Exemplar-based Kernel Preserving Embedding

by

Ahmed Elbagoury

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2016

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

With the rapid increase of available data, it becomes computationally harder to extract useful information, specially in the case of high-dimensional data. Choosing a representative subset of the data can be useful to overcome this challenge as these representatives can be used by data analysts or presented to end users to give them a grasp of the data nature and structure.

In this dissertation, first an Exemplar-based approach for topic detection is proposed, in which detected topics are represented using a few selected tweets. Using exemplar tweets instead of a set of keywords allows for an easy interpretation of the meaning of the detected topics. The approach is then extended to detect topics that emerge in new epochs of data. Experimental evaluation on benchmark Twitter datasets shows that the proposed topic detection approach achieves the best term precision. It does this while maintaining good topic recall and running times compared to other approaches for topic detection. Moreover, the proposed emerging extension achieves higher topic recall with improved running times when compared to recent emerging topic detection approaches.

To overcome the challenge of high-dimensional data, several techniques, like PCA and NMF, were proposed to embed high-dimensional data into low-dimensional latent space. However, data represented in latent space is difficult for data analysts to understand and grasp the information encoded in it. In addition, these techniques do not take the relations between the data points into account. This motivated the development of other techniques like MDS, LLE and ISOMAP which preserve the relations between the data instances, but they still use latent features. In this dissertation, a new embedding technique is proposed to mitigate the previous problems by projecting the data to a space described by few points (i.e., the exemplars) which preserves the relations between the data points. The proposed method **E**xemplar-**b**as**e**d **K**ernel Preserving (EBEK) embedding is shown theoretically to achieve the lowest reconstruction error of the kernel matrix. EBEK achieves a linear running time complexity in terms of the number of the samples. Using EBEK in the approximate nearest neighbor search task shows its ability to outperform related work by up to 60% in the recall while maintaining a good running time. In addition, empirical evaluation on clustering shows that EBEK achieves higher NMI than LLE and NMF by differences up to 40% and 15% respectively. It also achieves a comparable cluster quality to ISOMAP with a difference up to 3% in NMI and F-measure with a speedup up to 15×. In addition, our interpretability experiments show that EBEK's selected basis are more understandable than the latent basis in images datasets.

Acknowledgements

- My supervisors, Prof. Mohamed Kamel and Prof. Fakhri Karray, for the great opportunity they offered me to join the University of Waterloo, and for the freedom and trust they gave to me throughout the program.

- My professors at Alexandria University Prof. Nagwa El-Makky and Prof. Moustafa Youssef, for their great help and encourgment during my first research endeaveours.

- Waterloo professors, Prof. Ali Ghodsi, Prof. Tamer Ozsu and Prof. Christopher Batty for their great courses that helped me a lot during my research.

- Ahmed Farahat, Khaled Ammar and Hytham Abdlerhman for their advice during my first days at Waterloo.

Dedication

To the memory of my father and to my mother and my wife.

Table of Contents

List of Tables

List of Figures

List of Acronyms

ALS Alternating Least Squares

ANN Approximate Nearest Neighbor

EBEK Exemplar-based Kernel Preserving

ITQ ITerative Quantization

LDA Latent Dirichlet Allocation

LLE Locally Linear Embedding

LSH Locality-Sensitive Hashing

MDS Multidimensional Scaling

NMF Non-negative Matrix Factorization

PCA Principal Component Analysis

R1D Rank-One Downdate

SFM Soft Frequent Pattern Mining

SKLSH Shift-invariant Kernels Locality-Sensitive Hashing

SVD Singular Value Decomposition

tf-idf Term Frequency–inverse Document Frequency

Chapter 1

Introduction

1.1 Motivation

Recently, extracting useful information from a large volume of data has attracted many researchers in different areas like text, images, videos and more. Nonetheless, this large size of data has very high computational and memory demands; in addition it is hard for data analysts to have a grasp of large sized data that has many dimensions. Developing large scale techniques that select a subset of the data can approximate the whole data can be useful in many applications.

First, the selected samples can be presented to a data analyst to understand the nature of the data, or to the end-users as in topic detection task from twitter streams. Although many approaches have been proposed for topic detection from Twitter streams, they present each topic as a set of keywords that can be unrelated to each other. On the other hand, selecting a few tweets to represent the underlying topics will be more readable and interpretable by the users. Previous approaches that were developed in the literature for topic detection like [15, 1] focus on identifying terms that represent the topic regardless of how the terms can be properly connected so that they can be easily interpreted by an individual and regardless of whether or not noisy terms are included in the retrieved set. This motivates proposing a fast and accurate Exemplar-based approach to detect topics in Twitter based on representing each topic by a single tweet. This Exemplar-based representation alleviates the aforementioned problems and allows for easy understanding of the retrieved topics.

Second, this can be useful in the case of dimensionality reduction where the existing techniques, like Principal Component Analysis (PCA) [30], summarize the data by projecting it on some latent space. These latent features are difficult to interpret and may contain negative memberships. This motivated another class of techniques to emerge in the literature which is Non-negative Matrix Factorization (NMF) [34], which mitigates the negative membership problem by describing the data using features that do not contain negative values. However, NMF techniques still use latent basis making them difficult to interpret. Using exemplar-based embedding solves the aforementioned problems by projecting the data into a lower dimension space spanned by a subset of data points (i.e., the exemplars), which attains lucid features, as these features are related to explicit data points. In addition, these exemplars can be used by the data analysts to gain a better understanding of the data nature and structure.

One criterion for selecting the exemplars is minimizing the discrepancy between the original data matrix and the low rank approximation obtained by these exemplars, which is a combinatorial problem. Thus, many techniques have been proposed to solve it greedily as in [20]. One limitation of these techniques is not taking the data points relations and similarities into account, preserving such relations is shown to be effective in the similarity preserving dimensionality reduction techniques like Multidimensional Scaling (MDS) [50], Locally Linear Embedding (LLE) [47] and ISOMAP [53]. Additionally, preserving the pairwise similarities in the embedded data is very useful for the task of Approximate Nearest Neighbor (ANN) search, which is defined as finding the set of samples that have the smallest distance to a given query sample. Finding ANNs has a wide range of applications in machine learning and information retrieval [41].

In this dissertation, Exemplar-based Kernel Preserving (EBEK)) embedding is proposed to choose the exemplars that result in the best low rank approximation of the similarities of the data where the similarities are represented by the kernel matrix. In addition, formulating the problem as preserving the similarities between the data points obviates the need to solve a combinatorial problem as will be shown in the theoretical analysis. It is essential to develop techniques that can work on the kernel matrix, as not all types of data can be represented in numerical feature vectors form. For instance, there is a need to group users in social media based on their friendship relations and to group proteins in bioinformatics based on their structures [18]. Nonetheless, having only the kernel matrix but not the higher dimension representation of the data makes the development of exemplar embedding techniques more challenging. To alleviate this problem, we extend EBEK to support arbitrary kernels by inferring the needed information about the high-dimensional data from the kernel matrix.

1.2 Summary of Contributions

The contributions of the dissertation can be summarized as follows:

- Proposing an Exemplar-based approach for topic detection and extending it to detect emerging topics by introducing time slots and the notion of topic burst.
- Performing a comparative study between a wide range of various topic detection approaches.
- Performing an extensive evaluation against recent emerging topic detection approaches.
- Deriving a theoretical proof to show that Exemplar-based Linear Kernel Preserving embedding achieves the minimum reconstruction error for the kernel matrix.
- Evaluating the proposed approach in practical domains like the approximate nearest neighbors search.
- Showing the interpretability of the exemplars chosen by the proposed approach on images datasets.

1.3 Dissertation Organization

The dissertation is organized as follows. Chapter 2 provides the needed background and discusses some of the related work. Then, Chapter 3 presents the details of the Exemplar-based topic detection approach. After that, the **E**xemplar-**b**ased **K**ernel Preserving Embedding (EBEK) is proposed in Chapter 4. Finally, Chapter 5 concludes the dissertation and show some future research directions.

1.4 Notations

The following notations are used throughout the rest of the dissertation unless otherwise is stated. Scalars are denoted by small letters (e.g., m, n), sets are shown in script letters (e.g., $\mathcal{E}, \mathcal{H}$), vectors are denoted by small bold italic letters (e.g., $\boldsymbol{f}$, $\boldsymbol{g}$), and matrices are denoted by capital letters (e.g., A, S). In addition the following notations are used:

For a set $\mathcal{E}$:

$\lvert\mathcal{E}\rvert$	the size of the set.

For a vector $\boldsymbol{x} \in \mathbb{R}^{m}$:

$\boldsymbol{x}_i$	i-th element of $\boldsymbol{x}$.

For a matrix $A \in \mathbb{R}^{n\times m}$:

$A_{i,j}$	the (i, j)-th entry of A.
$A_{i,:}$	the i-th row of A.
$A_{:,j}$	the j-th column of A.
$A_{:,\mathcal{E}}$	the submatrix of A which consists of the set $\mathcal{E}$ of columns.
$A_{\eta,\mathcal{E}}$	the submatrix of A that consists of the set η of rows and the set $\mathcal{E}$ of columns.
A^T	the transpose of A.
$\tilde{A}$	the low rank approximation of A.
$\lvert\lvert A\rvert\rvert_F$	the Frobenius norm of A.

1.5 Summary

This chapter provided the motivation of this work along with the dissertation organization. The next chapter we will provide the necessary background and related work.

Chapter 2

Background and Related Work

2.1 Background

2.1.1 Low Rank Approximation

Given a data matrix $X \in \mathbb{R}^{d \times n}$, the matrix $\widetilde{X}$ is called a low rank approximation of X and can be expressed as

$$\widetilde{X} = BT$$

Where $B \in \mathbb{R}^{d \times m}$, $m \leq n$, represents the basis of the column space of the low-rank approximation matrix $\widetilde{X}$. And the elements of $T \in \mathbb{R}^{m \times n}$ represent the coefficients of the matrix $\widetilde{X}$ in the basis B.

Finding the best low-rank approximation is described as: giving a data matrix $X \in \mathbb{R}^{d \times n}$ and a positive integer k, find $\widetilde{X}$ such that:

$$\widetilde{X} = \text{arg} \min_{A, \text{rank}(A) \leq k} ||X - A||_F$$

The above equation measures the Frobenius norm of the discrepancy matrix between the original matrix and its approximation.

2.1.2 SVD

Singular Value Decomposition is a well known matrix factorization technique, where for every matrix $X \in \mathbb{R}^{d \times n}$ there exists two orthogonal matrices U and V and a diagonal matrix Σ such that $X = U \Sigma V^T$. Where, $\sigma_1 \geq \sigma_2 \geq \sigma_3 \geq ... \geq \sigma_n$ are the singular values of X.

For every matrix $X \in \mathbb{R}^{d \times n}$, $l = \min(d, n)$ and $r =$ number of non-zero eigenvalues, Singular Value Decomposition has the following properties:

- $\text{rank}(X) = r$.
- $\text{range}(X) = \text{span}(U_{:1}, U_{:2}, U_{:3} ... U_{:1n})$.
- $\text{null}(A) = \text{span}(V_{:,r+1}, V_{:,r+2}, ... V_{:,n})$.
- $||A||_2 = \sigma_1$.
- $||A||_F = \sqrt{\sigma_1^2 + \sigma_2^2 + ... + \sigma_r^2}$.

The truncated SVD can be used for obtaining rank-k approximation, which is computed as follows:

$$\widetilde{X} = U_{:,1:k} \Sigma_{k,1:k} (V_{:,1:k})^T$$

Where $U_{:,1:k}$ and $V_{:,1:k}$ are orthogonal matrices that represent the leading k left and right singular vector of the matrix X respectively and $\Sigma_{1:k,1:k}$ is a diagonal matrix with the leading k singular values on the diagonal. The product of these three matrices is known as the truncated SVD.

The low rank approximation obtained by SVD has the lowest reconstruction error in terms of Frobenius and spectral norms. More formally.

$$||X - \widetilde{X}_k||_2 = \sigma_{k+1}$$

$$||X - \widetilde{X}_k||_F = \sum_{i=1}^{r-k} \sigma_{k+i}$$

2.1.3 Stochastic Singular Value Decomposition

As computing singular value decomposition can take long time or be infeasible for a large number of data points. SVD can be performed by using stochastic singular value decomposition [27] , where stochastic singular value decomposition produces reduced a rank singular value decomposition by applying two steps:

1. Compute an approximate basis for the column space of X
    - Draw a random $n \times (k + p)$ matrix Ω

- $Y = X\Omega$ Get a subspace that approximates the columns space of X
- $Q = OrthCols(Y)$ Get an orthogonal basis for this subspace
- This is repeated q (a parameter to the algorithm) times to get better results

2. Given Q, compute approximate SVD of X

   - $[\tilde{U}, \tilde{\Sigma}, \tilde{V^T}] = svd(Q^T X)$

2.1.4 Principal Components Analysis (PCA)

Principal components analysis (PCA) [30] is a very popular technique for dimensionality reduction. It projects the data matrix into a subspace of latent features that retains the maximum variance of the data.

For a given data matrix $X \in \mathbb{R}^{d\times n}$ (n samples), the direction of maximum variation (i.e, the principal component) is given by the eigenvector associated with the largest eigenvalue of the covariance matrix $S = X^T X, S \in \mathbb{R}^{n\times n}$More formally given a data matrix $X \in \mathbb{R}^{d\times n}$.

PCA can be related to the SVD of a centered data matrix $X \in \mathbb{R}^{d\times n}$

$$X = U\Sigma V^T$$

The columns of the matrix U represents the eigenvectors of the covariance matrix X.

2.1.5 Approximate Nearest Neighbor

Approximate Nearest Neighbor (ANN) search, is defined as finding the set of samples that have the smallest distance to a given query sample. Finding ANNs has a wide range of applications in machine learning and information retrieval [41]. Both the straightforward solution, which computes the distances to all the samples and retrieve the nearest ones, and the Multi-dimensional indexing methods like k-d tree [24] are not efficient and sometimes infeasible for large dimensions. One way to alleviate the problem of large dimensions is given by Johnson-Lindenstrauss Lemma in [29], which states that the pairwise distances in small point set can be well-preserved in low-dimensional embedding. This is why several approaches have been proposed to project the data into a lower dimensional space and then utilize this lower dimensional space to compute the nearest neighbors of the data points as done in [26], [2] and [46].

2.1.6 Matrix Congruence and Sylvester's Law of Inertia

Congruent Matrices

Two square matrices A and $B \in \mathbb{R}^{n \times n}$ with real entries are said to be congruent, if there exists an invertible matrix $P \in \mathbb{R}^{n \times n}$ such that

$$A = P^T B P$$

Inertia of Matrices

The inertia of a Hermitian matrix A is defined to be the tuple

$$i(A) = \{n_+, n_0, n_-\}$$

Where, n_+ is the number of positive eigenvalues of A, n_0 is the number of zero eigenvalues of A, and n_- is the number of negative eigenvalues of A.

Sylvester's Law of Inertia

Two Hermitian matrices $A, B \in \mathbb{R}^{n \times n}$ are said to be congruent if and only if they have the same inertia.

One corollary of this theorem is that every Hermitian matrix is congruent to a diagonal matrix D with n_+ ones, n_- negative ones and n_0 zeros. In other words every Hermitian matrix $D = P^T A P$.

2.2 Related Work

First, this section covers some of the related work in the area of topic detection in subsection 2.2.1 and then covers some of the related work in the data embedding in subsections 2.2.2 and 2.2.3 respectively.

2.2.1 Topic Detection

A variety of techniques have been proposed for topic detection. One approach depends on applying matrix factorization techniques on the term-document matrix. In this approach, the Term Frequency–inverse Document Frequency (tf-idf) weighting data matrix X is factorized as $W \times H$, where W contains the membership of each tweet to each of the topics and H contains the weights of each term to each of the topics. Latent Semantic Analysis is a popular text analysis approach [33], which projects a data matrix X into a lower dimensional space whose basis are latent topics. This is done by representing the matrix X as the product of three matrices ($X = U\Sigma V^T$) using Singular Value Decomposition (SVD). SVD can take a long time or may become infeasible for a large number of data points. LSA can be performed using stochastic SVD [27] (We will refer to this approach as stochastic LSA.) Using LSA has two disadvantages: 1) The factorized matrices may have negative values which can not be easily interpreted. 2) The discovered topics are latent and do not have a clear meaning. This motivates using Non-negative Matrix Factorization (NMF) as in [4, 5], where a data matrix X is factorized to the product of two non-negative matrices $W \times H$. As the elements of the matrices W and H are non-negative, the membership of each tweet to each topic can be interpreted easily.

There are many algorithms for NMF; I will focus on two of them. The first one is ALS proposed in [4]. Alternating Least Squares (ALS) minimizes the reconstruction error of a data matrix X by minimizing: $||X - W \times H||_F$ where $||A||_F$ is the Frobenius norm of matrix A, such that the elements of W and H are non-negative. The algorithm applies a least squares step to find one matrix which is followed by another least squares step to find the other matrix in an alternating manner. The other NMF algorithm is R1D proposed in [5]. The algorithm is based on the observation that the leading singular vectors of a non-negative matrix are non-negatives which yields a rank-1 approximation. Rank-One Downdate (R1D) extends this observation to a higher rank approximation in an iterative fashion. At each iteration, the algorithm selects a rank-one sub-matrix that minimizes the reconstruction error, subtracts it from the original matrix and forces the negative residuals to zeros. However, NMF approaches describe the topic by a set of terms which results in topics that are not easily understood by the user and allows noisy terms to exist. Therefore, the proposed approach solves this problem by representing each topic by a real tweet (exemplar) which will not suffer from noisy terms as it is written by a human. In addition, real tweets can easily be understood by the user, thus users can directly understand the detected topics.

Other approaches for detecting topics are proposed by clustering the set of tweets where the tweets in the same cluster are assumed to discuss the same topic. K-means is a well known clustering algorithm. Its objective function is to minimize the sum of squares of distances between each cluster's data points and its centroid. K-means can be used for topic detection by considering that tweets in one cluster have the same topic, which is represented by the cluster centroid. On the other hand, clustering suffers from fragmented topics and representing the topic that a cluster discusses is an important task. This is solved in our approach by using exemplars (tweets) to represent the topic.

There are a number of emerging topic detection approaches that have been recently introduced in the literature like Latent Dirichlet Allocation (LDA), Document Pivot, Soft Frequent Pattern Mining (SFM) and Bngram. LDA proposed by [6] is a probabilistic topic modeling approach. It assumes that each document has a hidden distribution over the terms and that each topic has a hidden distribution over the documents. By observing the terms in each document, LDA can infer these hidden distributions. It was reported in [42, 39] that LDA has a problem with the data sparsity in short text. The document pivot approach is also used for emerging topic detection. It was introduced by [45]. Document pivot works incrementally, which means the newly arrived document is compared to the existing centroids, and if its similarity to the closest centroid is above a certain threshold, the document is assigned to this centroid. Otherwise, a new cluster is created that contains this document. Finally, clusters are sorted according to a score measure that favors clusters with new terms and the top clusters are considered as emerging topics. The main problem of the document pivot approach is that it usually generates fragmented clusters. Moreover, SFM [1] is used for emerging topic detection. Its objective is to detect patterns of co-occurrence between groups of terms and put terms together in the same topic if they usually appear together. Finally, Bngram [1] was also proposed for detecting emerging topics. Its main idea is to use Ngrams. The approach also proposed a new scoring measure based on time. This score measure focuses on choosing terms that their frequencies have increased during the current time slot compared to the previous time slots. There are many other approaches for emerging topic detection like feature pivoting and frequent pattern mining. Our proposed emerging topic detection approach inherits the same advantage of Exemplar topic detection. Thus it results in easily interpreted emerging topics and it does not suffer from selecting noisy terms. In addition, the Exemplar-based emerging topic detection approach keeps track of the topics evolution over time, while LDA, document pivot and SFM, do not track topics changes over time.

2.2.2 Data Embedding

In this subsection, I shed the light on some of the embedding techniques that have been proposed in the literature.

Locally Linear Embedding (LLE)

LLE [47] is another dimensionality reduction technique that aims to find a mapping, which preserves the local distances between the points, by trying to reconstruct the points only using their k-nearest neighbors. The algorithm starts by finding the set of nearest neighbors for each data point X_i which is denoted as $\mathcal{N}_{x_i}$. Then, it finds the cofficients w such that

$$W = \arg\min_{w} \sum_{i=1}^{n} ||X_{i:} - \sum_{j=1}^{k} w_{ij}\mathcal{N}_{x_i}(j)||^2$$

Where $\mathcal{N}_{x_i}(j)$ is the jth neighbor of the point x_i.

The weight matrix W is then used to find the optimal embedding by solving the following optimization problem

$$Y = \arg\min_{Y} \sum_{i=1}^{n} ||Y_{i:} - \sum_{j\mathcal{N}_{Y_{:i}} W_{ij} Y_{:j}} ||^2$$

It can be shown that the optimal solution is obtained by setting the columns of Y^T to the eigenvectors associated with the lowest eigenvalues of L where $L = (I - W)^T(I - W)$

Multidimensional Scaling (MDS)

Classical MDS [50] minimizes the difference between the Euclidean distances of the data points in the original space and the Euclidean distances of the projected data points in the lower dimensional space and hence maintains the relationships between the points. More formally, given a data matrix $X \in \mathbb{R}^{d\times n}$; MDS tries to find a an embedding matrix $Y \in \mathbb{R}^{p\times n}$ where $p << d$ such that Y is

$$\arg\min_{Y} \sum_{i=1}^{n}\sum_{j=1}^{n}(d_{ij}^{(X)} - d_{ij}^{(Y)})^2$$

Where $d_{ij}^{(X)} = ||X_{:i} - X_{j:}||$ and $d_{ij}^{(Y)} = ||Y_{:i} - Y_{j:}||$.

It can be shown that the solution is $Y = \Lambda^{\frac{1}{2}}V^T$, where V represents the eigenvectors of X^TX corresponding to the top d eigenvalues, and Λ is the top d eigenvalues of X^TX

ISOMAP

ISOMAP [53] adapts the same objective of MDS but using a different distance measure called geodesic distance. Geodesic distance is measured by the shortest path between the data points in a graph formed by connecting the points only to its k-nearest neighbour. The ISOMAP algorithm performs three steps

- Find the set of neighbors for each data point.
- Compute the pairwise geodesic distances between all points.
- Find the low-dimension embedding using MDS.

2.2.3 Approximate Nearest Neighbor

As computing Approximate Nearest Neighbor (ANNs) is a time and memory consuming task to be performed on the high-dimensional data. That is why, several approaches have been proposed to project the data into a lower dimensional space and then utilize this lower dimension space to compute the nearest neighbors of the data points as in [26], [2] and [46]. For example, a modified version of PCA called PCA-RR is used in [26] where the projection matrix W of PCA is multiplied by a random orthogonal matrix R and then the approach uses WR as lower dimension basis to project the data on. Yet the objective function of PCA does not preserve the similarities of the data, which limits its ability to find the best ANNs. Locality-Sensitive Hashing (LSH) [2] mitigates this problem by trying to preserve the local neighbors of the points using random hash functions that with high probability map similar data points to the same buckets. While, utilizing these buckets enables LSH to retrieve the ANNs, defining general random hash functions for LSH is a difficult task. Shift-invariant Kernels Locality-Sensitive Hashing (SKLSH) [46] modifies the objective function of LSH to approximate shift-invariant kernels using random feature mapping. ITerative Quantization (ITQ) [26] is another approach for finding ANNs, which tries to learn a similarity preserving binary coding using training data and then utilizes it to encode the data and compute the ANNs. While ITQ coding captures the data properties, it requires a lot of training data to find a good binary coding, in addition this training phase consumes a lot of time.

2.3 Summary

This chapter gave the necessary background about linear algebra, dimensionality reduction and topic detection. Then it showed some of the related work in the area of dimensionality reduction and topic detection. In the next chapter an Exemplar-based topic detection approach for Twitter Streams is proposed.

Chapter 3

Exemplar-based Topic Detection in Twitter Streams

Detecting topics in Twitter streams has been gaining an increasing amount of attention. It can be of great support for communities struck by natural disasters, and could assist companies and political parties to understand users' opinions and needs. Traditional approaches for topic detection focus on representing topics using terms and are negatively affected by length limitation and the lack of context associated with tweets. In this chapter, an Exemplar-based approach for topic detection is proposed, in which detected topics are represented using a few selected tweets. Using exemplar tweets instead of a set of keywords allows for an easy interpretation of the meaning of the detected topics. The approach is then extended to detect topics that emerge in new epochs of data. Experimental evaluation on benchmark Twitter datasets shows that the proposed topic detection approach achieves the best term precision. It does this while maintaining good topic recall and running time compared to other approaches for topic detection. Moreover, the proposed emerging extension achieves higher topic recall with an improved running time when compared to recent emerging topic detection approaches.

The rest of the chapter is organized as follows: the chapter starts by discussing some of the related work and then presenting the proposed approach for topic detection. This is followed by extending the proposed approach to detect emerging topics. After that, the implementation details are shown, experimental setup, results, and discussion.

3.1 Exemplar-based Topic Detection

Due to the text length limitation, topic detection in short text is more challenging than in long text. So most of the existing approaches are not suitable for detecting topics in short text, as the is case in LDA which will not be able to accurately infer topic distribution over tweets due to length limitations. Therefore there is a need to design new approaches for detecting topics in short text. The basic idea behind this work is to use an Exemplar-based approach to detect topics, where each detected topic is represented using the most representative tweet. This tweet (i.e., the exemplar) is much easier to be interpreted by the user as it contains related terms and it represents a topic that is of direct importance to the user.

In this section the Exemplar-based topic detection approach is proposed, then it is extended to handle the detection of emerging topics.

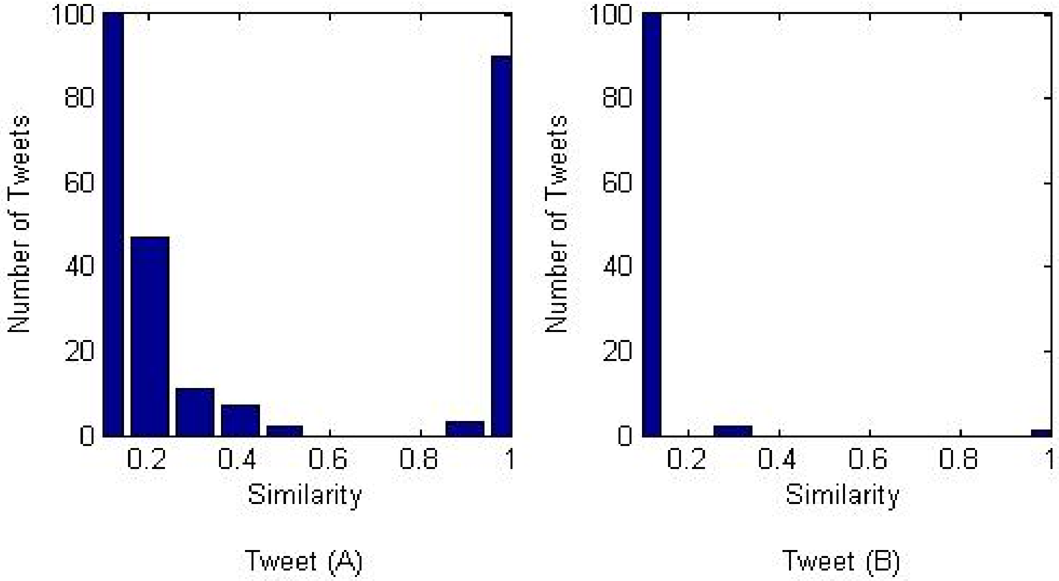


Figure 3.1: The similarity distributions of two tweets: Tweet (A) with a high variance and Tweet (B) with a low variance

3.1.1 Problem Formulation

Given a set of tweets $\mathcal{T}$ of size n, our goal is to detect the underlying topics in this set and represent each topic using only one tweet (exemplar). The selection criterion used to select this tweet should be able to detect a tweet for each topic such that each tweet is descriptive for one topic and discriminates this topic from other topics at the same time.

3.1.2 Exemplar Selection Criterion

The criterion used in this work is based on the following observation. A tweet which is similar to a set of tweets and dissimilar to the rest of the tweets is a good topic representative. This can be formulated by defining a similarity matrix $S_{n\times n}$ where S_{ij} is the similarity between tweet t_i and tweet t_j. The distribution of similarities between each tweet t_i and the rest of the tweets can be classified into three cases:

1. Tweet t_i is similar to many tweets. Therefore, its similarity distribution will have low sample variance
2. Tweet t_i is very similar to a set of tweets and less similar to the others. Therefore, its similarity distribution will have high sample variance
3. Tweet t_i is not similar to most of the other tweets. Therefore, its similarity distribution will have low sample variance

The tweets that fall in the second case are good candidates for representing topics, as each tweet is very similar to a set of tweets and therefore it can capture their underlying topic. On the other hand, each of these tweets is different from the rest of the tweets which means it can distinguish between its topic and the rest of the topics. This suggests using the variance of each tweet as a criterion for selecting topic representatives, where the sample variance of the similarities for each tweet t_i is computed as

$$\text{var}(S_{:i}) = \frac{1}{n-1}\sum_{j=1}^{n}(S_{ij} - \mu_i)^2 \ ,$$

where μ_i is the mean of similarities between t_i and other tweets:

$$\mu_i = \frac{1}{n}\sum_{j=1}^{n} S_{ij} \ .$$

Figure 3.1 supports our intuition using real tweets, where a tweet with a high variance is similar to a group of tweets that discuss the same topic and dissimilar to the rest of the tweets, while the other tweet that has a low variance is dissimilar to most of the tweets.

3.1.3 Exemplar Selection Algorithm

Choosing exemplars for detected topics can be done in an iterative manner by choosing the tweet with the highest variance in each iteration as an exemplar for a topic. One problem with this approach is that it does not guarantee the selected tweets are talking about different topics. So after choosing each exemplar, we have to remove its effect to ensure that no more tweets about the same topic will be selected as exemplars of another topic. One way to remove the effect of an exemplar t_i is to disqualify the tweets that are ϵ close to it from being exemplars, and consider the tweet that has the highest variance of similarity and is not ϵ close to t_i as the exemplar of the next topic. Figure 3.2 shows an example where each node represents a tweet and its ϵ close tweets are within the dotted circle. Tweets are sorted descendingly based on the variance of their similarities with the rest of the tweets and each tweet is labeled in accordance by this order. In this example, after choosing the first tweet as the first topic exemplar, tweets 2, 3 and 4 are not chosen as exemplars of new topics as they are very close to tweet 1 and do not represent new topics. Tweet 5 which is the tweet with the highest variance of similarity and not ϵ close to tweet 1 is chosen as an exemplar of a new topic. Similarly, tweets 6 and 7 are not chosen as exemplars while tweet 8 is chosen.

The set of exemplars $\mathcal{E}$ is constructed iteratively using the following objective function, at each iteration i:

$$\begin{aligned} &\max_{t_i \in \mathcal{T}} \text{var}(S_{:i}) \\ \text{s.t.} \quad & S_{ij} \leq \epsilon \quad \forall t_i, t_j \in \mathcal{E} \quad \textit{and} \quad i \neq j \end{aligned}$$

This objective function can be solved by iterating through the tweets in descending order of the variance of their similarities and consider the first tweet that is not ϵ-close to t_i as the exemplar of the next topic.

3.1.4 Speeding Up Calculations

Computing a similarity matrix between a large number of tweets is very complex in terms of running time and memory usage. Thus, to be able to handle large amounts of data, we approxi-

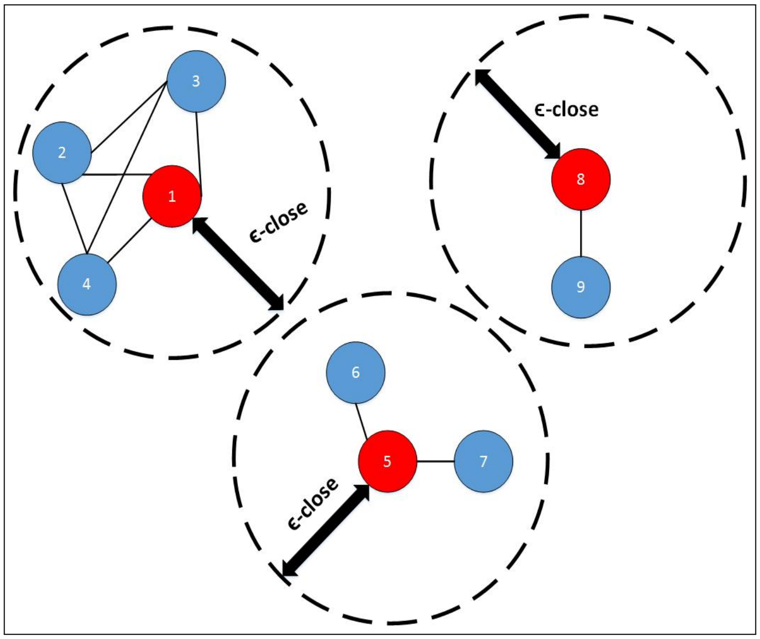


Figure 3.2: Each dotted circle shows the tweets that are ϵ close to the tweet in the center. The tweets are labeled in descending order of the variance of their similarities

mate the variance of the similarities of each tweet t_i using its similarity with fewer number m of tweets, where $m < n$. So the variance of each tweet is calculated as:

$$\mathrm{var}(S_{:i}) = \frac{1}{m-1}\sum_{j=1}^{m}(\widehat{S}_{ij} - \widehat{\mu}_i)^2$$

Where

$$\widehat{\mu}_i = \frac{1}{m}\sum_{j=1}^{m}\widehat{S}_{ij}$$

And $\widehat{S}$ is the similarity matrix between all tweets n and a random subset of size m.

It is shown empirically in the evaluation section that this approach achieves good results with acceptable run time. The pseudocode of the approach is shown in Algorithm 1.

Algorithm 1: Exemplar-based Topic Detection

```
Data: $\mathcal{T}$ set of tweets, $k$ number of topics, $m$ size of the random subset and $\epsilon$ similarity
      threshold
Result: $\mathcal{E}$ set of $k$ tweets each representing a topic
$\widehat{\mathcal{T}} \leftarrow$ select-random$(\mathcal{T}, m)$
                                                        // Select $m$ random tweets
$\widehat{S} \leftarrow$ similarity$(\mathcal{T}, \widehat{\mathcal{T}})$
$\widehat{\mathbf{v}} \leftarrow$ zeros$(n)$                                 // Vector of size n
$i \leftarrow 1$
while $i \leq n$ do
    $\widehat{\mathbf{v}}_{\mathbf{i}} \leftarrow$ var$(\widehat{S}_{:i})$
    $i \leftarrow i + 1$
$\widehat{\mathbf{v}} \leftarrow$ sort$(\widehat{\mathbf{v}}, "descending")$
$topic \leftarrow 1$
$i \leftarrow 1$
while $(topic < k)$ do
    $\mathcal{E}.add(\widehat{\mathbf{v}}_{\mathbf{i}})$
    $i \leftarrow i + 1$
    while similarity($\widehat{\mathbf{v}}_{\mathbf{i}}$, $\mathcal{E}(topic)$) $\geq \epsilon$ do
        $i \leftarrow i + 1$
    $topic \leftarrow topic + 1$
```

3.2 Emerging Topic Detection

To detect emerging topics, the notion of time slot is used to keep track of the evolution of each topic over time. Therefore, a Twitter stream is divided into time slots where each time slot contains the tweets that have occurred within its window. To detect emerging topics we need to select exemplars that: 1) Capture the underlying topics in the current time slot and 2) Differ from the topics that were discussed in the previous time slot. Based on these objectives, detecting emerging topics is completed in two steps:

- Select a candidate set $\mathcal{C}$ of tweets that represent the topics in the current time slot m
- Filter the candidate set $\mathcal{C}$ by excluding the exemplars that are discussing the same topics that appear in the previous time slot

The first step is achieved by applying the algorithm discussed in the previous section to the tweets in the current time slot. This will result in a candidate set $\mathcal{C}$ that contains candidate exemplars; these exemplars represent topics that are discussed in the current time slot m. After that, the set $\mathcal{C}$ is scanned to filter the candidate exemplars by comparing them to the topics in the previous time slot. Comparing the candidate exemplars to all the tweets in the previous time slot -to ensure and estimate its novelty- is a time consuming task. Therefore, each exemplar in the set $\mathcal{C}$ is compared to the exemplars chosen in the previous time slot $m-1$, as these exemplars summarize the topics in the previous time slot.

A notion of topic burst is needed to measure how novel the topic is. Therefore, a topic burst measure is introduced based on how the candidate topic is related to the previous topics. This is done by computing the topic similarity to topics detected in the previous time slot and choosing the topics that are not similar to the previously detected topics. As each time slot has its own dictionary and vocabulary list, it is time consuming to match the dictionaries of the different time slots to build a common dictionary. Thus, a simple similarity measure is defined as follows:

$$\text{similarity}(t_i, t_k) = \frac{t_i \cap t_k}{\mathit{max}(|t_i|, |t_k|)}$$

where

- $t_i \in \mathcal{E}_m$ and $t_j \in \mathcal{E}_{m-1}$
- $\mathcal{E}_m$ and $\mathcal{E}_{m-1}$ are the sets of selected exemplars at time slot m and $m-1$ respectively
- $t_i \cap t_k$ are the number of common terms between tweet t_i and tweet t_k
- $|t_i|$ and $|t_j|$ is the number of terms in t_i and t_j respectively

This similarity measure depends on the common words relative to the tweets lengths, which is simple and fast to calculate and does not involve relying on matching the time slot dictionaries. Therefore, the objective function is now modified for each iteration i as follows:

$$\max_{t_i \in \mathcal{T}_m} \text{var}(S_{:i,m})$$

$$\text{s.t.} \quad S_{ij,m} \leq \epsilon \quad \forall t_i, t_j \in \mathcal{E}_m \quad \text{and} \quad i \neq j$$

$$\text{s.t.} \quad \max \ \text{similarity}(t_i, t_k) \leq \delta$$

$$\forall t_i \in \mathcal{E}_m \text{ and } t_k \in \mathcal{E}_{m-1}$$

where

- $S_{:i,m}$ denotes the similarity between tweet t_i and all tweets in time slot m
- $\mathcal{T}_m$ is the set of tweets in time slot m
- δ is the maximum similarity threshold between an exemplar t_i at time slot m and an exemplar t_k in time slot $m-1$

Therefore, to solve the previous objective function, at each iteration i the tweet with the highest variance that is not similar to the previously selected $i-1$ tweets (topics) and not similar to topics detected at the previous time slot is selected. This will result in detecting emerging topics which have suddenly gained peoples' attention.

3.3 Experimental Results

Three Twitter datasets were used for evaluation in this chapter and were initially collected by [1]. The datasets correspond to three distinct events which include FAcup (39,282 tweets), Super Tuesday (707,300 tweets) and US Elections (1,157,674 tweets). I have re-constructed the three datasets as only tweet ids were provided by [1]. The ground truth topics of the datasets were constructed from news headlines reported during the events. Data is preprocessed using TMG Matlab tool [1] to remove stop words and convert it to TF-IDF representation.

The proposed topic detection approach is compared against five topic detection approaches, which are: Latent Semantic Analysis (LSA), stochastic LSA, Alternating Least Squares (ALS), Rank-1 Downdate (R1D) and K-means. In addition, the extension of emerging topic detection is compared to four recent emerging topic detection approaches, which are: Latent Dirichlet Allocation (LDA), Document Pivot, Soft Frequent Pattern Mining (SFM) and Bngram.

As topics in each time slot were represented by keywords, labels for each tweet were not provided. I have used the same measures and evaluation code used by [1] to evaluate the different topic detection approaches. These measures are:

- Topic recall: The number of successfully retrieved topics divided by the total number of topics that should have been retrieved.
- Term precision: The number of successfully retrieved keywords in the detected topics divided by the total number of keywords in these topics.

[1] http://scgroup20.ceid.upatras.gr:8000/tmg/ [Last visit 14/02/2016]

- Term recall: The number of successfully retrieved keywords divided by the total number of keywords that should have been retrieved.

Topic precision was not used by [1] as not all the topics covered by Twitter appear in news sources. For the other topic detection approaches, the top 15 keywords of each discovered topic is used as topic keywords. While in Exemplar-based approach, each exemplar (tweet) is used as a topic and its terms are used as topic representatives. The Exemplar-based approach uses a random set to approximate the similarity matrix. Therefore, the Exemplar approach was run 10 times.The average and the 95% confidence intervals of the results were reported in Figures 3.3a to 3.4c. Also, the similarity measure used was the cosine similarity and the size of the random subset of tweets m used by the Exemplar approach was set to 1000.

Figures 3.3a, 3.3b and 3.3c show the evaluation measures of applying different topic detection approaches in FAcup, Super Tuesday and US Elections datasets respectively. Exemplar-based was applied by setting the similarity threshold ϵ to 0.5 in FAcup, to 0.01 in US Elections and to 0.1 in Super Tuesday, where these values were tuned empirically. Moreover, Table 3.1 shows the running time for the topic detection approaches. For FAcup, LSA and stochastic LSA had the least term precision. While, Exemplar approach increased the term precision with some loss in term recall as it reduces the noise in the terms by enforcing the topics to be real tweets. Moreover, LSA and stochastic LSA had the least topic recall. However, the rest of the approaches had close topic recall values. For running time, Exemplar approach, K-means, LSA and stochastic LSA had a close running time. Note that, we were unable to run ALS on the US Elections and Super Tuesday datasets as it has a huge running time. In Super Tuesday dataset, Exemplar provided again the best term precision. Moreover, Exemplar was the best in term recall and topic recall. For running time, Exemplar and K-means approaches were the fastest. Finally, for the results of US Elections dataset, Exemplar also reached the best term precision. For topic recall, Exemplar was the best or second best in most cases, then LSA stochastic was the best in the first time slots. Term recall results were comparable for all the used approaches. For the running time, Exemplar approach and K-means were again the fastest.

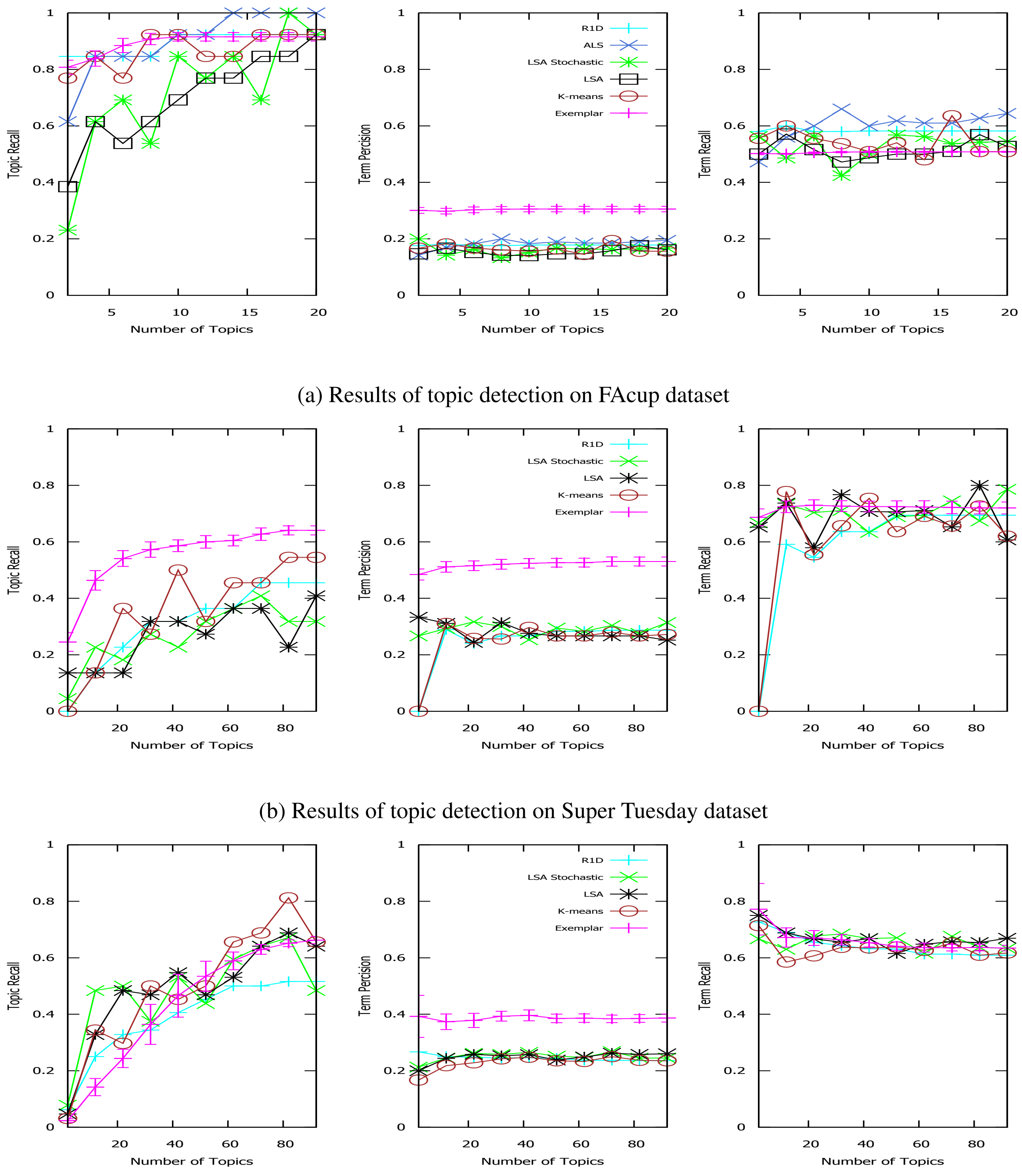
Figures 3.4a, 3.4b and 3.4c show the evaluation measures of applying different emerging topic detection approaches in FAcup, Super Tuesday and US Elections datasets respectively. The δ parameter of the Exemplar approach was set to 0.5 in all datasets. In addition, Table 3.2 shows the running time for the emerging topic detection approaches. For FAcup, LDA had the best topic recall, while Exemplar was the second best in topic recall and term precision. For term recall, Bngaram and SFM had the best term recall, however they were able to detect a few number of correct topics which was reflected in the topic recall results. For running time, again Exemplar approach was the fastest. In Super Tuesday dataset, Exemplar provided again the best topic recall and term recall and was the third in term precision after Bngram and SFM which again had a

Table 3.1: Running time in seconds for topic detection approaches

Dataset	**Number of Topics (N)**	**R1D**	**ALS**	**LSA**	**LSA Stochastic**	**K-means**	**Exemplar**
FAcup	**N = 2**	1.3133	38.3629	0.4229	0.2737	**0.0957**	0.8575
	N = 10	6.5546	104.2447	0.5111	0.4613	**0.1887**	0.9226
	N = 20	14.6239	220.6417	0.6565	0.9482	**0.3909**	0.8290
Super Tuesday	**N = 12**	114.7866	-	8.3473	25.7385	**6.4527**	15.0808
	N = 52	450.0996	-	38.5653	171.9562	27.8256	**20.5971**
	N = 92	732.1874	-	103.5337	387.4809	50.1507	**24.7621**
US Elections	**N = 12**	305.5000	-	14.4126	28.4466	**8.9827**	28.4967
	N = 52	1081.3000	-	67.8509	140.4287	42.8705	**38.6812**
	N = 92	1796.5000	-	209.6554	418.7506	63.8388	**45.5971**

low topic recall. For running time, SFM and Exemplar approach were the fastest approaches. However, Bngram approach was the slowest one in all the three datasets. Finally, for the results of US Elections dataset, Exemplar had the best topic recall and term recall. However, it was the third in term precision after Bngram and SFM which both had a low topic recall in most cases. For the running time, SFM and the Exemplar approach were the fastest. Moreover, our proposed topic detection and emerging topic detection approaches are stable, where increasing the number of topics either increases or maintains the same quality (topic recall, term precision and term recall). While, the other approaches show unstable behavior with the number of topics as their curves keep oscillating.

Table 3.3 shows sample topics detected by our proposed Exemplar approach and LDA in the three different events. As shown in the table, the detected topics by our approach can be easily interpreted and understood by the user while LDA detected topics are hard to interpret and usually contain noisy confusing terms.

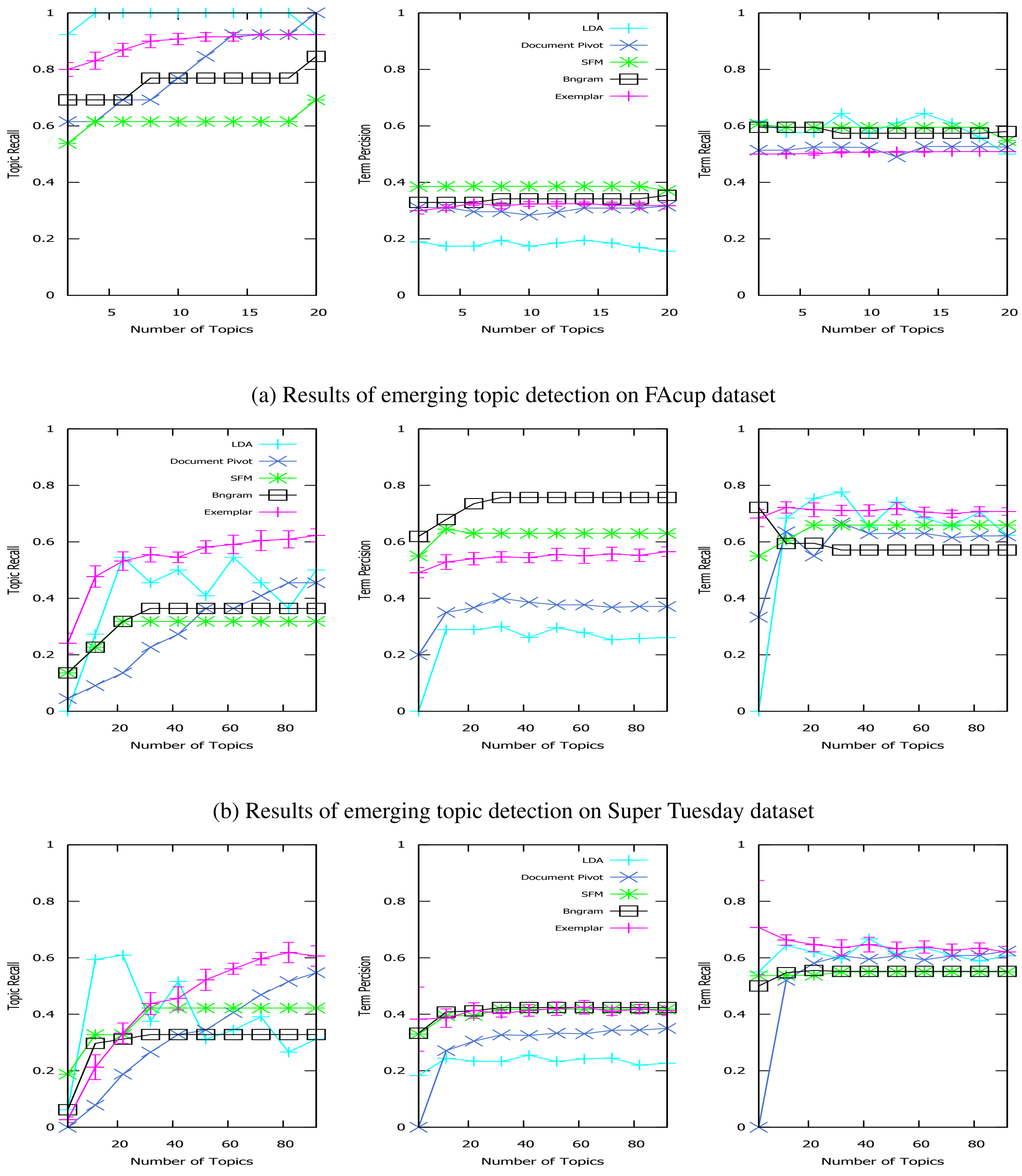


(a) Results of topic detection on FAcup dataset

(b) Results of topic detection on Super Tuesday dataset

(c) Results of topic detection on US Elections dataset

Figure 3.3: Results of topic detection



(a) Results of emerging topic detection on FAcup dataset

(b) Results of emerging topic detection on Super Tuesday dataset

(c) Results of emerging topic detection on US Elections dataset

Figure 3.4: Results of emerging topic detection

Table 3.2: Running time in seconds for emerging topic detection approaches

Dataset	**Number of Topics (N)**	**LDA**	**Document Pivot**	**SFM**	**Bngram**	**Exemplar**
FAcup	**N = 2**	3.3550	8.2910	28.7680	266.3780	**1.0102**
	N = 10	4.4020	8.2910	28.7680	266.3780	**1.5930**
	N = 20	4.2670	8.2910	28.7680	266.3780	**2.4871**
Super Tuesday	**N = 12**	103.0670	235.6230	49.6520	613.5750	**30.8725**
	N = 52	122.4840	235.6230	**49.6520**	613.5750	71.7208
	N = 92	144.1410	235.6230	**49.6520**	613.5750	122.2432
US Elections	**N = 12**	114.8130	156.5290	78.4430	1436.2410	**76.9178**
	N = 52	143.6980	156.5290	**78.4430**	1436.2410	182.4017
	N = 92	172.4850	156.5290	**78.4430**	1436.2410	285.6489

3.4 Summary

In this chapter, an Exemplar-based approach for topic detection in Twitter streams is proposed. The approach represents each topic using explicit data sample. The approach is extended to detect emerging topics in new epochs of data. The effectiveness of the approach is shown using three different datasets. In the next chapter EBEK, an Exemplar-based Kernel Preserving Embedding approach, is proposed.

Table 3.3: Sample topics detected by Exemplar-based and LDA approaches

Topics	Approach	FAcup	US Elections	Super Tuesday
Topic 1	Exemplar	RT @chelseafc: We've kicked off #CFCWembley #FACupFinal (SL)	RT @AP: AP RACE CALL: Obama wins Vermont; Romney wins Kentucky. #Election2012	BREAKING NEWT: Gingrich wins Georgia Republican primary (AP)
	LDA	chelsea liverpool final kick la sl kicked cup wembley de game reds fa anthem win	vote voted america president line years election voting time today make tonight good de en	mitt romney georgia newt ohio virginia gingrich oklahoma iran obama car santorum tennessee energy drive
Topic 2	Exemplar	Yellow card to Mikel #FACupFinal	RT @AP: AP RACE CALL: Romney wins North Carolina. #Election2012	RT @thinkprogress: Mike Allen reports (on @politico livestream) that Romney campaign ...
	LDA	final chelsea la cup liverpool de fa wembley el en red comienza ya fc phil	obama votes electoral carolina romney indiana kentucky north projected south florida vermont called win red	santorum ohio romney exit polls cnn rick poll voters gingrich exits winning show em republicans
Topic 3	Exemplar	RT @chelseafc: 2nd half kicked off #CFCWembley #FACupFinal (SL)	RT @TheEllenShow: What an amazing night. Congratulations @BarackObama! I'm proud of our country.	RT @AP: BREAKING NEWS: Romney wins Republican presidential primary in Ohio
	LDA	fans anthem national reds blues great easy booing players trending ramirez eh gaga lady direction	voting time va tomorrow work great lead white choice house start republicans clear predicting half	peyton tennessee manning colts breaking tomorrow strong character important moral money tonight won announce end

Chapter 4

EBEK: Exemplar-based Kernel Preserving Embedding

This chapter proposes Exemplar-based Kernel Preserving (EBEK) embedding to choose the exemplars that result in the best low rank approximation of the similarities of the data where the similarities are represented by the kernel matrix. In addition, formulating the problem as preserving the similarities between the data points obviates the need to solve a combinatorial problem as will be shown in our theoretical analysis. It is essential to develop techniques that can work on the kernel matrix, as not all types of data can be represented in numerical feature vectors form. For instance, there is a need to group users in social media based on their friendship relations and to group proteins in bioinformatics based on their structures[18].

Nonetheless, having only the kernel matrix but not the higher dimension representation of the data makes the development of exemplar embedding techniques more challenging. To alleviate this problem, we extend EBEK to support arbitrary kernels by inferring the needed information about the high-dimensional data from the kernel matrix.

The rest of the chapter is organized as follows: Section 4.1 shows the details of the proposed Exemplar-based Kernel Preserving embedding. Then, experimental evaluations are shown in section 4.2.

4.1 EBEK: Exemplar-based Kernel Preserving Embedding

We start this section by providing the details of embedding that preserves linear kernels in subsection 4.1.1 and then we explain the details of extending the approach to support arbitrary kernels in subsection 4.1.2.

4.1.1 Exemplar-based Linear Kernel Preserving Embedding

Our objective in this work is choosing a subset of columns (i.e, data points) that preserve the pairwise similarities as much as possible between the data embedded in the span of these columns. In addition, we would like these columns to be less similar to each other to ensure that these columns capture the different characteristics of the dataset. Given this objective the problem can be defined as follows.

Problem Definition Given a data matrix $A \in \mathbb{R}^{d \times n}$ (n samples in d dimensional space). Select a subset $\mathcal{E}$ of m columns, such that:

$$\text{arg} \min_{A_{:,\mathcal{E}}, T} ||S - \tilde{S}||_F = \text{arg} \min_{A_{:,\mathcal{E}}, T} ||A^T A - \tilde{A}^T \tilde{A}||_F$$

$$\text{s.t. } S(i,j) \leq \epsilon \quad \forall i,j \in \mathcal{E} \tag{4.1}$$

Where $\tilde{A}$ is the low rank approximation of A using the columns $A_{:,\mathcal{E}}$, $\tilde{A} = A_{:,\mathcal{E}} T$, $A_{:,\mathcal{E}} \in \mathbb{R}^{d \times m}$ and $T \in \mathbb{R}^{m \times n}$. T represents the coefficients used to reconstruct the n samples using the m selected samples and ϵ is a similarity threshold to ensure that the columns are not similar to each other.

At first we will drop the constrains on the columns similarities and try to minimize the objective function and then we will show how to minimize the objective function while preserving these constrains.

The goal of this objective function is to choose $A_{:,\mathcal{E}}$ and T that minimize the Frobenius norm of the difference between the pairwise similarity matrix of the original data ($S = A^T A$) and the pairwise similarity matrix of the low rank approximation data ($\tilde{S} = \tilde{A}^T \tilde{A}$), where the similarity here is defined by the linear kernel. Therefore, the problem in equation 4.1 can be reduced to:

$$\text{arg} \min_{A_{:,\mathcal{E}}, T} ||A^T A - \tilde{A}^T \tilde{A}||_F = \tag{4.2}$$

$$\text{arg} \min_{A_{:,\mathcal{E}}, T} ||A^T A - T^T A_{:,\mathcal{E}}^T A_{:,\mathcal{E}} T||_F$$

Let $S_{\mathcal{E},\mathcal{E}} = A_{:,\mathcal{E}}^T A_{:,\mathcal{E}} \in \mathbb{R}^{m\times m}$, which represents the pairwise similarities between the selected m samples. Then, equation 4.2 can be rewritten as:

$$\arg \min_{A_{:,\mathcal{E}},T} ||S - T^T S_{\mathcal{E},\mathcal{E}} T||_F \tag{4.3}$$

As the matrix S is symmetric positive semi-definite (by construction), then $S = V\Sigma^2 V^T$, where $A = U\Sigma V^T$, is the singular value decomposition of A. In addition, Σ is a diagonal matrix with $\texttt{rank}(S)$ positive elements and $n - \texttt{rank}(S)$ zero elements on the diagonal. Then equation 4.3 can be written as:

$$\arg \min_{A_{:,\mathcal{E}},T} ||V\Sigma^2 V^T - T^T S_{\mathcal{E},\mathcal{E}} T||_F \tag{4.4}$$

Lemma 4.1.1. *$||GBQ||_F = ||B||_F$ for any matrix B and orthogonal matrices G and Q.*

Proof.

$$\begin{aligned} ||GBQ||_F^2 &= tr((GBQ)^T GBQ) \\ &= tr(Q^T B^T G^T GBQ) \\ &= tr(QQ^T B^T G^T GB) = tr(B^T B) = ||B||_F \end{aligned}$$

As Q and G are orthogonal matrices, hence $QQ^T = I$ and $G^T G = I$.

□

Based on lemma 4.1.1, equation 4.4 can be re-written as:

$$\begin{aligned} &\arg \min_{A_{:,\mathcal{E}},T} ||V^T(V\Sigma^2 V^T - T^T S_{\mathcal{E},\mathcal{E}} T)V||_F \\ &= \arg \min_{A_{:,\mathcal{E}},T} ||\Sigma^2 - V^T T^T S_{\mathcal{E},\mathcal{E}} TV||_F \\ &= \arg \min_{A_{:,\mathcal{E}},T} ||\Sigma^2 - (TV)^T S_{\mathcal{E},\mathcal{E}} TV||_F \end{aligned} \tag{4.5}$$

Lemma 4.1.2. *$S_{\mathcal{E},\mathcal{E}} = P^{-T} D P^{-1}$, where $P \in \mathbb{R}^{m\times m}$ is an invertible matrix and $D \in \mathbb{R}^{m\times m}$ is a diagonal matrix that has only entries 0 and $+1$. The number of $+1$ in D equals r, where r is the rank of matrix $S_{\mathcal{E},\mathcal{E}}$.*

Proof. Using Sylvester's Law of Inertia [51], each symmetric matrix $E \in \mathbb{R}^{m\times m}$ is congruent to a diagonal matrix $D \in \mathbb{R}^{m\times m}$ which has only entries 0, $+1$ and -1 along the diagonal, where the number of zero diagonal elements is $m-p$, $p = \text{rank}(E)$, the number of positive diagonal elements, q, is the number of positive eigenvalues, the number of negative diagonal elements is the number of negative eigenvalues $p-q$. Which means that there exists an invertible matrix $P \in \mathbb{R}^{m\times m}$ such that: $P^T E P = D$. Applying this to the matrix $S_{\mathcal{E},\mathcal{E}}$ gives the following: $P^T S_{\mathcal{E},\mathcal{E}} P = D$, then

$$S_{\mathcal{E},\mathcal{E}} = P^{-T} D P^{-1} \tag{4.6}$$

As the matrix $S_{\mathcal{E},\mathcal{E}}$ is symmetric positive semi-definite, then it has r positive eigenvalues, where $r = \text{rank}(S_{\mathcal{E},\mathcal{E}}) = \text{rank}(A_{:,\mathcal{E}})$, and $m-r$ zero eigenvalues. The matrix P can be obtained by multiplying pairs of elementary transformations, one of which is with rows and the other is the corresponding transformation with the columns as explained in [35]. □

Theorem 4.1.3. *By setting $T = P(\Sigma_{1:m,:})V^T$, where P satisfies equation 4.6 and selecting a subset $\mathcal{E}$ of columns from matrix A that have the highest rank, the matrix $\tilde{S}$, which equals to $T^T A^T_{:,\mathcal{E}} A_{:,\mathcal{E}} T$, achieves the minimum low rank approximation of S.*

Proof. Using lemma 4.1.2, and by substituting equation 4.6 in equation 4.5, we get:

$$\begin{aligned} &\arg\min_{A_{:,\mathcal{E}},T} ||\Sigma^2 - (TV)^T P^{-T} D P^{-1} TV||_F \\ &= \arg\min_{A_{:,\mathcal{E}},T} ||\Sigma^2 - (P^{-1}TV)^T D P^{-1} TV||_F \end{aligned} \tag{4.7}$$

Our objective is to put the matrix D in canonical form such that:

$$D = \begin{bmatrix} I_r & & & \\ & 0 & & \\ & & \ddots & \\ & & & 0 \end{bmatrix} \tag{4.8}$$

Where I_r is r-by-r identity matrix. As the singular values of S are sorted along the diagonal of Σ, putting the matrix D in the form of equation 4.8 enables us to cancel the first r singular values (the largest ones), which means the error of equation 4.7 in terms of the Frobenius norm will be $\sqrt{\sum_{i=r+1}^{n} \sigma_i^4}$, where σ_i^2 is the i^{th} singular value of S. This can be achieved by setting the value $P^{-1}TV = (\Sigma_{1:m,:})$, where $\Sigma_{1:m,:}$ is the first m rows of the matrix Σ. This can be seen by substituting the value of $P^{-1}TV$ in equation 4.7 which will be:

$$\arg\min_{A_{:,\mathcal{E}},T} ||\Sigma^2 - (\Sigma_{1:m,:})^T D(\Sigma_{1:m,:})||_F \tag{4.9}$$

The term $(\Sigma_{1:m,:})^T D(\Sigma_{1:m,:})$ in equation 4.9 is:

.

$$\begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_m \\ 0 & \dots & 0 \\ \vdots & \ddots & \\ 0 & \dots & 0 \end{bmatrix} \begin{bmatrix} I_r & & & \\ & 0 & & \\ & & \ddots & \\ & & & 0 \end{bmatrix} \begin{bmatrix} \sigma_1 & 0 & & \dots & \dots & 0 \\ & \ddots & & & & \\ & & \sigma_m & 0 & \dots & 0 \end{bmatrix}$$

$$= \begin{bmatrix} \sigma_1^2 & & & & & & \\ & \sigma_2^2 & & & & & \\ & & \ddots & & & & \\ & & & \sigma_r^2 & & & \\ & & & & 0 & & \\ & & & & & \ddots & \\ & & & & & & 0 \end{bmatrix}$$

Thus, the optimal value for $P^{-1}TV = (\Sigma_{1:m,:})$, and $T = P(\Sigma_{1:m,:})V^T$. The error in this case is equal to the minimum achieved error using rank-k approximation obtained by SVD [60]. So to minimize 4.7, we need to:

- Choose subset of columns $\mathcal{E}$ from A that have the maximum rank.
- Set the value of T to $P(\Sigma_{1:m,:})V^T$.

□

To maximize the rank of $A_{:,\mathcal{E}}$, there are two cases:

- If $r \geq m$, any independent m columns can be chosen by reducing the matrix to its echelon column form and use the non-zero columns.
- If $r < m$, in this case the non-zero columns of the echelon form and any other $m - r$ columns are used, and in this case the error will be zero.

Recall that $\tilde{A} = A_{:,\mathcal{E}}T$. To obtain the lower dimension embedding of the data in the space spanned by $A_{:,\mathcal{E}}$ the matrix $A_{:,\mathcal{E}}$ is replaced by its QR factorization. $\tilde{A} = QRT$, where Q is the orthogonal bases of the space spanned by $A_{:,\mathcal{E}}$ and the lower dimension embedding of the data is RT.

Until now, we have only considered minimizing the objective function without the similarities constrains. As shown in the previous proof, choosing any subset of columns that has the maximum rank will be optimal for the objective function. Therefore, to decide which subset to choose, we employ the similarities constrains and choose a subset of columns with the maximum rank, such that the pairwise similarities between these columns are upper-bounded by a similarity threshold ϵ that can be chosen empirically.

Algorithm 2 shows the pseudo code of the algorithm. The method `getIndependentcol` returns m independent columns. This can be computed using echelon form. However, the set of independent columns can be computed more efficiently using algorithm 3. The algorithm starts with an arbitrary column, as the first independent column (line 2), then for each subsequent column j check if it has component orthogonal to the previously chosen columns or not (lines 5 to 7); if it has it will be included in the set of independent columns (lines 8 to 10).

EBEK running time complexity is $O(dnlogm + (d+n)m^2 + nmd + m^3)$, where $O(dnlogm + (d+n)m^2)$ is the time to compute the stochastic SVD decomposition of A, $O(nmd)$ for the independent columns selection and $O(m^3)$ for computing the matrix P.

Algorithm 2: Linear Kernel Preserving Embedding

```
Data: Matrix $A \in \mathbb{R}^{d \times n}$ and integer $m$
Result: $W \in \mathbb{R}^{m \times n}$, which represents the lower dimension embedding of the data
$[\Sigma, V] \leftarrow$ stochasticSVD($A$, $m$)
$\mathcal{E} \leftarrow$ getIndependentcol($A, m$)
$S_{\mathcal{E},\mathcal{E}} \leftarrow A^T_{:,\mathcal{E}} A_{:,\mathcal{E}}$
$P \leftarrow$ diagMatrix($S_{\mathcal{E},\mathcal{E}}$)
$T \leftarrow P \Sigma V^T$
$[Q, R] =$ orthogonalize($A_{:,\mathcal{E}}$)
$W \leftarrow RT$
```

4.1.2 Exemplar-based Kernel Preserving Embedding

The previous section illustrated the idea of how to choose the subset of columns in order to preserve the linear kernel similarities. In this section, we generalize our approach to work with

Algorithm 3: getIndependentcol: Independent Columns Selection

```
**Data:** Matrix $A \in \mathbb{R}^{d \times n}$, integer $m$
**Result:** $\mathcal{E}$ a set of indexes of $m$ independent columns in matrix $A$
size $\leftarrow 1$
$\mathcal{E} \leftarrow \{1\}$
**for** $i = 2$ : *min*$(m, n)$ **do**
    $a_i \leftarrow A_{:,i}$
    **for** $j = 1$ : *size* **do**
        $a_j \leftarrow A_{:,\mathcal{E}(j)}$
        $a_i \leftarrow a_i - \frac{<a_i,a_j>}{<a_j,a_j>} a_j$
    **if** $||a_i||_1 \neq 0$ **then**
        size $\leftarrow$ size $+ 1$
        $\mathcal{E} \leftarrow \mathcal{E} \cup i$
```

any type of kernel matrix K.

Recall that in general $K = \phi(A)^T \phi(A)$, where $\phi(A)$ is the representation of the matrix A in the high-dimensional space which is defined by the kernel function. Let the SVD decomposition of $\phi(A)$ be $U \Sigma V^T$, then the SVD decomposition of K equals $V \Sigma^2 V^T$ and hence the V and Σ can be computed in the first step of the algorithm using the SVD decomposition of K.

The objective of the second step of algorithm 2 is retrieving $m \times m$ submatrix of K that has the maximum rank. While this can be accomplished using the matrix A in the case of linear kernel, it can not be done in the same way in the case of general kernels, as the matrix $\phi(A)$ is not known. However, this objective is achieved by iteratively eliminating set of columns and rows from K, such that the rank of K is maximized upon the elimination each time, which maximizes the rank of $\phi(A)$, as $\texttt{rank}(K) = \texttt{rank}(\phi(A))$. To show the process of eliminating these columns and rows, we first introduce the following lemma.

Proposition 1. *For any vector $\mathbf{x_i} \in \mathbb{R}^n$ linearly depends on $\mathbf{x_j} \in \mathbb{R}^n$, sub($\mathbf{x_i}, \mathcal{M}$) linearly depends on sub($\mathbf{x_j}, \mathcal{M}$), where sub($\mathbf{x}, \mathcal{M}$) function returns a vector in $\mathbb{R}^{|\mathcal{M}|}$ containing elements indexed by the set $\mathcal{M}$ of $\mathbf{x}$.*

Proof. As $\mathbf{x_i}$ linearly depends on $\mathbf{x_j}$, then there exists scalars a_i and a_j not equal to zero, such that: $a_i \mathbf{x_i} + a_j \mathbf{x_j} = \mathbf{0}$ Applying the `sub` function to both sides of the previous equation and noting that function sub attains the superposition principle, we get:

$$\texttt{sub}(a_i \mathbf{x_i} + a_j \mathbf{x_j}, \mathcal{M}) = \texttt{sub}(\mathbf{0}, \mathcal{M})$$

Algorithm 4: diagMatrix: Diagonalize the input matrix

```
Data: Matrix $S \in \mathbb{R}^{m \times m}$
Result: $P$
$I \in R^{m \times m}$ identity matrix
for $i = 2 : m$ do
    $D_{i:} = S_{i:} / \sqrt{|S_{i,i}|}$
    $D_{:i} = S_{:i} / \sqrt{|S_{i,i}|}$
    $I_{i:} = I_{i:} / \sqrt{|S_{i,i}|}$
    for $j = i + 1 : m$ do
        mult $= -\frac{D_{j,i}}{D_{i,i}}$
        $D_{j:} =$ mult $\times D_{i:} + D_{j:}$
        $D_{:j} =$ mult $\times D_{:i} + D_{:j}$
        $I_{j:} =$ mult $\times I_{i:} + I_{j:}$
    $P = I^T$
```

$$\texttt{sub}(a_i\mathbf{x_i}, \mathcal{M}) + \texttt{sub}(a_j\mathbf{x_j}, \mathcal{M}) = \texttt{sub}(\mathbf{0}, \mathcal{M})$$

$$a_i\texttt{sub}(\mathbf{x_i}, \mathcal{M}) + a_j\texttt{sub}(\mathbf{x_j}, \mathcal{M}) = \texttt{sub}(\mathbf{0}, \mathcal{M})$$

Therefore, the vector $\texttt{sub}(\mathbf{x_i}, \mathcal{M})$ linearly depends on the vector $\texttt{sub}(\mathbf{x_j}, \mathcal{M})$.

□

The process starts by selecting an arbitrary column $K_{:,i}$ as the first independent column, then all columns $K_{:,j}, \forall j \in \mathcal{C}$ are eliminated, where $\mathcal{C}$ is the set containing all columns that linearly depend on $K_{:,i}$. Removing such columns does not reduce the rank of the remaining submatrix. To maintain the symmetry of the remaining matrix all the rows $K_{j,:}, \forall j \in \mathcal{C}$ should be eliminated too. As $K_{l,:} = K_{:,l}, \forall i \in \{1, 2, \dots n\}$ (by the symmetry of K), then the rows $K_{j,:}, \forall j \in \mathcal{C}$ are linearly depending on the row $K_{i,:}$. By proposition 1, the rows $\texttt{sub}(K_{j,:}, n-1), \forall j \in \mathcal{C}$ are linearly depending on the row $\texttt{sub}(K_{i,:}, n-1)$ too. Thus, the rank of the remaining submatrix after the elimination of these rows is not affected. Note that, the columns $\texttt{sub}(K_{:,j}, n-1), \forall j \in \mathcal{C}$, that were removed, are linearly depending on the column $\texttt{sub}(K_{:,i}, n-1)$ by the same lemma. The columns in the remaining submatrix do not depend on the column $K_{:,i}$, which means we can choose one of them to be the second independent column, and then repeat the process of eliminating the columns depending on it and their corresponding rows. The process terminates when either we achieve the size of the desired submatrix or we end up with submatrix of smaller size, which means that the original matrix has a rank less than m; in this case we can pad the remaining submatrix with dependent columns and its corresponding rows.

Algorithm 5: Kernel Preserving Embedding

```
Data: Matrix $K \in \mathbb{R}^{n \times n}$ and integer $m$
Result: $T \in \mathbb{R}^{m \times n}$, which represents the lower dimension embedding of the data
$[\Sigma^2, V] \leftarrow$ stochasticSVD($K$, $m$)
$\mathcal{E} \leftarrow$ getIndependentcolKernel$(K, m)$
$P \leftarrow$ diagMatrix$(K_{\mathcal{E},\mathcal{E}})$
$T \leftarrow P \Sigma V^T$
```

Algorithm 6: Independent Columns Selection for Kernel Preserving Embedding

```
Data: Matrix $K \in \mathbb{R}^{n \times n}$, integer $m$
Result: $\mathcal{E}$ a set of indexes of $m$ independent columns in matrix $K$
$\mathcal{E} \leftarrow \{\}$ // Set of independent columns
$\mathcal{N} \leftarrow$ set of all columns
for $i = 1 : min(m, n)$ do
    $i \leftarrow$ select column of $K$ that is in $\mathcal{N}$ and not in $\mathcal{E}$
    $\mathcal{E} \leftarrow \mathcal{E} \cup k_i$
    for $j = 1 : n$ do
        if column $j \notin \mathcal{N}$ or $j == i$ then
            continue
        $K_{\mathcal{N},j} \leftarrow K_{\mathcal{N},j} - \frac{<K_{\mathcal{N},i},K_{\mathcal{N},j}>}{<K_{\mathcal{N},i},K_{\mathcal{N},i}>} K_{\mathcal{N},i}$
        if $||K_{\mathcal{N},j}||_1 = 0$ then
            $\mathcal{N} \leftarrow \mathcal{N} \setminus j$
```

4.2 Experimental Results

The effectiveness of the proposed approach is evaluated on two tasks, approximate nearest neighbor search and clustering. Section 4.2.1 shows the setup and results for the ANN search task, and section 4.2.2 shows the experiments setup and results for the clustering task. In the rest of this section EBEK using linear kernel is referred to as EBEK; if other kernels are used, it will be stated explicitly. Then, Section 4.2.3 presents interpretability experiments to show that the exemplars chosen by EBEK are interpretable.

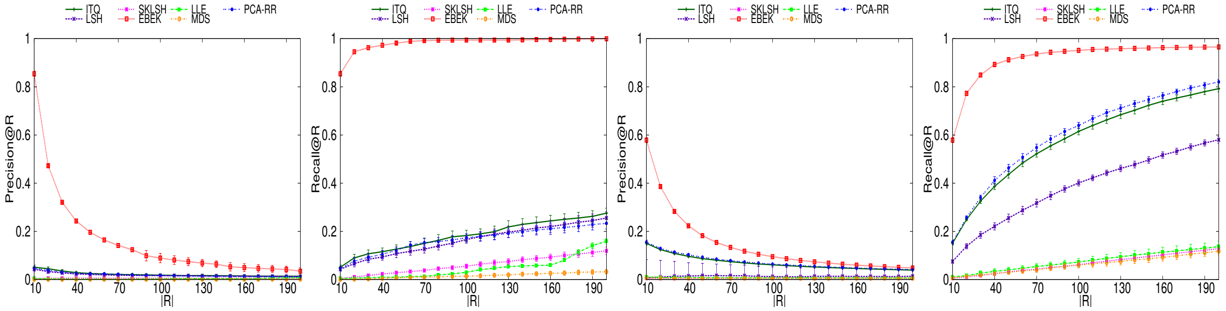
4.2.1 Approximate Nearest Neighbor Search

This subsection discusses the experimental setup and results for the task of ANNs search. To evaluate the effectiveness of our approach, the pairwise similarities between the lower dimension data is computed and the nearest neighbors are retrieved based on the lower dimension embedding.

To build the ground truth, the set $\mathcal{T}$ of nearest neighbors is retrieved by computing the distance to all queries and then applying the linear scan. The search quality for each approach is measured using Recall@$\mathcal{R}$ and Precision@$\mathcal{R}$ as in [59], where for each query the set of $\mathcal{R}$ nearest neighbors is retrieved and the recall is computed as the fraction of the samples in both set $\mathcal{T}$ and $\mathcal{R}$ and the size of the set $\mathcal{T}$, Recall@$\mathcal{R} = \frac{|\mathcal{R} \cap \mathcal{T}|}{|\mathcal{T}|}$ and Precision@$\mathcal{R} = \frac{|\mathcal{R} \cap \mathcal{T}|}{|\mathcal{R}|}$.

We have used four datasets, COIL20 which contains 1440 samples in 1024 dimensional space, ISOLET which contains 1560 samples in 617 dimensions, TDT2 which contains 9394 sample in 19677 dimensional space and a subset of 20 Newsgroups (20NG in short) containing 9990 samples in 29360 dimensional space [10]. The performance with $|\mathcal{T}| = 10$, and 50 is reported and each experiment is repeated 10 times each time using the same 100 query and the average and 95% confidence interval are provided. The observed behavior remains valid for other $|\mathcal{T}|$. Additionally, the number of the basis in the lower dimensional space m is set to 10 by default, unless otherwise stated. Note that the results in this subsection are not affected by the value of ϵ as discussed in section 4.1.1. Figure 4.1 shows the results of the different techniques in COIL20 and ISOLET datasets and as shown in the figure, EBEK was able to achieve the best Precision@R and Recall@R. After that, ITQ and PCA-RR were the second best in Precision@R and Recall@R. Moreover, table 4.1 shows the running time of the techniques in COIL20 and ISOLET datasets. The results show that LSH and SKLSH were the fastest approaches, while EBEK was the third fastest approach with a gap of at most 0.04 seconds to LSH.

Figures 4.2 and 4.3 show the effect of changing the $|\mathcal{T}|$ on the Recall@$\mathcal{R}$ and Precision@$\mathcal{R}$ for both TDT2 and 20NG datasets. Note that, MDS and LLE were omitted from TDT2 and 20NG datasets as they were taking more than 20 minutes to run. Table 4.2 shows the running time of obtaining the low dimension embedding and the bit-encoding (depending on the approach) in TDT2 and 20NG datasets. It is obvious that EBEK consistently achieves the highest precision and recall while achieving the lowest running time.



(a) COIL20 Dataset Precision@R (b) COIL20 Dataset Recall@R (c) ISOLET Dataset Precision@R (d) ISOLET Dataset Recall@R

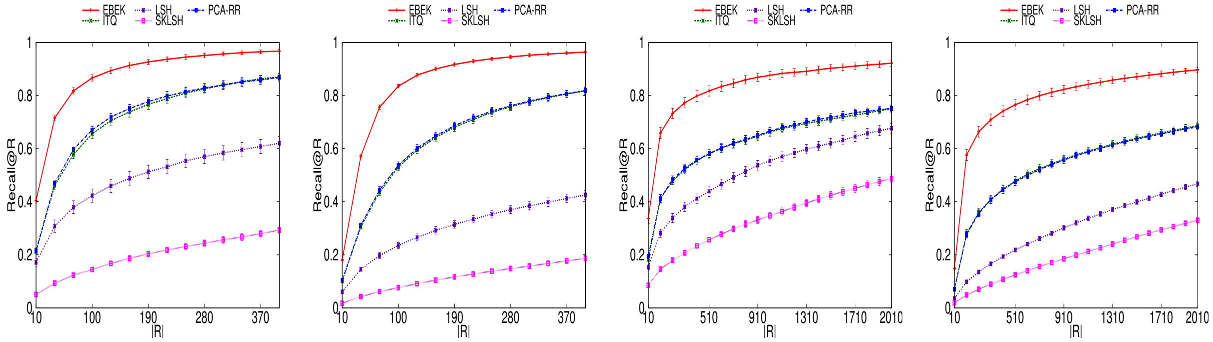
Figure 4.1: Precision@R and Recall@R for COIL20 and ISOLET Datasets

Table 4.1: Approximate Nearest Neighbors Running Time (in Seconds) Comparison for COIL20 and ISOLET Datasets

	COIL20	**ISOLET**
EBEK	0.07 ± 0.00	0.03 ± 0.01
ITQ	0.08 ± 0.00	0.05 ± 0.00
LSH	**0.03 ± 0.00**	**0.02 ± 0.00**
PCA-RR	0.07 ± 0.00	0.04 ± 0.00
SKLSH	**0.03 ± 0.00**	**0.02 ± 0.00**
MDS	129.39 ± 0.20	146.44 ± 1.23
LLE	0.43 ± 0.01	1.06 ± 0.05

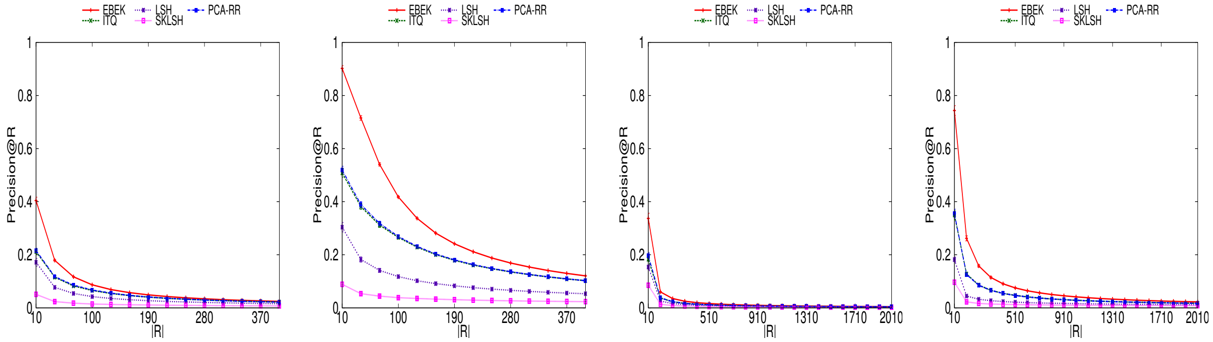
4.2.2 Clustering

To evaluate the effectiveness of the proposed algorithm, the data is projected to a lower dimension space using different approaches, then the projected data is clustered using K-means. K-means algorithm is used as it is very popular and has been used in previous work as in [11][21]. After the clustering is performed, the cluster labels are compared to ground-truth labels and the Normalized Mutual Information (NMI) between clustering labels and the class labels, F-measure and running time of the dimensionality reduction and clustering are reported. Experiments that depend on random variable are repeated 10 times then the average and 95% confidence interval are reported. The linear kernel is evaluated on four datasets, which have been used by [11][21]



(a) TDT2 Dataset, $|\mathcal{T}| = 10$ (b) TDT2 Dataset, $|\mathcal{T}| = 50$ (c) 20NG Dataset, $|\mathcal{T}| = 10$ (d) 20NG Dataset, $|\mathcal{T}| = 50$

Figure 4.2: Recall@R for TDT2 and 20NG Datasets



(a) TDT2 Dataset, $|\mathcal{T}| = 10$ (b) TDT2 Dataset, $|\mathcal{T}| = 50$ (c) 20NG Dataset, $|\mathcal{T}| = 10$ (d) 20NG Dataset, $|\mathcal{T}| = 50$

Figure 4.3: Precision@R for TDT2 and 20NG Datasets

for the feature selection task; table 4.3 shows the details of the datasets.[1] In this subsection we compare our approach against five other approaches: 1) PCA, which achieves the minimum reconstruction error. 2) NMF using multiplicative update algorithm [34]. We used the Matlab implementation with the default settings for PCA and NMF. 3) MDS; we used the Matlab implementation. 4) LLE, and 5) ISOMAP; for both we used the implementation provided by the authors.[2,3]

[1] The datasets are available at
http://www.cad.zju.edu.cn/home/dengcai/Data/FaceData.html
http://www.cs.columbia.edu/CAVE/software/softlib/coil-20.php
https://archive.ics.uci.edu/ml/datasets/ISOLET
https://www.otexts.org/1577

[2] https://www.cs.nyu.edu/ roweis/lle/code.html

[3] http://isomap.stanford.edu/IsomapR1.tar

Table 4.2: Approximate Nearest Neighbors Running Time Comparison.

	TDT2 $\lvert\mathcal{T}\rvert = 10$	**TDT2** $\lvert\mathcal{T}\rvert = 50$	**20NG** $\lvert\mathcal{T}\rvert = 10$	**20NG** $\lvert\mathcal{T}\rvert = 50$
EBEK	$\mathbf{5.17 \pm 0.07}$	$\mathbf{5.17 \pm 0.08}$	$\mathbf{6.21 \pm 0.07}$	$\mathbf{6.35 \pm 0.1}$
ITQ	32.77 ± 0.96	32.19 ± 1.66	260.40 ± 40.80	335.27 ± 74.82
LSH	14.47 ± 0.11	14.27 ± 0.18	268.09 ± 91.93	276.29 ± 80.68
PCA-RR	31.87 ± 1.12	31.70 ± 1.25	300.09 ± 58.29	451.66 ± 91.01
SKLSH	14.58 ± 0.27	14.16 ± 0.17	868.07 ± 130.50	716.33 ± 198.58

Table 4.3: The properties of datasets used for evaluation

Dataset	**# Instances**	**# Features**	**# Classes**
ORL	400	1024	40
COIL20	1440	1024	20
ISOLET	1560	617	26
USPS	9298	256	10

As each dataset contains the same number of samples in each class, except for USPS dataset, in LLE and ISOMAP we set the number of neighbors to the total number of samples divided by the number of classes. The intuition behind this choice in LLE is to keep the samples in the same class close to each other, while increasing the distance between the samples in different classes. For ISOMAP the intuition is to approximate the neighborhood of each point by the samples in the same class. For USPS dataset we set the number of neighbors to the floor of the number of total samples divided by the number of classes. It is worth noting that for LLE and ISOMAP we experimented smaller values for the number of neighbors than the aforementioned one. However, the quality of clusters degrades with insignificant reduction in running time. In addition, ISOMAP does not scale on the USPS dataset even when the number of neighbors is set to 2; it takes more than half an hour to project the data on 50 dimensions.

Figure 4.4 shows the clustering results. The NMI and F-Measure for EBEK is very close to those of PCA. Although PCA achieves running time less than EBEK, the later projects the data on basis that can be related to real samples rather than latent basis as in the former. Also, it can be seen that the porposed approach achieves NMI and F-Measure higher than LLE by a difference up to 30% and 38% respectively; we can also note that our approach scales well with the increase in the number of samples while LLE does not. While NMF achieves a good running time, the quality of the clusters declines with increasing the number of samples as in USPS dataset. ISOMAP achieves a higher NMI and F-Measure in some datasets by a difference up to

3.6% and 3% respectively, however, this increase comes with 40 times increase in the running time. Additionally, ISOMAP does not scale with the increase in the number of samples and it does not run on the USPS dataset. Moreover, our approach achieves a speedup over ISOMAP up to 2.8X. Our approach achieves a higher NMI and F-Measure than MDS by a difference up to 2% in both measures with significant difference in the running time. Note that MDS takes around 180 seconds in ORL, around 2900 seconds in ISOLET, around 2300 seconds in COIL20 and around 3700 seconds in USPS. However, the limit of the y-axis on these figures is small to show the difference between the running time of the rest of the algorithms.

In addition, RBF kernel is evaluated using the same kernel parameters reported in [12] on two datasets ORL and COIL20; table 4.4 shows the results using 50 and 100 dimensions. The RBF kernel does not improve the results, which is consisted with the work done in [28], that suggests using linear kernel for large number of features.

4.2.3 Interpretability Experiments

To show that the basis detected by EBEK are more understandable than the basis detected by the other approaches, two datasets are selected which are COIL20 and ORL datasets and the basis selected by EBEK and PCA-RR are drawn in figure 4.5. Note that the basis of PCA were similar to the basis detected by PCA-RR and PCA-RR has much better quality in the approximate nearest neighbor task, thus we only show PCA-RR basis. The value of ϵ was chosen empirically to yield the best visualization results and was set to 0.65 and 0.94 in COIL20 and ORL datasets respectively. As shown in the figure 4.5, in COIL20 dataset, EBEK basis were more interpretable than PCA basis, as it shows that COIL20 contains different objects in different orientations. While PCA produced understandable basis in the ORL dataset, still EBEK basis are more understandable. PCA basis point out that there is a change in the mouth area in the dataset images and as a viewer you do not know what are these changes. However, EBEK shows you these changes with men with breads and people with different mouth emotions. Additionally, EBEK basis capture characteristics that PCA can not capture, for example that the dataset contains different gender, different age and different color people.

To further show the interpretability of EBEK, we apply EBEK in the task of word embedding. Word embedding [40] is a vector-space word representation that captures both syntactic and semantic regularities between words in languages. Word embedding has showed interesting results in capturing word analogy between words [44] which not only measures the degree of similarity between the vector representation of pair of words, but in addition it captures other dimension of difference. An example in [44] is used to illustrate this as follows. The analogy ”king → queen as man → woman” is encoded in the vector space by the vector equation *king* −

Table 4.4: Different Kernel Types Comparison.

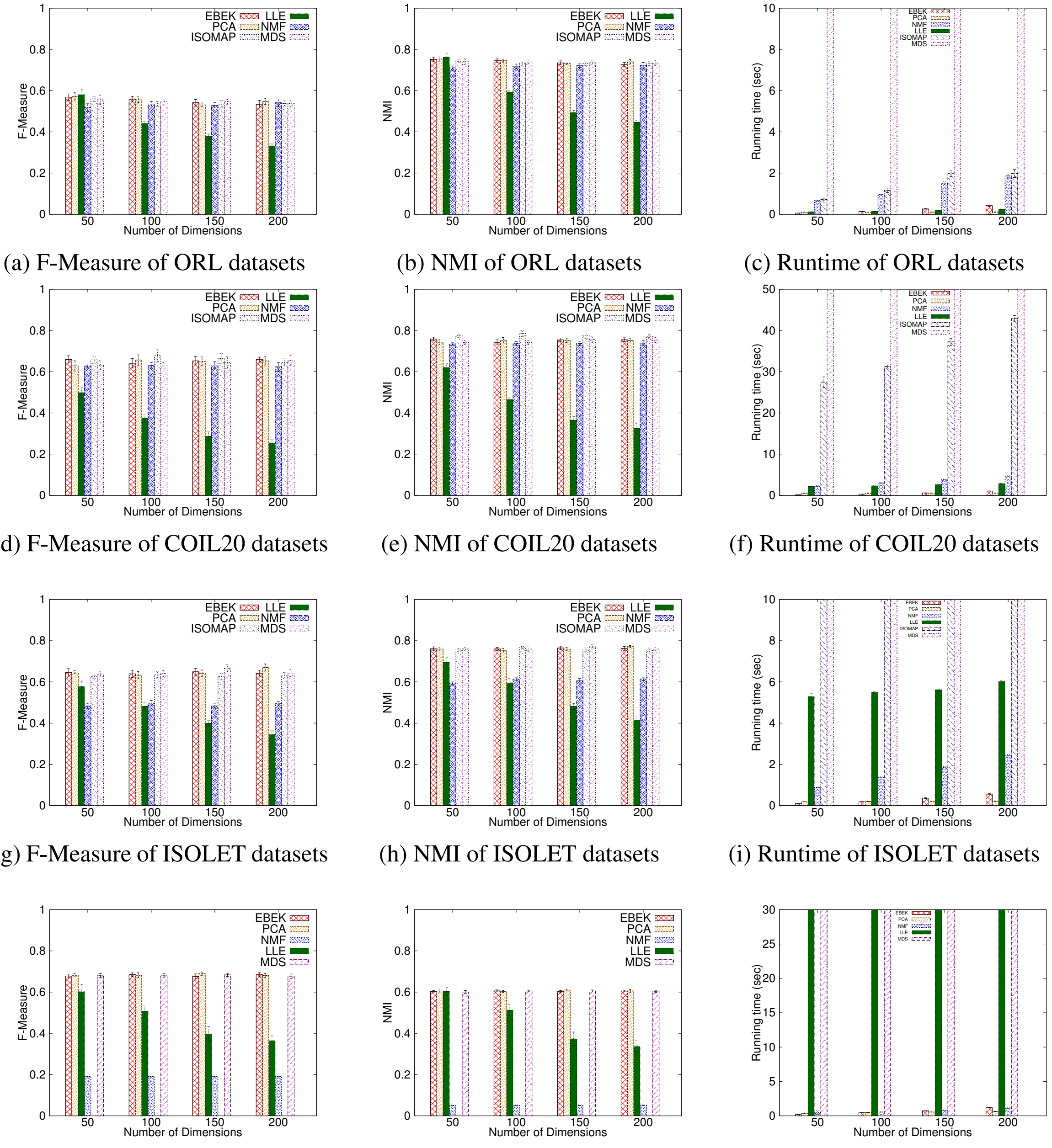
Dataset	**Measure**	**Linear**	**RBF**
ORL (m=50)	**Fmeasure**	0.56 ± 0.01	$\mathbf{0.58 \pm 0.01}$
	NMI	$\mathbf{0.75 \pm 0.01}$	$\mathbf{0.75 \pm 0.006}$
	Time	$\mathbf{0.05 \pm 0.003}$	0.29 ± 0.05
ORL (m=200)	**Fmeasure**	$\mathbf{0.53 \pm 0.01}$	0.51 ± 0.012
	NMI	$\mathbf{0.72 \pm 0.01}$	0.69 ± 0.01
	Time	$\mathbf{0.41 \pm 0.027}$	1.02 ± 0.02
COIL20 (m=50)	**Fmeasure**	$\mathbf{0.65 \pm 0.01}$	0.59 ± 0.02
	NMI	$\mathbf{0.75 \pm 0.008}$	0.69 ± 0.01
	Time	$\mathbf{0.13 \pm 0.005}$	1.72 ± 0.02
COIL20 (m=200)	**Fmeasure**	$\mathbf{0.65 \pm 0.01}$	0.59 ± 0.02
	NMI	$\mathbf{0.75 \pm 0.009}$	0.68 ± 0.01
	Time	$\mathbf{1.04 \pm 0.028}$	7.05 ± 0.02

$queen = man - woman$. This enables word embedding to overcome the deficiencies in Latent Semantic Analysis (LSA) [33], that LSA performs poorly on the word embedding task. Although word embedding has many advantages over other techniques. It still embeds the data in a latent space with basis that can not easily be interpreted. In this experiment we use EBEK as a post-processing step for the vector representation obtained by word embedding techniques to embed these vectors in a space of explicit words. Keeping in mind that EBEK preserves the relations between the data instance, which enables it to preserve the advantages of the word embedding while using explicit features at the same time.

Two word embedding techniques Glove tool [44] and word2vec [25] to first construct the embedding of words in 50 latent space and then EBEK is used on this output to embed the data into 10 exemplars (i.e, words). Figure 4.6 shows the resulting embedding of samples words using EBEK, the vertical axis represents the basis and the horizontal one represents samples. Words are chosen to be in two groups: location and animal. As shown in the figure EBEK was able to preserve the similarities between the location words and the animal words, for example in Figures 4.6a and 4.6b you can see that the vectors of *cat* and *dog* are not only similar to each other but different than the vectors of *Seattle* and *Bosont* .

4.3 Summary

In this chapter the method EBEK, Exemplar-based Kernel Preserving embedding, is proposed. EBEK is shown theoretically to achieve the lowest reconstruction error of the kernel matrix with linear running time complexity. The effectiveness of the approach is shown on the approximate nearest neighbor and clustering tasks, where EBEK performs better than the related work by up to 60% in the recall while maintaining a good running time in the former case. and by up tp 40% in the latter case. The next chapter concludes the work and gives some future research directions.



(a) F-Measure of ORL datasets

(b) NMI of ORL datasets

(c) Runtime of ORL datasets

(d) F-Measure of COIL20 datasets

(e) NMI of COIL20 datasets

(f) Runtime of COIL20 datasets

(g) F-Measure of ISOLET datasets

(h) NMI of ISOLET datasets

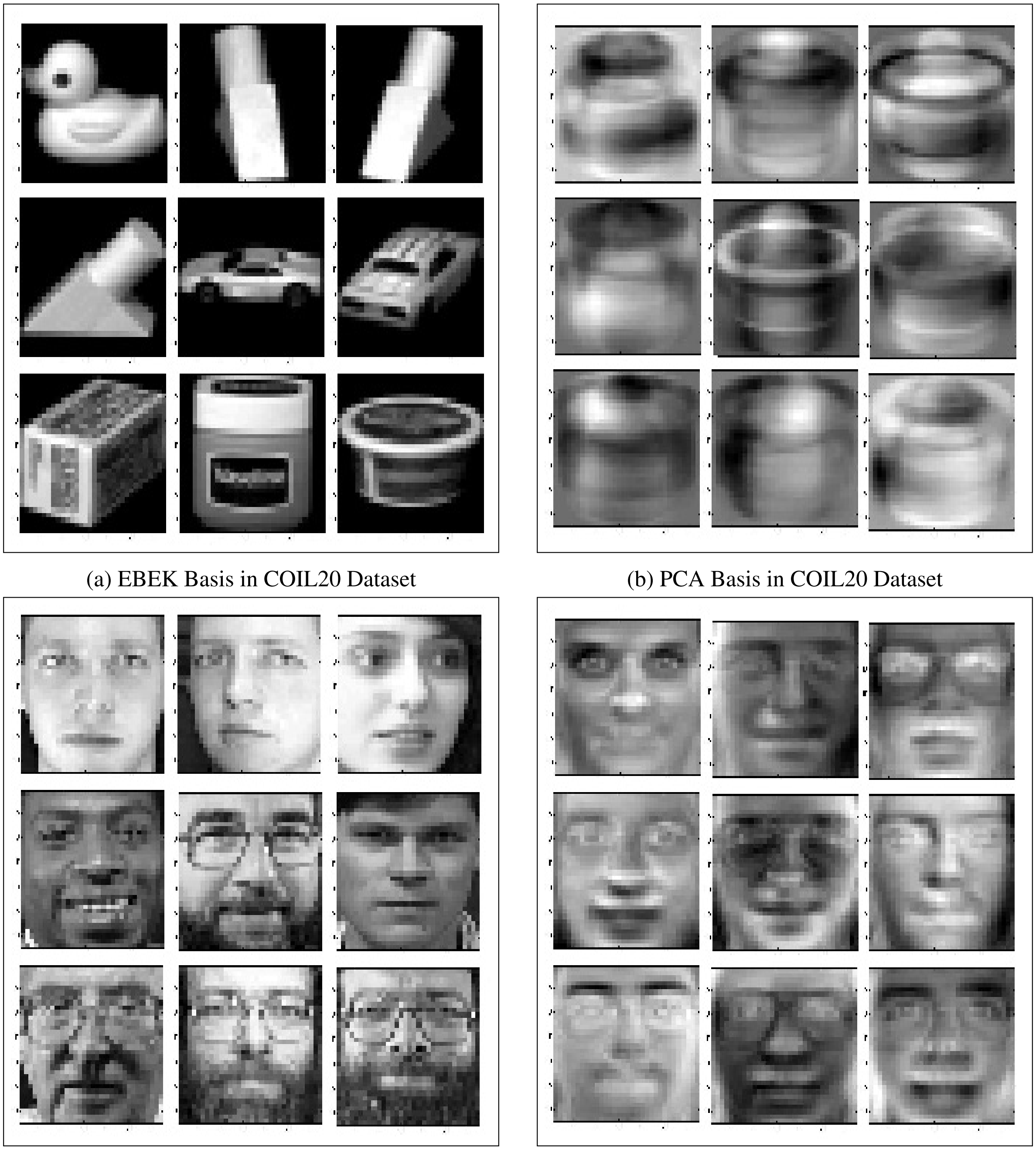
(i) Runtime of ISOLET datasets

(j) F-Measure of USPS datasets

(k) NMI of USPS datasets

(l) Runtime of USPS datasets

Figure 4.4: Clustering results on different datasets



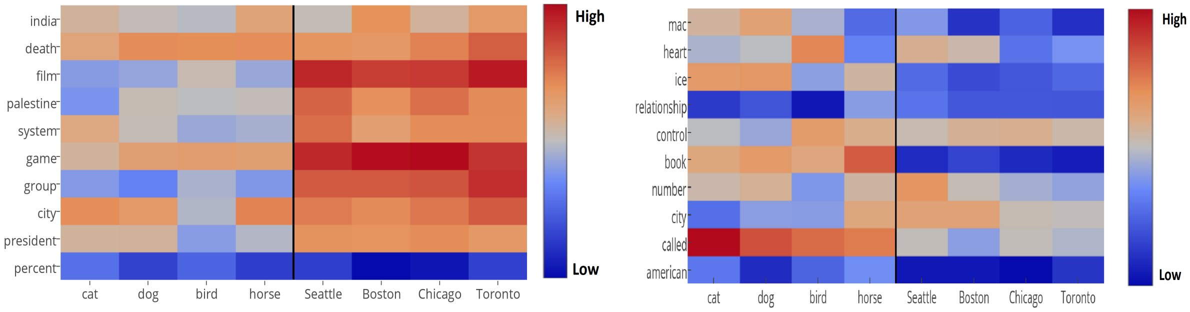
(a) EBEK Basis in COIL20 Dataset

(b) PCA Basis in COIL20 Dataset

(c) EBEK Basis in ORL Dataset

(d) PCA Basis in ORL Dataset

Figure 4.5: EBEK and PCA Basis in COIL20 and ORL Datasets



(a) Embedding of Glove vectors

(b) Embedding of word2vec vectors

Figure 4.6: EBEK results on word embedding

Chapter 5

Conclusions and Future Work

5.1 Conclusion

This dissertation focuses on presenting techniques for selecting the representatives samples from large datasets. It starts by showing the importance of using exemplars to represent data. Then, an Exemplar-based approach for detecting frequent and emerging topics in Twitter streams. The approach selects exemplar tweets as representatives for the detected topics based on the variance of the similarity between exemplars and other tweets. Our Exemplar-based approach for topic detection achieved the best term precision as it selects real tweets as topic representatives and therefore reduces the noisy terms in the topics representatives. Moreover, the proposed emerging extension achieved a better topic recall, term precision and running time compared to recently proposed approaches for emerging topic detection.

In addition, an **E**xemplar-**b**as**e**d **K**ernel Preserving (EBEK) embedding is proposed and shown theoretically to achieve the lowest reconstruction error of the kernel matrix. Stochastic SVD is employed to achieve an efficient implementation of EBEK, which runs in linear time in terms of number of samples. Evaluation shows that EBEK exceeds the related work in the retrieved ANNs quality, while maintaining a good running time. Moreover, in clustering EBEK outperforms the related work in running time, while achieving a good clustering quality. Our future work includes exploiting distributed systems, like Spark to scale out the proposed approach, along with a detailed complexity analysis of the distributed approach.

5.2 Future Work

This subsection gives some directions to extend this work.

5.2.1 Developing a sampling method for Nyström

Developing sampling techniques for Nyström approximation is a challenging problem [32, 52, 17]. One way to extend this work is to extend it to sample data for Nyström approximation. Given a data matrix $X \in \mathbb{R}^{d \in n}$, each sample has d components in different directions, if we were to use one sample $X_{:i}$ to reconstruct the whole data, the error will be the sum of the components of the other samples in the directions that sample $X_{:i}$ does not span. So, based on this we can assign an error to each sample and choose the set of samples that has the minimum error. However, this approach has two drawback:

- The need to build a kernel matrix, which will be inefficient or infeasible for large data, which is not feasible in case of large data [43].
- We need to update the error function for each sample upon choosing a new representative which is very costly.

To mitigate the aforementioned problems, we can use a two-step algorithm for choosing representatives. The first step clusters the samples and uses a representative for each cluster to build a smaller similarity matrix of representatives instead of the whole matrix. This is based on the following assumption: the samples that are in different clusters are less similar to each than those in the same cluster, so we can approximate the similarities between samples from different clusters as zero. The second step assigns a weight to each cluster representative as described above. Then subset of these representatives will be chosen.

The challenges are:

- How to cluster the data such that the sizes of clusters are close to each other (i.e, balanced clusters [3]) and at the same time each cluster is coherent so that we can assume the similarity between samples from different clusters are zero.
- In one extreme, the number of clusters will equal the number of samples which mean we will build a complete accurate Kernel matrix, which is infeasible for large data. On the other extreme we will build m kernel matrix to approximate the original kernel, where m is the number of chosen samples. The question is how to relate the number of clusters to the error?

5.2.2 Increase the sparsity of the EBEK embedding

Chapter 4 presenting an embedding technique that eases the interpretation of the data by embedding the data into a space spanned by explicit features. One way to improve the interpretability of the embedded vector is increasing the sparsity in the embedded vectors [22]. The sparsity can be enforced using two ways:

- Consider any value in the embedded vectors below a fixed threshold to be zero. However, using this method the resulting error can not be quantified.
- Incorporate a regularization term in equation 4.1 in section 4.1.1 that increases the more the embedding is sparse. This can be achieved usign LASSO regularization [54, 55]

5.2.3 Scaling out the implementation of EBEK

Processing large volume of data on a single machine may be infeasible on one machine due to memory and computational demands, which makes the use of distributed system a need. Many frameworks have been developed to eliminate the challenges of distributed systems implementation like MapReduce [14], Spark [56], Giraph, which is an open source implmentation of Pregel [38] and GraphLab [36]. One extention is to implement EBEK on one of these distributed systems.

References

[1] Luca Maria Aiello, Georgios Petkos, Christian Martin, David Corney, Symeon Papadopoulos, Ryan Skraba, Ayse Goker, Ioannis Kompatsiaris, and Aldo Jaimes. Sensing trending topics in twitter. *Multimedia, IEEE Transactions on*, 15(6):1268–1282, 2013.

[2] Alexandr Andoni and Piotr Indyk. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. In *Foundations of Computer Science, 2006. FOCS'06. 47th Annual IEEE Symposium on*, pages 459–468. IEEE, 2006.

[3] Arindam Banerjee and Joydeep Ghosh. Scalable clustering algorithms with balancing constraints. *Data Mining and Knowledge Discovery*, 13(3):365–395, 2006.

[4] Michael W Berry, Murray Browne, Amy N Langville, V Paul Pauca, and Robert J Plemmons. Algorithms and applications for approximate nonnegative matrix factorization. *Computational statistics & data analysis*, 52(1):155–173, 2007.

[5] Michael Biggs, Ali Ghodsi, and Stephen Vavasis. Nonnegative matrix factorization via rank-one downdate. In *Proceedings of the 25th international conference on Machine learning*, pages 64–71. ACM, 2008.

[6] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022, 2003.

[7] Christos Boutsidis, Petros Drineas, and Malik Magdon-Ismail. Near-optimal column-based matrix reconstruction. *SIAM Journal on Computing*, 43(2):687–717, 2014.

[8] Christos Boutsidis, Michael W Mahoney, and Petros Drineas. An improved approximation algorithm for the column subset selection problem. In *Proceedings of the twentieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 968–977. Society for Industrial and Applied Mathematics, 2009.

[9] Christos Boutsidis, Jimeng Sun, and Nikos Anerousis. Clustered subset selection and its applications on it service metrics. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 599–608. ACM, 2008.

[10] Deng Cai, Xuanhui Wang, and Xiaofei He. Probabilistic dyadic data analysis with local and global consistency. In *Proceedings of the 26th Annual International Conference on Machine Learning (ICML'09)*, pages 105–112, 2009.

[11] Deng Cai, Chiyuan Zhang, and Xiaofei He. Unsupervised feature selection for multi-cluster data. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 333–342. ACM, 2010.

[12] Radha Chitta, Rong Jin, Timothy C Havens, and Anil K Jain. Approximate kernel k-means: Solution to large scale kernel clustering. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 895–903. ACM, 2011.

[13] Ali Civril and Malik Magdon-Ismail. Column subset selection via sparse approximation of svd. *Theoretical Computer Science*, 421:1–14, 2012.

[14] Jeffrey Dean and Sanjay Ghemawat. Mapreduce: simplified data processing on large clusters. *Communications of the ACM*, 51(1):107–113, 2008.

[15] Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391, 1990.

[16] Amit Deshpande, Luis Rademacher, Santosh Vempala, and Grant Wang. Matrix approximation and projective clustering via volume sampling. In *Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm*, pages 1117–1126. Society for Industrial and Applied Mathematics, 2006.

[17] Petros Drineas and Michael W Mahoney. On the nyström method for approximating a gram matrix for improved kernel-based learning. *The Journal of Machine Learning Research*, 6:2153–2175, 2005.

[18] Ahmed Elgohary, Ahmed K Farahat, Mohamed S Kamel, and Fakhri Karray. Embed and conquer: Scalable embeddings for kernel k-means on mapreduce. SIAM.

[19] Ahmed Elgohary, Ahmed K Farahat, Mohamed S Kamel, and Fakhri Karray. Embed and conquer: Scalable embeddings for kernel k-means on mapreduce. SIAM.

[20] Ahmed K Farahat, Ahmed Elgohary, Ali Ghodsi, and Mohamed S Kamel. Distributed column subset selection on mapreduce. In *Data Mining (ICDM), 2013 IEEE 13th International Conference on*, pages 171–180. IEEE, 2013.

[21] Ahmed K Farahat, Ali Ghodsi, and Mohamed S Kamel. An efficient greedy method for unsupervised feature selection. In *Data Mining (ICDM), 2011 IEEE 11th International Conference on*, pages 161–170. IEEE, 2011.

[22] Manaal Faruqui, Yulia Tsvetkov, Dani Yogatama, Chris Dyer, and Noah Smith. Sparse overcomplete word vector representations. *arXiv preprint arXiv:1506.02004*, 2015.

[23] Yoav Freund, Sanjoy Dasgupta, Mayank Kabra, and Nakul Verma. Learning the structure of manifolds using random projections. In *Advances in Neural Information Processing Systems*, pages 473–480, 2007.

[24] Jerome H Friedman, Jon Louis Bentley, and Raphael Ari Finkel. An algorithm for finding best matches in logarithmic expected time. *ACM Transactions on Mathematical Software (TOMS)*, 3(3):209–226, 1977.

[25] Yoav Goldberg and Omer Levy. word2vec explained: Deriving mikolov et al.'s negative-sampling word-embedding method. *arXiv preprint arXiv:1402.3722*, 2014.

[26] Yunchao Gong and Svetlana Lazebnik. Iterative quantization: A procrustean approach to learning binary codes. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 817–824. IEEE, 2011.

[27] Nathan Halko, Per-Gunnar Martinsson, and Joel A Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM review*, 53(2):217–288, 2011.

[28] Chih-Wei Hsu, Chih-Chung Chang, Chih-Jen Lin, et al. A practical guide to support vector classification, 2003.

[29] William B Johnson and Joram Lindenstrauss. Extensions of lipschitz mappings into a hilbert space. *Contemporary mathematics*, 26(189-206):1, 1984.

[30] IT Jolliffe. Principal components as a small number of interpretable variables: some examples. *Principal Component Analysis*, pages 63–77, 2002.

[31] Shiva Prasad Kasiviswanathan, Prem Melville, Arindam Banerjee, and Vikas Sindhwani. Emerging topic detection using dictionary learning. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 745–754. ACM, 2011.

[32] Sanjiv Kumar, Mehryar Mohri, and Ameet Talwalkar. On sampling-based approximate spectral decomposition. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 553–560. ACM, 2009.

[33] Thomas K Landauer, Peter W Foltz, and Darrell Laham. An introduction to latent semantic analysis. *Discourse processes*, 25(2-3):259–284, 1998.

[34] Daniel D Lee and H Sebastian Seung. Algorithms for non-negative matrix factorization. In *Advances in neural information processing systems*, pages 556–562, 2001.

[35] Seymour Lipschutz and Marc Lipson. *Schaum's Outline of Linear Algebra*. McGraw-Hill Education, 5 edition, 2012.

[36] Yucheng Low, Danny Bickson, Joseph Gonzalez, Carlos Guestrin, Aapo Kyrola, and Joseph M Hellerstein. Distributed graphlab: a framework for machine learning and data mining in the cloud. *Proceedings of the VLDB Endowment*, 5(8):716–727, 2012.

[37] Michael W Mahoney. An improved approximation algorithm for the column subset selection problem. 2010.

[38] Grzegorz Malewicz, Matthew H Austern, Aart JC Bik, James C Dehnert, Ilan Horn, Naty Leiser, and Grzegorz Czajkowski. Pregel: a system for large-scale graph processing. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*, pages 135–146. ACM, 2010.

[39] Rishabh Mehrotra, Scott Sanner, Wray Buntine, and Lexing Xie. Improving lda topic models for microblogs via tweet pooling and automatic labeling. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, pages 889–892. ACM, 2013.

[40] Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. Linguistic regularities in continuous space word representations. In *HLT-NAACL*, pages 746–751, 2013.

[41] Jorge Moraleda. Gregory shakhnarovich, trevor darrell and piotr indyk: Nearest-neighbors methods in learning and vision. theory and practice. *Pattern Analysis and Applications*, 11(2):221–222, 2008.

[42] David Newman, Edwin V Bonilla, and Wray Buntine. Improving topic coherence with regularized topic models. In *Advances in neural information processing systems*, pages 496–504, 2011.

[43] Raajen Patel, Thomas A Goldstein, Eva L Dyer, Azalia Mirhoseini, and Richard G Baraniuk. oasis: Adaptive column sampling for kernel matrix approximation. *arXiv preprint arXiv:1505.05208*, 2015.

[44] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. *Proceedings of the Empiricial Methods in Natural Language Processing (EMNLP 2014)*, 12:1532–1543, 2014.

[45] Saša Petrović, Miles Osborne, and Victor Lavrenko. Streaming first story detection with application to twitter. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 181–189. Association for Computational Linguistics, 2010.

[46] Maxim Raginsky and Svetlana Lazebnik. Locality-sensitive binary codes from shift-invariant kernels. In *Advances in neural information processing systems*, pages 1509–1517, 2009.

[47] Sam T Roweis and Lawrence K Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.

[48] Ankan Saha and Vikas Sindhwani. Learning evolving and emerging topics in social media: a dynamic nmf approach with temporal regularization. In *Proceedings of the fifth ACM international conference on Web search and data mining*, pages 693–702. ACM, 2012.

[49] G Shakhnarovich, T Darell, and P Indyk. Nearest-neighbors methods in learning and vision, 2006.

[50] Vin D Silva and Joshua B Tenenbaum. Global versus local methods in nonlinear dimensionality reduction. In *Advances in neural information processing systems*, pages 705–712, 2002.

[51] James Joseph Sylvester. A demonstration of the theorem that every homogeneous quadratic polynomial is reducible by real orthogonal substitutions to the form of a sum of positive and negative squares. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 4(23):138–142, 1852.

[52] Ameet Talwalkar and Afshin Rostamizadeh. Matrix coherence and the nystrom method. *arXiv preprint arXiv:1004.2008*, 2010.

[53] Joshua B Tenenbaum, Vin De Silva, and John C Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

[54] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

[55] Robert Tibshirani, Michael Saunders, Saharon Rosset, Ji Zhu, and Keith Knight. Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(1):91–108, 2005.

[56] Reynold S Xin, Josh Rosen, Matei Zaharia, Michael J Franklin, Scott Shenker, and Ion Stoica. Shark: Sql and rich analytics at scale. In *Proceedings of the 2013 ACM SIGMOD International Conference on Management of data*, pages 13–24. ACM, 2013.

[57] Tan Xu and Douglas W Oard. Wikipedia-based topic clustering for microblogs. *Proceedings of the American Society for Information Science and Technology*, 48(1):1–10, 2011.

[58] Xiaohui Yan, Jiafeng Guo, Shenghua Liu, Xue-qi Cheng, and Yanfeng Wang. Clustering short text using ncut-weighted non-negative matrix factorization. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 2259–2262. ACM, 2012.

[59] Ting Zhang, Chao Du, and Jingdong Wang. Composite quantization for approximate nearest neighbor search. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 838–846, 2014.

[60] Zhenyue Zhang, Hongyuan Zha, and Horst Simon. Low-rank approximations with sparse factors i: Basic algorithms and error analysis. *SIAM Journal on Matrix Analysis and Applications*, 23(3):706–727, 2002.