which each row corresponds to a document and each column corresponds to a word. Therefore, each row vector describes how many times each word appears in a particular document. If weighted with the TF-IDF formula, then this may be referred to as the Weighted TDM.
- **Text Mining** - Extraction of important information from a collection of textual data sources.
- **Tuning Parameter** - A parameter which is required to adjust how a particular method processes an input.

- **Unsupervised Method** - A method which does not require manual tweaking in order to run (e.g. does not require a tuning parameter).
- **Weighted Term-Document Matrix** - See Term-Document Matrix.

Appendix B

Supporting Matrix Calculations

B.1 Normalizing Matrices

Many dimensionality reduction techniques normalize their input prior to performing any calculations. A summary of such normalizations, including centering and scaling, are defined here.

Input: Matrix M $(r \times c)$

Column-Centering: Subtract the mean of each column from each column. The average or mean value of column j in M:

$$\overline{M_{*,j}} = avg(M_{1,j} : M_{r,j}) = \frac{\sum_{i=1}^{r} M_{i,j}}{r} \tag{B.1}$$

Let C be the column-centered data from the input matrix M:

$$C_{i,j} = M_{i,j} - \overline{M_{*,j}} \tag{B.2}$$

Row-Centering: Subtract the mean of each row from each row. The average or mean value of row i in M:

$$\overline{M_{i,*}} = avg(M_{i,1} : M_{i,c}) = \frac{\sum_{j=1}^{c} M_{i,j}}{c} \tag{B.3}$$

Let C be the row-centered data from the input matrix M:

$$C_{i,j} = M_{i,j} - \overline{M_{i,*}} \tag{B.4}$$

Double-Centering: Subtract the mean value of each row from each row, and each column from each column. This can be accomplished by applying both the column-centering Equation B.2 and row centering Equation B.4. Double-centering can also be accomplished by the following:

The centering matrix H is an $n \times n$ matrix whose values are $-n^{-1}$ off the diagonal and $1 - n^{-1}$ on the diagonal. This is defined as:

$$H = I - n^{-1} \tag{B.5}$$

The data M can then be double-centered by the following (let C be the double-centered data):

$$C = HMH \tag{B.6}$$

Scaling: Let the scaled matrix S be the division of each value in M by the root mean square value of its column. This helps avoid computations with extremely small values that could otherwise look like dividing by zero to a computer since precision is limited. These new scaled values in S lie in a computationally satisfactory range. The scaled value $S_{i,j}$ for $M_{i,j}$ is computed as follows:

Root mean square of column j in M:

$$rms(M, j) = \sqrt{\frac{\sum_{i=1}^{d} M_{i,j}^2}{d-1}} \tag{B.7}$$

Scaled value:

$$S_{i,j} = \frac{M_{i,j}}{rms(M, j)} \tag{B.8}$$

Scaled and Column-Centered:

$$S_{i,j} = \frac{M_{ij} - \overline{M_{*,j}}}{rms(M, j)} \tag{B.9}$$

B.2 Euclidean Embeddings

A Euclidean embedding utilizes a special kind of distance - Euclidean distance. In general, distance is defined as how far apart two things are. Though this sounds straightforward, there are many different ways of defining distance. The only constraints on a distance is that it is required to be non-negative and symmetric. Euclidean distance, on the other hand, also expresses linearity. In other words, it is required to satisfy the triangle inequality.

Euclidean embeddings are preferred to other embeddings because of that extra linearity property. The reason linearity is so important is because dissimilarity matrices of Euclidean distances are guaranteed to have positive eigenvalues. As explained in Section 3.2, eigenvalues and their associated eigenvectors are needed in order to effectively project high dimensional inputs into low dimensional spaces.

The dissimilarity matrices described in Section 3.1 typically contain negatives and make no effort to satisfy the triangle inequality in most cases. As a result, they are technically proximities and not distances. They can be transformed into distances by adding some constant which translates all the proximities to values greater than or equal to zero, but this does not guarantee a Euclidean embedding because it is likely that triangle inequalities are not met.

However, a standard procedure exists to embed a non-Euclidean symmetric matrix into a Euclidean space. There always exists some constant c large enough that when added to all the distances, the triangle inequality is satisfied [3]. There is a procedure for computing the smallest possible constant c which can be added and still coerce all the distances in the matrix to satisfy the triangle inequality. The smallest possible c is desirable because as c

grows, the number of dimensions needed to represent the data grows too.

A matrix can be embedded in a Euclidean space as follows:

Input: Symmetric Matrix M $(n \times n)$

1) Construct the $2n \times 2n$ supermatrix S:

Let D contain the squared values from the double-centering of matrix M as defined by Equation B.6.

$$S = \begin{bmatrix} 0 & 2D \\ -I & -4M \end{bmatrix} \tag{B.10}$$

2) Determine the minimum constant c: Perform an eigenvalue decomposition of S and find the largest real eigenvalue. The value of that eigenvalue is the minimal constant c.

3) Compute $M*$, the $n \times n$ Euclidean embedding of the input matrix M:

Let H be the centering matrix defined by Equation B.5 for the number of rows n.

$$M* = M + 2cD + \frac{c^2}{2}H \tag{B.11}$$

Appendix C

Experimental Support Information

C.1 Stopper Words List

Table C.1: Stopper Words

a	clearly	generally	later	open	sees	turn
about	come	get	latest	opened	several	turned
above	concentrated	gets	least	opening	shall	turning
across	concentration	give	less	opens	she	turns
after	concentrations	given	let	or	should	two
again	could	gives	lets	order	show	u
against	d	go	like	ordered	showed	under
all	develop	going	likely	ordering	showing	until
all	developed	good	long	orders	shows	up
almost	developing	goods	longer	other	side	upon
alone	develops	got	longest	others	sides	us
along	did	great	ltd	our	since	use
already	differ	greater	m	out	small	use
also	different	greatest	made	over	smaller	used
although	differently	group	make	p	smallest	used
always	do	grouped	making	paper	so	uses
among	does	grouping	man	papers	some	uses
an	done	groups	many	part	somebody	using
and	down	h	mass	parted	someone	v
another	down	had	may	parting	something	very
any	downed	has	me	parts	somewhere	w
anyone	downs	having	members	perhaps	states	wanted

Table C.2: Stopper Words, cont.

anything	during	he	men	place	still	wanting
anywhere	e	her	method	places	still	wants
are	each	here	methods	point	studied	was
area	early	herself	might	pointed	studies	way
areas	effect	high	mol	pointing	study	ways
around	effected	high	more	points	studying	we
as	effecting	high	most	possible	such	well
ask	effects	higher	mostly	present	sure	wells
asked	either	highest	mr	presented	system	went
asking	elsevier	him	mrs	presenting	systems	were
asks	elseviers	himself	much	presents	t	what
at	end	his	must	problem	take	when
away	ended	how	my	problems	taken	where
b	ending	however	myself	put	technique	whether
back	ends	i	n	puts	techniques	which
backed	enough	if	necessary	q	test	while
backing	even	important	need	quite	tested	who
backs	evenly	in	needed	r	testing	whole
be	ever	inc	needing	rather	tests	whose
became	every	increase	needs	really	than	why
because	everybody	increased	never	reserved	that	will
become	everyone	increases	new	result	the	with
becomes	everything	increasing	new	resulted	their	within
been	everywhere	interest	newer	resulting	them	without
before	f	interested	newest	results	then	work
began	face	interesting	next	right	there	worked
behind	faces	interests	no	right	therefore	working
being	fact	into	nobody	rights	these	works
beings	facts	is	non	room	they	would
best	far	it	noone	rooms	thing	x
better	felt	its	not	s	things	y
between	few	itself	nothing	said	think	year
big	find	j	now	same	thinks	years
both	finds	just	nowhere	sample	this	yet
but	first	k	number	sampled	those	you
by	for	kcal	numbers	samples	though	young
c	four	keep	o	sampling	thought	younger
came	from	keeps	of	saw	thoughts	youngest
can	full	kind	off	say	three	your
cannot	fully	knew	often	says	through	yours
cause	further	know	old	second	thus	z
caused	furthered	known	older	seconds	to	
causes	furthering	knows	oldest	see	today	
causing	furthers	l	on	seem	together	
certain	g	large	once	seemed	too	
certainly	gave	largely	one	seeming	took	

C.2 Sample Article

The Best of SCIENCE NEWS

January 27, 1996

MEDICAL SCIENCES

Olestra: Too good to be true? Researchers flush out health risks of fake fat.

By KATHLEEN FACKELMANN

The dream of a guiltfree potato chip may be crumbling. Olestra, the fat substitute that tastes like the rich stuff but has zero calories, has been known to cause diarrhea, cramping, and other nasty side effects in some people. Now, scientists have added more serious health risks to that list.

Olestra is the brainchild of Procter and Gamble, the Cincinnati-based company that holds the patent on this artificial fat, which it calls Olean. But Procter and Gamble needs the Food and Drug Administration's approval before it can market a line of olestra-containing snacks such as potato chips, tortilla chips, and crackers.

In November, olestra passed muster with two panels assigned by FDA to review the scientific evidence. At press time, the decision rested with FDA Commissioner David Olestra: Too good to be true? Researchers flush out health risks of fake fat.

Ordinarily, a product that sails through two levels of FDA review would almost certainly win Kessler's stamp of approval. However, the growing chorus of opposition to olestra may change the situation.

The beauty of olestra, a synthetic mixture of sugar and vegetable oil, is that it passes through the body without being digested or absorbed. Potato chips that contain the no-cal olestra end up having less than half the calories and none of the fat contained in regular chips. Olestra's ability to pass through the body intact poses a danger, however. Researchers say olestra binds and helps flush away certain key nutrients believed to protect against chronic diseases.

"The public needs to know more about olestra," says Walter C. Willett, an epidemiologist at the Harvard School of Public Health in Boston. Willett helped organize a scientific meeting on olestra held there last week. "The public is being asked by Procter and Gamble and the

FDA advisory committee to participate in a vast, uncontrolled national experiment," Willett says. He adds that olestra products, if approved, would be consumed by many people, including children, without adequate safety studies.

Procter and Gamble agrees that olestra helps carry away fat-soluble vitamins such as A, D, E, and K. Indeed, the firm plans to add those vitamins to snack foods containing olestra.

But the fake fat would also sweep out of the body nutrients called carotenoids, the yellow, orange, or red pigments found in many fruits and vegetables. There are about 500 nutrients in the carotenoid family–too many to add back to a bag of chips. Yet some carotenoids are thought to shield people against a wide range of diseases, including an eye condition and prostate cancer.

From data generated by Procter and Gamble, epidemiologist Meir J. Stampfer estimated that people who ate just three small olestra-containing snacks per week could expect at least a 10 percent drop in concentrations of carotenoids in their blood. He described the potential impact of such carotenoid reduction at the Boston meeting.

Stampfer, also at the Harvard School of Public Health, turned his attention first to age-related macular degeneration, a disorder that causes blurry vision and blindness. In 1994, a Boston team provided compelling evidence that two carotenoids, lutein and zeaxanthin, help prevent this devastating disorder (SN: 11/12/94, p. 310).

A 10 percent drop in concentrations of lutein and zeaxanthin would result in 390 to 800 additional cases of macular degeneration per year in the United States, Stampfer estimates.

Prostate cancer may also be prevented with a diet rich in certain carotenoids. At last week's meeting, Edward Giovannucci of Harvard Medical School in Boston presented data showing that lycopene, a carotenoid found in tomato-based products, may help protect men from developing cancer of the prostate, the nut-sized gland surrounding the urethra.

Calculations by Stampfer showed that olestra snacking could lead to 2,400 to 9,800 additional cases of prostate cancer each year.

Evidence from dietary studies has linked fruits and vegetables containing carotenoids to protection from heart disease and cancer. The most recent studies on a particular carotenoid, beta carotene, taken in supplement form, did not report such protection (see p. 55). If further studies strengthen the link, however, Stampfer calculates that a 10 percent drop in

carotenoids could cause 32,000 extra deaths in the United States per year.

Stampfer is the first to admit that scientists have yet to prove conclusively that carotenoids protect against such diseases. But if olestra is approved and further research does confirm the tie, "we're in for some serious consequences."

Procter and Gamble's Greg Allgood says the scientific evidence on carotenoids is not persuasive. He points to the two large studies that panned the ability of beta carotene to stave off cancer or heart disease.

Allgood says such results cast doubt on the entire lot of carotenoids, not just beta carotene.

According to Procter and Gamble, "it is not possible to conclude that a reduction in serum carotenoid concentration will present a public health concern."

Stampfer calls the company's focus on beta carotene alone a "smokescreen," adding that researchers have gathered proof that several other carotenoids protect human health. Future research may uncover still more with disease prevention prowess, he says.

The two FDA panels assigned to review olestra agreed with Procter and Gamble's favorable assessment. FDA spokesperson Brad Stone says that the majority of panel members were reasonably certain that no harm would result from approval.

One panel member who did object to the majority view said, "I don't think the advisory panel was objective from the beginning." Joan Gussow went on to tell Science News that some of the experts who spoke before the panel were consultants to Procter and Gamble but did not clearly identify themselves as such.

Stampfer also disagreed with the FDA panel. He says that Procter and Gamble's own studies show the drop in carotenoids; therefore, olestra is likely to be harmful.

Whether Procter and Gamble wins FDA approval or not, many consumers may still want the fake fat chips, despite their gastrointestinal side effects and the risk of carotenoid depletion. The scientific opposition to olestra, however, is unlikely to melt away.

C.3 Experiment Runner Manual

Dound Quick Runner
How to Use: v1.17

```
usage  java -jar DoundQRunner.jar [<options>]

Runs the specified classification experiment and records results to the console.  The
following files also get results
      File Prefix:  YYYY-MM-DD—INPUT_SRC-DR_TYPE-DISSIM_COMP-DIST_METRIC-MIND-MAXD-MINDRK-
                    MAXDRK-CLASSIFIER[MINK,MAXK]-VALIDATION_TYPE.

      Extensions: .res => results   .cm => confusion matrices     .eigs => eigenvalues
                  .dat => TDM/IPDM  .pts => 3-D points
```

Input Switches

-?, --help	display this usage information	
-d, --dr_type=TYPE	specify the type of DR to do (all, pca, mds, lle, isomap, lafon, none, or alln)	[*default=all*]
-i, --input_source=NAME	name of the corpus to work with	[*default=ScienceNews*]
-dist, --distance_metric=METRIC	distance type: euclid, or cos	[*default=cos*]
-dc, --dissim_comp=TYPE	dissimilarity type: raw, scaled, or exp	[default=raw]
-mind, --min_dimensions=INT	min # of dimensions to test	[*default=30*]
-maxd, --max_dimensions=INT	max # of dimensions to test	[*default=30*]
-C, --classifier=TYPE	either knn, linear, or quad	[*default=knn*]
-mink, --min_classifier_k=INT	min # of knn neighbors to test	[*default=9*]
-maxk, --max_classifier_k=INT	max # of knn neighbors to test	[*default=9*]
-mindrk, --min_dr_neighbors=INT	min # of neighbors to use in Isomap/LLE	[*default=10*]
-maxdrk, --max_dr_neighbors=INT	max # of neighbors to use in Isomap/LLE	[*default=10*]
-e, --embed	embed input in a Euclidean space	
-l, --leave_one_out	do leave-one-out vice 2-fold cross-validation	

Output Switches

-c, --output_to_console	only output to the console (no file logging)
-D, --print_ipdm	print the IPDM to the .dat file
-E, --print_eigvals	print eigenvalues of the decomposition to the .eig file
-M, --print_cm	print the confusion matrices to the .cm file
-P, --print_points	print truncated points (and their categories) of the decomposition to the .pts file

C.4 Groups Encoder Manual

Dound Corpus Grouping
How to Use: v1.00

usage: java -jar dound-grouper.jar PATH [<options>]

Groups documents in a corpus into a specified number of randomly generated groups which can be constrained to ensure an even distribution of documents in each category across all groups.

Order of arguments is unimportant (other than PATH, which must come first). All switches which set values expect the next argument to be the value assigned to the parameter designated by the switch.

Switches:

-n, --name=NAME	name of the group set (will be used as the filename if nothing else is specified).
-o, --output_fn=FILENAME	where to save the group set.
-e, --dont_equalize	do not equalize the groups so that each has the same number of documents from each category. *[default=do equalize]*
-g, --num_groups=INTEGER	number of groups to divide the documents up into *[default=2]*

C.5 Corpus Encoder Manual

Dound Text Encoding
How to Use: v1.10

usage: java -jar Dound.jar PATH1 [path2] [pathN] [<options>]

Computes the dissimilarity matrix, term-document matrix (TDM), or weighted TDM as specified from the documents in the specified paths.

Order of arguments is unimportant. You may specify as many paths to load documents from as you like. All switches which set values expect the next argument to be the value assigned to the parameter designated by the switch.

Input Switches:

-conf, --config=FILENAME	configuration is loaded from the specified file; all other options are ignored; input paths may be added
-R, --recursive	look in paths recursively for files
-r, --recursive_depth	set how deep to recursively look for files (-1 means no limit and is equivalent to -R) *[default=0]*
-sp, --supress_config_paths	do not input from any paths saved in the loaded configuration. Only applies if -conf is specified.

Output Switches:

-c, --console=VERBOSITY_LEVEL	how much to log to standard out. Ranges from 0 (none) to 4 (verbose). 1=status reports. 2=dissimilarity matrix only. 3=combination of 1 and 2. *[default=2]*
-F, --format=FORMAT	how to format numbers; printf style *[default=%.7f]*
-f, --file_prefix=PREFIX	prefix of the files to log to; old files in the way are renamed. *$C* will be replaced by abbreviated configuration info. *[default=no file logging]*
-S, --separator=STRING	how printed items are separated (`space` will be interpreted as a space character) *[default:tab]*
-sr, --show_rows	print row numbers on the dissimilarity matrix
-s, --sort	sort documents in ascending order
-T, --terms	log terms in the corpus if a file prefix is set
-v, --verbose	enable verbose logging to files

End-Goal Output Switches: (will print to standard out or the specified file)

note: by default, a dissimilarity matrix is computed

-tdm, --term_doc_matrix	compute the term-document matrix (word counts)
-tdmw, --term_doc_matrix_weighted	compute the weighted term-document matrix

Configuration Switches:

-dc, --dissimilarity_comp=TYPE	how to use the specified distance metric to compute dissimilarity: `RawDistance` (use metric as is), `Scaled Distance` (c-c*dist/max_dist)^(1/c) where c is the specified coefficient (usually 1 or 2), `ExpOfNegDistance` (e^-dist), or `WordCountScaled Distance` (dist/sqrt(#ofWordsInDoc1*#ofWordsInDoc2). *[default=RawDistance]*
-C, --coefficient=DOUBLE	coefficient for the `ScaledDistance` dissimilarity computer *[default=1.0]*
-dist, --distance_comp_type=TYPE	distance metric to use in the dissimilarity computer type, either: `DistanceCosineEnMass` (efficiently computes cosine distances [advisable in text mining]) or `DistanceEuclidean` *[default=DistanceCosineEnMass]*
-l, --min_word_length=INTEGER	minimum # of characters in a word *[default=1]*
-m, --min_words=INTEGER *-M, --max_words=INTEGER*	set the min/max n-Gram size *[default=1,1 (unigrams)]*
-ns, --no_stem	turns stemming off *[default=stemming on]*
-sf, --stopper_file=FILENAME	the file to get stopper words from *[default: none]*
-t, --threshold=DOUBLE	sets the weight threshold to keep words *[default=0.0]*
-w, --doc_weighter=WEIGHTER_TYPE	how to weight words in documents, either `tfidf` (uses the term-frequency inverse-document-frequency formula), `counts` (n-gram counts), or `bincounts` (uses whether or not an n-gram appears in a document [0s and 1s]) *[default=tfidf]*
-wf, --word_format=INTEGER	what constitutes a word (0=alphanumeric or numeric, 1=alphanumeric, 2=letters only) *[default=1]*

Word Usage Configuration Switches:

* = of documents a word must/may be used in to be considered (i.e. not thrown out)

-mu, --min_usages=INTEGER *-Mu, --max_usages=INTEGER*	minimum/maximum number * *[default=1]* *[default=0=no limit]*
-mup, --min_usages_percent=DOUBLE *-Mup, --max_usages_percent=DOUBLE*	minimum/maximum percentage * *[default=0.0]* *[default=1.0=100%=no max]*
-mf, --min_freq=DOUBLE *-Mf, --max_freq=DOUBLE*	minimum/maximum frequency a word must appear in the corpus *[default=0.0]* *[default=0.0=no maximum]*

clarification:

frequency	a word's total # of appearances	/ # docs in corpus
usages percent	# of docs a word appears in	/ # docs in corpus
usages	# of docs a word appears in	