T.C. DOĞUŞ UNIVERSITY
INSTITUTE OF SCIENCE AND TECHNOLOGY
COMPUTER AND INFORMATION SCIENCES DEPARTMENT

HIGHER-ORDER SEMANTIC SMOOTHING FOR TEXT CLASSIFICATION

M.S THESIS

Mitat POYRAZ
201091002

Thesis Advisor:
Assoc. Prof. Dr. Murat Can GANİZ

JANUARY 2013
ISTANBUL

T.C. DOĞUŞ UNIVERSITY
INSTITUTE OF SCIENCE AND TECHNOLOGY
COMPUTER AND INFORMATION SCIENCES DEPARTMENT

HIGHER-ORDER SEMANTIC SMOOTHING FOR TEXT CLASSIFICATION

M.S THESIS

Mitat POYRAZ
201091002

Thesis Advisor:
Assoc. Prof. Dr. Murat Can GANİZ

JANUARY 2013
ISTANBUL

HIGHER-ORDER SEMANTIC SMOOTHING FOR TEXT CLASSIFICATION

APPROVED BY:

Assoc. Prof. Dr. Murat Can GANİZ ....

(Thesis Advisor)

Prof. Dr. Selim AKYOKUŞ ....

(Doğuş University)

Assoc. Prof. Dr. Yücel SAYGIN ....

(Sabancı University)

DATE OF APPROVAL: 29.01.2013

PREFACE

In my thesis, a novel semantic smoothing method named Higher Order Smoothing (HOS) for the Naïve Bayes algorithm is presented. HOS is built on a graph based data representation which allows semantics in higher-order paths to be exploited. This work was supported in part by The Scientific and Technological Research Council of Turkey (TÜBİTAK) grant number 111E239. Points of view in this document are those of the authors and do not necessarily represent the official position or policies of the TÜBİTAK.

Istanbul, January 2013 Mitat POYRAZ

ABSTRACT

Text classification is the task of automatically sorting a set of documents into classes (or categories) from a predefined set. This task is of great practical importance given the massive volume of online text available through the World Wide Web, Internet news feeds, electronic mail and corporate databases. Existing statistical text classification algorithms can be trained to accurately classify documents, given a sufficient set of labeled training examples. However, in real world applications, only a small amount of labeled data is available because expert labeling of large amounts of data is expensive. In this case, making an adequate estimation of the model parameters of a classifier is challenging. Underlying this issue is the traditional assumption in machine learning algorithms that instances are independent and identically distributed (IID). Semi-supervised learning (SSL) is the machine learning concept concerned with leveraging explicit as well as implicit link information within data to provide a richer data representation for model parameter estimation.

It has been shown that Latent Semantic Indexing (LSI) takes advantage of implicit higher-order (or latent) structure in the association of terms and documents. Higher-order relations in LSI capture "latent semantics". Inspired by this, a novel Bayesian framework for classification named Higher Order Naïve Bayes (HONB), which can explicitly make use of these higher-order relations, has been introduced previously. In this thesis, a novel semantic smoothing method named Higher Order Smoothing (HOS) for the Naïve Bayes algorithm is presented. HOS is built on a similar graph based data representation of HONB which allows semantics in higher-order paths to be exploited. Additionally, we take the concept one step further in HOS and exploited the relationships between instances of different classes in order to improve the parameter estimation when dealing with insufficient labeled data. As a result, we have not only been able to move beyond instance boundaries, but also class boundaries to exploit the latent information in higher-order paths. The results of experiments demonstrate the value of HOS on several benchmark datasets.

Key Words: Naïve Bayes, Semantic Smoothing, Higher Order Naïve Bayes, Higher Order Smoothing, Text Classification

ÖZET

Metin sınıflandırma, bir dokümanlar kümesini daha önceden tanımlanan sınıflara ya da kategorilere otomatik olarak dahil etme işlemidir. Bu işlem, Web sayfalarında, Internet haber kaynaklarında, e-posta iletilerinde ve kurumsal veri tabanlarında mevcut olan çok büyük miktardaki elektronik metin nedeniyle, giderek büyük önem kazanmaktadır. Hâlihazırdaki metin sınıflandırma algoritmaları, yeterli sayıda etiketli eğitim kümesi verildiği taktirde dokümanları doğru sınıflandırmak üzere eğitilebilir. Oysaki gerçek hayatta, büyük miktarda verilerin uzman kişilerce etiketlenmesi pahalı olduğundan çok az sayıda etiketli veri mevcuttur. Bu durumda, sınıflandırıcının model parametreleri ile ilgili uygun bir kestirim yapmak zordur. Bunun temelinde, makine öğrenimi algoritmalarının, veri içerisindeki örneklerin dağılımının bağımsız ve özdeş olduğunu varsayması yatar. Yarı öğreticiyle öğrenme kavramı, model parametre kestirimi için, veri içerisindeki hem açık hem de saklı ilişkilerden yararlanıp, onu daha zengin bir şekilde temsil etmeyle ilgilenir.

Saklı Anlam Indeksleme'nin (LSI) dokümanların içerdiği terimler arasındaki yüksek dereceli ilişkileri kullanan bir teknik olduğu ortaya konulmuştur. LSI tekniğinde kullanılan yüksek dereceli ilişkilerden kasıt, terimler arasındaki gizli anlamsal yakınlıktır. Bu teknikten esinlenerek, Higher Order Naïve Bayes (HONB) adı verilen, metnin içerisindeki yüksek dereceli anlamsal ilişkileri kullanan, yeni bir metod literatürde yer almaktadır. Bu tezde Higher Order Smoothing (HOS) adı verilen, Naïve Bayes algoritması için yeni bir anlamsal yumuşatma metodu ortaya konmuştur. HOS metodu, HONB uygulama çatısında yer alan, metin içerisindeki yüksek dereceli anlamsal ilişkileri kullanmaya imkan veren grafik tabanlı veri gösterimine dayanmaktadır. Ayrıca HOS metodunda, aynı sınıfların örnekleri arasındaki ilişkilerden faydalanma noktasından bir adım öteye geçilerek, farklı sınıfların örnekleri arasındaki ilişkilerden de faydalanılmıştır. Bu sayede, etiketli veri kümesinin yetersiz olduğu durumlardaki parametre kestirimi geliştirilmiştir. Sonuç olarak, yüksek dereceli anlamsal bilgilerden faydalanmak için, sadece örnek sınırlarının ötesine geçmekle kalmayıp aynı zamanda sınıf sınırlarının da ötesine geçebiliyoruz. Farklı veri kümeleriye yapılan deneylerin sonuçları, HOS metodunun değerini kanıtlamaktadır.

Anahtar Kelimeler: Naïve Bayes, Anlamsal Yumuşatma, Higher Order Naïve Bayes, Higher Order Smoothing, Metin Sınıflandırma

ACKNOWLEDGMENT

I would like to express my deep appreciation and gratitude to my advisor, Yrd. Dr. Murat Can GANİZ, for the patient guidance and mentorship he provided to me, all the way from when I was first considering applying to the M.S program in the Computer Engineering Department, through to completion of this degree.

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

$n(d,w_i)$	Term frequency of word w_i in document d
V	Vertex of a graph
E	Edge of a graph
$\hat{G}$	Tripartite graph
$\varphi(w_i,D_j)$	Number of higher-order paths between word w_i and documents belongs to c_j
$\phi(D_j)$	Number of higher-order paths extracted from the documents of c_j
$\partial(w_i,c_j)$	Number of higher-order paths between word w_i and class label c_j
$\Phi(D)$	Number of higher-order paths between all terms and all class terms in D
X_d^t	Boolean document-term data matrix
O_1	First-order co-occurrence matrix
O_2	Second-order co-occurrence matrix
X_{clb}	Class-binarized data matrix

ABBREVIATIONS

AUC	Area Under the ROC Curve
CBSGC	Consistent Bipartite Spectral Graph Co-partitioning
HONB	Higher Order Naïve Bayes
HOS	Higher Order Smoothing
HOSVM	Higher Order Support Vector Machines
IID	Independent and Identically Distributed
IR	Information Retrieval
JM	Jelinek-Mercer Smoothing
k-NN	K-Nearest Neighbors
LSI	Latent Semantic Indexing
MNB	Multinomial Naïve Bayes
MVNB	Multivariate Bernoulli Naïve Bayes
NB	Naïve Bayes
ODP	Open Directory Project
SDP	Semi-definite Programming
SSL	Semi-supervised Learning
SVD	Singular Value Decomposition
SVM	Support Vector Machines
TF	Term Frequencies
TS	Training Set Size
VSM	Vector Space Model

TABLE OF CONTENTS

1. INTRODUCTION

1.1. Scope and objectives of the Thesis

A well-known problem in real-world applications of machine learning is that they require a large, often prohibitive, number of labeled training examples to learn accurately. However, often in practice, it is very expensive and time consuming to label large amounts of data as they require the efforts of skilled human annotators. In this case, making an adequate estimation of the model parameters of a classifier is challenging. Underlying this issue is the traditional assumption in machine learning algorithms that instances are independent and identically distributed (IID) (Taskar et.al, 2002). This assumption simplifies the underlying mathematics of statistical models and allows the classification of a single instance. However in real world datasets, instances and attributes are highly interconnected. Consequently, the IID approach does not fully make use of valuable information about relationships within a dataset (Getoor and Diehl, 2005). There are several studies which exploit explicit link information in order to overcome the shortcomings of IID approach (Chakrabarti et.al, 1998; Neville and Jensen, 2000; Taskar et.al, 2002; Getoor and Diehl, 2005). However, the use of explicit links has a significant drawback; in order to classify a single instance, an additional context needs to be provided. There is another approach which encounters this drawback, known as higher-order learning. It is a statistical relational learning framework which allows supervised and unsupervised algorithms to leverage relationships between different instances of the same class (Edwards and Pottenger, 2011). This approach makes use of implicit link information (Ganiz et.al, 2006; Ganiz et.al, 2009; Ganiz et.al, 2011). Using implicit link information within data provides a richer data representation. It is difficult and usually expensive to obtain labeled data in real world applications. Using implicit links is known to be effective especially when we have limited labeled data. In one of these studies, a novel Bayesian framework for classification named Higher Order Naïve Bayes (HONB) has been introduced (Ganiz et.al, 2009; Ganiz et.al, 2011). HONB is built on a graph based data representation which leverages implicit higher-order links between attribute values across different instances (Ganiz et.al, 2009; Lytkin, 2009; Ganiz et.al, 2011). These implicit links are defined as higher-order paths. Attributes or features such as terms in documents of a text collection are richly connected by higher order paths of this kind. HONB exploits this rich connectivity (Ganiz et.al, 2009).

In this thesis, we follow the same practice of exploiting implicit link information by developing a novel semantic smoothing method for Naïve Bayes (NB). We call it Higher Order Smoothing (HOS).

1.2. Methodology of the Thesis

HOS is built on novel graph based data representation which is inspired from the data representation of HONB. However in HOS, we take the concept one step further and exploit the relationships between instances of different classes. This approach improves the parameter estimation in the face of sparse data conditions by reducing the sparsity. As a result, we move beyond instance boundaries and class boundaries as well to exploit the latent information in higher-order paths.

We perform extensive experiments by varying the size of the training set in order to simulate real world settings and compare our algorithm with different smoothing methods and other algorithms. Our results on several benchmark datasets show that HOS significantly boosts the performance of Naïve Bayes (NB) and on some datasets it even outperforms Support Vector Machines (SVM).

2. LITERATURE REVIEW

Text classification is defined as the task of automatically assigning a document to one or more predefined classes (or categories), based on its content. Documents are usually represented with the Vector Space Model (VSM) (Salton et al., 1975), a model borrowed from Information Retrieval (IR). In this model, documents are represented as a vector where each dimension corresponds to a separate word in the corpus dictionary. Therefore, the document is represented as a matrix where each row is a document and each column is a word. If a term occurs in the document then its value in the matrix is non-zero. In literature, several different ways of computing these values, also known as term weights, have been developed.

Generally, a large number of words exist in even a moderately sized set of documents; for example, in one data set we use (WebKb4) 16,116 words exist in 4,199 documents. However, each document typically contains only a small number of words. Therefore document-term matrix is a high-dimensional, typically very sparse matrix with almost 99% of the matrix entries being zero. Several studies have shown that, with the increase of dimensionality, inference based on pairwise distances becomes increasingly difficult (Beyer et.al, 1998; Verleysen and François, 2005). Although VSM is widely used, most of the commonly used classification algorithms such as k-nearest neighbors (k-NN), Naïve Bayesian and Support Vector Machines (SVM) rely on pairwise distances, hence suffer from the curse of dimensionality (Bengio et.al,2006). In order to overcome this problem, several approaches exploiting the latent information in higher-order co-occurrence paths between features within datasets have been proposed (Ganiz et.al, 2009; Ganiz et.al, 2011).

The underlying analogy of the concept 'higher-order' is that human do not necessarily use the same vocabulary when writing about the same topic. For example, in their study, Lemaire and Denhièr (2006) found 131 occurrences of word "internet", 94 occurrences of word "web", but no co-occurrences at all, in a 24-million words French corpus from the daily newspaper Le Monde. Obviously it can be seen that these two words are strongly associated and this relationship can be brought to light if the two words co-occur with other words in the corpus. For instance, consider a document set containing noteworthy number of co-occurrences between the words "quantum" and "computer", "computer" and "microprocessor". We could infer that there is a conceptual relationship beween the words "quantum" and "microprocessor", although they do not directly co-occur in any document. Relationships between "quantum" and "computer", "computer" and "microprocessor" is called as a first-order co-occurrence. The conceptual relationship between "quantum" and "microprocessor" is called a second-order co-occurence which can be generalized to higher (3rd, 4th, 5th, etc) order co-occurrences. Many algorithms have been proposed in order to exploit higher-order occurences between words such as the Singular Value Decomposition (SVD) based Latent Semantic Indexing (LSI).

At the very basic level, we are motivated by the LSI algorithm (Deerwester et.al, 1990), which is a widely used technique in text mining and IR. It has been shown that LSI takes advantage of implicit higher-order (or latent) structure in the association of words and documents. Higher-order relations in LSI capture "latent semantics" (Li et.al, 2005). There are several disadvantages of using LSI in classification. It is a highly complex, unsupervised, black box algorithm.

In their study, Kontostathis and Pottenger (2006) mathematically prove that LSI implicitly depends on higher-order co-occurrences. They also demonstrate empirically that higher-order co-occurrences play a key role in the effectiveness of systems based on LSI. Terms which are semantically similar lie closer to one another in the LSI vector space, so latent relationships among terms can be revealed.

Titles:	
c1:	*Human*machine *interface*for Lab ABC *computer* applications
c2:	A *survey* of *user* opinion of *computer system response time*
c3:	The *EPS user interface* management *system*
c4:	*System* and *human system* engineering testing of *EPS*
c5:	Relation of *user*-perceived *response time* to error measurement
m1:	The generation of random, binary, unordered *trees*
m2:	The intersection *graph* of paths in *trees*
m3:	*Graph minors* IV: Widths of *trees* and well-quasi-ordering
m4:	*Graph minors*: A *survey*

Figure 2. 1 Example document collection (Deerwester et al., 1990)

	t1	t2	t3	t4	t5	t6	t7	t8	t9	t10	t11	t12
human(t1)	x	1	1	0	2	0	0	1	0	0	0	0
interface(t2)	1	x	1	1	1	0	0	1	0	0	0	0
computer(t3)	1	1	x	1	1	1	1	0	1	0	0	0
user(t4)	0	1	1	x	2	2	2	1	1	0	0	0
system(t5)	2	1	1	2	x	1	1	3	1	0	0	0
response(t6)	0	0	1	2	1	x	2	0	1	0	0	0
time(t7)	0	0	1	2	1	2	x	0	1	0	0	0
EPS(t8)	1	1	0	1	3	0	0	x	0	0	0	0
survey(t9)	0	0	1	1	1	1	1	0	x	0	1	1
trees(t10)	0	0	0	0	0	0	0	0	0	x	2	1
graph(t11)	0	0	0	0	0	0	0	0	1	2	x	2
minors(t12)	0	0	0	0	0	0	0	0	1	1	2	x

Figure 2. 2 Deerwester Term-to-Term Matrix (Kontostathis and Pottenger, 2006)

	t1	t2	t3	t4	t5	t6	t7	t8	t9	t10	t11	t12
human(t1)	x	0.54	0.56	0.94	1.69	0.58	0.58	0.84	0.32	-	-	-
interface(t2)	0.54	X	0.52	0.87	1.50	0.55	0.55	0.73	0.35	-	-	-
computer(t3	0.56	0.52	x	1.09	1.67	0.75	0.75	0.77	0.63	0.15	0.27	0.20
user(t4)	0.94	0.87	1.09	x	2.79	1.25	1.25	1.28	1.04	0.23	0.42	0.31
system(t5)	1.69	1.50	1.67	2.79	x	1.81	1.81	2.30	1.20	-	-	-
response(t6)	0.58	0.55	0.75	1.25	1.81	x	0.89	0.80	0.82	0.38	0.56	0.41
time(t7)	0.58	0.55	0.75	1.25	1.81	0.89	x	0.80	0.82	0.38	0.56	0.41
EPS(t8)	0.84	0.73	0.77	1.28	2.30	0.80	0.80	x	0.46	-	-	-
survey(t9)	0.32	0.35	0.63	1.04	1.20	0.82	0.82	0.46	x	0.88	1.17	0.85
trees(t10)	-	-	0.15	0.23	-	0.38	0.38	-	0.88	x	1.96	1.43
graph(t11)	-	-	0.27	0.42	-	0.56	0.56	-	1.17	1.96	x	1.81
minors(t12)	-	-	0.20	0.31	-	0.41	0.41	-	0.85	1.43	1.81	x

Figure 2. 3 Deerwester Term-to-Term matrix (Kontostathis and Pottenger, 2006)

Let's consider a simple document collection given in Figure 2.1 where document $c1$ has the words {human, interface} and $c3$ has {interface, user}. As can be seen from the co-occurrence matrix in Figure 2.2, the terms "human" and "user" do not co-occur in this example collection. After applying LSI, however, the reduced representation co-occurrence matrix in Figure 2.3 has a non-zero entry for "human" and "user" thus implying a similarity between the two terms. This is an example of second-order co-occurrence; in other words, there is a second-order path between "human" in $c1$ and "user" in $c3$ through "interface" (common to both $c1$ and $c3$). This second-order path implicitly links $c1$ to $c3$

, violating the IID assumption. The results of experiments reported in (Kontostathis and Pottenger, 2006) show that there is a strong correlation between second-order term co-occurrence, the values produced by SVD algorithm used in LSI, and the performance of LSI measured in terms of F-measure , the harmonic mean of precision and recall. As noted, the authors also provide a mathematical analysis which proves that LSI does in fact depend on higher-order term co-occurrence (Ganiz et.al, 2011).

A second motivation stems from the studies in link mining which utilize explicit links (Getoor and Diehl, 2005). Several studies in this domain have shown that significant improvements can be achieved by classifying multiple instances collectively (Chakrabarti et.al, 1998; Neville and Jensen, 2000; Taskar et.al, 2002). However, use of explicit links requires an additional context for classification of a single instance. This limitation restricts the applicability of these algorithms. There are also several studies which exploit implicit link information in order to improve the performance of machine learning models (Ganiz et.al, 2006; Ganiz et.al, 2009; Ganiz et.al, 2011). Using implicit link information within data provides a richer data representation and it is shown to be effective especially under the scarce training data conditions. In one of these a novel Bayesian framework for classification named Higher Order Naïve Bayes (HONB) is introduced (Ganiz et.al, 2009; Ganiz et.al, 2011).

HONB employs a graph based data representation and leverages co-occurrence relations between attribute values across different instances. These implicit links are named as higher-order paths. Attributes or features such as terms in documents of a text collection are richly connected by such higher-order paths. HONB exploits this rich connectivity (Ganiz et.al, 2009). Furthermore, this framework is generalized by developing a novel data driven space transformation that allows vector space classifiers to take advantage of relational dependencies captured by higher-order paths between features (Ganiz et.al, 2009). This led to the development of Higher Order Support Vector Machines (HOSVM) algorithm. Higher-order learning which a statistical relational learning framework consists of several supervised and unsupervised machine learning algorithms in which relationships between different instances are leveraged via higher order paths (Li et.al, 2005; Lytkin, 2009; Edwards and Pottenger, 2011).

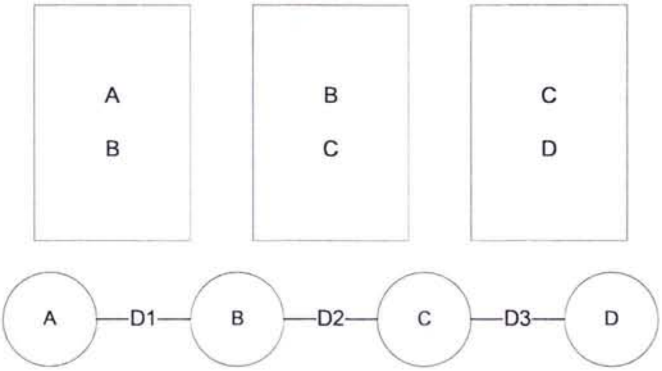

Figure 2. 4 Higher order co-occurrence (Kontostathis and Pottenger, 2006)

A higher order path is shown in Figure 2.4 (reproduced from (Kontostathis and Pottenger, 2006)). This figure depicts three documents, $D1$, $D2$ and $D3$, each containing two terms represented by the letters A, B, C and D. Below the three documents there is a higher-order path that links term A with term D through B and C. This is a third-order path since three links, or "hops," connect A and D. Similarly, there is a second order path between A and C through B. A co-occurs with B in document $D1$, and B co-occurs with C in document $D2$. Even if terms A and C never co-occur in any of the documents in a corpus, the regularity of these second order paths may reveal latent semantic relationship such as synonymy. As well as HONB, several studies in different areas of natural language processing have employed graph based data representation for decades. These areas include, among others, document clustering and text classification.

Text classification is the task of assigning a document to appropriate classes or categories in a predefined set of categories. However, in the real world, as the number of documents explosively increases, the number of categories reaches a significantly large number so it becomes much more difficult to browse and search the categories. In order to solve this problem, categories are organized into a hierarchy like Open Directory Project (ODP) and the Yahoo! directory. Hierarchical classifiers are widely used when categories are organized in hierarchy; however, many data sets are not organized in hierarchical forms in real world.

To handle this problem, authors of the study (Gao et.al, 2005), propose a novel algorithm to automatically mine hierarchical taxonomy from the data set in order to take advantage of hierarchical classifier. In their approach, they model the relationship between categories, documents and terms by a tripartite graph and partition it using consistent bipartite spectral graph co-partitioning (CBSGC) algorithm. They use two bipartite graphs for representing relationships between categories-documents and documents-terms. As can be seen in Figure 2.5, a document is used as a bridge between these two bipartite graphs to generate a category–document–term tripartite graph. CBSGC is a recursive algorithm to partition the tripartite graph which terminates when subsets of the leaf nodes contains only one category. Their experiments show that, CBSGC discover very reasonable hierarchical taxonomy and improves the classification accuracy on 20 Newsgroups dataset.

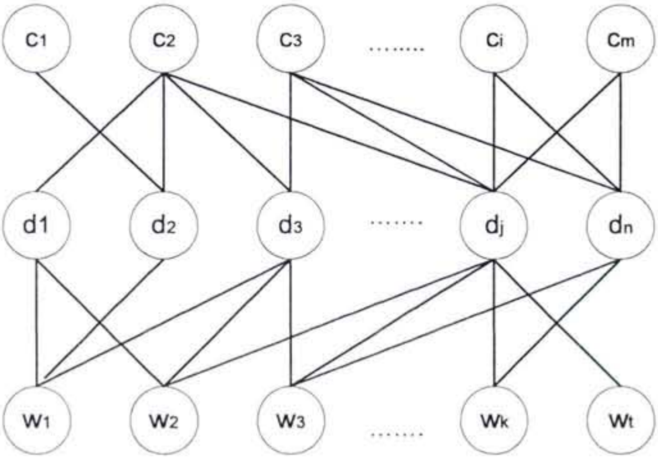

Figure 2. 5 The category-document-term tripartite graph (Gao et.al, 2005)

In another study, Mengle and Goharian (2010) intend to discover the relationships among document categories which are represented in the form of a concept hierarchy. In their approach, they represent such relationships in a graph structure called Relationship-net shown in Figure 2.6, where categories are the vertices of this graph and edges are the relationship among them. In a category hierarchy, only the relationships among categories sharing the same parent are represented. Therefore, identifying relationships among non-sibling categories (categories with different parents) is limited. In Relationship-net, relationships between non-sibling categories as well as sibling categories are presented so authors identify more relationships than a hierarchical taxonomy does. To identify the relationships among categories, a text classifier's misclassification information is utilized. This approach relies on the finding that categories which mostly are misclassified as each other indeed are relevant. They evaluate 20Newsgroup, ODP and SIGIR data sets in their experiments and results show that Relationship-net based on misclassification information statistically significantly outperforms the CBSCG approach.

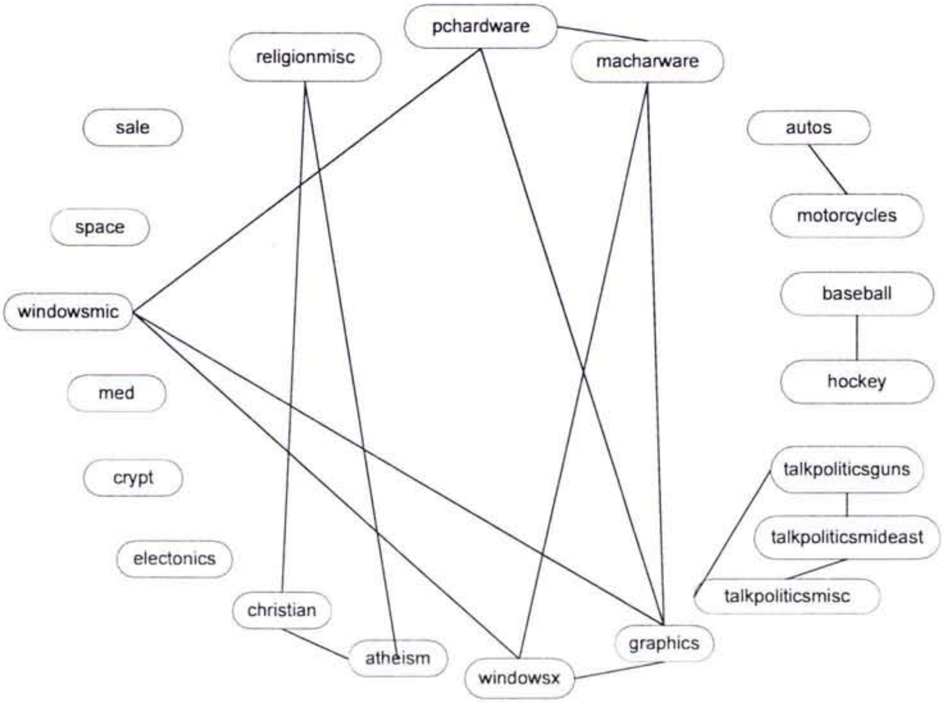

Figure 2. 6 Relationship-net for the 20NG data set (Mengle and Goharian, 2010)

Besides using graph structure in hierarchical taxonomy, several works employing graphs for clustering documents have been proposed. Clustering is the task of partitioning a set of objects into groups (or clusters) such that similar objects are in the same cluster while dissimilar objects are in different clusters. Homogeneous data clustering has been studied for years in the literature of machine learning and data mining, however, heterogeneous data clustering has attracted more and more attention in recent years. Underlying this issue is that the similarities among one type of objects sometimes can only be defined by the other type of objects especially when these objects are highly interrelated. For instance, documents and terms in a text corpus, reviewers and movies in movie recommender systems, are highly interrelated heterogeneous objects. In these examples, traditional clustering algorithms might not work very well. In order to avoid this problem, many researchers started to extend traditional clustering algorithms and propose graph partitioning algorithms to co-cluster heterogeneous objects simultaneously (Dhillon, 2001; Zha et.al, 2001).

In his study, Dhillon (2001) considers the problem of simultaneous co-clustering of documents and words. Most of the existing algorithms based on separate clustering, either documents or words but not simultaneously. Document clustering algorithms, cluster documents based upon their word distributions whereas word clustering algorithms uses words' co-occurrences in documents. Therefore, there is a dual relationship between document and word clustering as they both induce each other. This characterization is recursive because document clusters determine word clusters, which in turn determine (better) document clusters. In his approach, he represents a document collection as a bipartite graph shown in Figure 2.7 and proposes an algorithm to solve this dual clustering problem.

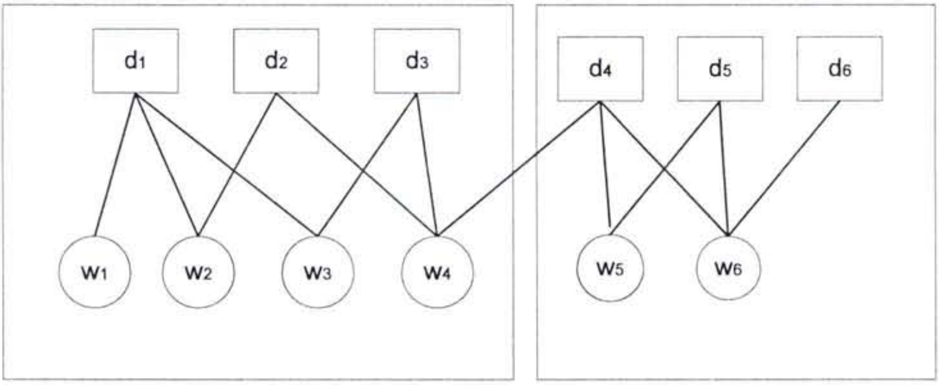

Figure 2. 7 A bipartite graph of document and words (Dhillon, 2001)

It is obvious that, better word and document clustering can be achieved by partitioning the graph such that the crossing edges between partitions have minimum weight. Therefore, simultaneous clustering problem become a bipartite graph partitioning problem. His algorithm partitions documents and words simultaneously by finding minimum cut vertex partitions in this bipartite graph, and provides good global solution in practice. He uses popular Medline (1033 medical abstracts), Cranfield (1400 aeronautical systems abstracts) and Cisi (1460 information retrieval abstracts) data sets in their experiments and his results verify that proposed co-clustering algorithm works well on real examples.

In another study (Zha et.al, 2001), the authors also represent documents and terms as vertices in a bipartite graph, where edges of the graph are the co-occurrence of the term and document. In their approach, they propose a clustering method based on partitioning this bipartite graph. Unlike from (Dhillon, 2001), normalized sum of edge weights between unmatched pairs of vertices of the bipartite graph is minimized to partition the graph. They show that, by computing partial SVD of the associated edge weight matrix of the bipartite graph, an approximate solution to the minimization problem can be obtained. In their experiments they apply their technique successfully on document clustering.

Clustering methods based on pair-wise similarity of data points, such as spectral clustering methods require finding eigenvectors of the similarity matrix. Therefore, even though these methods shown to be effective on a variety of tasks, they are prohibitively expensive when applying on large-scale text datasets. To tackle this problem, authors of the study (Frank and Cohen, 2010), represent a text data set as a bipartite graph in order to propose a fast method for clustering big text datasets. Documents and words correspond to vertices in the bipartite graph and the number of paths between two document vertices is used as a similarity measure. According to their results, even if proposed method runs much faster from previous methods, it works as well as them in clustering accuracy.

As distinct from above studies clustering documents, authors of the study (Caimei et.al, 2011), propose a novel clustering method called "Tripartite Clustering" which clusters a social tagging data set. Sets of users, resources and tags are elements of a social tagging system hence it is naturally based on a tripartite form. In their tripartite graph representation shown in Figure 2.8, each of these elements corresponds to vertices and a vertex is characterized by its link to the other two types of vertices. They compare Tripartite Clustering with K-means in their experiments and results show that, their method achieves equivalent or better performances and produces more useful information.

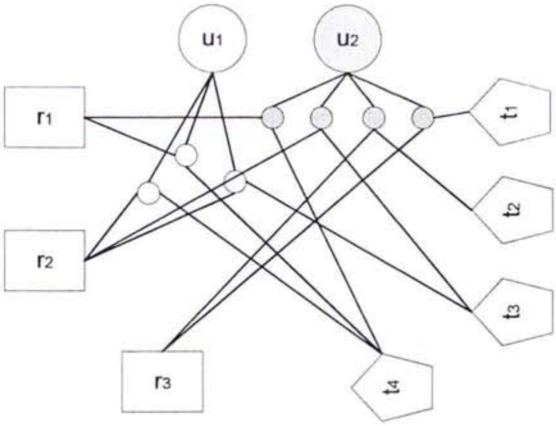

Figure 2. 8 Example of the tripartite network of social tagging system (Caimei et.al, 2011)

Although there are several works for co-clustering two types of heterogeneous objects (denoted by pair wise clustering) such as documents and terms, works for co-clustering more types of heterogeneous data (denoted by higher-order clustering) is still very limited. In their study, Gao et.al (2005) work on co-clustering higher-order objects in which there is a central object connecting the other objects so as to form a star structure. According to them, this structure could be a very good abstract for real-world situations, such as co-clustering categories, documents and terms in text mining where the central object for the star is documents. Their premise for the star structure is that, they treat co-clustering categories, documents and terms problem as a union of multiple pair wise co-clustering problems with the constraint of the star structure. Therefore, they develop an algorithm based on semi-definite programming (SDP) for efficient computation of the clustering results. In their experiments on toy problems and real data, they verify the effectiveness of their proposed algorithm.

Clustering algorithms are described as unsupervised machine learning algorithms because they are not provided with a labeled training set. On the other hand, Hussain and Bisson (2010) propose a two-step approach for expanding the unsupervised X-Sim co-clustering algorithm to deal with text classification task. In their approach, firstly, they introduce a priori knowledge by introducing class labels into the training dataset while initializing X-Sim. Underlying concept of this is that X-Sim algorithm exploits higher-order similarities within a data set hence adding class labels will force higher-order co-occurrences. Secondly, they introduce a method to reduce the similarity values of documents in different classes. Therefore, the influence of higher-order co-occurrences between documents in different categories is promoted. According to their experiment results, the proposed approach which is an extension of the X-Sim co-clustering algorithm gain performance equal or better to both traditional and state-of-the-art algorithms like k-NN, supervised LSI and SVM.

In another study, Radev (2004) proposes a tripartite updating method for a number classification task which is especially important in question answering systems. In his study, he defines a bipartite graph shown in Figure 2.9 where features are vertices and these vertices are connected with labeled and unlabeled examples.

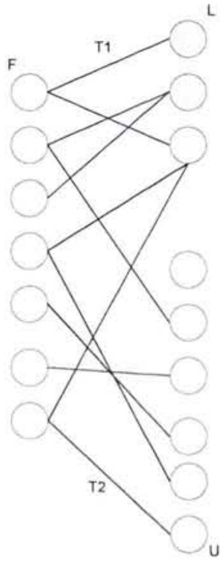

Figure 2. 9 Bipartite representation (Radev, 2004)

In order to evaluate the performance of proposed method, he compares tripartite updating with a weakly supervised classification algorithm based on graph representation, spectral partitioning. This algorithm is known as weakly supervised in the literature because it requires a small number of labeled examples. His experimental results show that, tripartite updating outperforms spectral partitioning even though they both require minimal labeled data. The results also indicate that, both methods scale well to different ratios between the number of labeled training examples and the number of unlabeled examples.

The authors of study (Gharahmani and Lafferty, 2003) also use a weighted graph to introduce a new classification method based on the Gaussian random field. Labeled and unlabeled instances are the vertices of this graph where edge weights represent the similarity between them. In order to identify the labeled node that is closest to a given unlabeled instance based on the graph topology, they apply belief propagation. They perform experiments on text and digit classification and promising results demonstrate that, proposed method has the potential to draw advantage from unlabeled data efficiently to improve classification accuracy.

There are two commonly referred event models in Naïve Bayes for text categorization; multivariate Bernoulli (MVNB) and multinomial models (MNB). The first one is also known as binary independence model. In this model presence and absence of the terms is represented respectively "1" and "0". On the other hand in multinomial model is a unigram language model with integer term counts. Thus, each class can be defined as a multinomial distribution. Multinomial model is actually a unigram language model (McCallum and Nigam, 1998). McCallum and Nigam (1998) compare multivariate Bernoulli and multinomial model on several different data sets. Their experimental results show that the multivariate Bernoulli event model represents better performance at smaller vocabulary sizes, whereas the multinomial model generally performs well with large vocabulary sizes. Most of the studies about Naïve Bayes text classification employs multinomial model based on the recommendation of the well-known paper of McCallum and Nigam (McCallum and Nigam, 1998). However, there are some interesting studies using binary data. For instance, MNB is shown to perform better with binary data in some cases such as spam detection (Schneider, 2004; Metsis et.al, 2006). In another study Kim et.al (2006), propose a multivariate Poisson Naïve Bayes text classification model with weight-enhancing method to improve performances on rare categories. Their experiments show that, this model is a good alternative to traditional Naïve Bayes classifier because it allows more reasonable parameter estimation when a low number of training documents is available.

In general, NB parameter estimation drastically suffer from sparse data because it has so many parameters to estimate in text classification problems $(|V||C|+|C|)$ where $|V|$ denotes the dictionary and $|C|$ denotes the set of class labels (McCallum and Nigam, 1999). Most of the studies on NB text classification employ Laplace smoothing by default. There are a few studies that attempt to use different smoothing methods. For instance Juan and Ney (2002) use multinomial model with several different smoothing techniques which origin from statistical language modeling field and generally used with n-gram language models. These include absolute discounting with unigram backing-off and absolute discounting with unigram interpolation. They state that absolute discounting with unigram interpolation gives better results than Laplace smoothing. They also consider document length normalization. Peng et.al (2004), augment NB with n-grams and advanced smoothing methods from language modeling domain such as linear interpolation, absolute smoothing, Good-Turing Smoothing and Witten–Bell smoothing.

In another study (Chen and Goodman, 1998), authors propose a semantic smoothing method based on the extraction of topic signatures. Topic signatures correspond to multi-word phrases such as n-grams or collocations that are extracted from the training corpus. After having topic signatures and multiword phrases they used them in semantic smoothing background collection model to smooth and map the topic signatures. They demonstrate that when the training data is small, the NB classifier with semantic smoothing outperforms better than NB with background smoothing (Jelinek-Mercer) and Laplace smoothing.

SVM is a popular large margin classifier. This machine learning method aims to find a decision boundary that separates points into two classes thereby maximizing margin (Joachims, 1998). SVM projects data points into a higher dimensional space so that the data points become linearly separable by using kernel techniques. There are several kernels that can be used SVM algorithm. Linear kernel is known to perform well on text classification. We include SVM results in our experiments for comparison reasons.

3. METHODOLOGY

3.1. Theoretical Background

In this section we review the Naïve Bayes event models and data representations. Although our method is not restricted to a particular application domain we focus on textual data.

3.2. Naïve Bayes Event Models

Naïve Bayes is one of the most popular and commonly used machine learning algorithms in text classification due to its easy implementation and low complexity. There are two generative event models that are commonly used with Naïve Bayes (NB) for text classification. First and the less popular one is multivariate Bernoulli event model which is also known as binary independence NB model (MVNB). In this model, documents are considered as events and they are represented a vector of binary attributes indicating occurrence of terms in the document. Given a set of class labels $C=\{c_1,....,c_k\}$ and the corresponding training set D_j of documents representing class c_j, for each j $\{1,....,K\}$. The probability that a document in class c_j, will mention term w_i. With this definition (Chakrabarti, 2002),

$$P(d|c_j)=\prod_{w\in d}\frac{P(w_i|c_j)}{1-P(w_i|c_j)}\prod_{w\in W}\left(1-P(w_i|c_j)\right) \tag{3.1}$$

Conditional probabilities $P(w_i\,|\,c_j)$ are estimated by

$$P(w_i\,|\,c_j)=\frac{1+\sum_{d\in D_j}^{|D|}w_i(d)}{2+|D_j|} \tag{3.2}$$

which is ratio of the number of documents that contain term w_i, in class c_j, to the total number of documents in class c_j. The constants in numerator and denominator in (3.2) are introduced according to Laplace's rule of succession in order to avoid zero-probability terms (Ganiz et.al, 2009). Laplace smoothing adds a pseudo count to every word count. The main disadvantage of Laplace is to give too much probability mass to previously unseen events.

Second NB event model is multinomial model (MNB) which can make use of term frequencies. Let term w_i occur $n(d,w_i)$ times in document d, which is said to have length $\ell_d = \sum_{w_i} n(d,w_i)$. With this definition

$$\begin{aligned} P(d|c_j) &= P(L=\ell_d|c_j)\,P(d \mid \ell_d, c_j) \\ &= P(L=\ell_d|c_j)\binom{\ell_d}{\{n(d,w_i)\}}\prod_{w_i \in d}\theta_t^{n(d,w_i)} \end{aligned} \tag{3.3}$$

Class conditional term probabilities are estimated using (3.4).

$$P(w_i|c_j) = \frac{1 + \sum_{d \in D_j}^{|D|} n(d,w_i)}{|W| + \sum_{d \in D_j, w_i \in d} n(d,w_i)} \tag{3.4}$$

where $|W|$ is vocabulary (total number of words) (Chakrabarti, 2002).

Because of sparsity in training data, missing terms (unseen events) in the document can cause "zero probability problem" in NB. To eliminate this, we need to distribute some probability mass to unseen terms. This process is known as smoothing. The most common smoothing method in NB is Laplace smoothing. Formulas of the NB event models in (3.2) and (3.4) already included Laplace smoothing. In the next section, we provide details of a more advanced smoothing method which perform well especially on MVNB.

3.2.1. Jelinek-Mercer Smoothing

In Jelinek-Mercer smoothing method, the maximum estimate is interpolated with the smoothed lower-order distribution (Chen and Goodman, 1998). This is achieved by linear combination of maximum likelihood estimate in (3.5) with the collection model in (3.6) as shown in (3.7). In (3.6), $|D|$ represents the whole training set, including the documents from all classes.

$$P_{ml}\left(w_i \middle| c_j\right) = \frac{\sum_{d \in D_j}^{|D|} w_i(d)}{\left|D_j\right|} \tag{3.5}$$

$$P\left(w_i \middle| D\right) = \frac{\sum_{d}^{|D|} w_i(d)}{|D|} \tag{3.6}$$

$$P\left(w_i \middle| c_j\right) = (1-\beta) \times P_{ml}\left(w_i \middle| c_j\right) + \beta \times P(w_i|D) \tag{3.7}$$

3.2.2. Higher Order Data Representation

Data representation we built on is initially used in (Ganiz et.al, 2011). In this study, it is indicated that definition of a higher-order path is similar to the one in graph theory, which states that given a non-empty graph $G=(V,E)$ of the form $V=\{x_0, x_1, ..., x_k\}$, $E=\{x_0x_1, x_1x_2, ..., x_{k-1}x_k\}$ with nodes x_i distinct, two vertices x_i and x_k are linked by a path P where the number of edges in P is its length.

A different approach is given in (Ganiz et.al, 2009) by using a bipartite graph. In this approach a bipartite graph $G=((V_{D,}V_W),E)$ is built from a set of D documents for a better representation. As it can be seen in Figure 3.1, in this graph, vertices in V_D correspond to documents and vertices in V_W correspond to terms.

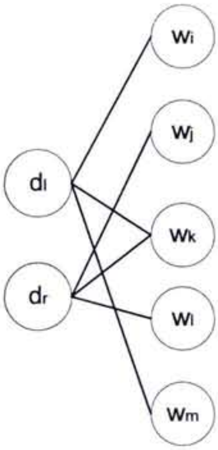

Figure 3. 1 Bipartite graph representation of documents and terms

"There is an edge (d,w) between two vertices where $d \in V_D$ and $w \in V_W$ iff word w occurs in document d. In this representation, a higher-order path in dataset D can be considered as a chain sub graph of G. For example a chain $w_i - d_l - w_k - d_r - w_j$ which is also denoted as $(w_i, d_l, w_k, d_r, w_j)$ is a second-order path since it spans through two different document vertices. Higher-order paths simultaneously capture term co-occurrences within documents as well as term sharing patterns across documents, and in doing so provide a much richer data representation than the traditional feature vector form" (Ganiz et.al, 2009).

3.2.3. Higher Order Naïve Bayes

Rich relational information between terms and documents can be exploited by using higher-order paths. In Higher Order Naïve Bayes (HONB) this valuable information is integrated into multivariate Bernoulli Naïve Bayes algorithm (MNVB) by estimating parameters from higher-order paths instead of documents (Ganiz et.al, 2009). Formulation of parameter estimates are given in (3.8) and (3.9) which are taken from (Ganiz et.al, 2009).

$$P(w_i|c_j) = \frac{1+\varphi(w_i, D_j)}{2+\phi(D_j)} \tag{3.8}$$

$$P(c_j) = \frac{\phi(D_j)}{\sum_{k=1}^{K} \phi(D_k)} \tag{3.9}$$

The number of higher-order paths containing term w_i given the set of documents that belongs c_j is represented by $\varphi(w_i, D_j)$. On the other hand, $\phi(D_j)$ denote the total number of higher-order paths extracted from the documents of c_j. In (3.8) the Laplace smoothing is included in order to avoid zero probability problem for the terms that do not exist in c_j.

3.3. Higher Order Smoothing

In this section we present a novel semantic smoothing method called Higher Order Smoothing (HOS) by following the same approach of exploiting implicit link information. HOS is built on a graph-based data representation from the previous algorithms in higher-order learning framework such as HONB (Ganiz et.al, 2009; Ganiz et.al, 2011). However, in HONB, higher-order paths are extracted in the context of a class. Therefore we cannot exploit relations between terms and documents in different classes.

In HOS we take the concept one step further and exploit the relationships between instances of different classes in order to improve the parameter estimation. As a result, we are not only moving beyond document boundaries but also class boundaries to exploit the latent semantic information in higher-order co-occurrence paths between terms (Poyraz et.al, 2012). We accomplish this by extracting higher-order paths from the whole training set including all classes of documents. Our aim is to reduce sparsity especially in the face of insufficient labeled data conditions.

In order to do so, we first convert the nominal class attribute to a number of binary attributes each representing a class label. For instance, in WebKb4 dataset 'Class' attribute has the following set of values $C = \{course, faculty, project, staff, student\}$. We add these four class labels as new terms (i.e. columns to our document by term matrix). We call them "class labels". Each of these labels indicates if the given document belongs to a particular class or not.

After this transformation, we slightly modify the higher-order data representation by characterize a set of D documents as a tripartite graph. In this tripartite graph $\hat{G}=((V_W,V_C,V_D),E)$, vertices in V_D correspond to documents, vertices in V_W correspond to terms, and finally vertices in V_C correspond to class terms or in other words class labels. Figure 3.1 shows such a tripartite graph which represents relationship between terms, class labels, and documents. Similarly, to previous higher-order data representation with bipartite graph, a higher-order path in dataset D can be considered as a chain sub graph of $\hat{G}$. However, we are interested in such chain sub graphs that start with a term vertex from V_W, spans through different document vertices in V_D, and terminate with a class term vertex in V_C. $w_i-d_s-w_k-d_r-c_j$ is such a chain which we denote by (w_i,d_s,w_k,d_r,c_j). This chain corresponds to a second-order path since it spans through two document vertices. These paths have potential to cross class boundaries and capture latent semantics. We enumerate higher-order paths between all the terms in the training set and the class terms. These higher-order paths capture the term co-occurrences within a class of documents as well as term relation patterns across classes. As a result, they provide more dense data representation than the traditional vector space. This is the basis of our smoothing algorithm.

Let's consider $w_1-d_1-w_2-d_2-c_1$ which is an example chain is in the tripartite graph given in Figure 3.2. This chain is indicated with red bold lines and it corresponds to a second-order path. In this example let's assume that w_1 never occurs in the documents of c_1. We still can estimate parameter value of w_1 for c_1 using such paths. This is achieved by intermediate terms such as w_2 that co-occurs with w_1 (given w_2 occurs in the documents of c_1). As can be seen from the example, this new data representation and the new definition of higher-order paths allow us to calculate class conditional probabilities for some of the terms that do not occur in documents of a particular class. This framework serves as a semantic smoothing method for estimating model parameters of previously unseen terms given the fact that higher-order paths reveal latent semantics (Kontostathis and Pottenger, 2006).

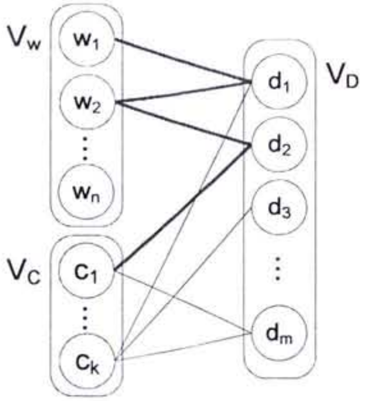

Figure 3. 2 Data representation for HO paths using tripartite graph (Poyraz et.al, 2012)

Based on this representation and modified definition of higher-order paths we can formulate HOS. Let $\partial(w_i, c_j)$ denote the number of higher-order paths that is between term w_i and class label c_j in the dataset D, and $\Phi(D)$ denote the total number of higher-order paths between all terms and all class terms in D. Please note that D represents all documents from all classes. This is one of the important differences between the formulation of HONB and HOS. The parameter estimation equation of the proposed HOS is given in (3.10). Although HOS has the potential to estimate parameters for terms that do not occur in the documents of a class but occurs in other classes in training data, there can be terms that occur only in test set. In order to avoid zero probability problems in these cases, we apply Laplace smoothing in (3.10). Class priors are calculated according to multivariate Bernoulli model using documents.

$$P(w_i|c_j) = \frac{1 + \partial(w_i, c_j)}{2 + \Phi(D)} \tag{3.10}$$

We recognize that different orders of paths may have different contribution to semantics and provide even richer data representation. Similar to the linear interpolation (a.k.a. Jelinek-Mercer) we can combine estimates calculated from different order of paths. In (3.11) the linear combination of first-order paths (just co-occurrences) with second-order paths is shown. We use this formulation in our experiments. We set β to 0.5 experimentally since for majority of our datasets and training set size percentages, 0.5 performs best.

$$P(w_i \mid c_j) = (1-\beta) \times P_{fo}(w_i \mid c_j) + \beta \times P_{so}(w_i \mid c_j) \tag{3.11}$$

The overall process of extracting second-order paths for HOS is described in Algorithm 1. It is based on the enumeration algorithm proposed in (Ganiz et.al, 2009) which is described in detail in (Lytkin, 2009).

Algorithm 1 : Enumerating second-order paths for HOS

```
Input : Boolean document-term data Matrix $X = X_d^t$

Output: $O_2$ matrix which stores the number of second-order paths in data Matrix $X$

1. Initialize vector $l = (l^1, \ldots, l^n)$, which will store class labels of given data matrix $X$

2. for each row $i$ in data matrix $X$

        2a. $l^i = X_i^{t-1}$

3. Initialize class labels binary matrix $C_{lb} = C_{lb\,d}^{\,c}$ which will represent each class value as binary where $c$ is the number of classes in data matrix $X$

4. for each row $i$ in $C_{lb}$ matrix

    4a. for each column $c$ in $C_{lb}$ matrix

        4b. if $l^i$ is equal to $j$

            4c. set $C_{lb}(i, j)$ equal to 1

5. Compute matrix $X_{clb} = X_{clb\,d}^{\,t+c}$ by appending binary class valued matrix $C_{lb}$ to data matrix $X$

6. Compute first-order co-occurrence matrix $O_1 = X_{clb}^{\,T} X_{clb}$

7. Compute second-order co-occurrence matrix $O_2 = O_1 O_1$

8. for each row $i$ in first-order co-occurrence matrix $O_1$

    8a. for each column $j$ in first-order co-occurrence matrix $O_1$
    8b. Compute scalar $s$, to eliminate paths $t_1, d_1, t_2, d_1, t_3$, where both document vertices ($d_1$) are same
        $s = O_2(i,j) - (O_1(i,j) * (O_1(i,i) + O_1(j,j)))$
    8c. Update the element of second-order co-occurrence matrix, $O_2(i,j) = O_2(i,j) + s$
9. Return $O_2$
```

In algorithm 1, first, class labels are removed from given Boolean document-term data matrix and stored in a vector. Then, using class labels vector, a binary class labels matrix which represents each class value as binary, is built. Afterwards, class labels removed data matrix and binary class labels matrix are combined. In this instance, we have a new matrix called class-binarized matrix X_{clb} which stores the input data matrix and its binary class values. We use X_{clb} to calculate the first and second order paths. First order paths matrix is calculated by multiplying X_{clb} by its transpose. Second order paths matrix is calculated by multiplying first order paths by itself. Finally, scalar s if computed in order to eliminate paths, where both document vertices d_1 are same and second order paths matrix is updated using this scalar value.

4. CONCLUSION

4.1. Experiment Results

In order to analyze the performance of our algorithm for text classification, we use three widely used benchmark datasets. First one is a variant of 20 Newsgroups[1] dataset. It is called 20News-18828 and it has fewer documents from the original 20 Newsgroup dataset since duplicates postings are removed. Additionally for each posting headers are deleted except "From" and "Subject" headers. Our second dataset is the WebKB[2] dataset which includes web pages collected from computer science departments of different universities. There are seven categories which are student, faculty, staff, course, project, department and other. We use four class version of the WebKB dataset which is used in (McCallum and Nigam, 1998). This dataset is named as WebKB4. Third dataset is 1150Haber dataset which consists of 1150 news articles in five categories namely economy, magazine, health, politics and sport collected from Turkish online newspapers (Amasyalı and Beken, 2009). We particularly choose a dataset in different language in order to observe efficiency of higher-order algorithms in different languages. Similar to LSI, we expect higher-order paths based algorithms HONB and HOS to perform well on different languages without any need for tuning. More information about this data set and text classification on Turkish documents can be found in (Torunoğlu et.al, 2011). One of the most important differences between WebKB4 and other two datasets is the class distribution. While 20News-18828 and 1150Haber have almost equal number of documents per class, WebKB4 have highly skewed class distribution. For the statistics given in Table 4.1, we apply no stemming or stop word filtering. We only filter infrequent terms whose document frequency is less than three. Descriptions of the datasets, under these conditions are given in Table 4.1 including number of classes $(|C|)$, number of documents $(|D|)$ and the vocabulary size $(|V|)$.

[1] http://people.csail.mit.edu/people/jrennie/20Newsgroups

[2] http://www.cs.cmu.edu/~textlearning

Table 4. 1 Descriptions of the datasets with no preprocessing

DATA SET	|C|	|D|	|V|
20NEWS-18828	20	18,828	50,570
WEBKB4	4	4,199	16,116
1150HABER	5	1150	11,038

As can be seen from Algorithm 1, complexity of the higher-order path enumeration algorithm is proportional to the number of terms. In order avoid unnecessary complexity and to finish experiments on time we reduce the dictionary size of all three datasets by applying stop word filtering and stemming using Snowball stemmer. Finally, dictionary sizes are fixed to 2,000 by selecting the most informative terms using Information Gain feature selection method. All of these preprocessing operations are widely applied in the literature and it has been known that they usually improve the performance of traditional vector space classifiers. For that reason, we are actually giving a considerable advantage to our baseline classifier NB and SVM. Please note that HOS is expected to work well when the data is very sparse. In fact, these preprocessing operations reduce sparsity. As mentioned before we vary the training set size by using following percentages of the data for training and the rest for testing: 1%, 5%, 10%, 30%, 50%, 70%, 80% and 90%. These percentages are indicated with "ts" prefix to avoid confusion with accuracy percentages. We take class distributions into consideration while doing so. We run algorithms on 10 random splits for each of the training set percentages and report average of these 10 results augmented by standard deviations. While splitting data into training and test set, we employ stratified sampling. This approach is similar to (McCallum and Nigam, 1998) and (Rennie et.al, 2003) where they use 80% of the data for training and 20% for test.

Our dataset include term frequencies (tf). However, higher-order paths based classifiers HONB and HOS currently can only work with binary data. Therefore they convert term frequencies to binary values in order to enumerate higher-order paths. We use up to second-order paths based on the experiment results of previous studies (Ganiz et.al, 2009; Ganiz et.al, 2011). Since we use binary data, our baseline classifier is multivariate Bernoulli NB (MVNB) with Laplace smoothing. This is indicated as MVNB in the results.

We also employ more advanced smoothing method with MVNB which is Jelinek-Mercer (JM). Furthermore, we compare our results with HONB and state of the art text classifier SVM. We used linear kernel in SVM since it has been known to perform well in text classification. Additionally, we optimize soft margin cost parameter C by using the set of $\{10^{-3},....,1,10^{1},...,10^{3}\}$ of possible values. We picked the smallest value of C which resulted in the highest accuracy. We observed that C value is usually 1 when the training data is small (e.g. up to 10%) and it is usually 10^{-2} when training data increase (e.g. after 10%) with the exception of 1150Haber which is our smallest dataset. In 1150Haber, best performing C value is 1 in all training set percentages except 90%.

We use accuracy augmented by standard deviations as our primary evaluation metric. Tables show accuracies of algorithms. We only provide F-measure (F1) and AUC values for 80% training data level due to length restrictions. However, we observe that they exhibit same patterns. We also provide statistical significance tests in several places by using Student's t-Test especially when accuracies of different algorithms are close to each other. We use $\alpha = 0.05$ significance level and consider the difference is statistically significant if the probability associated with Student's t-Test is lower.

Our experiments show that HOS demonstrate remarkable performance on 20 Newsgroups dataset. This can be seen in Table 4.2 and Figure 4.1. HOS statistically significantly outperforms our baseline classifier MVNB (with Laplace smoothing) by a wide margin in all training set percentages. Moreover, HOS statistically significantly outperforms all other algorithms including NB with Jelinek-Mercer smoothing (MVNB+JM), HONB, and even SVM.

Table 4. 2 Accuracy and standard deviations of algorithms on 20 Newsgroups dataset

TS	MVNB	MVNB+JM	HOS	HONB	SVM
1	24.77±2.49	48.01±1.37	42.92±3.61	44.09±2.04	32.65±1.75
5	55.68±1.26	69.10±0.68	65.81±1.57	64.65±0.92	56.16±1.11
10	65.01±1.57	72.95±1.42	**76.70±0.79**	69.93±0.62	65.15±0.61
30	72.83±0.74	75.66±0.63	**81.97±0.33**	76.12±0.38	75.99±0.61
50	75.11±0.58	76.64±0.68	**83.06±0.29**	78.53±0.37	79.35±0.34
70	75.65±0.64	76.81±0.67	**83.33±0.54**	79.92±0.34	81.53±0.32
80	76.29±0.58	77.01±0.71	**83.59±0.41**	80.49±0.50	82.07±0.46
90	76.21±1.18	76.50±1.02	**83.26±0.84**	80.11±0.65	82.38±1.15

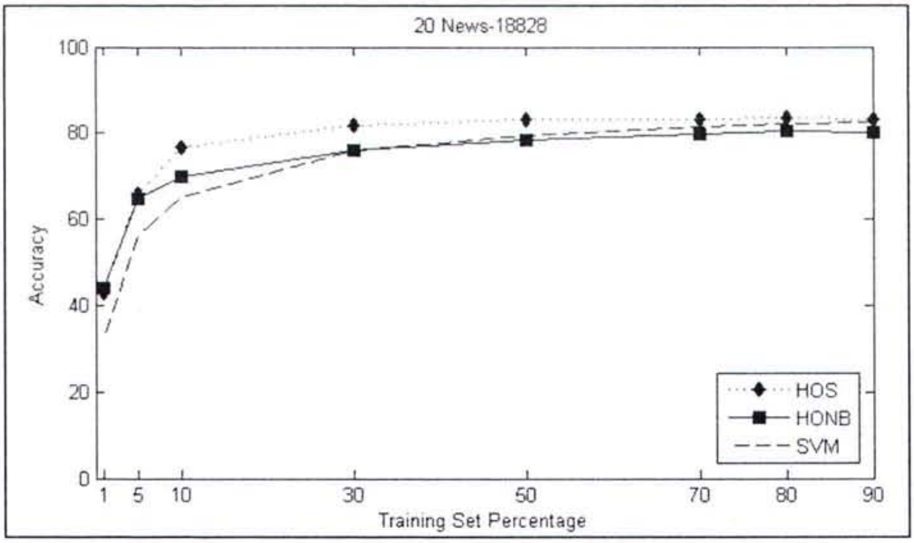

Figure 4. 1 Accuracy of HOS, HONB and SVM on 20News-18828

Table 4.3 and Figure 4.2 show the performance of HOS on WebKB4 dataset. Although not as visible as 20 Newsgroups dataset, HOS still outperforms our baseline MVNB starting from 10% training set level. The performance differences are statistically significant.

Additionally, HOS statistically significantly outperforms MVNB with JM smoothing starting from 30% level. Interestingly, HONB performs slightly better than HOS on this dataset. On the other hand SVM is significantly the best performing algorithm. We attribute the better performance of HONB and especially SVM to the skewed class distribution of the dataset. This the main difference of WebKB dataset from our other datasets.

Table 4. 3 Accuracy and standard deviations of algorithms on WebKB4 dataset

TS	MVNB	MVNB+JM	HOS	HONB	SVM
1	44.48±1.03	69.96±3.15	30.08±6.56	70.58±3.80	60.57±1.82
5	68.17±2.49	79.33±2.15	61.15±6.51	77.68±2.94	79.01±1.33
10	74.46±1.36	80.76±1.54	**77.71±2.33**	80.83±1.35	83.48±1.14
30	81.53±1.05	83.02±0.92	**85.24±0.75**	86.83±0.58	89.43±0.55
50	82.57±0.83	82.81±0.81	**86.08±0.55**	87.64±0.75	91.04±0.47
70	83.53±0.98	83.19±1.08	**87.01±0.87**	88.53±0.75	91.69±0.72
80	83.14±1.17	82.85±1.23	**86.47±1.25**	88.79±0.85	91.78±0.64
90	84.17±2.10	83.41±1.61	**87.01±1.20**	88.36±1.42	92.20±1.00

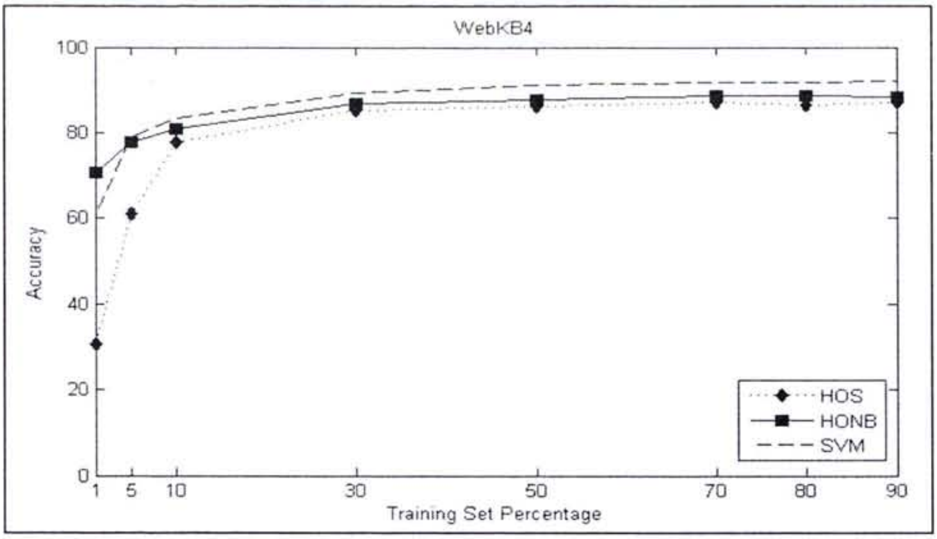

Figure 4. 2 Accuracy of HOS, HONB and SVM on WebKB4

The performance of HOS on 1150Haber dataset, which can be seen in Table 4.4 and Figure 4.3, is somewhat similar to 20 Newsgroups. HOS statistically significantly outperforms baseline MVNB starting from 10% and MVNB with JM smoothing from 30% level. HONB and HOS show a very similar performance on this dataset with the exception of small training set sizes (i.e. up to 30%) where HONB performs better. After 30% the differences between accuracies of HONB and HOS are not statistically significant. Please note that this dataset is our smallest dataset and there is relatively less data to infer from especially for HOS. Additionally, HOS statistically significantly outperform SVM starting from 10% level.

Table 4. 4 Accuracy and standard deviations of algorithms on 1150Haber dataset

TS	MVNB	MVNB+JM	HOS	HONB	SVM
1	35.70±7.64	48.40±5.04	32.09±11.1	30.32±12.7	38.92±3.03
5	65.06±12.6	81.01±6.95	**67.00±11.9**	88.25±0.93	67.47±4.24
10	72.95±3.83	86.01±2.03	**83.13±4.12**	91.61±0.85	76.27±2.71
30	87.64±1.14	91.49±0.71	**93.79±0.31**	94.20±0.59	87.39±1.21
50	88.73±0.65	91.10±0.63	**94.42±0.42**	94.73±0.57	89.55±1.12
70	89.97±0.88	91.39±0.83	**95.01±0.85**	95.30±0.96	90.55±1.49
80	89.70±2.40	90.83±2.50	**94.96±1.84**	95.91±1.60	91.91±2.39
90	90.78±2.73	91.48±2.42	**94.35±2.14**	95.22±1.75	90.78±1.93

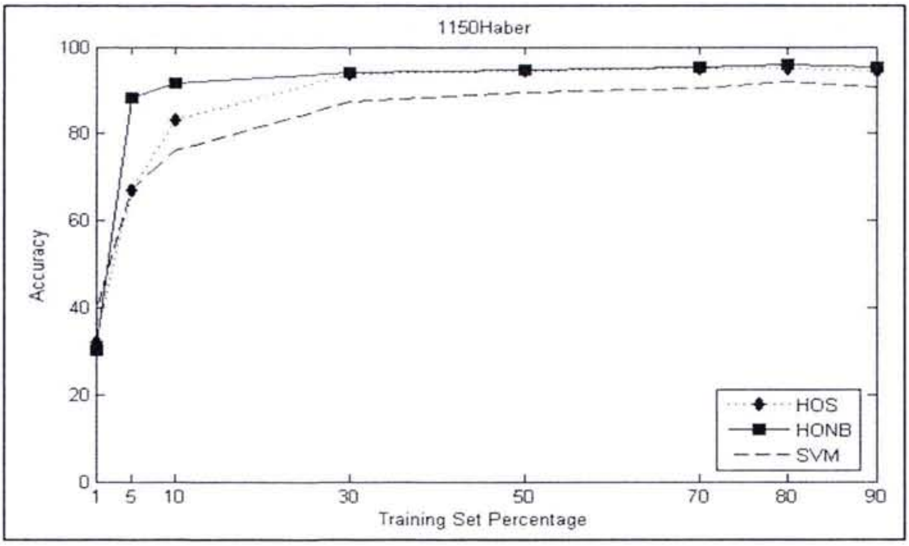

Figure 4. 3 Accuracy of HOS, HONB and SVM on 1150Haber

In order to quantify the level of the performance improvement over other algorithms, we define the following performance gain for the accuracy.

$$gain_{HOS} = \frac{p_{HOS} - p_X}{p_X} \tag{4.1}$$

where p_{HOS} is the Higher Order Smoothing algorithm's accuracy result and p_X stands for the result of the other algorithms (MVNB, MVNB+JM, HONB, or SVM). We present performance improvements of HOS over other algorithms on Table 4.5, 4.6 and Table 4.7. Improvements are most visible in 20 Newsgroups dataset. In Table 4.5, we can see that HOS improves upon MVNB and SVM about 17% at 10% training set size (ts) level on 20 Newsgroups dataset. We can observe improvements in all ts levels on this dataset. Table 4.6 and 4.7 shows the improvements on WebKB4 and 1150haber datasets respectively.

Table 4. 5 Performance improvement of HOS over other methods on 20 Newsgroups

TS	MVNB	MVNB+JM	HONB	SVM
10	17.98	5.14	9.68	17.73
30	12.55	8.34	7.69	7.87
50	10.58	8.38	5.77	4.68
70	10.15	8.49	4.27	2.21
80	9.57	8.54	3.85	1.85
90	9.25	8.84	3.93	1.07

Table 4. 6 Performance improvement of HOS over other methods on WebKB4 dataset

TS	MVNB	MVNB+JM	HONB	SVM
10	4.36	-3.78	-3.86	-6.91
30	4.55	2.67	-1.83	-4.69
50	4.25	3.95	-1.78	-5.45
70	4.17	4.59	-1.72	-5.10
80	4.01	4.37	-2.61	-5.79
90	3.37	4.32	-1.53	-5.63

Table 4. 7 Performance improvement of HOS over other methods on 1150Haber dataset

TS	MVNB	MVNB+JM	HONB	SVM
10	13.95	-3.35	-9.26	8.99
30	7.02	2.51	-0.44	7.32
50	6.41	3.64	-0.33	5.44
70	5.60	3.96	-0.30	4.93
80	5.86	4.55	-0.99	3.32
90	3.93	3.14	-0.91	3.93

We present the results of more evaluation metrics at the 80% training set level. This percentage is commonly used in random trial experiments (McCallum and Nigam, 1998; Rennie et.al, 2003). Table 4.8 shows F-measure (F1) performance of algorithms at 80% training set level. Similar trend can also be seen in here. HOS outperforms baseline MVNB for all the datasets. Table 4.9 presents AUC values of the algorithms in this training set percentage level. Again, HOS outperforms baseline MVNB for all the datasets. One interesting observation from this table is the results of algorithms on WebKB4 dataset. Although SVM is by far the best performing algorithm in this dataset in terms of accuracy, it has been outperformed by HOS in terms of AUC.

Table 4. 8 F-measure performance of algorithms at 80% training set level

ALGORITHM	**20NEWS-18828**	**WEBKB4**	**1150HABER**
HONB	79.96 ± 0.75	88.34 ± 0.97	95.92 ± 1.60
HOS	**83.02 ± 0.72**	**85.32 ± 1.74**	**94.95 ± 1.84**
MVNB	76.41 ± 0.59	82.80 ± 1.23	89.79 ± 2.40
MVNB+JM	77.39 ± 0.81	82.43 ± 1.31	90.96 ± 2.5
SVM	82.02 ± 0.47	90.81 ± 1.21	091.92 ± 2.39

Table 4. 9 AUC performance of algorithms at 80% training set level

ALGORITHM	**20NEWS-18828**	**WEBKB4**	**1150HABER**
HONB	98.18 ± 0.07	97.58 ± 0.27	99.57 ± 0.24
HOS	**98.57 ± 0.09**	**96.90 ± 0.46**	**99.56 ± 0.25**
MVNB	97.67 ± 0.17	96.17 ± 0.51	99.25 ± 0.38
MVNB+JM	97.74 ± 0.19	96.19 ± 0.54	99.43 ± 0.31
SVM	90.32 ± 0.25	93.41 ± 0.72	94.95 ± 1.50

4.2. Discussion

The use of higher-order paths for estimation of conditional term probabilities have been discussed in (Lytkin, 2009; Ganiz et.al, 2011). It is observed that highly discriminative terms exhibit much stronger influence on classification by HONB than by NB. Additionally, HONB tends to place more emphasis on the presence of terms in a document being classified (Ganiz et.al, 2011). Since HOS is based on higher-order paths, it enjoys some of these benefits. However, in HOS we are enumerating much fewer number of higher-order paths because paths need to end with a class label. Therefore, we have less data to extract patterns from. As a result, HONB has a higher performance on small training set sizes compare to HOS especially at 1% and 5% levels. Yet, HOS quickly catches up about at 10% level and outperforms HONB especially in 20 Newsgroups dataset. This dataset has relatively large number of classes (having 20 classes compare to six classes in WebKB4, and five classes in 1150Haber). With 20 class labels, we can extract much stronger patterns from higher-order paths using HOS. It would be interesting to observe the performance of HOS on real world datasets with much larger number of categories. Results on 1150haber dataset suggest that HONB and HOS may also perform well on different languages without additional tuning similar to LSI. This is an important advantage compare to the natural language processing based semantic methods such as (Zhou et.al, 2008).

In terms of training time complexity, an $O(n^2(m+n))$ algorithm is given in previous studies for obtaining counts of higher-order paths in a dataset with m instances and n dimensions (Lytkin, 2009; Ganiz et.al, 2009). In training phase, HONB forms a document by term matrix with dimensions (mxn) for each class, and use this algorithm to obtain counts of higher-order paths between all terms. Our approach which is given in Algorithm 1, suggests using the same principles; therefore, it has the same computational complexity. When we examine Algorithm 1 more closely, the computation of matrices O_1 and O_2 in steps 6 and 7, respectively, takes $O(mn^2+n^3)$ time. The loops in step 4 and 8, iterates over C_{lb} and O_1 matrices, respectively, and takes $O(n^2)$time. The computational complexity of

Algorithm 1 is dominated by computation matrices O_1 and O_2, therefore the complexity is $O(mn^2 + n^3)$

However, given the fact that we have based our computations on a single document by term matrix and we have used the paths ending only with class labels, we are enumerating much fewer numbers of paths. So in practice, HOS runs faster than HONB. Both HOS and HONB share the low classification time complexity of Naïve Bayes.

4.3. Future Work

It has been shown that LSI takes advantage of implicit higher-order (or latent) structure in the association of terms and documents. Higher-order relations in LSI capture "latent semantics" (Kontostathis and Pottenger, 2006). Motivated by this, a novel Bayesian framework for classification named Higher Order Naïve Bayes (HONB) is introduced (Ganiz et.al, 2009; Ganiz et.al, 2011). HONB can explicitly make use of these higher-order relations in the context of a class.

We present a novel semantic smoothing method named Higher Order Smoothing (HOS) for Naïve Bayes algorithm. HOS is built on a novel graph based data representation which allows us to exploit semantic information in higher-order paths. HOS exploits the relationships between instances of different classes in order to improve the parameter estimation in the face of sparse data. As a result, we do not only move beyond instance boundaries but also class boundaries to exploit the latent information in higher-order co-occurrence paths.

We have performed experiments on several benchmark datasets and compared HOS with several different classifiers. HOS significantly outperforms the baseline classifier (MVNB) in all datasets under different data conditions. Furthermore, it even outperforms SVM by a wide margin in the well-known 20 Newsgroups dataset. Our results demonstrate the value of HOS as a semantic smoothing algorithm.

As future work, we are planning to perform more detailed analysis in order to understand the reasons for the improved performance of HOS. Additionally, we would like to get insights about under which conditions and what type of datasets HOS performs well. We are also planning to advance the current higher-order learning framework, which works on binary data, so it can make use of term frequencies. Several studies emphasize the importance of different varieties of normalizations such as document-length normalization in improving Naïve Bayes performance (Rennie et.al, 2003; Eyheramendy et.al, 2003; Kolcz and Yih, 2007). Thus, we would like to analyze HOS by incorporating document length and weight normalization in the future.

REFERENCES

[1] Amasyalı, M.F., and Beken, A., (2009), "Türkçe Kelimelerin Anlamsal Benzerliklerinin Ölçülmesi ve Metin Sınıflandırmada Kullanılması", *IEEE 17th Signal Processing and Communications Applications Conference (SIU)*, April, Antalya, Turkey.

[2] Bengio, Y., Delalleau, O., and Le Roux, N., (2006), "The Curse of Highly Variable Functions for Local Kernel Machines", *In Advances in Neural Information Processing Systems*, vol. 18, pp. 107-114.

[3] Beyer, K., Goldstein, J., Ramakrishnan, R., and Shaft, U., (1998), "When is Nearest Neighbor Meaningful?", *In 7th International Conference on Database Theory*, vol. 1540, pp. 21-235.

[4] Chakrabarti, S., Dom, B., and Indyk, P., (1998), "Enhanced Hypertext Categorization using Hyperlinks", *Proc. International Conference on Management of Data (ACM SIGMOD)*, June, Seattle, Washington,, pp. 307-318.

[5] Chakrabarti, S., (2002), *Mining the Web: Discovering Knowledge from Hypertext Data*, Morgan Kaufmann Publishers, 2002, pp. 148-151.

[6] Chen, S.F., and Goodman, J., (1998), "An Emprical Study of Smoothing Techniques for Language Modeling", Technical Report, Harvard University Center for Research in Computing Technology, 1998.

[7] Deerwester, S., Dumais, S.T., and Harshman, R., (1990), "Indexing by Latent Semantic Analysis", *Journal of the American Society for Information Science*, vol. 41, pp. 391-407.

[8] Dhillon, I.S., (2001), "Co-clustering Documents and Words using Bipartite Spectral Graph Partitioning", *In Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* , August, San Francisco, California, USA, pp. 269-274.

[9] Edwards, A., and Pottenger, W.M., (2011), "Higher Order Q-Learning", *Proc. Adaptive Dynamic Programming and Reinforcement Learning (ADPRL)*, vol. 1, pp.128-134.

[10] Eyheramendy, S., Lewis, D.D, and Madigan, D., (2003), "On the Naïve Bayes Model for Text Categorization", 9th *International Workshop on Artificial Intelligence and Statistics (AISTATS)*, January, Florida, USA, pp. 332-339.

[11] Ganiz, M., Pottenger, W.M., Kanitkar, S., and Chuah, M.C., (2006), "Detection of Interdomain Routing Anomalies Based on Higher-Order Path Analysis", *Proc. IEEE International Conference on Data Mining (ICDM)*, December, Hong Kong, China, pp. 874-879.

[12] Ganiz, M., Lytkin, N., and Pottenger, W.M., (2009), "Leveraging Higher Order Dependencies between Features for Text Classification", *Proc. European Conference on Machine Learning and Pronciples and Practice of Knowledge Discovery in Databases (ECML/PKDD)*, September, Bled, Slovenia, pp. 375-390.

[13] Ganiz, M., George, C., and Pottenger, W.M., (2011), "Higher Order Naïve Bayes: A Novel Non-IID Approach to Text Classification", *IEEE Transactions on Knowledge and Data Engineering*, vol. 23, pp. 1022-1034.

[14] Gao, B., Liu, T., Feng, G., Qin, T., Cheng, Q., and Ma, W., (2005), "Hierarchical Taxonomy Preparation for Text Categorization using Consistent Bipartite Spectral Graph Co-partitioning", *IEEE Transactions on Knowledge and Data Engineering*, vol.17, no. 9, pp. 1263-1273.

[15] Gao, B., Liu, T.Y., Zheng, X., Cheng, Q.S., and Ma, W.Y., (2005), "Consistent Bipartite Graph Co-partitioning for Star-structured High-order Heterogeneous Data Co-clustering", *In Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, August, Chicago, USA, pp.41–50.

[16] Getoor, L., and Diehl, C.P., (2005), "Link Mining: A Survey", *Proc. International Conference on Knowledge Discovery and Data Mining (ACM SIGKDD)*, August, Chicago, USA, pp. 3-12.

[17] Gharahmani, Z., and Lafferty, J., (2003), "Semi-supervised Learning using Gaussian Felds and Harmonic Functions", *Proceedings of the 20th International Conference on Machine Learning*, August, Washington, USA, pp. 912–919.

[18] Hussain, S.F., and Bisson, G., (2010), "Text Categorization using Word Similarities Based on Higher Order Co-occurrences", *Proceedings of the SIAM International Conference on Data Mining (SDM),* April 29 - May 1, Columbus, Ohio, USA.

[19] Joachims, T., (1998), "Text Categorization with Support Vector Machines: Learning with Many Relevant Features", *10th European Conference on Machine Learning (ECML)*, April, Chemnitz, Germany, pp.137-142.

[20] Juan, A., and Ney, H., (2002), "Reversing and Smoothing the Multinomial Naïve Bayes Text Classifier", *In Proceedings of the 2nd International Workshop on Pattern Recognition in Information Systems (PRIS)*, April, Alacant, Spain, pp. 200-212.

[21] Kim, S.B., Han, K.S., Rim, H.C., and Myaeng, S.H., (2006), "Some Effective Techniques for Naïve Bayes Text Classification", *IEEE Transactions on Knowledge and Data Engineering*, vol.18, pp. 1457–1466.

[22] Kolcz, A., and Yih, W., (2007), "Raising the Baseline for High-Precision Text Classifiers", *In Proceedings of the 13th International Conference on Knowledge Discovery and Data Mining (ACM SIGKDD)*, August, California, USA, pp. 400-409.

[23] Kontostathis, A., and Pottenger, W.M., (2006), "A Framework for Understanding LSI Performance", *Journal of the Information Processing and Management*, vol. 42, pp. 56-73.

[24] Lemaire, B., and Denhière, G., (2006), "Effects of High-order Co-occurrences on Word Semantic Similarities", *Current Psychology Letters# Behaviour, Brain and Cognition*, vol. 18, pp. 1.

[25] Li, S., Wu, T., and Pottenger, W.M., (2005), "Distributed Higher Order Association Rule Mining using Information Extracted from Textual Data", *In SIGKDD Explorations Newsletter*, vol. 7, pp. 26-35.

[26] Lin, F., and Cohen, W.W., (2010), "A Very Fast Method for Clustering Big Text Datasets", *In ECAI*, August, Lisbon, Portugal, pp. 303-308.

[27] Lu, C., Hu, X., and Park, J.R., (2011), "Exploiting the Social Tagging Network for Web Clustering", *In Proc. IEEE Transactions on Systems, Man and Cybernetics*, vol.41, no. 5.

[28] Lytkin, N., (2009), *Variance-based Clustering Methods and Higher Order Data Transformations and Their Applications*, Ph.D. Thesis, Rutgers University, NJ, U.S.A.

[29] McCallum, A., and Nigam, K., (1998), "A Comparison of Event Models for Naïve Bayes Text Classification", *Proc. AAAI 1998 Workshop on Learning for Text Categorization*, July, Winconsin, USA, pp. 41-48.

[30] McCallum, A., and Nigam, K., (1999), "Text Classification by Bootstrapping with Keywords, EM and Shrinkage", *In Working Notes of ACL 1999 Workshop for the Unsupervised Learning in Natural Language Processing (ACL)*, June, Maryland, USA, pp. 52-58.

[31] Mengle, S., and Goharian, N., (2010), "Detecting Relationships among Categories using Text Classification", *JASIST* , vol. 61, no. 5, pp. 1046-1061.

[32] Metsis, V., Androutsopoulos, I., and Paliouras, G., (2006), "Spam Filtering with Naïve Bayes – Which Naïve Bayes?", *3rd Conference on Email and Anti-Spam (CEAS)*, July, California, USA.

[33] Neville, J., and Jensen, D., (2000), "Iterative Classification in Relational Data", *Proc. AAAI 2000 Workshop on Learning Statististical Models from Relational Data (LSR)*, July, Texas, USA, pp. 13-20.

[34] Peng, F., Schuurmans, D., and Wang, S., (2004), "Augmenting Naïve Bayes Classifiers with Statistical Language Models", *Information Retrieval*, vol. 7, pp. 317–345.

[35] Poyraz, M., Kilimci, Z.H., and Ganiz, M.C., (2012), "A Novel Semantic Smoothing Method Based on Higher Order Paths for Text Classification", *In Data Mining (ICDM)*, December, Brussels, Belgium, pp. 615-624.

[36] Radev, D., (2004), "Weakly Supervised Graph Based Methods for Classification", Technical Report CSE-TR-500-04, University of Michigan.

[37] Rennie, J.D.M., Shih, L., Teevan, J., Karger, D.R., (2003), "Tackling the Poor Assumptions of Naïve Bayes Text Classifiers", *20th International Conference on Machine Learning*, August, Washington, USA, pp. 616-623.

[38] Salton, G., Wong, A., and Yang, C.S., (1975), "A Vector Space Model for Information Retrieval", *Journal of the American Society for Information Science*, vol. 18, pp. 613-620.

[39] Schneider, K.M., (2004), "On Word Frequency Information and Negative Evidence in Naïve Bayes Text Classification", *4th International Conference on Advances in Natural Language Processing (EsTAL)*, October, Alacant, Spain, pp. 474–485.

[40] Taskar, B., Abbeel, P., and Koller, D., (2002), "Discriminative Probabilistic Models for Relational Data", *Proc. Uncertainty in Artificial Intelligence (UAI'02)*, August, Alberta, Canada, pp. 485-492.

[41] Torunoğlu, D., Çakırman, E., Ganiz, M.C, Akyokuş, S., Gürbüz, M.Z, (2011), "Analysis of Preprocessing Methods on Classification of Turkish Texts", *Proc. International Symposium on Innovations in Intelligent Systems and Applications (INISTA)*, June, Istanbul, Turkey, pp. 112-118.

[42] Verleysen, M. and François, D., (2005), "The Curse of Dimensionality in Data Mining and Time Series Prediction", *Proceedings of the 8th International Conference on Artificial Neural Networks*, vol. 3512, pp. 758-770.

[43] Zha, H., Ding, C., and Gu, M., (2001), "Bipartite Graph Partitioning and Data Clustering", *In proceedings of CIKM'01*, November, Atlanta, Georgia, USA, pp. 25-32.

[44] Zhou, X., Zhang, X., and Hu, X., (2008), "Semantic Smoothing for Bayesian Text Classification with Small Training Data", *Proc. International Conference on Data Mining (SIAM)*, April, Atlanta, Georgia, USA, pp. 289–300.

CV

Place and Date of Birth		Konya - 12.06.1986
High School	2000-2004	İzmir Atatürk Anatolian High School
B.S Degree	2004-2009	İzmir Institute of Technology Computer Engineering Department
M.S Degree	2009-	Doğuş University Computer Engineering Department

Work Experiences:

	2009-	Doğuş University Research Assistant

Publications:

Poyraz, M., Kilimci, Z.H., Ganiz, M.C., (2012), "A Novel Semantic Smoothing Method based on Higher Order Paths for Text Classification", *IEEE International Conference on Data Mining (ICDM 2012)*, December 10-13, 2012, Brussels, Belgium

Poyraz, M., Ganiz, M.C., Akyokus, S., Gorener, B., Kilimci, Z.H., (2012), "Exploiting Turkish Wikipedia as a Semantic Resource for Text Classification", *INISTA 2012*, July 2-4, 2012, Trabzon, Turkey.

Taylan, D., Poyraz, M., Akyokus, S., Ganiz, M.C., (2011), "Intelligent Focused Crawler: Learning which Links to Crawl", *INISTA 2011*, June 15-18, 2011, Istanbul, Turkey.

Gökçen, G., Yaman,M.Y., Akın,S., Aytas, B., Poyraz, M., Kala, M.E., Toksoy, M., (2009), "Konutlarda Enerji Performansı Standard Değerlendirme Metodu, (KEP-SDM) İçin Geliştirilen Enerji Sertifikalandırma Yazılımı (KEP-İYTE-ESS)", *IX. Ulusal Tesisat Mühendisliği Kongresi*, 6-9 May, İzmir, Turkey.

Gökçen, G., Yaman,M.Y., Akın,S., Aytas, B., Poyraz, M., Kala, M.E., Toksoy, M., (2009), "Konutlarda Enerji Performansı Standard Değerlendirme Metodu, (KEP-SDM) İçin Geliştirilen Enerji Sertifikalandırma Yazılımı (KEP-İYTE-ESS)", *Journal of DTK (Dogalgaz-Tesisat-Klima)*, November 2009, vol.13, no. 151, pp. 113-123.