VARIANCE-BASED CLUSTERING METHODS AND HIGHER ORDER DATA TRANSFORMATIONS AND THEIR APPLICATIONS

BY NIKITA I. LYTKIN

A dissertation submitted to the

Graduate School—New Brunswick

Rutgers, The State University of New Jersey

in partial fulfillment of the requirements

for the degree of

Doctor of Philosophy

Graduate Program in Computer Science

Written under the direction of

Casimir A. Kulikowski

and approved by

_____

_____

_____

_____

New Brunswick, New Jersey

October, 2009

ABSTRACT OF THE DISSERTATION

Variance-based Clustering Methods and Higher Order Data Transformations and Their Applications

by Nikita I. Lytkin

Dissertation Director: Casimir A. Kulikowski

Two approaches have been proposed in statistical and machine learning communities in order to address the problem of uncovering clusters with complex structure. One approach relies on the development of clustering criteria that are able to accommodate increasingly complex characteristics of the data. The other approach is based on simplification of structure of data by mapping it to a different feature space via a non-linear function and then clustering in the new space.

This dissertation covers three related studies: development of a novel multi-dimensional clustering method, development of non-linear mapping functions that leverage higher-order co-occurrences between features in boolean data, and applications of these mapping functions for improving the performance of clustering methods. In particular, we treat clustering as a combinatorial optimization problem of finding a partition of the data so as to minimize a certain criterion. We develop a novel multi-dimensional clustering method based on a statistically-motivated criterion proposed by J. Neyman for stratified sampling from one-dimensional data. We show that this criterion is more reflective of the underlying data structure than the seemingly similar K-means criterion

when second order variability is not homogeneous between constituent subgroups. Furthermore, experimental results demonstrate that generalization of the Neyman's criterion to multi-dimensional spaces and development of the associated clustering algorithm allow for statistically efficient estimation of the grand mean vector of a population.

In the framework of the mapping-based approach to discovering complex cluster structures, we introduced a novel adaptive non-linear data transformation termed Unsupervised Second Order Transformation (USOT). The novelties behind USOT are (a) that it leverages in a unsupervised manner, higher-order co-occurrences between features in boolean data, and (b) that it considers each feature in the context of probabilistic relationships with other features. In addition, USOT has two desirable properties. USOT adaptively selects features that would influence the mapping of a given feature, and preserves the interpretability of dimensions of the transformed space. Experimental results on text corpora and financial time series demonstrate that by leveraging higher-order co-occurrences between features, clustering methods achieved statistically significant improvements in USOT space over the original boolean space.

Acknowledgements

I thank professors Ilya Muchnik, William M. Pottenger and Casimir Kulikowski for their wisdom, mentorship, support and encouragement during my Ph.D. studies. The past several years have been very formative and life changing. I greatly appreciate all your efforts and help, without which a successful completion of this dissertation would not have been possible. I also wish to thank professor Eugene Bauman for insights and discussions of the polynomial clustering framework, and professor Michael Pazzani for initial support by a graduate research assistantship.

I consider myself to have been blessed with being born into an amazing family with a wise, caring and compassionate mother Kamilla, father Igor, sister Elena, grandmother Valentina and aunt Irana. Your love, encouragement, advice and humor accompanied by an occasional push continue to help me stay afloat in the sometimes turbulent waters of everyday existence.

I would also like to express deep gratitude to my dear Ayelet for cheering me up in the crankier moments of the graduate school experience and most importantly, for being loving, supportive, and an inspiring influence in my life.

Coming from abroad and entering a completely different social culture is always challenging. The presence of generous and understanding people on such a journey is invaluable. I have been extremely fortunate to have met Belida Han Uckun and the Kaczmarek family – Irene, Andrew, Rafal and Tomasz – who quickly became a central part of my social circle in the United States and whose kindness and friendship made the adjustment to the new environment much more pleasant.

Dedication

To the one light that shines through all.

Table of Contents

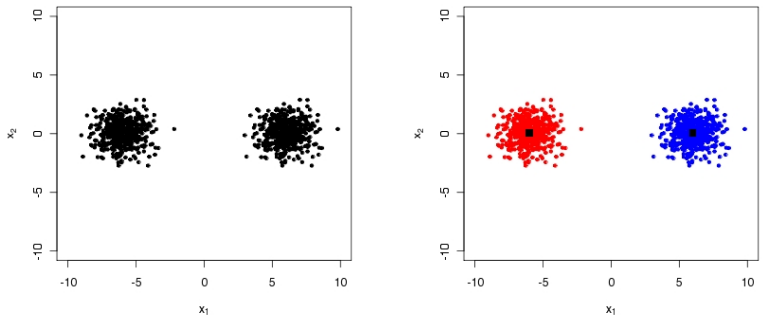
Chapter 1

Introduction

Cluster analysis is a subarea of machine learning that studies methods of unsupervised discovery of homogeneous subsets of data instances from heterogeneous datasets. Given a heterogeneous set of objects (e.g. time series of returns of mutual funds, or text documents covering various topics), the objects are automatically clustered such that objects within a cluster are very similar while objects from different clusters are highly dissimilar. Methods of cluster analysis have been successfully applied in a wide spectrum of areas of science and engineering including biology [11, 19, 65], physics [47], finance [52, 58], image analysis [8, 15, 27, 42, 53, 68], information retrieval and text mining [3, 6, 59], and cybersecurity [13].

The multitude of methods of cluster analysis [14, 16, 22, 23, 29, 31, 32, 34, 49, 54, 61, 62] developed to date can be divided into two broad categories: heuristic methods and formal methods based on mathematical formulations of clustering as an optimization problem. An example of a heuristic method is clustering by identifying connected components in a graph that somehow represents a given dataset. Unlike the heuristic approach to clustering, a mathematical formulation allows for systematic study of existing clustering methods and for development of novel approaches based on established theoretical results. In this work, we follow the formal approach and consider clustering as an optimization problem. In this view, a clustering method is comprised of a criterion (an objective function), which measures the quality of a clustering, and an algorithm for optimization of the criterion. A well-known member of this category of clustering methods is K-means. K-means criterion is minimized by clusterings comprised of tight

groups of points with each group centered around one of K points characterizing a qualitatively different subgroup of data objects. Figure 1.1a demonstrates an example of such cluster structure comprised of two rounded clouds of points on a two-dimensional plane. As its name suggests, K-means partitions the data solely on the basis of locations of the cluster means. A point is associated with that cluster to whose mean it is closest as measured by the squared Euclidean distance. As we show in Appendix C, this is equivalent to separating each cluster from all others by a hyperplane whose norm is determined only by the cluster's mean vector.

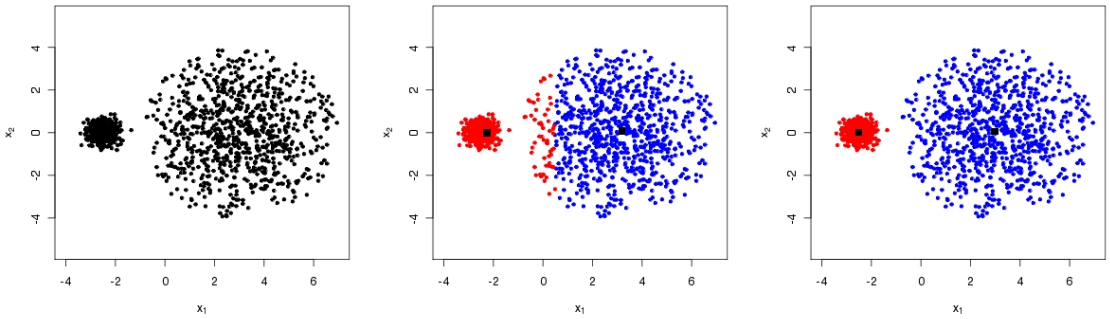


(a) Input dataset (b) Clustering by K-means

Figure 1.1: A simple dataset and its clustering by K-means. Cluster means are indicated by squares.

Often in practice, however, data clusters have more complex structure than shown in Figure 1.1. Increased complexity of the data comes in the form requiring that a clustering method takes into account additional characteristics of the clusters. Consider Figure 1.2 for instance. While the locations of cluster means are certainly an important feature for adequately partitioning the data in Figure 1.2a, inherent inability of K-means to account for cluster variances results in a poor clustering shown in Figure 1.2b. Hence, a clustering criterion that in addition to cluster means also takes into account cluster variances would be more appropriate for producing the desirable clustering shown in Figure 1.2c.

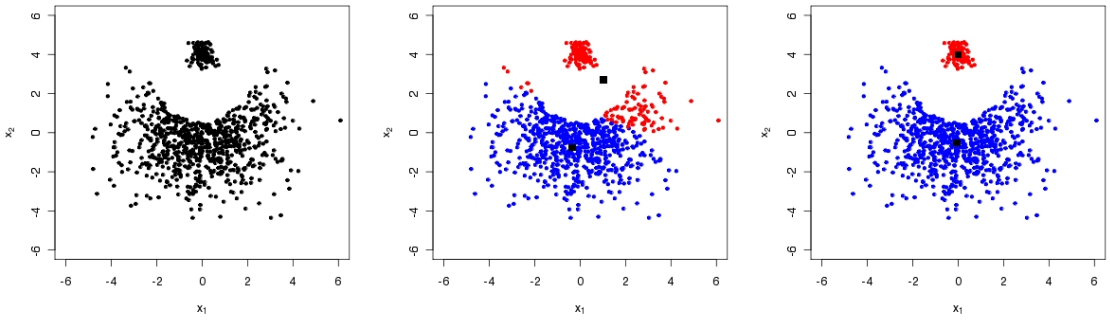
A common feature of the datasets in Figures 1.1 and 1.2 is that both clusters are



(a) Input dataset (b) Clustering by K-means (c) Ideal clustering

Figure 1.2: K-means' inherent inability to account for cluster variances results in a poor clustering

separable by a linear discriminant boundary. However, a typical real-world data often exhibits more complex, non-linear cluster structure as exemplified by Figure 1.3. Applying K-means on this data produces an unsatisfactory clustering shown in Figure 1.3b. In this case, a more flexible clustering criterion that is able to accommodate varying cluster scatter as well as non-linearity within the data would have better chances of discovering the more intuitive clustering shown in Figure 1.3c.



(a) Input dataset (b) Clustering by K-means (c) Ideal clustering

Figure 1.3: A more complex dataset exhibiting a non-linear cluster structure

Two approaches have been proposed in statistical and machine learning communities in order to address the problem of uncovering complex clusters. One approach relies on the development of clustering criteria that are able to accommodate increasingly

complex characteristics of the data. The other approach is based on the simplification of structure of data by mapping it to a different feature space via a non-linear function and then clustering in the new space. It is hoped that such mapping will increase separability between the "true" clusters thus making them more obvious for discovery by simple clustering criteria. However, since different datasets may exhibit drastically different internal structure, the mapping function applied must be adaptive to the data. In order to get a better understanding of what makes each cluster distinct from others, clusters are often analyze as to how well do they capture specifics of individual or groups of features. For example, do values of a given feature vary equally within clusters, or does this feature exhibit different behavior in different clusters and what domain knowledge can be inferred from that? To be able to answer such questions, it is important that dimensions of the new feature space into which the data is mapped, maintain their interpretability in terms of the original features.

In this work we make contributions to both of these approaches. In Section 2.1, we develop first multi-dimensional clustering algorithm for a criterion that was proposed by Neyman in [51] for stratified sampling from one-dimensional data, but has never before been applied for clustering in multi-dimensional spaces. We then show that this criterion is more reflective of the underlying data structure than the seemingly similar K-means criterion when second order variability is not homogeneous between constituent subgroups. Neyman's criterion takes into account cluster means and variances, and, in general, produces non-linear cluster boundaries. We also discover that K-means and Neyman's criteria produce identical clusterings when cluster variances are equal.

Then, in Section 2.2, we introduce a novel adaptive non-linear data transformation termed Unsupervised Second Order Transformation (USOT). USOT maps data from a boolean[1] space to a real space thereby emphasizing specifics of the various

[1]Many methods of mapping real-valued data to boolean spaces exist, but their development is beyond the scope of this dissertation. We did, however, use some of these methods in the experiments in Section 3.5.

homogeneous subgroups of data instances. USOT leverages probabilistic dependencies estimated based on indirect co-occurrences between features in the dataset. In our work [26] on supervised learning, we found these links, termed higher-order paths, to be an abundant source of extremely valuable information that allowed higher-order classifiers to consistently outperform the traditional methods. The novelties behind USOT are (a) that it leverages in a unsupervised manner, higher-order co-occurrences between features, and (b) that it considers each feature in the context of probabilistic relationships with other features. USOT has two desirable properties. USOT adaptively selects features that would influence the mapping of a given feature. If a feature j exhibits the same distribution regardless of the value of a feature i, then feature j will have no effect on mapping feature i. Moreover, interpretability of dimensions of the USOT space is retained due to one-to-one correspondence with the original boolean features.

The intuition behind USOT originated from our work on higher-order classifiers [26], and in particular from the Supervised Second Order Transformation (SSOT) described in Section 2.2.3. SSOT is a novel data transformation that requires the knowledge of true class labels of the instances comprising a training set. Both USOT and SSOT are defined over the space of higher-order paths. However, aside from SSOT being a supervised transformation, the main difference between USOT and SSOT lies in the way the two mappings use the higher-order paths. While USOT considers probabilistic dependencies between a feature and all other features, SSOT makes use of probabilistic dependencies between a class indicator variable and the features.

In Section 2.2.5, we develop a $O\left((m+n)n^2\right)$ time algorithm for obtaining the counts of higher-order paths used by USOT and SSOT. This algorithm improves over the $O\left(m^2n^3\right)$ complexity of a straight-forward path counting algorithm also given in Section 2.2.5.

Overall, this dissertation covers three related studies: development of a novel multi-dimensional clustering method based on the Neyman's criterion, development of non-linear mapping functions that leverage higher-order co-occurrences between features in boolean data, and applications of these mapping functions for improving the performance of clustering methods. In Section 2.1, we develop a novel multi-dimensional clustering method based on the Neyman's criterion. We discuss the related work on clustering criteria in Section 2.1.4. Since criteria discussed in this work are functions of cluster variances, we refer to methods based on these criteria as Variance-Based Clustering. In Section 2.2, we describe the proposed adaptive non-linear data transformations USOT and SSOT. Related work is discussed in Section 2.2.6. Evaluation of the proposed clustering method on simulated data is presented in Section 3.1. In Section 3.2, we present experimental results on estimation of the mean vector by stratified sampling in multi-dimensional spaces. In Section 3.4, we present an approach to unsupervised text categorization by applying the proposed methods. In Section 3.5, we carry out a Return-Based Style Analysis of approximately 7,000 mutual funds. Chapter 4 concludes this work and outlines further research directions.

Chapter 2

Methodology

2.1 Variance-Based Clustering

In this work, we treat clustering as a combinatorial optimization problem of minimizing a certain objective function by partitioning a set of points in a n-dimensional Euclidean into a pre-specified number of disjoint clusters. In particular, we consider two clustering criteria

$$I_1 = \sum_{\alpha=1}^{K} p_\alpha \sigma_\alpha^2, \tag{2.1}$$

and

$$I_2 = \sum_{\alpha=1}^{K} p_\alpha \sigma_\alpha, \tag{2.2}$$

where K is the number of clusters sought, p_α denotes the probability, or relative weight, of cluster $\alpha = 1, \ldots, K$, and σ_α^2 denotes its variance. Intuitively and mathematically, criteria (2.1) and (2.2) seem very similar. In fact, both of these criteria are minimized by clusterings comprised of congregations of points tightly centered around the cluster means. Criterion (2.1) is the well-known and studied objective function of the K-means method. Clusterings minimizing (2.1) are Voronoi diagrams constructed on the basis of the given dataset. An efficient minimization algorithm for criterion (2.1) was given by [44]. It should be noted that in one dimension, globally optimal clusterings for criteria (2.1) and (2.2) can be obtained by a dynamic programming approach [7, 8]. Unfortunately, in higher dimensions the problem becomes NP-hard and one resorts to considering locally optimal solutions such as produced by the K-means algorithm [44].

Criterion (2.2) was proposed in [51] for stratified sampling from one-dimensional

data. However, no algorithm for minimization of (2.2) in spaces of dimensionality higher than one was given to date. Moreover, the behavior of criterion (2.2) in multi-dimensional spaces has not been studied in the literature, perhaps due to the lack of a minimization algorithm for this criterion.

The main contributions of this section are the following. We develop an efficient clustering algorithm for criterion (2.2) in high-dimensional spaces. In order to do that, we prove that (2.2) is a strictly concave function of cluster moments, and then show how the framework of [5] can be applied for minimization of this criterion. We then study the behavior of criterion (2.2) in multi-dimensional spaces. We show that criterion (2.2) is more complex than (2.1) and, in general, produces non-linear cluster boundaries. We also identify another major difference between these clustering criteria. While (2.1) takes into account only the locations of cluster means[1], criterion (2.2) also considers the cluster variances. Moreover, we uncover a condition under which criteria (2.1) and (2.2) produce identical clusterings.

In Section 2.1.1, we introduce the necessary foundational concepts used for the development of clustering algorithms in Section 2.1.2. In Section 2.1.3, we derive the cluster membership functions for criterion (2.2) and provide an analytical comparison with criterion (2.1). The related work is discussed in Section 2.1.4.

2.1.1 Mathematical Foundations: Criteria of Optimality for Clustering

Let $\mathcal{X}$ denote an n-dimensional Euclidean space where each distinct data instance is uniquely characterized by a vector $x \in \mathcal{X}$. A clustering H is a partition of space $\mathcal{X}$ into K disjoint regions, and is determined by a set of characteristic functions $H =$

[1]In Appendix C, we rewrite criterion (2.1) to show that it effectively only depends on the cluster means and not the variances.

$(h_1(x), \ldots, h_K(x))$, where

$$h_\alpha(x) = \begin{cases} 1, & \text{if } x \text{ belongs to cluster } \alpha \\ 0, & \text{otherwise.} \end{cases}$$

We denote by $\mathcal{H}$ the set of all possible clusterings into K non-empty clusters. The key role in this work is played by a specific type of clusterings where cluster boundaries in space $\mathcal{X}$ are specified by smooth[2] functions. First, we illustrate the relationship between a clustering criterion and cluster boundaries for the case of two clusters ($K = 2$), and then provide a generalization to an arbitrary number of clusters $K \geq 2$.

A clustering $H = (h_1(x), h_2(x))$ is specified by a *discriminant function* $F(x)$ as follows[3]:

$$h_1(x) = \begin{cases} 1, & \text{if } F(x) \geq 0 \\ 0, & \text{otherwise,} \end{cases} \quad \text{and} \quad h_2(x) = \begin{cases} 1, & \text{if } F(x) < 0 \\ 0, & \text{otherwise.} \end{cases} \tag{2.3}$$

The boundary between clusters is the *discriminant surface* $F(x) = 0$. Further, we only consider clusterings into two clusters such that certain criteria (functionals) achieve extremal values on these clusterings.

Functionals considered in this work are differentiable functions of the non-normalized cluster moments, of order up to r, of the probability distribution function $P(x)$. The non-normalized cluster moments of the l-th order ($l = 0, \ldots, r$) are defined as

$$M_1^{(l)} = \int_{\mathcal{X}} x^l h_1(x) dP(x) \quad \text{and} \quad M_2^{(l)} = \int_{\mathcal{X}} x^l h_2(x) dP(x),$$

where x^l denotes the scalar $\|x\|^l$ when l is even and the vector $x\|x\|^{l-1}$ when l is odd. The distribution $P(x)$ need not be known and no assumptions are made regarding its type. It is only assumed that the probability density function $\Pr(x)$ of occurrence of points $x \in \mathcal{X}$ exists, is continuous and is concentrated in a compact set R of space $\mathcal{X}$, i.e., $\Pr(x) = 0, \ \forall x \notin R$.

---

[2]As will be shown later, cluster boundaries discussed in this work are specified by polynomial functions.

[3]To avoid ambiguity, points of the discriminant surface $F(x) = 0$ are always assigned to cluster 1.

Below, we state a theorem published in [2] that given a functional of a general form, characterizes the corresponding smooth discriminant functions.

Theorem 1. *Let the quality of a clustering $H \in \mathcal{H}$ be measured by a functional of the form*

$$I\left(M_1^{(0)}, M_1^{(1)}, \ldots, M_1^{(r)}, M_2^{(0)}, M_2^{(1)}, \ldots, M_2^{(r)}\right), \tag{2.4}$$

where I is a differentiable function of the non-normalized cluster moments of order up to and including r, and the probability density $\Pr(x)$ is a continuous function that is zero outside a compact set R of space $\mathcal{X}$. Then:

1. *if functional (2.4) achieves an extremum on some discriminant function, the same extremum is achieved on a polynomial discriminant function of degree r defined as:*

$$F(x) = f_2(x) - f_1(x) = \sum_{l=0}^{r}\left(c_2^{(l)}, x^l\right) - \sum_{l=0}^{r}\left(c_1^{(l)}, x^l\right) = \sum_{l=0}^{r}\left(c_2^{(l)} - c_1^{(l)}, x^l\right), \tag{2.5}$$

where

$$c_1^{(l)} = \frac{\partial I}{\partial M_1^{(l)}} \quad \text{and} \quad c_2^{(l)} = \frac{\partial I}{\partial M_2^{(l)}} \tag{2.6}$$

2. *the discriminant function defined by (2.5) and (2.6) endows functional (2.4) with a stationary value.*

In Theorem 1, $c_\alpha^{(l)}$ denote scalars when l is even and vectors with coordinates $\frac{\partial I}{\partial M_{\alpha,i}^{(l)}}$ when l is odd, where $\alpha \in \{1, 2\}$ is the cluster index and $M_{\alpha,i}^{(l)}$ is the i-th component of the vector $M_\alpha^{(l)}$; $\left(c_\alpha^{(l)}, x^l\right)$ denotes multiplication of scalars $c_\alpha^{(l)}$ and $\|x\|^l$ when l is even and the scalar product of vectors $c_\alpha^{(l)}$ and $x\|x\|^{l-1}$ when l is odd.

We note that Theorem 1 is concerned with partitions of the compact set R of space $\mathcal{X}$ rather than of the entire space $\mathcal{X}$. We also note that a functional of the form (2.4) can be constructed such that clusterings minimizing it are of interest. In this case, *polynomial membership functions*

$$f_1(x) = \sum_{l=0}^{r}\left(c_1^{(l)}, x^l\right) \quad \text{and} \quad f_2(x) = \sum_{l=0}^{r}\left(c_2^{(l)}, x^l\right), \tag{2.7}$$

are regarded as measures of distance between a point and a cluster. On the other hand, a functional of the form (2.4) can be constructed such that clusterings maximizing it are sought. Under this condition, membership functions (2.7) are regarded as measures of affinity between a point and a cluster. The corresponding discriminant function (2.5), in this case, has to be taken with a negative sign in definition (2.3) of characteristic functions.

We now consider a more general problem of finding a clustering minimizing a functional of the form

$$I\left(M_1^{(0)}, M_1^{(1)}, \ldots, M_1^{(r)}, \ldots, M_K^{(0)}, M_K^{(1)}, \ldots, M_K^{(r)}\right), \tag{2.8}$$

where

$$M_\alpha^{(l)} = \int_{\mathcal{X}} x^l h_\alpha(x) dP(x),\ \alpha = 1, \ldots, K, \tag{2.9}$$

denotes the l-th ($l = 0, \ldots, r$) non-normalized moment of cluster α.

Let $c = \left(c_1^{(0)}, c_1^{(1)}, \ldots, c_1^{(r)}, \ldots, c_K^{(0)}, c_K^{(1)}, \ldots, c_K^{(r)}\right)$ denote a vector of coefficients, where $c_\alpha^{(l)}$ denote scalars when l is even and n-dimensional vectors when l is odd. Vector c specifies polynomial membership functions $f_1(x), f_2(x), \ldots, f_K(x)$, where

$$f_\alpha(x) = \sum_{l=0}^{r} \left(c_\alpha^{(l)}, x^l\right). \tag{2.10}$$

For a given vector c, the *polynomial clustering* $H^c = (h_1^c(x), \ldots, h_K^c(x))$ is specified via membership functions (2.10) as follows:

$$h_\alpha^c(x) = \begin{cases} 1, & \text{if } f_\alpha(x) = \min\limits_{i=1,\ldots,K} f_i(x),\ \alpha = \min\limits_{i=1,\ldots,K} \{i : f_i(x) = f_\alpha(x)\} \\ 0, & \text{otherwise.} \end{cases} \tag{2.11}$$

For convenience, let $\mu(H) = \left(M_1^{(0)}, M_1^{(1)}, \ldots, M_1^{(r)}, \ldots, M_K^{(0)}, M_K^{(1)}, \ldots, M_K^{(r)}\right)$ denote the vector of the non-normalized cluster moments under a clustering $H \in \mathcal{H}$. Functional (2.8) can then be rewritten as

$$I = I\left(\mu(H)\right). \tag{2.12}$$

Two clusterings H and H^* are *equivalent* if $\mu(H) = \mu(H^*)$. A generalization of Theorem 1 to clusterings into an arbitrary number of clusters ($K \geq 2$), follows.

Theorem 2. *Let $I(\mu(H))$ be a strictly concave functional that attains a local minimum on a clustering H^*. Then a polynomial clustering H^c, equivalent to H^*, exists for which the vector $c = \left(c_1^{(0)}, c_1^{(1)}, \ldots, c_1^{(r)}, \ldots, c_K^{(0)}, c_K^{(1)}, \ldots, c_K^{(r)}\right)$ of coefficients is determined as a supergradient[4] of the functional $I(\mu(H))$ at the point $\mu(H^*)$.*

The proof of Theorem 2 is provided in Appendix B, and rests on the following lemma, which is proved in Appendix A.

Lemma 1. For an arbitrary vector c and an arbitrary clustering $H \in \mathcal{H}$, the following inequality holds:

$$(c, \mu(H^c) - \mu(H)) \leq 0.$$

In Appendix B we also show that set $Z = \{\mu(H) : H \in \mathcal{H}\}$ of vectors of the non-normalized cluster moments of all possible clusterings $H \in \mathcal{H}$ is bounded, closed and convex. Therefore, all local minima of a strictly concave functional (2.12) are attained on the boundary points of set Z. Lemma 1 states that polynomial clusterings correspond to the boundary points of set Z. Theorem 2 specifies the form of the polynomial clusterings minimizing a strictly concave functional (2.12). A variant of Theorem 2 for the case of maximization of a convex functional $I\left(M_1^{(0)}, M_1^{(1)}, \ldots, M_K^{(0)}, M_K^{(1)}\right)$ was first published in [5].

2.1.2 Algorithms of Search for Extrema of Clustering Criteria

In this section, we present a framework for constructing clustering algorithms based on the mathematical foundations given in Section 2.1.1. Within this framework, we develop a clustering algorithm for the Neyman's criterion (2.2), whose strict concavity (see Section 2.1.3 for a proof) allows for application of Theorem 2.

---

[4] *A supergradient of a concave functional I at a point z^* is a vector q satisfying the condition $I(z) - I(z^*) \leq (q, z - z^*)$ for any point z in the domain of functional I.*

The input data for a clustering algorithm is assumed to be given in the form of a finite sample $X = \{x_1, x_2, \ldots, x_m\}$ of points. We denote by $\tilde{\mathcal{H}}$ the set of all possible clusterings into K non-empty clusters constructed on the basis of the sample X. Additionally, we denote by p_α the zeroth non-normalized moment $M_\alpha^{(0)}$, i.e., the probability of cluster α. Given a clustering $H = (h_1(x), \ldots, h_K(x))$, $H \in \tilde{\mathcal{H}}$, the vector $\tilde{\mu}(H) = \left(\tilde{p}_1, \tilde{M}_1^{(1)}, \ldots, \tilde{M}_1^{(r)}, \ldots, \tilde{p}_K, \tilde{M}_K^{(1)}, \ldots, \tilde{M}_K^{(r)}\right)$ of the non-normalized sample cluster moments is estimated over the sample X as follows:

$$\begin{aligned} \tilde{p}_\alpha &= \frac{1}{m}\sum_{i=1}^{m} h_\alpha(x_i) = \frac{m_\alpha}{m}, \\ \tilde{M}_\alpha^{(l)} &= \frac{1}{m}\sum_{i=1}^{m} x_i^l h_\alpha(x_i),\ l = 1, \ldots, r,\ \alpha = 1, \ldots, K, \end{aligned}$$

where m_α is the number of points in cluster α.

In general, for a given functional $I(\mu(H))$, we are interested in finding a clustering H^* such that

$$H^* = \arg\min_{H \in \tilde{\mathcal{H}}} I\left(\tilde{\mu}(H)\right).$$

However, an exhaustive enumeration of the set $\tilde{\mathcal{H}}$ of all possible partitions of m points into K clusters is infeasible in most cases, because the number

$$S(m, K) = \frac{1}{K!}\sum_{\alpha=1}^{K}(-1)^{K-\alpha}\begin{pmatrix} K \\ \alpha \end{pmatrix}\alpha^m$$

of distinct partitions grows rapidly with K and m. For example, there are $S(10, 4) = 34,105$ partitions of ten objects into four clusters, while there are $S(19, 4) \approx 10^{10}$ partitions of nineteen objects into four clusters [34]. We, therefore, resort to search for clusterings that provide functional I with local minima.

From Theorem 2 follows directly that, in cases when functional I is a strictly concave differentiable[5] function of the non-normalized cluster moments, the Basic Gradient Descent (BGD) procedure (Algorithm 1) is guaranteed to converge to a clustering that

[5]If a concave functional I is differentiable at a point z^*, then there exists a unique supergradient of I at the point z^*, namely the gradient of functional I at the point z^*.

provides functional I with a local minimum. In Algorithm 1, $\nabla I(\mu(H))$ denotes the gradient

$$\nabla I = \left(\frac{\partial I}{\partial M_1^{(0)}}, \frac{\partial I}{\partial M_1^{(1)}}, \ldots, \frac{\partial I}{\partial M_1^{(r)}}, \ldots, \frac{\partial I}{\partial M_K^{(0)}}, \frac{\partial I}{\partial M_K^{(1)}}, \ldots, \frac{\partial I}{\partial M_K^{(r)}} \right)$$

of a functional I evaluated at a point $\mu(H)$. Step 7 of the BGD avoids degenerate solutions that contain clusters with fewer points than a predefined threshold $b \geq 0$. As we will see in Section 2.1.3, Neyman's criterion (2.2) assumes non-zero cluster variances, estimation of which requires at least two distinct points to be present in each cluster ($b = 2$). K-means criterion (2.1), on the other hand, is only concerned with cluster means. In this case, we allow singleton clusters ($b = 1$) to be present in the K-means solution.

Algorithm 1: Basic Gradient Descent (BGD)

Input: Sample $X = \{x_1, x_2, \ldots, x_m\}$ of distinct points
Input: Initial (arbitrary) clustering H
Input: Minimum cluster size b
Output: Clustering H^*

1 **repeat**
2 $H^* \leftarrow H$
3 Compute vector $\tilde{\mu}(H)$ of the non-normalized sample cluster moments
4 Compute vector $c = \nabla I(\tilde{\mu}(H))$ of coefficients
5 Construct the polynomial clustering H^c using characteristic functions $h_\alpha^c(x)$ defined by (2.11)
6 $H \leftarrow H^c$
7 **for** $\alpha = 1, \ldots, K$ **do**
8 **if** $\sum_{x \in X} h_\alpha^c(x) < b$ **then**
9 Put into cluster α, $\left(b - \sum_{x \in X} h_\alpha^c(x) \right)$ closest points as measured by the corresponding membership function (2.10)
  end
  end
  until $\tilde{\mu}(H^*) = \tilde{\mu}(H)$
10 **return** *Clustering* H^*

The overall form of the clustering algorithm proposed in this work for functional (2.2) is the same as that of the K-means algorithm. The difference between the two algorithms lies in the membership functions (2.10) according to which clusterings are constructed

in step 5 of the BGD. The precise form of membership functions for criterion (2.2) is derived in Section 2.1.3, where comparisons with the K-means membership functions are also drawn. The general clustering algorithm for functionals (2.1) and (2.2) is given by Algorithm 2, which acts as a wrapper around the BGD. By executing the BGD starting from N different randomly generated partitions of the data, Algorithm 2 obtains a deeper minimum of the clustering criterion. Algorithm 2 then outputs a clustering giving the smallest value of the criterion.

Algorithm 2: Clustering Algorithm for Functionals (2.1) and (2.2)

Input: Sample $X = \{x_1, x_2, \ldots, x_m\}$ of distinct points
Input: Number K of clusters
Input: Number N of iterations
Output: The best clustering H^* found during the N iterations
if *Functional (2.1)* **then**
    b = 1
end
if *Functional (2.2)* **then**
    b = 2
end
Initialize the set $\mathcal{H}^*$ of locally optimal clusterings: $\mathcal{H}^* = \emptyset$
for $i = 1, \ldots, N$ **do**
    Generate a random assignment H_i of points to clusters (for functional (2.1) each cluster must be non-empty; for functional (2.2) each cluster must contain at least two points)
    Execute BGD initialized with H_i: $H_i^* = \mathrm{BGD}(X, H_i, b)$
    $\mathcal{H}^* = \mathcal{H}^* \cup \{H_i^*\}$
end
return $H^* = \arg\min_{H \in \mathcal{H}^*} I(\tilde{\mu}(H))$

The total computational complexity of Algorithm 2 is $O(NtKmn)$ scalar additions and multiplications, where t is the number of iterations performed by the BGD during the N iterations in Algorithm 2.

2.1.3 Cluster Membership Functions for The Neyman's Criterion

As was mentioned earlier, the Neyman's criterion (2.2) was originally developed [51] for stratified sampling from one-dimensional data. We generalize functional (2.2) to multi-dimensional data as follows. Let $\mathcal{M}_\alpha^{(l)}$ denote the l-th normalized moment of cluster α,

$$\mathcal{M}_\alpha^{(l)} = \frac{M_\alpha^{(l)}}{p_\alpha},$$

where $p_\alpha = M_\alpha^{(0)}$ is the probability of cluster α. Functional (2.2) can then be rewritten as

$$I_2 = \sum_{\alpha=1}^{K} p_\alpha \sigma_\alpha = \sum_{\alpha=1}^{K} p_\alpha \sqrt{\left(\mathcal{M}_\alpha^{(2)} - \left(\mathcal{M}_\alpha^{(1)}\right)^2\right)}, \tag{2.13}$$

where $\left(\mathcal{M}_\alpha^{(1)}\right)^2$ denotes the scalar product $\left(\mathcal{M}_\alpha^{(1)}, \mathcal{M}_\alpha^{(1)}\right)$ of the mean vector of cluster α with itself.

We now prove that functional (2.13) is strictly concave, which makes the application of Theorem 2 possible. We assume that for any clustering $H \in \mathcal{H}$, cluster variances are positive, i.e., $\sigma_\alpha^2 > 0,\ \alpha = 1, \ldots, K$.

Claim 1. Functional I_2 is strictly concave.

Proof. We prove the claim by showing that the α-th functional $I_{2\alpha} = p_\alpha \sigma_\alpha$ in summation (2.13) is strictly concave, from which it follows that functional I_2 is strictly concave. First, we compute the gradient $\nabla I_{2\alpha} = \left(c_\alpha^{(0)}, c_\alpha^{(1)}, c_\alpha^{(2)}\right)$ of functional $I_{2\alpha}$:

$$\begin{array}{rcccccl} c_\alpha^{(0)} & = & \frac{\partial I_2}{\partial p_\alpha} & = & \frac{M_\alpha^{(2)}}{2 p_\alpha \sigma_\alpha} & = & \frac{\mathcal{M}_\alpha^{(2)}}{2\sigma_\alpha}, \\ c_\alpha^{(1)} & = & \frac{\partial I_2}{\partial M_\alpha^{(1)}} & = & -\frac{M_\alpha^{(1)}}{p_\alpha \sigma_\alpha} & = & -\frac{\mathcal{M}_\alpha^{(1)}}{\sigma_\alpha}, \\ c_\alpha^{(2)} & = & \frac{\partial I_2}{\partial M_\alpha^{(2)}} & = & \frac{1}{2\sigma_\alpha}. & & \end{array} \tag{2.14}$$

Let the non-normalized cluster moments of cluster α under a clustering $H \in \mathcal{H}$ be denoted by $\mu_\alpha(H) = \left(p_\alpha, M_\alpha^{(1)}, M_\alpha^{(2)}\right)$. For any two clusterings $H \in \mathcal{H}$ and $\hat{H} \in \mathcal{H}$ we

have

$$\begin{aligned}
\Big(\nabla I_{2\alpha}(\mu_\alpha(H)), \mu_\alpha(\hat{H}) - \mu_\alpha(H)\Big) &= \left(\frac{\mathcal{M}_\alpha^{(2)}}{2\sigma_\alpha}(\hat{p}_\alpha - p_\alpha) - \frac{1}{\sigma_\alpha}\left(\mathcal{M}_\alpha^{(1)}, \hat{M}_\alpha^{(1)} - M_\alpha^{(1)}\right) + \frac{1}{2\sigma_\alpha}\left(\hat{M}_\alpha^{(2)} - M_\alpha^{(2)}\right)\right) \\
&= \frac{1}{2\sigma_\alpha}\left(\mathcal{M}_\alpha^{(2)}\hat{p}_\alpha - M_\alpha^{(2)} - 2\left(\mathcal{M}_\alpha^{(1)}, \hat{M}_\alpha^{(1)}\right) + 2p_\alpha\left(\mathcal{M}_\alpha^{(1)}\right)^2 + \hat{M}_\alpha^{(2)} - M_\alpha^{(2)}\right) \\
&= \frac{\hat{p}_\alpha}{2\sigma_\alpha}\left(\mathcal{M}_\alpha^{(2)} - 2\left(\mathcal{M}_\alpha^{(1)}, \hat{\mathcal{M}}_\alpha^{(1)}\right) + \hat{\mathcal{M}}_\alpha^{(2)}\right) - p_\alpha\sigma_\alpha \\
&= \frac{\hat{p}_\alpha}{2\sigma_\alpha}\left(\mathcal{M}_\alpha^{(2)} - \left(\mathcal{M}_\alpha^{(1)}\right)^2 + \left(\mathcal{M}_\alpha^{(1)} - \hat{\mathcal{M}}_\alpha^{(1)}\right)^2 - \left(\hat{\mathcal{M}}_\alpha^{(1)}\right)^2 + \hat{\mathcal{M}}_\alpha^{(2)}\right) - p_\alpha\sigma_\alpha \\
&= \frac{\hat{p}_\alpha}{2\sigma_\alpha}\left(\sigma_\alpha^2 + \left(\mathcal{M}_\alpha^{(1)} - \hat{\mathcal{M}}_\alpha^{(1)}\right)^2 + \hat{\sigma}_\alpha^2\right) - p_\alpha\sigma_\alpha,
\end{aligned}$$

and

$$I_{2\alpha}(\mu_\alpha(\hat{H})) - I_{2\alpha}(\mu_\alpha(H)) = \hat{p}_\alpha\hat{\sigma}_\alpha - p_\alpha\sigma_\alpha.$$

By subtracting the first equation from the second and simplifying, we obtain the following inequality

$$\begin{aligned}
&I_{2\alpha}(\mu_\alpha(\hat{H})) - I_{2\alpha}(\mu_\alpha(H)) - \Big(\nabla I_{2\alpha}(\mu_\alpha(H)), \mu_\alpha(\hat{H}) - \mu_\alpha(H)\Big) = \\
&= \frac{\hat{p}_\alpha}{2\sigma_\alpha}\left(2\hat{\sigma}_\alpha\sigma_\alpha - \sigma_\alpha^2 - \left(\mathcal{M}_\alpha^{(1)} - \hat{\mathcal{M}}_\alpha^{(1)}\right)^2 - \hat{\sigma}_\alpha^2\right) \\
&= -\frac{\hat{p}_\alpha}{2\sigma_\alpha}\left((\sigma_\alpha - \hat{\sigma}_\alpha)^2 + \left(\mathcal{M}_\alpha^{(1)} - \hat{\mathcal{M}}_\alpha^{(1)}\right)^2\right) < 0,\ \mu_\alpha(\hat{H}) \neq \mu_\alpha(H).
\end{aligned}$$

From the definition of a strictly concave function follows that functional $I_{2\alpha}$ is strictly concave. Therefore, functional $I_2 = \sum_{\alpha=1}^{K} I_{2\alpha}$ is strictly concave. □

Using the gradient (2.14) for specifying membership functions (2.10) yields

$$\begin{aligned}
f_\alpha(x) &= c_\alpha^{(0)} + \left(c_\alpha^{(1)}, x\right) + c_\alpha^{(2)}x^2 \\
&= \tfrac{\mathcal{M}_\alpha^{(2)}}{2\sigma_\alpha} - \tfrac{1}{\sigma_\alpha}\left(\mathcal{M}_\alpha^{(1)}, x\right) + \tfrac{x^2}{2\sigma_\alpha} \\
&= \tfrac{1}{2\sigma_\alpha}\left(\mathcal{M}_\alpha^{(2)} - \left(\mathcal{M}_\alpha^{(1)}\right)^2\right) + \tfrac{1}{2\sigma_\alpha}\left(\left(\mathcal{M}_\alpha^{(1)}\right)^2 - 2\left(\mathcal{M}_\alpha^{(1)}, x\right) + x^2\right) \\
&= \tfrac{\sigma_\alpha}{2} + \tfrac{1}{2\sigma_\alpha}\left(x - \mathcal{M}_\alpha^{(1)}\right)^2.
\end{aligned} \tag{2.15}$$

The term $\left(x - \mathcal{M}_\alpha^{(1)}\right)^2$ in (2.15) is the squared Euclidean distance between a point $x \in \mathcal{X}$ and the cluster's mean vector $\mathcal{M}_\alpha^{(1)}$. These squared Euclidean distances are, in fact, the cluster membership functions $f_\alpha^{\mathrm{KM}}(x)$ for the K-means criterion (2.1):

$$f_\alpha^{\mathrm{KM}}(x) = \left(x - \mathcal{M}_\alpha^{(1)}\right)^2. \tag{2.16}$$

A detailed derivation of membership functions (2.16) can be found in Appendix C along with a proof of strict concavity of the K-means criterion (2.1).

Membership functions (2.16) and (2.15) elucidate a key difference between criteria (2.1) and (2.2). Note that membership functions (2.16) do not depend on the second non-normalized moments $M_\alpha^{(2)}$. This stems from the fact that, as illustrated by equation (C.2) in Appendix C, the K-means criterion (2.1) depends only on the first two non-normalized moments p_α and $M_\alpha^{(1)}$, and is independent of $M_\alpha^{(2)}$. The Neyman's criterion (2.13), on the other hand, is more complex since it depends on all three non-normalized moments p_α, $M_\alpha^{(1)}$ and $M_\alpha^{(2)}$. As a result, cluster membership functions (2.15) depend not only on the squared Euclidean distance between a point and the cluster's mean $\mathcal{M}_\alpha^{(1)}$, but also on the cluster's scatter as measured by its standard deviation σ_α.

In order to show the exact role cluster variances play in criterion (2.2) and to further underline its differences from criterion (2.1), consider the discriminant surface

$$\begin{aligned} F(x) = f_\alpha(x) - f_\beta(x) &= \tfrac{\sigma_\alpha}{2} + \tfrac{1}{2\sigma_\alpha}\left(x - \mathcal{M}_\alpha^{(1)}\right)^2 - \tfrac{\sigma_\beta}{2} - \tfrac{1}{2\sigma_\beta}\left(x - \mathcal{M}_\beta^{(1)}\right)^2 \\ &= (\sigma_\beta - \sigma_\alpha)\, x^2 + 2\left(\sigma_\alpha \mathcal{M}_\beta^{(1)} - \sigma_\beta \mathcal{M}_\alpha^{(1)}, x\right) + \\ &\quad + \sigma_\beta \left(\mathcal{M}_\alpha^{(1)}\right)^2 - \sigma_\alpha \left(\mathcal{M}_\beta^{(1)}\right)^2 + \sigma_\alpha \sigma_\beta \left(\sigma_\alpha - \sigma_\beta\right) = 0. \end{aligned} \tag{2.17}$$

specified by membership functions (2.15) between two clusters α and β, and compare (2.17) with the K-means discriminant surface

$$\begin{aligned} F^{\mathrm{KM}}(x) = f_\alpha^{\mathrm{KM}}(x) - f_\beta^{\mathrm{KM}}(x) &= \left(x - \mathcal{M}_\alpha^{(1)}\right)^2 - \left(x - \mathcal{M}_\beta^{(1)}\right)^2 \\ &= 2\left(\mathcal{M}_\beta^{(1)} - \mathcal{M}_\alpha^{(1)}, x\right) + \left(\mathcal{M}_\alpha^{(1)}\right)^2 - \left(\mathcal{M}_\beta^{(1)}\right)^2 = 0. \end{aligned} \tag{2.18}$$

specified by (2.16). Equations (2.17) and (2.18) reveal the following relationship between criteria (2.1) and (2.2). When cluster variances are equal, discriminant surface (2.17) coincides with (2.18). Thus, criterion (2.2) produces the same clustering as criterion (2.1). Furthermore, surface (2.18) is the hyperplane that contains the mid point $x = \frac{1}{2}\left(\mathcal{M}_\beta^{(1)} + \mathcal{M}_\alpha^{(1)}\right)$ of the line segment connecting the cluster means $\mathcal{M}_\alpha^{(1)}$

and $\mathcal{M}_\beta^{(1)}$, and whose norm $2\left(\mathcal{M}_\beta^{(1)} - \mathcal{M}_\alpha^{(1)}\right)$ is collinear with that line segment. In contrast, (2.17) is a quadratic surface that contains a point $x = (1-\tau)\mathcal{M}_\alpha^{(1)} + \tau\mathcal{M}_\beta^{(1)}$, $\tau \in (0,1)$, where

$$\tau = \frac{\sigma_\alpha - \sqrt{\sigma_\alpha\sigma_\beta\left(\frac{\left(\sigma_\beta - \sigma_\alpha\right)^2}{\left(\mathcal{M}_\beta^{(1)} - \mathcal{M}_\alpha^{(1)}\right)^2} + 1\right)}}{\sigma_\alpha - \sigma_\beta}. \tag{2.19}$$

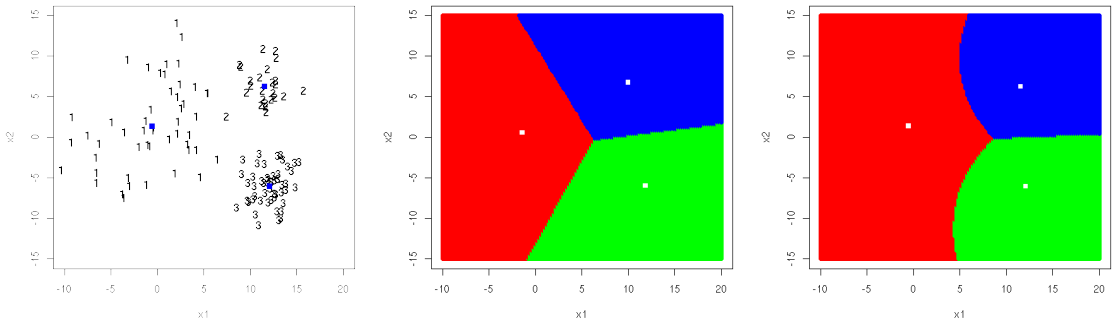
The differences between discriminant surfaces (2.17) and (2.18) are illustrated by Figure 2.1. The dataset shown in Figure 2.1a consisted of three clusters, each generated by a Gaussian distribution. The data generator parameters were:

- Cluster probabilities: $p_1 = 0.4,\ p_2 = 0.2,\ p_3 = 0.4$,
- Cluster means: $\mu_1 = (0,0),\ \mu_2 = (12,6),\ \mu_3 = (12,-6)$,
- Cluster covariance matrices: $\Sigma_1 = 4.5I,\ \Sigma_{\{2,3\}} = 2I$.

The same set of $N = 50$ randomly generated initial assignments of points to clusters was used by each algorithm. Clusterings yielding the smallest values of criteria (2.1) and (2.2) are shown in Figures 2.1b and 2.1c, respectively. As can be seen from Figure 2.1b, membership functions (2.16) produced linear discriminant surfaces regardless of cluster probabilities and variances. In contrast, due to unequal variances of the red and the other two clusters, membership functions (2.15) produced quadratic discriminant surfaces shown in Figure 2.1c. Since the variance of the red cluster was greater, the corresponding discriminant surfaces were shifted further away from its mean. The discriminant surface between the blue and the green clusters remained linear due to variances of these clusters being equal.

2.1.4 Related Work

Despite the fact that (2.8) encompasses a large family of functions, the only clustering criterion known to fall under the theoretical framework of [5] was the K-means functional (2.1). In this work, we found that there is another criterion, namely (2.2), that



(a) Input dataset (b) Criterion I_1 (K-means) (c) Criterion I_2

Figure 2.1: Clusterings obtained on the dataset shown in Figure 2.1a. Cluster means are indicated by squares.

also fits within the framework of [5]. We proved that criterion (2.2) is strictly concave and developed the first multi-dimensional clustering algorithm for minimization of this criterion.

Another variance-based clustering criterion

$$I_3 = \sum_{\alpha=1}^{K} p_\alpha^2 \sigma_\alpha^2, \tag{2.20}$$

was proposed in [37]. A globally optimal clustering minimizing criterion (2.20) can be obtained in one dimension by a dynamic programming approach [7]. In multi-dimensional spaces, however, minimization of (2.20) is challenging. As we showed in [46], criterion (2.20) is non-convex and therefore does not fall within the framework considered in this work. Hence, a different method for minimizing (2.20) in multi-dimensional spaces is required.

While in this work we consider efficient local minimization of criteria (2.1) and (2.2), a number of approximation algorithms were recently proposed [4, 17, 56] for global minimization of clustering criteria of the general form

$$J(X, H) = \sum_{\alpha=1}^{K} \Phi_\alpha = \sum_{\alpha=1}^{K} \sum_{x,y \in X} \phi(x, y) h_\alpha(x) h_\alpha(y), \tag{2.21}$$

where $\phi(x, y)$ is a non-negative "cost" of placing points x and y into the same cluster. Unfortunately, none of the aforementioned approximations schemes are applicable for

minimization of criteria (2.1) and (2.2) as they are not of the form (2.21). By letting $\phi(x, y) = ||x-y||^2$ be the squared Euclidean distance and after some algebra, we obtain

$$\Phi_\alpha = \sum_{x,y \in X} ||x - y||^2 h_\alpha(x) h_\alpha(y) = 2m_\alpha \sum_{x \in X} ||x - \bar{x}_\alpha||^2 h_\alpha(x) = 2m_\alpha^2 \sigma_\alpha^2, \tag{2.22}$$

where m_α is the number of points in cluster α, $\bar{x}_\alpha = \frac{1}{m_\alpha} \sum_{x \in X} x h_\alpha(x)$ is its mean vector and σ_α^2 is its variance. From (2.22) follows that

$$p_\alpha \sigma_\alpha^2 = \frac{m_\alpha}{m} \sigma_\alpha^2 = \frac{\Phi_\alpha}{2m_\alpha m},$$

where $m = \sum_{\alpha=1}^{K} m_\alpha$ is the total number of points in X. Therefore, criterion (2.1) is not of the form (2.21). An analogous argument can be applied to show that since

$$p_\alpha \sigma_\alpha = \frac{\sqrt{\Phi_\alpha}}{\sqrt{2}m},$$

criterion (2.2) also is not of the form (2.21).

2.2 Higher Order Transformations

Real-world heterogenous data often have complex structure comprised of overlapping homogeneous subgroups of data instances that are not linearly separable from the rest of the dataset. Presence of non-linear dependencies in the data precludes simple clustering criteria such as (2.1) from identifying adequate partitions. This problem can be addressed by either applying a more complex, non-linear clustering criterion, e.g. (2.2), or by projecting the data into a different feature space using non-linear mapping functions. The latter approach aims to increase separability between the underlying "true" clusters, thus simplifying the structure of the data and making it more suitable for clustering.

One of the key contributions of this chapter is the Unsupervised Second Order Transformation (USOT) described in Section 2.2.4. USOT is an adaptive non-linear

function that maps data from a boolean[6] space to a real space thereby emphasizing specifics of the various homogeneous subgroups of data instances. When mapping a data instance, USOT uses two types of information – local and global. The local information is extracted directly from the data instance being mapped. The global component of USOT comes in the form of probabilistic dependencies between features. The dependencies are estimated based on indirect co-occurrences between features in the dataset. As will be explained in more detail in Section 2.2.1, a pair of features i and j may never co-occur within a single data instance. In fact, such co-occurrences, termed first-order paths [25], are often very sparse. However, there may exist indirect links, or higher-order paths between i and j through some intermediate features. In our work [26] on supervised learning, we found higher-order paths to be an abundant source of extremely valuable information that allowed higher-order classifiers to consistently outperform the traditional methods.

The novelty behind USOT is (a) that it leverages in a unsupervised manner, higher-order co-occurrences between features, and (b) that it considers each feature in the context of probabilistic relationships with other features. USOT has two desirable properties. USOT adaptively selects features that would influence the mapping of a given feature. If a feature j exhibits the same distribution regardless of the value of a feature i, then feature j will have no effect on mapping feature i. Moreover, interpretability of dimensions of the USOT space is retained due to one-to-one correspondence with the original boolean features.

The intuition behind USOT originated from our work on higher-order classifiers [26], and in particular from the Supervised Second Order Transformation (SSOT) described in Section 2.2.3. SSOT is a novel data transformation that requires the knowledge of true class labels of the instances comprising a training set. Both USOT and SSOT

[6]Many methods of mapping real-valued data to boolean spaces exist, but their development is beyond the scope of this dissertation. We did, however, use some of these methods in the experiments in Section 3.5.

are defined over the space of higher-order paths. However, aside from SSOT being a supervised transformation, the main difference between USOT and SSOT lies in the way the two mappings use the higher-order paths. While USOT considers probabilistic dependencies between a feature and all other features, SSOT makes use of probabilistic dependencies between a class indicator variable and the features.

Another contribution of this chapter is a $O\left((m+n)n^2\right)$ time algorithm for obtaining the counts of second-order paths for each feature in a dataset with m instances and n features. This algorithm improves over the $O\left(m^2n^3\right)$ complexity of a straight-forward path counting algorithm. Both algorithms are described in Section 2.2.5.

The rest of this chapter is organized as follows. Section 2.2.1 describes the data representation underlying the proposed non-linear mapping functions SSOT and USOT, and defines the notion of a higher-order path. Probabilistic characterization of features in the space of second-order paths is given in Section 2.2.2. Section 2.2.3 describes the SSOT and illustrates the effects of transitioning from the traditional feature vector representation to the space of higher-order paths. USOT is presented in Section 2.2.4. Related work is discussed in Section 2.2.6.

2.2.1 Data Representation by a Bipartite Graph

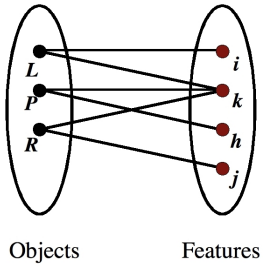
Below, we introduce the data representation that will be used in the following sections. We assume that all data instances are provided in the form of n-dimensional boolean vectors. The notions of a dataset and a data matrix are therefore assumed to be equivalent. Rows of a data matrix $X = ||x_L^i||$ correspond to objects, while columns correspond to features. We denote object indices by capital letters, e.g. L, and feature indices by small letters, e.g. i.

An $m \times n$ data matrix X can be viewed as a bipartite graph $G = (V_{\mathrm{O}} \cup V_{\mathrm{F}}, E)$. Vertices in V_{O} correspond to objects, while vertices in V_{F} correspond to features. Two vertices $L \in V_{\mathrm{O}}$ and $i \in V_{\mathrm{F}}$ are connected by an edge $(L, i) \in E$ iff object L contains

the feature i, i.e., iff $x_L^i = 1$. An illustration of a data matrix and the corresponding bipartite graph is given in Figure 2.2.

	i	k	h	j
L	1	1	0	0
P	0	1	1	0
R	0	1	0	1

(a)



(b)

Figure 2.2: (a) A boolean dataset, and (b) its representation by a bipartite graph

Traditional learning methods operating in vector spaces typically consider each feature independently of others by, for example, computing frequency of occurrence of an individual feature without regard for patterns of co-occurrence of this feature with others. Frequency of occurrence of feature i can be obtained from graph G by taking the degree of the feature vertex $i \in V_{\mathrm{F}}$. The degree of a vertex, however, is only a small subset of the vast amount of information reflected by G. In order to make use of this rich information, we depart from the traditional approach by considering (indirect) co-occurrences between features. Such co-occurrences are captured by chain subgraphs of graph G. We follow the terminology of Ganiz et al. [24, 25] who termed such subgraphs as paths and further classified them by the number of object vertices they span. The number of object vertices determines the order of a path. Below we give formal definitions of the first- and second-order paths that will be used in the following sections. We also provide an illustration of patterns of connectivity between features as a function of the path order.

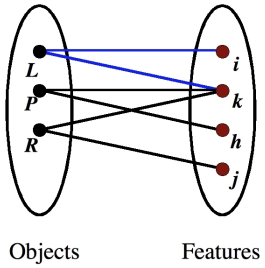
Definition 1. A first-order path (i, L, k) between features i and k is a chain subgraph where feature vertices i and k are linked through some common object vertex L.

As shown in Figure 2.3, first-order path (i, L, k) captures the co-occurrence between

features i and k within a single data instance L. We refer to feature co-occurrences within a single data instance as first-order co-occurrences. The number of first-order paths connecting features i and k in a dataset equals the frequency of their first-order co-occurrence.

	i	k	h	j
L	**1**	**1**	0	0
P	0	1	1	0
R	0	1	0	1

(a)



(b)

Figure 2.3: A first-order path (i, L, k) between features i and k is a chain subgraph that captures the first-order co-occurrence between features i and k within a single data instance L

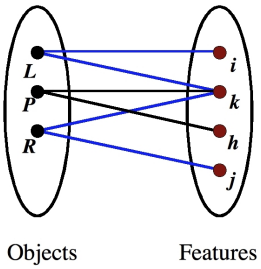
Definition 2. A second-order path (i, L, k, R, j) between features i and j is a chain subgraph where feature vertices i and j are linked through an intermediate feature vertex k and two distinct object vertices L and R.

A second-order path is exemplified in Figure 2.4. Second-order paths are able to capture indirect co-occurrences between features that may not co-occur within a single data instance. In the example shown in Figure 2.4, features i and j do not have a first-order co-occurrence, but there exists a second-order co-occurrence (i, L, k, R, j) between them. In fact, second-order co-occurrences are much more abundant than first-order. Figures 2.5a and 2.5b visualize the frequencies of first- and second-order co-occurrences, respectively. The black regions in Figure 2.5a indicate the absence of first-order paths between the corresponding pairs of features. The total of roughly 75% of pairs of features in Figure 2.5a have no first-order co-occurrences. Figure 2.5b, on the other hand, demonstrates a drastically different sparsity pattern of the second-order paths in the same data graph. Virtually every pair of features became connected by at

least one path as a result of transitioning from first to second order.

	i	k	h	j
L	**1**	**1**	0	0
P	0	1	1	0
R	0	**1**	0	**1**

(a)



(b)

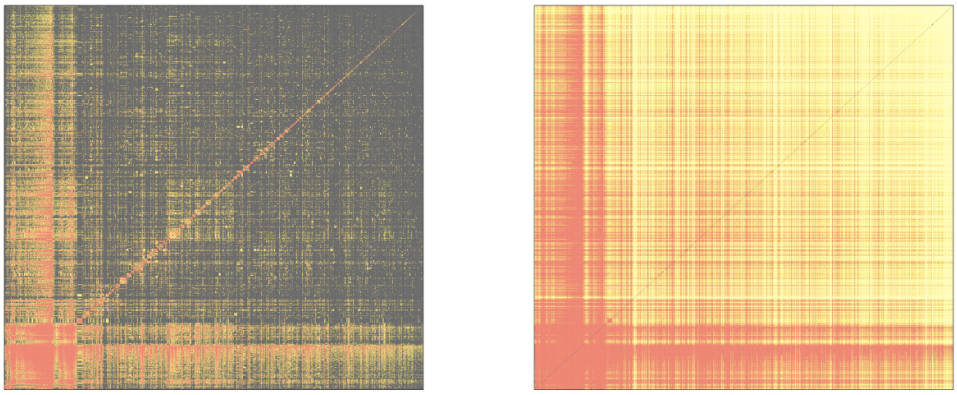
Figure 2.4: A second-order path (i, L, k, R, j) between features i and k is a chain subgraph that captures the first-order co-occurrence between features i and k within a single data instance L

Second-order paths simultaneously capture feature co-occurrences within objects as well as feature sharing patterns across objects, and in doing so provide a much richer data representation than the traditional feature vector form. As will be demonstrated by experimental results in Chapter 3, this richness of representation plays a crucial role in significantly improving the performance of pattern classifiers and clustering methods.

Our experimental results [26] demonstrate that the use of first-order paths does not improve classification accuracy. At the same time, while the use of second-order paths yields statistically significant performance improvements, the effect of incorporating third- and higher-order paths is insignificant. Moreover, computation of paths of order higher than two adds considerably to the algorithmic complexity. In order to keep the computational complexity of our algorithms manageable while retaining the desired performance improvements, we restrict our attention to the second-order paths.

2.2.2 Probabilistic Characterization of Features by Second Order Paths

We begin this section with a description of the traditional (i.e., zero-order) probabilistic characterization of boolean features and then show an extension, first proposed in [24],



(a) First-order (b) Second-order

Figure 2.5: Feature co-occurrence matrices of the RELIGION dataset. Rows and columns of the matrices were rearranged for visualization. Red color indicates higher co-occurrence frequency than indicated by yellow color. Black color indicates no co-occurrence (i.e., zero frequency). Note that colors are not comparable across plots due to different inter-quantile ranges of the co-occurrence frequencies.

of this characterization into the space of higher-order paths. The probabilistic higher-order characterization will be used for developing novel data transformations in Sections 2.2.3 and 2.2.4.

The zero-order probability mass function $P(x^i|X)$ of feature i is defined over two events: presence of feature i in a randomly chosen object from a dataset X, and the absence of that feature. The corresponding conditional probabilities are estimated using the frequency of occurrence of feature i in dataset X by

$$P(x^i = 1|X) = \frac{|\{x : x^i = 1,\ x \in X\}|}{|X|} \quad \text{and} \quad P(x^i = 0|X) = 1 - P(x^i = 1|X). \tag{2.23}$$

Probabilistic characterization (2.23) was extended into the space of higher-order paths by [24] who defined events over sets of higher-order paths rather than individual data instances. We now describe this extension.

Let $\Phi(X)$ denote the set of all second-order paths in a dataset X. Further let

$\varphi(i, X) \subseteq \Phi(X)$ denote the subset of second-order paths that contain feature i in dataset X. Set $\varphi(i, X)$ defines an event that a randomly chosen second-order path contains feature i. Together, sets $\Phi(X)$ and $\varphi(i, X)$ allow for characterization of each feature i by a probability mass function $\hat{P}(x^i|X)$ defined over two events: presence of feature i in a randomly chosen second-order path, and the absence of that feature from a randomly chosen second-order path. The corresponding conditional second-order probabilities can then be estimated by

$$\hat{P}(x^i = 1|X) = \frac{|\varphi(i, X)|}{|\Phi(X)|}, \quad \text{and} \quad \hat{P}(x^i = 0|X) = 1 - \hat{P}(x^i = 1|X). \tag{2.24}$$

2.2.3 Supervised Second Order Transformation

In this section, we present a novel data transformation that allows any classifier operating in vector spaces to take advantage of higher-order co-occurrences between features. We describe our approach for the case of binary classification. This, however, does not limit the applicability of the proposed approach, because numerous methods for multi-class classification based on binary classifiers have been proposed (see [55] for an overview). The proposed data transformation proceeds as follows.

Let $C = \{c_1, \ldots, c_K\}$ denote the set of class labels. Given two sets X_j and X_k of (training) objects from classes c_j and c_k, respectively, the class conditional second-order feature probabilities (2.24) are computed. Let us denote the corresponding conditional log likelihood ratios as

$$\phi_i^{(1)} = \log \frac{\hat{P}(x^i|X_j)}{\hat{P}(x^i|X_k)}, \tag{2.25}$$

and

$$\phi_i^{(0)} = \log \frac{1 - \hat{P}(x^i|X_j)}{1 - \hat{P}(x^i|X_k)}. \tag{2.26}$$

Each binary vector $x = (x^1, \ldots, x^n)$, $x \in X_j \cup X_k$, is then transformed into a real vector

$\hat{x} = (\hat{x}^1, \ldots, \hat{x}^n)$, where

$$\hat{x}^i = \begin{cases} \frac{\phi_i^{(1)}}{\sqrt{|\phi_i^{(1)}|}}, & \text{if } x^i = 1, \phi_i^{(1)} \neq 0 \\ \frac{\phi_i^{(0)}}{\sqrt{|\phi_i^{(0)}|}}, & \text{if } x^i = 0, \phi_i^{(0)} \neq 0 \\ 0, & \text{otherwise.} \end{cases} \tag{2.27}$$

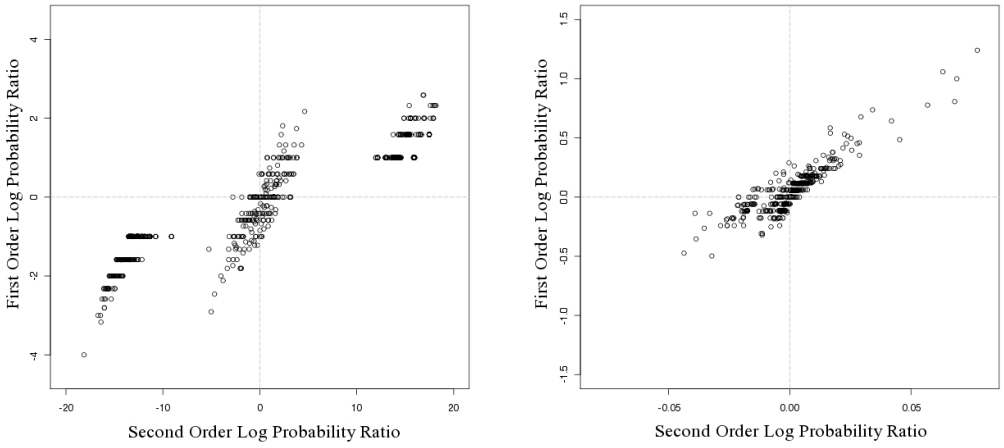
Finally, the resulting dataset $\hat{X}_j \cup \hat{X}_k$ is used as input for training a binary classifier for classes c_j and c_k.

Data transformation (2.27) assigns weights that are high in absolute values for highly discriminative features present in an object. The normalizing factors[7] in (2.27) moderate the spread of values of each feature in order to allow less discriminative features to retain a certain level of influence over the classification. This level of influence depends on the discriminative power of a feature as measured by (2.25) and (2.26).

In order to gain a deeper understanding of the effect of using higher-order paths for estimation of conditional feature probabilities (2.24), let us consider Figure 2.6 generated using a 5% (25 documents per class) random sample from "alt.atheism" and "soc.religion.christian" classes of the RELIGION dataset. For every term x^i, Figure 2.6a presents a plot of the conditional log probability ratio (2.25) obtained from higher-order probabilities (2.24) (horizontal axis) versus the log ratio obtained from zero-order probabilities (2.23) (vertical axis). Notice the differences in scales of values on the axes of Figure 2.6a: $[-20, 20]$ for higher-order log ratios versus $[-4, 4]$ for zero-order log ratios. Additionally, three distinct groups of terms appeared as a result of estimating conditional feature probabilities in the space of higher-order paths. We found that terms that fell into the right (left) most group are highly-discriminative terms that appeared in documents of only one of the classes. Figure 2.6a reveals that due to drastic increase in scale of values of higher-order log ratios, the variance along highly-discriminative dimensions increases dramatically by several orders of magnitude. In effect, this leads

[7]It is possible to omit the normalizing factors in (2.27). However, we have found experimentally that the normalized transformation, on average, yields slightly higher classification accuracies.

to increase in separability between classes and allows highly-discriminative terms to exert stronger influence on classification in higher- than in zero-order spaces.



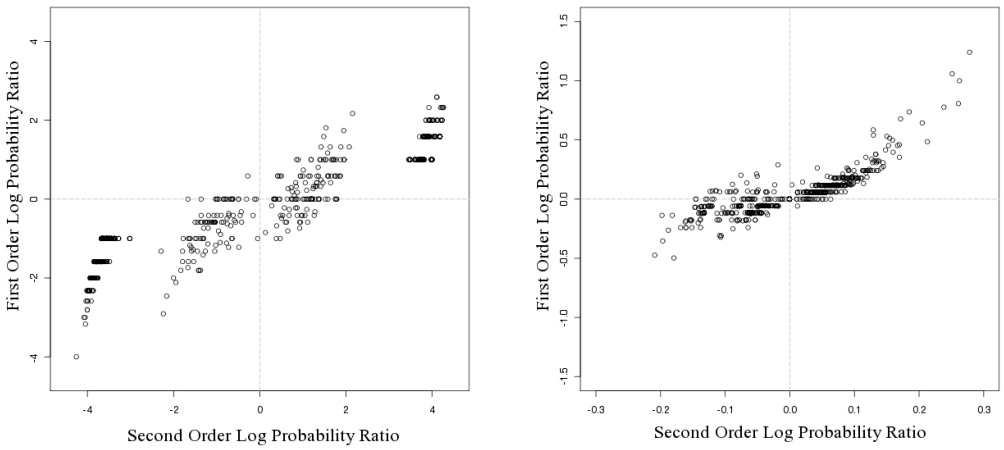
(a) (b)

Figure 2.6: Conditional log probability ratios obtained from higher-order probabilities (2.24) (horizontal axes) versus the log ratios obtained from zero-order probabilities (2.23) (vertical axes) on "alt.atheism" and "soc.religion.christian" classes of one of the 5% (25 documents per class) training samples from the RELIGION dataset

Figure 2.6b shows a plot of the conditional log probability ratios (2.26) of non-occurrence of a term in a higher-order model versus a zero-order model. An important feature in this figure is the one order of magnitude difference in values on the axes: $[-0.1, 0.1]$ for higher-order log ratios on the horizontal axis versus $[-1.5, 1.5]$ for zero-order log ratios on the vertical axis. Together, Figures 2.6a and 2.6b illustrate that while both first- and higher-order models take into account presence of terms as well as their absence, higher-order models tend to place more emphasis on the presence of terms in a document being classified.

As was noted earlier, the normalizing factors in (2.27) were introduced in order to allow features that may appear in multiple classes, but are still good discriminators as measured by the log likelihood ratios (2.25) and (2.26), to have a non-negligible impact during classification. The effect of these normalizing factors can be seen by comparing

Figures 2.6a and 2.6b with Figures 2.7a and 2.7b.



(a) (b)

Figure 2.7: Conditional log probability ratios obtained from higher-order probabilities (2.24) and normalized as in (2.27) (horizontal axes) versus the log ratios obtained from zero-order probabilities (2.23) (vertical axes) on the same dataset as in Figure 2.6

On the vertical axes in Figures 2.7a and 2.7b are plotted the same zero-order conditional log probability ratios as in Figures 2.6a and 2.6b, respectively. Plotted on the horizontal axes of Figures 2.7a and 2.7b are the higher-order conditional log probability ratios shown on the horizontal axes of Figures 2.6a and 2.6b, respectively, and normalized as in (2.27). Note the change in scales of the horizontal axes once normalization has been applied: $[-20, 20]$ before normalization (Figure 2.6a) versus $[-4, 4]$ after, and $[-0.1, 0.1]$ before normalization (Figure 2.6b) versus $[-0.3, 0.3]$ after. In addition, the middle group of terms seen in Figure 2.6a split into two subgroups in Figure 2.7a as a result of normalization. This split coupled with the change in scale of values of the higher-order log probability ratios allowed good discriminator terms from the middle group in Figure 2.6a to increase their relative significance during classification.

Although it is trivial to identify strongly discriminative features in a given training set, the question remains of how to weight those features for pattern classification. The supervised transformation proposed in this section addresses this question by leveraging

higher-order co-occurrences between features.

2.2.4 Unsupervised Second Order Transformation

In this section we present a novel Unsupervised Second Order Transformation (USOT) that aims to simplify the structure of heterogeneous data by increasing separability between various homogeneous subgroups of data instances thus making the data more adequate for clustering. The overall scheme of USOT is as follows. Each feature i partitions a n-dimensional boolean space $\mathcal{X} = \{0,1\}^n$ into two subspaces: $\mathcal{X}_1(i) = \{x : x^i = 1, x \in \mathcal{X}\}$ and $\mathcal{X}_0(i) = \mathcal{X} \setminus \mathcal{X}_1(i)$. In each of the subspaces $\mathcal{X}_1(i)$ and $\mathcal{X}_0(i)$, we represent feature i as a probabilistic function of all features. This probabilistic function is defined over the space of higher-order paths. Another function then unifies these representations across subspaces $\mathcal{X}_1(i)$ and $\mathcal{X}_0(i)$, and produces the final transformation. The unifying function also acts as a filter and prevents features that exhibit the same distribution in both subspaces $\mathcal{X}_1(i)$ and $\mathcal{X}_0(i)$, from influencing the transformation of feature i. This is a desirable property since features that follow the same distribution regardless of the value of feature i can be seen as independent of feature i and should not influence its mapping.

The novelty behind USOT is (a) that it leverages in a unsupervised manner, higher-order co-occurrences between features, and (b) that it considers each feature in the context of probabilistic relationships with other features. Unlike SSOT introduced in Section 2.2.3, USOT does not require any knowledge of the "true" class labels. Aside from SSOT being a supervised transformation, the main difference between USOT and SSOT lies in the way the two transformations use the higher-order paths. While USOT considers probabilistic dependencies between a feature and all other features, SSOT makes use of probabilistic dependencies between a class indicator variable and each feature independently.

Given a set $X \subseteq \mathcal{X}$ of n-dimensional boolean vectors, we denote by $X_1(i) = \{x :$

$x^i = 1, x \in X\}$ and $X_0(i) = X \setminus X_1(i)$, the two disjoint subsets determined by some feature i. Here, as before, $\varphi(i, X)$ will denote the subset of second-order paths that contain feature i in a dataset X, while $\Phi(X)$ will denote the set of all second-order paths in X.

In order to capture the probabilistic relationships between feature i and other features, we define the conditional higher-order probability mass function

$$P'\left(x^i|x^1,\ldots,x^n\right) = \frac{P'\left(x^1,\ldots,x^n|X_{x^i}(i)\right)P'\left(X_{x^i}(i)\right)}{P'\left(x^1,\ldots,x^n\right)}, \tag{2.28}$$

where the higher-order probability $P'\left(X_{x^i}(i)\right)$ of subset $X_{x^i}(i)$ is estimated by the ratio of the number of second-order paths in that subset,

$$P'\left(X_{x^i}(i)\right) = \frac{|\Phi\left(X_{x^i}(i)\right)|}{|\Phi\left(X_1(i)\right)| + |\Phi\left(X_0(i)\right)|}. \tag{2.29}$$

To make computation of the joint probability $P'\left(x^1,\ldots,x^n|X_{x^i}(i)\right)$ tractable, we make the common (naive) assumption of conditional independence of features given the value of feature i. It follows that

$$P'\left(x^1,\ldots,x^n|X_{x^i}(i)\right) = \prod_{j=1}^{n} P'\left(x^j|x^i\right), \tag{2.30}$$

where the conditional second-order probability mass function $P'\left(x^j|x^i\right)$ is estimated by

$$P'\left(x^j = 1|x^i\right) = \frac{|\varphi\left(j, X_{x^i}(i)\right)|}{|\Phi\left(X_{x^i}(i)\right)|}. \tag{2.31}$$

The probability mass function $P'\left(x^j|x^i\right)$ is completely defined by (2.31), since we have $P'\left(x^j = 0|x^i\right) = 1 - P'\left(x^j = 1|x^i\right)$.

The proposed USOT is a non-linear mapping $Z = \left(z^1(x),\ldots,z^n(x)\right) : \{0,1\}^n \rightarrow \Re^n$, from a n-dimensional boolean space $\mathcal{X}$ to a n-dimensional real space $\mathcal{Z}$. A notable feature of this mapping is that dimensions of space $\mathcal{Z}$ correspond to the original features and, therefore, maintain their interpretability. Function Z maps each boolean feature i to the real domain by a non-linear function $z^i\left(x^1,\ldots,x^n\right) : \{0,1\} \rightarrow \Re^n$. Mapping

functions z^i are defined over the space of second-order paths as

$$z^i\left(x^1,\ldots,x^n\right)=\frac{P'\left(x^i=1|x^1,\ldots,x^n\right)}{P'\left(x^i=0|x^1,\ldots,x^n\right)}=\prod_{j=1}^{n}\frac{P'\left(x^j|x^i=1\right)}{P'\left(x^j|x^i=0\right)}\frac{P'\left(X_1(i)\right)}{P'\left(X_0(i)\right)}. \tag{2.32}$$

For convenience of numerical computation, in practice we use log transformation of the mapping functions (2.32)

$$\log z^i\left(x^1,\ldots,x^n\right)=\sum_{j=1}^{n}\frac{P'\left(x^j|x^i=1\right)}{P'\left(x^j|x^i=0\right)}+\log\frac{P'\left(X_1(i)\right)}{P'\left(X_0(i)\right)}. \tag{2.33}$$

A relationship with the supervised learning theory can be noted here. It is easy to recognize the mapping function (2.33) as the Naive Bayes discriminant function defined over the space of second-order paths rather than the traditional feature frequencies, and where feature i plays a role of the class indicator.

Our framework (2.33) allows the use of feature frequencies, in which case the probability mass function $P\left(x^j|x^i\right)$ is estimated by

$$P\left(x^j=1|x^i\right)=\frac{|\{x:x^j=1,x\in X_{x^i}(i)\}|}{|X_{x^i}(i)|}, \tag{2.34}$$

instead of the second-order probability (2.31). Similarly, the probability $P\left(X_{x^i}(i)\right)$ of subset $X_{x^i}(i)$ is estimated by the ratio of the number of data instances in that subset,

$$P\left(X_{x^i}(i)\right)=\frac{|X_{x^i}(i)|}{|X_1(i)|+|X_0(i)|}, \tag{2.35}$$

rather than by the ratio (2.29) of the number of second-order paths. We refer to such transformation as Unsupervised Zero Order Transformation (UZOT).

2.2.5 Algorithms for Counting Second Order Paths

The number of second-order paths for each feature in a boolean dataset X with m objects and n features can be obtained in $O\left(m^2n^3\right)$ time by Algorithm 3. This algorithm is trivial to implement and requires no additional memory space beyond the $O(mn)$ space needed to store the dataset X.

Algorithm 3: Count Second-Order Paths

Input: Boolean $m \times n$ matrix $X = ||x_L^i||$
Output: Vector $p = \left(p^1, \ldots, p^n\right)$, whose i-th coordinate p^i equals the number of second-order paths that contain feature $i = 1, \ldots, n$
Output: The total number t of second-order paths

Initialize vector p to be a zero vector: $p^i := 0$, $i = 1, \ldots, n$
Initialize the path counter $t := 0$
for $i \in \{1, \ldots, n\}$ **do**
  for $k \in \{1, \ldots, n\} \setminus \{i\}$ **do**
    for $j \in \{1, \ldots, n\} \setminus \{i, k\}$ **do**
      for $L \in \{1, \ldots, m-1\}$ **do**
        for $M \in \{L+1, \ldots, m\}$ **do**
          $a := x_L^i x_L^k x_M^k x_M^j$
          $p^i := p^i + a$
          $p^k := p^k + a$
          $p^j := p^j + a$
          $t := t + a$
        end
      end
    end
  end
end
return (p, t)

However, the $O\left(m^2n^3\right)$ computational complexity can be reduced to $O\left((m+n)n^2\right)$ if enough memory is available to store two additional symmetric $n \times n$ matrices, which we denote by A and A^2. Matrix $A = X^TX$ holds the number of first-order paths between every pair of features, and is used to compute the matrix $A^2 = AA$. The ij-th element $a_{ij}^{(2)}$ of matrix A^2 holds the upper bound on the number of second-order paths where features i and j are the two end vertices. In order to obtain the exact number of such second-order paths, the value $a_{ij}^{(2)}$ must be corrected by first subtracting the number of paths where one of the features appears more than once (e.g. (i, L, i, M, j)) and then subtracting the number of paths where the same object vertex appears twice (e.g. (i, L, k, L, j)). The first correction can be accomplished by setting to zero all the diagonal elements of matrix A prior to computing A^2. Matrix A can also be used for obtaining for each feature i, the count of second-order paths where feature i is the middle vertex (e.g. (k, L, i, M, j)).

Algorithm 4 implements this faster, but more memory consuming approach. The two correction steps mentioned above are implemented by steps 6 and 12-14. Computation of matrices A and A^2 in steps 5 and 8, respectively, of Algorithm 4 takes $O(mn^2 + n^3)$ time. The loop in step 9 takes $O(mn^2)$ due to step 12, which iterates over the data instances. The computational complexity of Algorithm 4 is dominated by computation of matrices A and A^2, and is therefore $O\left((m+n)n^2\right)$. Because of its lower computational complexity, Algorithm 4 was used in all the experiments reported in this work.

2.2.6 Related Work

Our motivation for using higher-order co-occurrences between features stems from advances in the areas of link mining [28] and information retrieval. In addition to (or sometimes instead of) using the more traditional data representation by feature vectors characterizing each data instance independently of the others, link-based approaches

Algorithm 4: Count Second-Order Paths

Input: Boolean $m \times n$ matrix $X = ||x_L^i||$
Output: For each feature $i = 1, \ldots, n$, output the number of second-order paths that include feature i
Output: The total number of second-order paths in X

Initialize vector $p = \left(p^1, \ldots, p^n\right)^T$ of per-feature path counts to be a zero vector
$p^i := 0$, $i = 1, \ldots, n$

Initialize scalar t, which will store the total number of second-order paths
$t := 0$

Compute vector $l = \left(l^1, \ldots, l^m\right)^T$ of ℓ_1 norms of object (row) vectors of X
$l^L = \sum_{i=1}^{n} x_L^i$, $L = 1, \ldots, m$

Compute vector $r = \left(r^1, \ldots, r^m\right)^T$ of numbers of pairs of non-zero features within each data instance with one feature removed
$r^L = \frac{1}{2}(l^L - 1)(l^L - 2)$, $L = 1, \ldots, m$

Compute the first-order feature co-occurrence matrix $A = ||a_{ij}||$
$A := X^T X$

Set all diagonal elements a_{ii} of A to zero
$\mathrm{diag}(A) := 0$

Compute vector $c = \left(c^1, \ldots, c^n\right)^T$ of squared column sums of A
$c^i = \left(\sum_{j=1}^{n} a_{ji}\right)^2$, $i = 1, \ldots, n$

Compute the second-order feature co-occurrence matrix $A^2 = ||a_{ij}^{(2)}||$
$A^2 := AA$

for $i = 1, \ldots, n$ **do**
    if $i < n$ **then**
        for $j = i + 1, \ldots, n$ **do**
            Compute the number s of paths (i, L, k, L, j), $k \in \{1, \ldots, n\} \setminus \{i, j\}$, where feature i (j) is an end vertex and where both object vertices are the same
            $s = \sum_{L=1}^{m} x_L^i x_L^j \left(l^L - 2\right)$
            $p^i := p^i + a_{ij}^{(2)}$ - s
            $p^j := p^j + a_{ij}^{(2)}$ - s
            $t := t + a_{ij}^{(2)}$ - s
        end
    end
    Add to p^i the number of paths (k, L, i, M, j), $k, j \neq i$, $k \neq j$, $L \neq M$, where feature i is the middle vertex
    Let $b = \left(b^1, \ldots, b^n\right)^T$ be a vector of cumulative sums of elements of the i-th column $a_{\cdot i}$ of matrix A, i.e., $b^h = \sum_{g=1}^{h} a_{gi}$, $h = 1, \ldots, n$
    $p^i := p^i + c^i - a_{\cdot i}^T b - r^T x^i$
end
return (p, t)

[45, 50, 60] to collective classification leverage explicit dependencies, or links, within networked data [50]. Several studies [12, 33, 60] have shown that collective classification can achieve significant reductions in classification errors by performing inferences about multiple data instances simultaneously. However, such methods are context-dependent and are therefore not designed to classify single data instances. This restriction significantly limits the domain of applicability of link-based classifiers.

In contrast with link mining approaches, the proposed transformations SSOT and USOT leverage higher-order dependencies in the form of implicit links between features. Unlike collective classifiers, methods presented in this work maintain the ability to transform single data instances without requiring any additional context information. In case of SSOT, parameters of the transformation are estimated using the training data and then used to map each individual test instance to the SSOT feature space. Similarly, parameters of the USOT are estimated using a given dataset and can then be used to map any additional data instances as they become available.

Another motivation for our work originates from the results of [39], who gave a mathematical proof supported by empirical results of the dependence of Latent Semantic Indexing (LSI) [18], a technique often used in text mining and information retrieval, on higher-order term co-occurrences. Specifically, [39] showed that two terms will have a non-zero value in the LSI term co-occurrence matrix if and only if there exists at least one co-occurrence (be it of first- or higher-order) between these terms.

Higher-order relations play an important role in many other systems for text mining, information retrieval and network analysis. In [43], higher-order associations between entities (i.e., record-value pairs) in distributed databases were used for identification of records to be consolidated at a single site and subsequently mined for association rules. Higher-order term co-occurrences in lexical networks were used in [21] for solving a component of the problem of lexical choice, which identifies most typical synonyms in a given context. In another effort, [67] used second-order co-occurrences for extracting

a potentially relevant subset of documents to be processed by LSI, thus improving its runtime performance. Higher-order co-occurrences have also been used in other applications including word sense disambiguation [57] and stemming [66].

The famous web search ranking algorithm HITS [38] also exploits higher-order associations between sites in the World Wide Web. Given a query, HITS first extracts all web pages that contain the query terms. This simplistic retrieval procedure typically misses a large number of highly authoritative sources on the subject. As was noted in [38], authoritative pages often do not contain the exact query terms. However, there exist a number of "hub" sites that contain the query terms and actively link to the authoritative sources. In order to provide high-quality ranking of search results, HITS identifies hub and authoritative sources relevant to the query by expanding the set of potentially relevant sites to include those that either link to, or are linked from the pages containing the query terms.

In the context of social network analysis, [36] proposed a social status index based on higher-order paths between individuals casting votes for each other's popularity. Similarly to the feature co-occurrence matrix considered in this work, [36] considered a square choice matrix $C = ||c_{ij}||$ whose rows and columns correspond to individuals in a group. The ij-th element c_{ij} of the choice matrix C equals one if individual i voted for individual j. However, unlike the feature co-occurrence matrix, the choice matrix is not necessarily symmetric (i may vote for j, but j may not vote for i) and therefore encodes a directed graph. The standing of individual i as defined by [36] is proportional to the total number of paths terminating in i. The contribution of a path to the standing index decreases exponentially with increasing length, i.e., order, of the path. In essence, the standing index [36] takes into account not only the individual (zero-order) vote counts for an individual i, but also the vote counts of individuals that voted for i and of individuals that voted for those who voted for i, etc.

In a more recent effort, a supervised collective classification method termed Higher

Order Path Analysis (HOPA) was proposed in [25]. Unlike discriminative classifiers such as Support Vector Machine [64], HOPA constructs a separate model for each class independently of the others. Given a set of training data instances of one class, HOPA collects statistics on the distribution of counts of same-frequency third-order sub-paths extracted from the fourth-order paths present in the data. No assumption is made regarding the type of this distribution; its empirical estimate is used as the class model. Given a test set, HOPA extracts the counts of same-frequency third-order sub-paths and compares the distribution of these counts with the class model using the t-test. If, given a user-specified confidence level, the means of the two distributions are found to be significantly different, all the instances in the test set are classified as not belonging to the modeled class.

There are several crucial conceptual differences between HOPA and the transformations proposed in this work. First, the class model constructed by HOPA is based on aggregate frequencies of higher-order paths. The aggregation of path statistics makes it impossible to interpret HOPA's class model in terms of the original features. SSOT and USOT, on the other hand, maintain a one-to-one correspondence between dimensions of the original space and the higher-order feature space, thus retaining the interpretability. Furthermore, unlike SSOT and USOT, HOPA is context-dependent since it can only classify sets of instances, but not the individual instances independently of each other. Finally, HOPA is a classifier in and of itself and does not permit a straightforward integration with other learning methods as do SSOT and USOT. Nevertheless, HOPA was able to detect and classify anomalous events in the Border Gateway Protocol and further confirmed the value of using higher-order paths for pattern classification.

Another higher-order classifier termed Higher Order Naive Bayes (HONB) was proposed in [24]. HONB extends the Naive Bayes (NB) classifier for binomial data by estimating conditional feature probabilities over the space of higher-order paths (2.24) instead of individual data instances (2.23). HONB drastically outperformed NB on

a series of text classification problems, especially when training samples were small [24, 26]. SSOT also makes use of the higher-order conditional probabilities (2.24) and further generalizes HONB by allowing any classifier operating in vector spaces to take advantage of the higher-order co-occurrence relations.

Chapter 3

Experimental Results

3.1 Comparative Study of the K-means and Neyman's Clustering Criteria on Simulated Data

As was established in Section 2.1.3, criteria (2.1) and (2.2) produce identical clusterings when cluster variances are equal. Therefore, studying the behavior of these criteria under the conditions of unequal cluster variances is of most value. In this section, we present results on simulated data generated by two five-dimensional Gaussian distributions $\mathcal{N}(\mu_1, \Sigma_1)$ and $\mathcal{N}(\mu_2, \Sigma_2)$, which we will refer to as classes. Each class contained fifteen thousand points. The covariance matrices Σ_1 and Σ_2 were diagonal with varying standard deviations along each dimension. The standard deviations are shown in Table 3.1. Mean μ_1 of the first class was held fixed at the origin, while mean of the other

Table 3.1: Standard deviations of the data generator

Dimension	Class 1	Class 2
1	1.2	6
2	1.4	7
3	1.6	8
4	1.8	9
5	2	10

class was $\mu_2 = t\mathbf{1}$, where $\mathbf{1}$ is a five-dimensional vector of ones and t is a real-valued parameter. Varying t allowed us to observe the behavior of clustering criteria (2.1) and (2.2) as a function of distance between mean vectors of the two classes. For each value of t, the corresponding dataset was clustered into two clusters by each of the

criteria. The results are shown in confusion Tables 3.2–3.6 and are summarized by the class reconstruction accuracies plotted in Figure 3.1. These accuracies were obtained by solving an optimal assignment problem [1] over each confusion table. The cost of assigning cluster α to class i was the number of points from class i that were placed into cluster α. An optimal assignment of clusters to classes was that which maximized the total cost over all clusters and assigned each cluster to exactly one class to which no other cluster was assigned. Clustering results visualized in several two-dimensional subspaces of the five-dimensional space are shown in Figure 3.2.

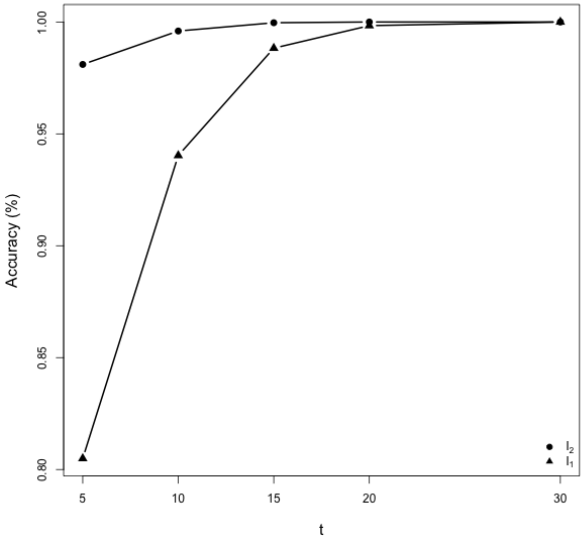


Figure 3.1: Class reconstruction accuracies

Accuracies reported in Figure 3.1 demonstrate that as the means of the two classes came closer together with decreasing t, K-means criterion (2.1) "misclassified" an increasing number of points from the larger-variance class two. This degradation in performance of criterion (2.1) is a direct consequence of its inherent disregard for cluster variances. In contrast, criterion (2.2) was able to accommodate the discrepancy in cluster variances and much more accurately reflected the underlying data structure as made evident by Figures 3.1 and 3.2 and by the confusion Tables 3.2–3.6.

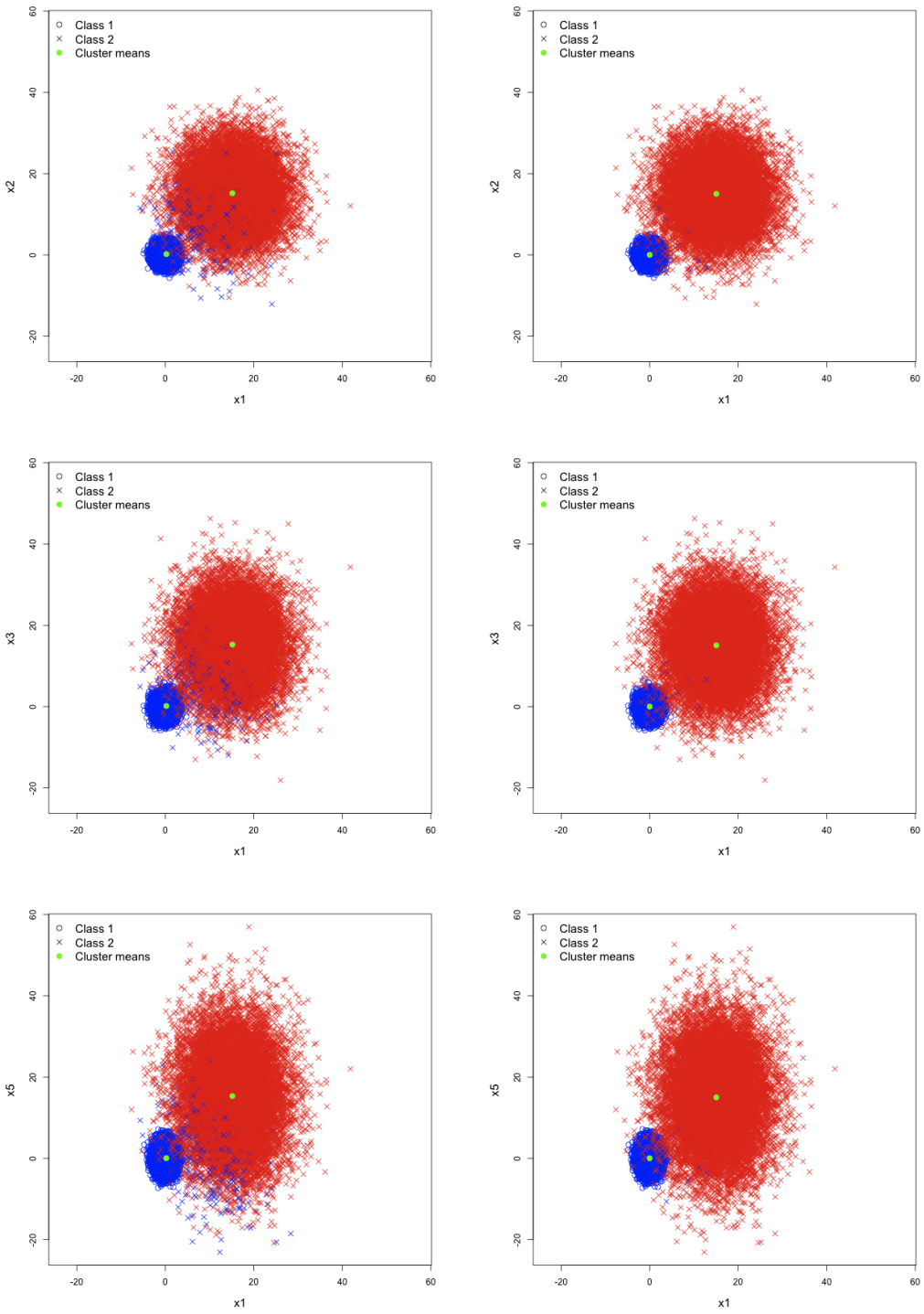


Figure 3.2: Clusterings obtained by criteria (2.1) (left) and (2.2) (right), and visualized in several two-dimensional subspaces of the five-dimensional space. The value of parameter t that resulted in the dataset shown was fifteen.

Table 3.2: Confusion tables for $t = 5$

	I_1		I_2	
Class \ Cluster	1	2	1	2
1	0	15000	0	15000
2	9148	5852	14431	569

Table 3.3: Confusion tables for $t = 10$

	I_1		I_2	
Class \ Cluster	1	2	1	2
1	0	15000	0	15000
2	13210	1790	14879	121

Table 3.4: Confusion tables for $t = 15$

	I_1		I_2	
Class \ Cluster	1	2	1	2
1	0	15000	0	15000
2	14649	351	14990	10

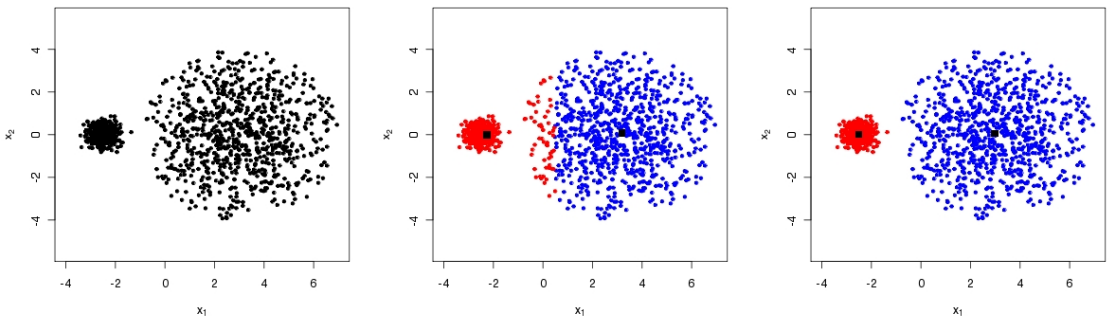
Table 3.5: Confusion tables for $t = 20$

	I_1		I_2	
Class \ Cluster	1	2	1	2
1	0	15000	0	15000
2	14951	49	15000	0

Table 3.6: Confusion tables for $t = 30$

	I_1		I_2	
Class \ Cluster	1	2	1	2
1	0	15000	0	15000
2	15000	0	15000	0

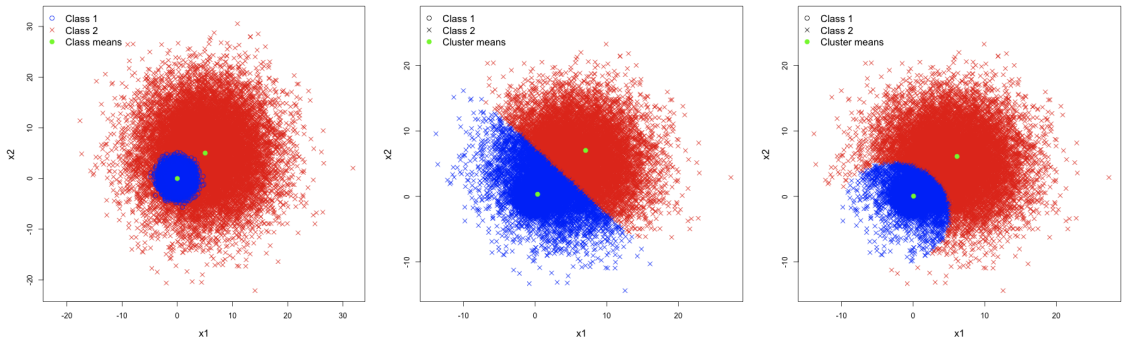
Two particular aspects of criterion (2.2) allowed it to accurately recover the generated class structure for each value of t. In cases where class means were close, but the data points from the two classes did not overlap as exemplified in Figure 3.3, the shift in the discriminant surface (2.17) away from the larger-variance cluster was enough to allow criterion (2.2) to reconstruct the class structure with high accuracy. As class means moved closer together, nonlinearity of criterion (2.2) started to play an increasingly important role. In case of an extreme overlap where points from the lower-variance class became absorbed by the cloud of points from the larger-variance class, as illustrated by the two-dimensional dataset shown in Figure 3.4, nonlinearity of criterion (2.2) became crucial for maintaining the high accuracies shown in Figure 3.1. Note the dramatic 18% decrease in accuracy suffered by the K-means criterion (2.1) once t decreased below fifteen causing points of class one to become engulfed by the points of class two. As a result, criterion (2.1) misclassified an increasingly larger number of points as can be seen from Tables 3.2 and 3.3. Meanwhile, criterion (2.2) was able to maintain the roughly 98% accuracy.



(a) Input dataset (b) Clustering by criterion (2.1) (c) Clustering by criterion (2.2)

Figure 3.3: The shift in the discriminant surface (2.17) away from the larger-variance cluster allowed criterion (2.2) to reconstruct the underlying class structure with high accuracy

Another reason for the high accuracies attained by criterion (2.2) despite the extreme overlap between points of the two classes lies in its ability to discover clusters

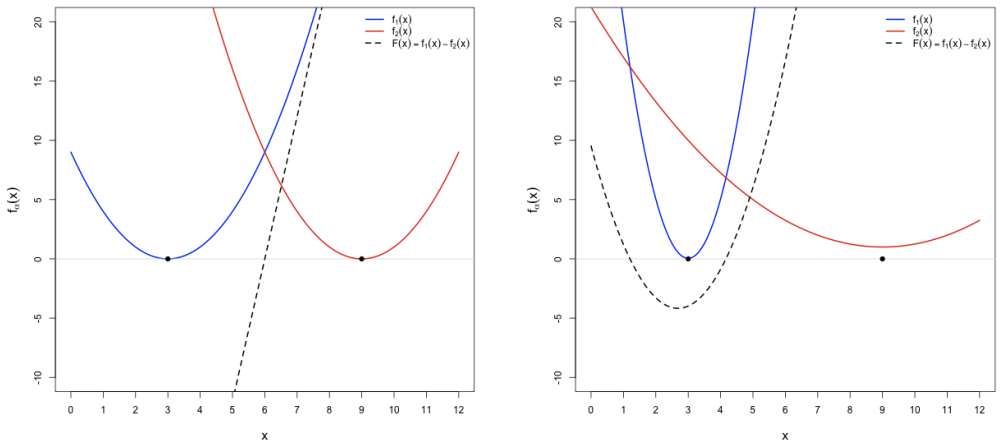


(a) Ground truth (b) Clustering by criterion (2.1) (c) Clustering by criterion (2.2)

Figure 3.4: Clustering under the conditions of an extreme overlap between points of the underlying classes. Points of class one (marked by circles) are completely covered by the points of class two (marked by crosses).

within clusters. In particular, clusterings produced by criterion (2.2) are not Voronoi tessellations as is the case with criterion (2.1). This aspect is best illustrated with a one-dimensional example shown in Figure 3.5. Suppose there are two clusters with means $\mathcal{M}_1^{(1)} = 3$ and $\mathcal{M}_2^{(1)} = 9$ and standard deviations $\sigma_1 = 0.1$ and $\sigma_2 = 2$. By plotting the discriminant function (2.15) for each of the clusters, in Figure 3.5b we can see that there are two intervals $(-\infty, 1.203)$ and $(4.166, +\infty)$ on the x axis where points would be assigned to the larger variance cluster two. These are the intervals where the distance from a point to cluster two is smaller than to cluster one, i.e. where function $F(x) = f_1(x) - f_2(x)$, also shown in Figure 3.5b, takes on positive values. Cluster one, in this case, covers the interval $[1.203, 4.168]$. In comparison, the K-means criterion (2.1) would partition the data axis x half way between the two cluster means into two consecutive intervals $(-\infty, 6]$ and $(6, +\infty)$ corresponding to clusters one and two, respectively, as shown in Figure 3.5a.

Sensitivity of criterion (2.2) to cluster variance makes this criterion a promising alternative to the K-means criterion (2.1), when used for initialization of the Expectation Maximization (EM) procedure for reconstruction of hidden mixtures of Gaussians [48]. EM is known for its sensitivity to initial conditions. Hence, successful reconstruction



(a) Criterion (2.1) (b) Criterion (2.2)

Figure 3.5: Solid lines show discriminant functions (2.16) (left) and (2.15) (right) for two clusters with means $\mathcal{M}_1^{(1)} = 3$ and $\mathcal{M}_2^{(1)} = 9$ (marked by black circles) and standard deviations $\sigma_1 = 0.1$ and $\sigma_2 = 2$. Dashed lines show the difference $F(x) = f_1(x) - f_2(x)$ between the discriminant functions.

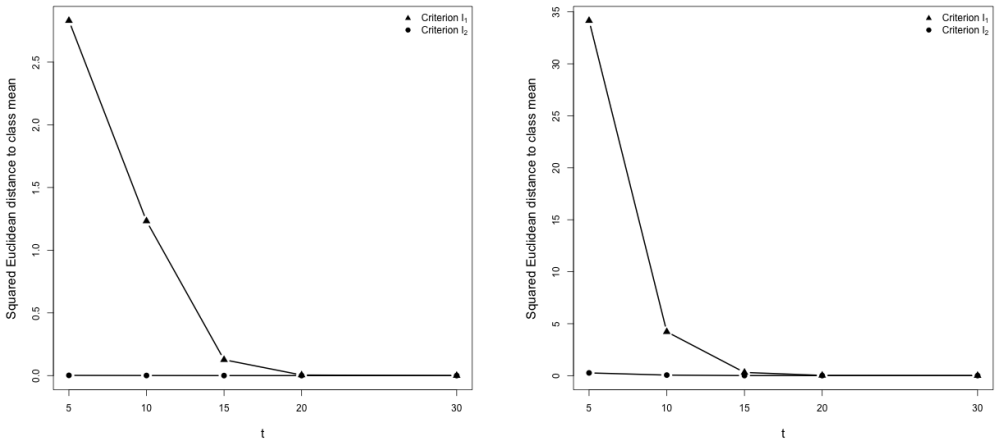
by the Gaussian Mixture Model (GMM) of the underlying data structure relies heavily on an adequate initialization of the EM algorithm. Figure 3.6 shows that criterion (2.2) more accurately reconstructed locations of the class means than did criterion (2.1). In addition, criterion (2.2) also reconstructed class covariance matrices with higher accuracy. The error in reconstruction of class covariance matrices can be measured by a non-negative discrepancy score

$$\left(\frac{\tilde{\sigma}_i}{\sigma_i} - 1\right)^2, \tag{3.1}$$

where σ_i is the standard deviation of the generator model along the i-th dimension (see Table 3.1) and $\tilde{\sigma}_i$ is an empirical estimate of σ_i. If a clustering coincides with the class labeling of points (e.g. see Table 3.6), the discrepancy scores (3.1) will be close[1] to zero across all dimensions and clusters. Scores (3.1) increase with increasing deviation of the cluster structure from the class labeling (e.g. see clustering by criterion (2.1) in Table

[1]When cluster structure coincides with the class labeling, scores (3.1) would not be exactly zero, because the data points being clustered are only a sample generated by the theoretical model. See Tables 3.6 and 3.11 for example.

3.2). Tables 3.7–3.11 show the discrepancy (3.1) for each criterion, dimension, class and across a range of values of t. These tables demonstrate that discrepancy scores for criterion (2.2) were consistently smaller than for criterion (2.1) until the point ($t = 30$) where mean vectors of the two classes became so far apart that both clustering criteria produced identical partitions of the data points. The difference between criteria (2.1) and (2.2) was especially apparent when points of the two classes overlapped strongly ($t < 15$). Discrepancy scores for criterion (2.2), in such cases, were several orders of magnitude smaller than for criterion (2.1). These results demonstrate that criterion (2.2) was able to recover the underlying class means and covariance matrices with higher accuracy than criterion (2.1). Although real-world data typically have more complex structure than was simulated here, the obtained experimental results suggest that criterion (2.2) is a potentially superior alternative to criterion (2.1) for initialization of the EM algorithm for GMM. We intend to investigate this further as a part of our future work.



(a) Class one (b) Class two

Figure 3.6: Squared Euclidean distance from a cluster mean to the corresponding generator class mean as a function of distance between the two class means

Table 3.7: Reconstruction error of class covariance matrices for $t = 5$

	Class one				
Criterion \ Dimension	1	2	3	4	5
I_1	4.428	3.525	2.913	2.474	1.768
I_2	0.017	0.009	0.007	0.004	0.004

	Class two				
Criterion \ Dimension	1	2	3	4	5
I_1	2×10^{-4}	0.001	0.001	0.006	0.029
I_2	9×10^{-6}	5×10^{-6}	7×10^{-6}	5×10^{-6}	8×10^{-5}

Table 3.8: Reconstruction error of class covariance matrices for $t = 10$

	Class one				
Criterion \ Dimension	1	2	3	4	5
I_1	2.504	1.448	0.942	0.579	0.465
I_2	0.003	0.002	0.001	10^{-4}	4×10^{-4}

	Class two				
Criterion \ Dimension	1	2	3	4	5
I_1	3×10^{-4}	0.001	0.001	0.002	0.004
I_2	10^{-4}	8×10^{-5}	10^{-4}	6×10^{-5}	4×10^{-6}

Table 3.9: Reconstruction error of class covariance matrices for $t = 15$

	Class one				
Criterion \ Dimension	1	2	3	4	5
I_1	0.62	0.267	0.116	0.046	0.033
I_2	2×10^{-4}	10^{-4}	2×10^{-5}	3×10^{-5}	7×10^{-6}

	Class two				
Criterion \ Dimension	1	2	3	4	5
I_1	2×10^{-4}	2×10^{-4}	4×10^{-4}	4×10^{-4}	4×10^{-4}
I_2	4×10^{-5}	6×10^{-5}	10^{-4}	5×10^{-5}	2×10^{-6}

Table 3.10: Reconstruction error of class covariance matrices for $t = 20$

	Class one				
Criterion \ Dimension	1	2	3	4	5
I_1	0.056	0.024	0.006	9×10^{-4}	7×10^{-4}
I_2	10^{-7}	3×10^{-6}	10^{-7}	4×10^{-5}	6×10^{-8}

	Class two				
Criterion \ Dimension	1	2	3	4	5
I_1	5×10^{-5}	8×10^{-5}	10^{-4}	9×10^{-5}	3×10^{-5}
I_2	4×10^{-5}	4×10^{-5}	9×10^{-5}	4×10^{-5}	5×10^{-7}

Table 3.11: Reconstruction error of class covariance matrices for $t = 30$

	Class one				
Criterion \ Dimension	1	2	3	4	5
I_1	10^{-7}	3×10^{-6}	10^{-7}	4×10^{-5}	6×10^{-8}
I_2	10^{-7}	3×10^{-6}	10^{-7}	4×10^{-5}	6×10^{-8}

	Class two				
Criterion \ Dimension	1	2	3	4	5
I_1	4×10^{-5}	4×10^{-5}	9×10^{-5}	4×10^{-5}	5×10^{-7}
I_2	4×10^{-5}	4×10^{-5}	9×10^{-5}	4×10^{-5}	5×10^{-7}

3.2 Estimation of the Mean Vector of Multi-dimensional Data by Stratified Sampling

The problem of estimation of the mean value of a scalar variable has been extensively studied [9, 10, 51, 37] by researchers in the statistical community. Given a random variable $\mathcal{X}$, an unbiased estimator $\bar{x}$ of the expectation $E(\mathcal{X})$ is

$$\bar{x} = \frac{1}{k}\sum_{x \in X} x, \tag{3.2}$$

where X is an independent and identically distributed sample of size k. The variance $D(\bar{x})$ of estimator (3.2) is

$$D(\bar{x}) = \frac{1}{k}\sigma^2, \tag{3.3}$$

where σ^2 is the variance of $\mathcal{X}$. Suppose a complete partition $H = (h_1(x), \ldots, h_K(x))$ into K disjoint intervals of the range of values of variable $\mathcal{X}$ is also given. As before, $h_\alpha(x)$ denotes the characteristic function of the α-th interval. The estimator (3.2) can be rewritten as

$$\bar{x} = \sum_{\alpha=1}^{K} p_\alpha \bar{x}_\alpha, \tag{3.4}$$

where p_α is the probability of interval α, $\bar{x}_\alpha = \frac{1}{k_\alpha}\sum\limits_{x \in X} x h_\alpha(x)$ is the estimator of the mean value of variable $\mathcal{X}$ in the interval α and k_α is the number of sample points from that interval. Since the exact knowledge of the probability p_α is typically not available, it is estimated from the sample X by $p_\alpha = \frac{k_\alpha}{k}$.

In 1926, Bowley [9] proposed the following sampling scheme. Given a partition H and a sample size k, randomly sample from each interval α

$$k_\alpha = p_\alpha k \tag{3.5}$$

number of points and apply estimator (3.4). Under this sampling scheme, variance $D_{\mathrm{B}}(\bar{x})$ of the estimator (3.4) becomes

$$D_{\mathrm{B}}(\bar{x}) = \frac{1}{k}\sum_{\alpha=1}^{K} p_\alpha \sigma_\alpha^2, \tag{3.6}$$

where σ_α^2 is the variance of $\mathcal{X}$ in the interval α. Therefore, (3.6) is minimized when the range of values of variable $\mathcal{X}$ is partitioned by criterion (2.1). Since (3.6) is always smaller than (3.3), Bowley's sampling scheme is more efficient than the simple random sample (3.2) [9, 10].

Bowley's scheme was further refined by Neyman who found [51] that the optimal number of points to be sampled from each interval α while minimizing the variance of estimator (3.4) is

$$k_\alpha = \frac{p_\alpha \sigma_\alpha}{\sum_{\beta=1}^{K} p_\beta \sigma_\beta} k. \tag{3.7}$$

In this case, variance $D_{\mathrm{N}}(\bar{x})$ of estimator (3.4) becomes

$$D_{\mathrm{N}}(\bar{x}) = \frac{1}{k} \left(\sum_{\alpha=1}^{K} p_\alpha \sigma_\alpha \right)^2. \tag{3.8}$$

It follows that partitioning the range of $\mathcal{X}$ so as to minimize criterion (2.2), minimizes (3.8). Theoretically, Neyman's sampling is more efficient than Bowley's, because (3.8) is no greater than (3.6) [10, 51]. We thus have

$$D_{\mathrm{N}}(\bar{x}) \leq D_{\mathrm{B}}(\bar{x}) \leq D(\bar{x}).$$

In this section, we test whether Neyman's sampling scheme is more efficient when the data is sampled from a multi-dimensional space. The data generator used in the experiments consisted of three five-dimensional Gaussian distributions with diagonal covariance matrices $\Sigma_1 = 100I$, $\Sigma_2 = 25I$ and $\Sigma_3 = I$. Means of these distributions are shown in Table 3.12. A sample X of m points representing the entire population to be sampled from was generated and clustered by criteria (2.1) and (2.2). Then, a number $k \ll m$ of points were sampled N times according to Bowley and Neyman's schemes (3.5) and (3.7), respectively. Using the estimator (3.4), for each sample of k points we computed the estimate $\bar{x}_{\mathrm{s}}$ of the grand mean vector $\bar{x} = \frac{1}{m} \sum_{x \in X} x$. Finally, we evaluated the efficiency of the sampling schemes by the squared Euclidean distance between $\bar{x}_{\mathrm{s}}$ and $\bar{x}$, averaged over the N trials. The resulting mean squared distances are shown in

Figure 3.7 for varying sample size k. In order to evaluate the sampling schemes under the conditions of varying class densities, we varied the number of points generated by each of the three Gaussian distributions.

Table 3.12: Means of the distributions comprising the data generator

Distribution	Mean vector
1	$(-20, 30, 0, 0, 0)^T$
2	$(15, 10, 0, 0, 0)^T$
3	$(0, 0, 0, 0, 0)^T$

As can be seen from Figure 3.7, Neyman's sampling scheme (3.7) consistently outperformed Bowley's scheme (3.5) across the range of sample sizes and class densities. In all cases, both schemes performed better than the simple random sample (3.2), particularly when the sample size k was small (e.g. $k = 20, m = 3000$). All differences were statistically significant at the 5% level. The number of samples drawn for each value of k was $N = 200$. Robust performance of sampling schemes (3.5) and (3.7) is especially encouraging since financial and temporal constraints often prohibit the acquisition of larger samples in most real-world applications.

The problem of efficient estimation of the mean value of a scalar variable was extensively studied in prior work [9, 10, 51]. Experimental results reported in this section demonstrate that generalization of criterion (2.2) to multi-dimensional spaces and development of the associated clustering algorithm in Section 2.1 allowed for efficient estimation of the mean vector. The results also suggest that aside from estimation of the mean vector of multi-dimensional data, criterion (2.2) would be particularly useful for construction of training samples for supervised machine learning (classification and regression). Small training samples that accurately capture the underlying distribution of the data are crucial for practical applications of computationally demanding methods such as Support Vector Machine [63] on massive real-world datasets comprised of millions of data instances. Low computational complexity of the proposed clustering

algorithm for criterion (2.2) opens the possibility of construction of small training samples from large-scale datasets. Further investigation of this topic constitutes one of the future research directions stemming from this work.

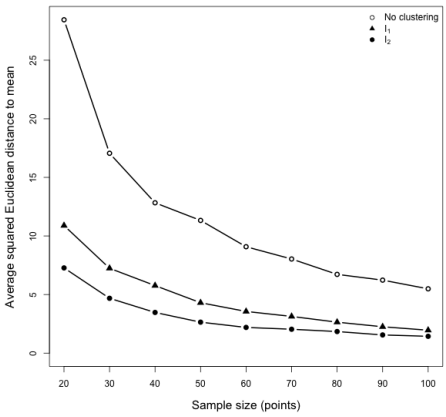
3.3 Supervised Second Order Transformation in Text Classification

Experimental evaluation of the Supervised Second Order Transformation (SSOT) describer in Section 2.2.3 was carried out on three widely-used text corpora: RELIGION, SCIENCE and POLITICS subsets of the 20 News Groups (20NG) [41] benchmark data. Our preprocessing procedures closely followed those commonly used in the literature [59, 35]. First, all cross-postings in the 20NG data were removed. Then, for each dataset we performed stop word removal, stemming and removal of all terms that occurred in fewer than three documents in the dataset. The remaining terms were ranked by Information Gain. The top 2000 terms were selected. Finally, 500 documents were sampled at random from each class to comprise the 20NG datasets used in our experiments. A summary description of the datasets is provided in Table 3.13.

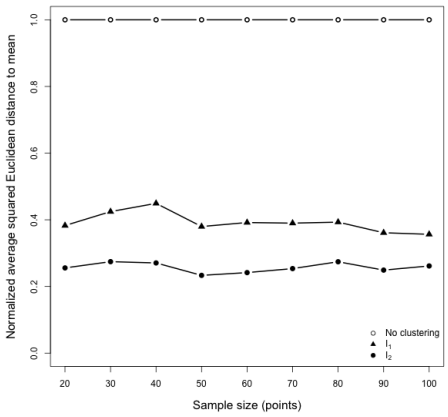
Table 3.13: Datasets used in the experiments

Dataset	Classes
RELIGION (3)	alt.atheism, soc.religion.christian, talk.religion.misc
SCIENCE (4)	sci.crypt, sci.electronics, sci.med, sci.space
POLITICS (3)	talk.politics.guns, talk.politics.mideast, talk.politics.misc

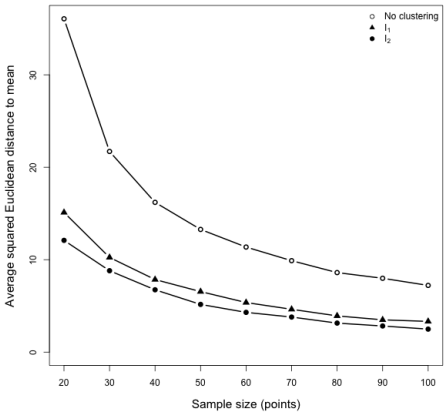
Support Vector Machine (SVM) [63] was chosen as the base classifier for evaluation of the SSOT. SVM with the linear kernel has been shown [35] to perform well on text classification problems. The linear kernel allowed us to observe directly the impact of leveraging higher-order co-occurrences, without any additional data transformations as performed implicitly by other kernel functions. Multi-class problems were addressed using the "one-against-one" classification scheme [40]. Under this scheme, a binary SVM classifier is constructed for every pair of classes. A data instance is then classified by



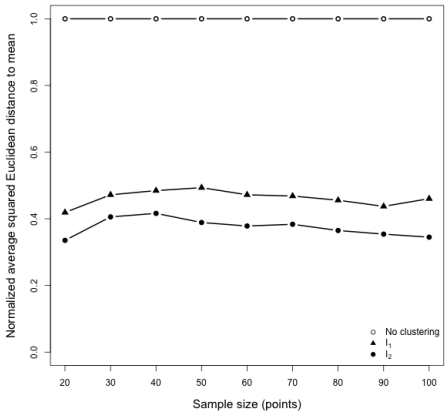
(a) $m_1 = 1000, m_2 = 1000, m_3 = 1000$



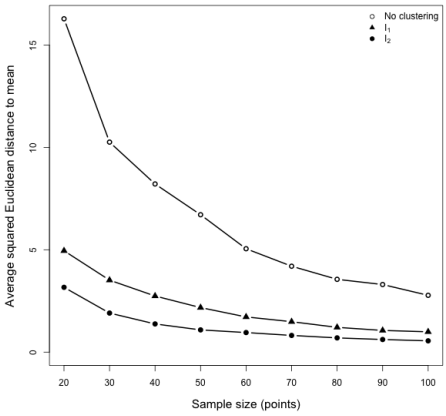
(b) $m_1 = 1000, m_2 = 1000, m_3 = 1000$



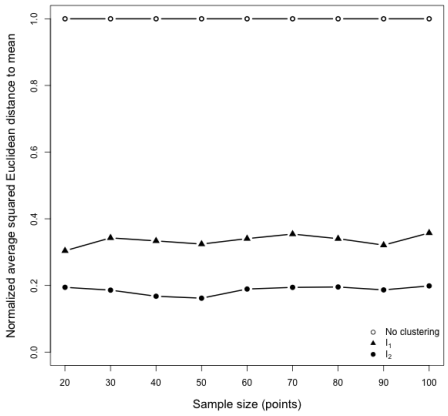
(c) $m_1 = 1000, m_2 = 500, m_3 = 250$



(d) $m_1 = 1000, m_2 = 500, m_3 = 250$



(e) $m_1 = 250, m_2 = 500, m_3 = 1000$



(f) $m_1 = 250, m_2 = 500, m_3 = 1000$

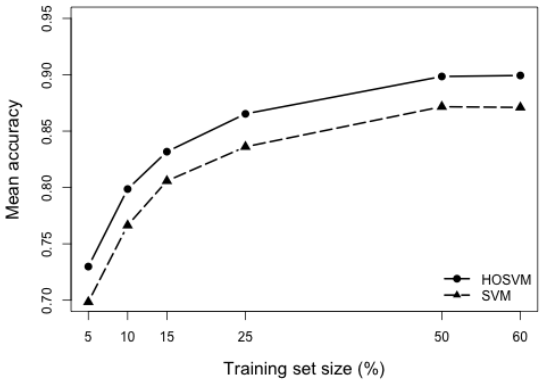
Figure 3.7: Average squared Euclidean distance between the grand mean $\bar{x}$ and the mean $\bar{x}_s$ obtained by different sampling schemes. Plots in the left column show the absolute values, while plots in the right column show values normalized by the average distance attained by the simple random sample (3.2). The number m_α of points generated by each Gaussian distribution is also shown.

each binary classifier and the final classification is determined by the majority vote over the assigned class labels. We refer to a SVM classifier constructed on the transformed data as Higher Order SVM (HOSVM).

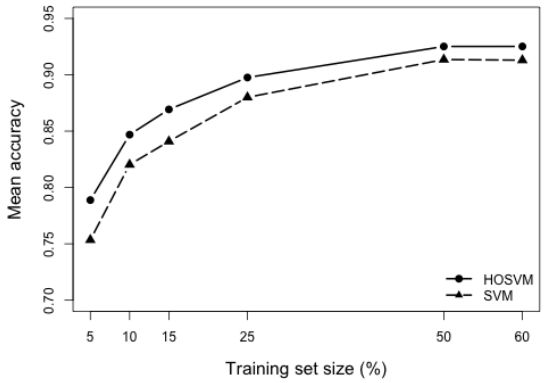
Figure 3.8 shows mean classification accuracies obtained by varying training set size from 5% (25 documents per class) up to 60%. For each training set size, eight trials were performed. On each trial, a set of documents were randomly sampled from each class for training, while the rest were used for testing. On every trial, all terms that did not appear in any of the training documents were disregarded. The classifiers were then trained in the corresponding subspace of the original term space. When selecting the value of the soft margin cost parameter C for SVM, we considered the set $\{10^{-4}, 10^{-3}, \ldots, 10^{4}\}$ of possible values. On every trial, we picked the smallest value of C which resulted in the highest accuracy obtained on the training set.

As can be seen from Figures 3.8a–3.8c, HOSVM consistently outperformed SVM across varying training set sizes and datasets. All accuracy improvements were statistically significant at the 5% level. Consistent and statistically significant accuracy improvements attained by HOSVM even on small training sets led us to explore this aspect further. In order to simulate a real-world scenario where only a few labeled data instances are available, we focused our attention on 5% training samples. This corresponded to training on 25 documents per class and testing on the other 475 documents per class. Classification accuracies averaged over eight trials are reported in Table 3.14. Highest accuracies attained on each dataset are highlighted in bold. The corresponding standard deviations are also reported in Table 3.14.

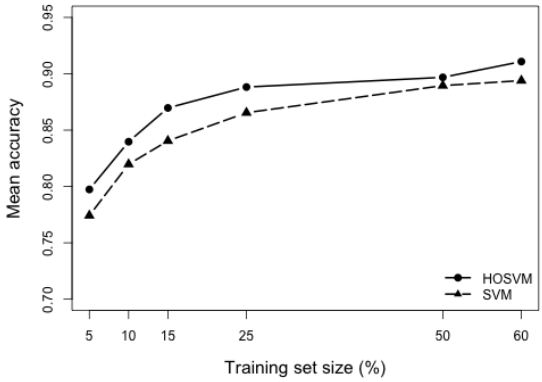
The obtained results indicate that leveraging higher-order co-occurrences lead to significant improvements in classification accuracies. HOSVM consistently outperformed SVM by an average of 3.1%. The improvements of HOSVM over SVM were statistically significant at the 5% level on all but one dataset. Although the difference in SVM and HOSVM accuracies on the POLITICS dataset was significant at level $\alpha = 0.158$,



(a) RELIGION



(b) SCIENCE



(c) POLITICS

Figure 3.8: Scalability across training set size

Table 3.14: Mean classification accuracies

	SVM		HOSVM	
Dataset	Acc.	St. dev.	Acc.	St. dev.
RELIGION (3)	0.699	0.022	**0.723**	0.023
SCIENCE (4)	0.751	0.029	**0.792**	0.039
POLITICS (3)	0.763	0.03	**0.793**	0.047

HOSVM outperformed SVM on seven out of eight trials on that data by an average of 3%.

In order to further verify the value of leveraging higher-order co-occurrences within the data, additional experiments were conducted. In these experiments, prior to training an SVM classifier with the linear kernel, transformation (2.27) was performed using the zero-order conditional probabilities (2.23) instead of the higher-order probabilities (2.24). The resulting approach is referred to as ZOSVM. Mean classification accuracies attained by ZOSVM are shown in Table 3.15. Comparison of Tables 3.14 and 3.15 makes it clear that ZOSVM performed worse than both SVM and HOSVM. These results indicate that taking advantage of higher-order co-occurrences was indeed crucial for achieving the performance improvements attained by HOSVM.

Table 3.15: Mean classification accuracies of ZOSVM

Dataset	ZOSVM	HOSVM
RELIGION (3)	0.678	**0.723**
SCIENCE (4)	0.745	**0.792**
POLITICS (3)	0.759	**0.793**

We have also conducted experiments with the Radial Basis Function (RBF) kernel for the HOSVM and SVM classifiers. The results were consistent with the findings of [35]. Namely, there were no significant differences between classification accuracies attained with the linear kernel and those attained with the RBF kernel.

3.4 Clustering Text Documents

In this chapter we evaluate the efficacy of the Unsupervised Second Order Transformation (USOT) proposed in Section 2.2.4. We use datasets with known class labels and cluster the data with criteria (2.1) and (2.2). If USOT was successful at emphasizing the specifics of the various homogenous subgroups of data instances and at increasing

separability between those subgroups, we would expect both clustering methods to reproduce the known classification with higher accuracy in the USOT space than in the original boolean space.

Experimental evaluation was carried out on four benchmark text corpora. Three of these datasets were the RELIGION, SCIENCE and POLITICS subsets of the 20 News Groups (20NG) [41] benchmark data with all cross postings and stop words removed and all other words stemmed. To keep the computation manageable, 500 documents were sampled at random from each class to comprise the 20NG datasets used in our experiments. The other dataset, BBC [30], contained 2225 news stories from the British Broadcasting Corporation (BBC). Each news story belonged to one of five classes: business, entertainment, politics, sport, or tech. The BBC dataset was preprocessed by the authors of [30] who removed stop words and stemmed the remaining words. For each dataset, we selected those terms whose minimum-frequency value covered at least five percent of data instances in the dataset. Other terms would have low variability and would therefore be largely ignored by the clustering process. A summary description of the datasets is provided in Table 3.16.

Table 3.16: Four datasets used in the experiments

Dataset	Classes	Dimensionality
RELIGION (3)	alt.atheism, soc.religion.christian, talk.religion.misc	429
SCIENCE (4)	sci.crypt, sci.electronics, sci.med, sci.space	505
POLITICS (3)	talk.politics.guns, talk.politics.mideast, talk.politics.misc	290
BBC (5)	business, entertainment, politics, sport, tech	635

As before, we clustered each dataset into the same number of clusters as there are classes. Each cluster was then assigned a unique class label by the optimal assignment method. In order to assess statistical significance of results, we ran each clustering algorithm ten times ($M = 10$) in the original boolean space and in the USOT space. Each run was initialized with ten random partitions. We then computed the average

class reconstruction error rates $\bar{E}_{\text{Bool}}$ and $\bar{E}_{\text{USOT}}$, and assessed statistical significance of their difference $\bar{E}_{\text{Bool}} - \bar{E}_{\text{USOT}}$ using the t-criterion,

$$t = \frac{\bar{E}_{\text{Bool}} - \bar{E}_{\text{USOT}}}{\sqrt{\frac{s^2_{\text{Bool}}}{M} + \frac{s^2_{\text{USOT}}}{M}}},$$

where s^2_{Bool} and s^2_{USOT} are the unbiased estimates of variances of the classification errors.

Class reconstruction errors attained by clustering criteria (2.1) and (2.2) in boolean (original), UZOT and USOT spaces are reported in Table 3.17. All performance improvements attained as a result of applying USOT were statistically significant at the 5% level. Moreover, the improvements were consistent across datasets and clustering criteria, which indicates that USOT was able to increase separability between the various homogeneous subgroups of data instances. Table 3.17 also demonstrates that using only the zero-order (i.e., term frequency) information was not sufficient for increasing separability between the underlying subgroups. This is consistent with our findings published in [26] and presented in Section 3.3 on using higher-order paths for supervised pattern classification.

Table 3.17: Average classification errors

	I_1			I_2		
Data	Bool.	UZOT	USOT	Bool.	UZOT	USOT
REL(3)	0.632	0.627	**0.56**	0.641	0.634	**0.579**
POL(3)	0.642	0.643	**0.523**	0.633	0.642	**0.573**
SCI(4)	0.587	0.7	**0.545**	0.689	0.698	**0.525**
BBC(5)	0.222	0.284	**0.185**	0.297	0.291	**0.178**

3.5 Return Based Style Analysis of Mutual Funds

A fundamental element of dynamic qualitative analysis of portfolios of mutual funds is the Return Based Style Analysis (RBSA). As suggested by its name, RBSA is concerned with analysis of styles of management of mutual based on time series of their returns.

A mutual fund's management style is reflected in continuous adjustments of the fund's portfolio performed by the manager in order to balance risk and profit while, at the same time, keeping with the stated investment objectives of the fund. The information regarding the fund's management style is of great value to individual investors and financial institutions, but is generally not revealed in detail by the managers. Only a broad overall objective of a mutual fund is typically stated in the fund's prospectus. What makes the style analysis more challenging is that, according to the current U.S. regulations, mutual funds are required to report the composition of their portfolios only four times per year (quarterly). In contrast, the return on a fund's portfolio is declared daily. The pattern of returns (Figure 3.9) of an individual mutual fund is generally thought of as a stochastic process. However, we hypothesize that funds with similar portfolios and dynamics of their adjustments performed by the managers would tend to have similar patterns of behavior of the funds' returns.

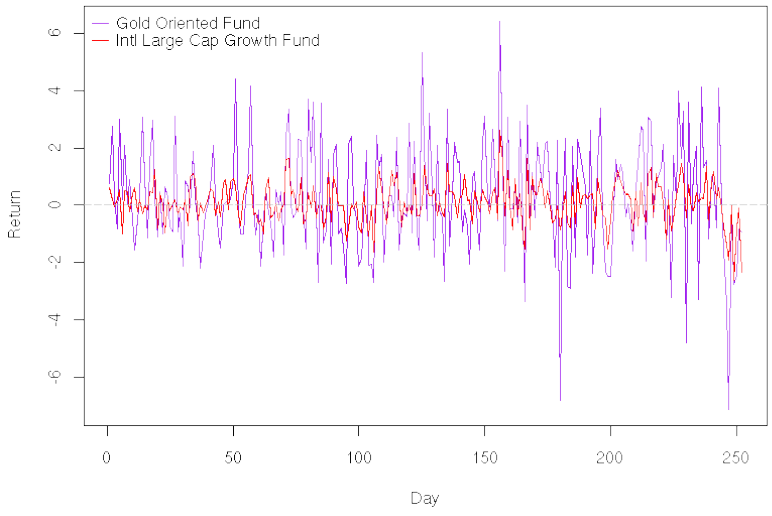


Figure 3.9: Time series of returns of two mutual funds (May 2005 – May 2006)

In this section, we present initial experimental results demonstrating that mutual funds can be grouped based on time series of their returns such that funds within a group reflect similar management styles. We follow the evaluation methodology described in

Section 3.4 and apply clustering criteria (2.1) and (2.2) in conjunction with USOT. The dataset used in the experiments consisted of time series of weekly returns of 6665 mutual funds spanning the period from May 2005 until May 2006. Return r_i^t of fund i at week t is determined as

$$r_i^t = \frac{p_i^t - p_i^{t-1}}{p_i^{t-1}},$$

where p_i^t is the closing share price of fund i at the end of week t. It is well-known that returns of mutual funds are often highly correlated with the market, a phenomenon known as the market trend. In fact, for 42% of the funds in our dataset at least 80% of the variability in returns was explained by the market's return represented by the Standard & Poor's[2] 500 (S&P500) index. S&P500 is a leading economic indicator that models the economy of the U.S. and is a value weighted index that includes 500 leading companies across major industries of the U.S. Deviations of the funds' returns from the market are caused by particular management decisions and reflect the specifics of management styles. Therefore, the market trend had to be removed from the time series of returns in order to reveal the specifics of management styles of the funds. We addressed this problem by constructing for each fund i a least squares linear regression model

$$r_i = a_i s + b_i + \varepsilon_i, \tag{3.9}$$

where s is the vector of returns of the S&P500 index. The residual vector ε_i reflects the behavior of the fund's return that is not explained by the market trend, and contains a more refined representation of the fund's management style. The residual vectors were subsequently used as the input data in our experiments.

Let $Y = ||y_L^i||$ denote a $m \times n$ real data matrix of residuals of returns. Rows of Y indexed by capital letters correspond to mutual funds, while columns indexed by small letters correspond to time points, or dimensions of the time series. To allow application of USOT, criterion (2.1) was used to convert real-valued residuals of returns

[2]http://www2.standardandpoors.com

into boolean from. The conversion was achieved by first clustering mutual funds into five[3] clusters along each dimension i independently using a dynamic programming algorithm [7] for criterion (2.1). As was noted earlier, this dynamic programming algorithm gives a globally optimal clustering of one-dimensional data. Let such clustering along the i-th dimension be denoted by $H^i = \left(h_1^i(y^i), \ldots, h_5^i(y^i)\right)$. Five boolean variables $x^{i1}, \ldots, x^{i5}$ were then created corresponding to the five clusters (i.e. intervals) along the i-th dimension. Finally, each data point $y_L^i, L = 1, \ldots, m$, was mapped to a boolean representation

$$x_L^{ij} = \begin{cases} 1, & \text{if } h_j^i(y_L^i) = 1,\ j = 1, \ldots, 5 \\ 0, & \text{otherwise,} \end{cases}$$

that encoded which interval did a fund fell into along the i-th dimension.

During the evaluation of experimental results, we used an external classification of mutual funds. This classification was provided by financial experts from Lipper[4], a financial analytics company. Each fund was assigned by the experts to one of seven classes shown in Table 3.5. We clustered the funds into seven clusters in the boolean and USOT spaces. The average class reconstruction errors shown in Table 3.19 demonstrate a consistent and statistically significant (at the 5% level) reductions in class reconstruction errors as a result of applying USOT. Similarly to the results obtained in Section 3.4, the performance improvements were attained across both clustering criteria indicating that USOT was successful at increasing separability between the underlying subgroups of mutual funds. Results obtained by clustering mutual funds based on the non-booleanized residuals provide a baseline for comparison with the boolean and USOT representation and indicate the plausibility of grouping mutual funds based on time series of their returns such that funds within a group have similar management styles. These results also suggest that the employed booleanization scheme introduced

[3]The number of clusters was chosen equal to five in order to provide enough resolution while keeping dimensionality of the resulting boolean space manageable for application of USOT.

[4]http://www.lipperweb.com/

additional noise that resulted in higher error rates on the boolean and USOT representations than on the original real-valued residuals. Although the problem of optimal booleanization of real data is beyond the scope of this thesis, in the future we plan to investigate if other booleanization approaches would be more effective. It is possible, however, that booleanization may result in appearance of spurious clusters in the data. As a part of future research, we intend to extend higher-order transformations beyond boolean data in order to circumvent potential problems associated with booleanization.

Table 3.18: Classes of mutual funds

Class	Description
Domestic	Invest in companies inside the U.S.
Fixed Income	Invest at least 65% of their assets in debt issues
Global	Invest at least 75% of their assets in companies both inside and outside of the U.S.
International	Invest at least 75% of their assets in companies strictly outside of the U.S.
Mixed Equity	Maintain a mix of stocks, bonds and money market instruments
Region-specific	Invest in specific regions
Sector-specific	Invest in specific economic sectors

Table 3.19: Average class reconstruction errors

	I_1			I_2		
	$\Re^n$	Bool.	USOT	$\Re^n$	Bool.	USOT
$\bar{E}$	**0.466**	0.57	0.495	**0.398**	0.61	0.581

Chapter 4

Conclusion

Two approaches have been proposed in statistical and machine learning communities in order to address the problem of uncovering complex clusters. One approach relies on the development of clustering criteria that are able to accommodate increasingly complex characteristics of the data. The other approach is based on the simplification of structure of data by mapping it to a different feature space via a non-linear function and then clustering in the new space. It is hoped that such mapping will increase separability between the "true" clusters thus making them more obvious for discovery by simple clustering criteria. However, since different datasets may exhibit drastically different internal structure, the mapping function applied must be adaptive to the data. In order to get a better understanding of what makes each cluster distinct from others, clusters are often analyze as to how well do they capture specifics of individual or groups of features. For example, do values of a given feature vary equally within clusters, or does this feature exhibit different behavior in different clusters and what domain knowledge can be inferred from that? To be able to answer such questions, it is important that dimensions of the new feature space into which the data is mapped, maintain their interpretability in terms of the original features.

This dissertation covers three related studies: development of a novel multi-dimensional clustering method, development of non-linear mapping functions that leverage higher-order co-occurrences between features in boolean data, and applications of these mapping functions for improving the performance of clustering methods. In particular, we developed first multi-dimensional clustering algorithm for the Neyman's criterion (2.2)

that was proposed in [51] for stratified sampling from one-dimensional data, but has never before been applied for clustering in multi-dimensional spaces. We then showed that this criterion is more reflective of the underlying data structure than the seemingly similar K-means criterion (2.1) when second order variability is not homogeneous between constituent subgroups. Unlike criterion (2.1), criterion (2.2) takes into account cluster means and variances, and, in general, produces non-linear cluster boundaries. We also discovered that criteria (2.1) and (2.2) produce identical clusterings when cluster variances are equal. Experimental results on simulated data generated by a mixture of multi-dimensional Gaussian distributions with different diagonal covariance matrices demonstrated that criterion (2.2) was able to recover the underlying class means and covariance matrices with higher accuracy than criterion (2.1). Although real-world data typically have more complex structure than was simulated in the aforementioned experiments, the obtained experimental results suggest that criterion (2.2) is a potentially superior alternative to criterion (2.1) for initialization of the Expectation Maximization procedure for reconstruction of hidden mixtures of Gaussians. We intend to investigate this possibility further as a part of our future work.

Development of criterion (2.2) was motivated by the problem of efficient estimation of the mean value of a scalar variable [10, 51]. A series of experimental results reported in this work demonstrate that generalization of criterion (2.2) to multi-dimensional spaces and development of the associated clustering algorithm allowed for efficient estimation of the grand mean vector of a population. Criterion (2.2) and the associated sampling scheme consistently and statistically significantly outperformed criterion (2.1) across the range of sample sizes and class densities when estimating the mean vector in multi-dimensional spaces. In all cases, both sampling schemes performed better than the simple random sample, particularly when the sample size was small. Robust performance of the stratified sampling schemes is especially encouraging since financial and temporal constraints often prohibit the acquisition of larger samples in most real-world

applications. The results also suggest that aside from estimation of the mean vector of multi-dimensional data, criterion (2.2) would be particularly useful for construction of training samples for supervised machine learning (classification and regression). Small training samples that accurately capture the underlying distribution of the data are crucial for practical applications of computationally demanding methods such as Support Vector Machine [63] on massive real-world datasets comprised of millions of data instances. Low computational complexity of the proposed clustering algorithm for criterion (2.2) opens the possibility of construction of small training samples from large-scale datasets. Further investigation of this topic constitutes one of the future research directions stemming from this work.

Recently, locally adaptive distance functions were proposed [20] as a data transformation that takes into account local differences in variance along each dimension for clustering. Locally Adaptive Clustering (LAC) proposed in [20] is essentially a modification of criterion 2.1. For each cluster and each dimension, LAC associates a weight that is used to adjust the Euclidean distance with respect to variance characteristics of the particular cluster. The weighted distances are then used in an iterative clustering procedure. LAC brings forward two issues that we intend to address in a future work related to criterion 2.2. The generalization of criterion (2.2) considered in this work aggregates the cluster variances across all dimension and does not treat them separately as does LAC. This behavior may produce unwanted results when the true clusters have ellipsoidal shapes highly elongated along certain dimensions. We therefore plan to investigate possible generalizations of criterion (2.2) that would work with arbitrary, not necessarily diagonal, covariance matrices instead of scalars characterizing clusters' variances. We also intend to investigate ways of introducing local weighting functions akin to LAC, but this time for clustering by criterion (2.2).

Since both criteria (2.1) and (2.2) are strictly concave, it follows from the definition

of a strictly concave function that criterion

$$I_4 = \gamma I_1 + (1-\gamma) I_2,\ \gamma \in [0,1], \tag{4.1}$$

is also strictly concave and can therefore be minimized by Algorithm 1. The utility of criterion (4.1) is unclear at this point, although this criterion may find its use in an interactive system for explorative analysis of data. By varying parameter γ, a user would be able to better understand the degree of separability between constituent subgroups. Subgroups that are well separated from the rest will tend to be stable with respect to the value of γ, i.e. would form separate clusters regardless of the value of γ. Clusterings of more commingled subgroups, on the other hand, would be sensitive to the setting of γ. Investigation of utility of criterion (4.1) and development of efficient minimization algorithms for the statistically-motivated non-convex criterion (2.20) proposed in [37] constitute two additional directions for future work.

In the framework of the mapping-based approach to discovering complex cluster structures, we introduced a novel adaptive non-linear data transformation termed Unsupervised Second Order Transformation (USOT). USOT maps data from a boolean space to a real space thereby emphasizing specifics of the various homogeneous subgroups of data instances. The novelties behind USOT are (a) that it leverages in a unsupervised manner, higher-order co-occurrences between features, and (b) that it considers each feature in the context of probabilistic relationships with other features. In addition, USOT has two desirable properties. USOT adaptively selects features that would influence the mapping of a given feature, and preserves the interpretability of dimensions of the transformed space.

The intuition behind USOT originated from our work on higher-order classifiers [26], and in particular from the Supervised Second Order Transformation (SSOT) also presented in this work. SSOT is a novel data transformation that requires the knowledge of true class labels of the instances comprising a training set. Both USOT and SSOT are defined over the space of higher-order paths. However, aside from SSOT being a

supervised transformation, the main difference between USOT and SSOT lies in the way the two mappings use the higher-order paths. While USOT considers probabilistic dependencies between a feature and all other features, SSOT makes use of probabilistic dependencies between a class indicator variable and the features.

We also developed a $O\left((m+n)n^2\right)$ time algorithm for obtaining the counts of higher-order paths used by USOT and SSOT. This algorithm improves over the $O\left(m^2n^3\right)$ complexity of a straight-forward path counting algorithm.

Experimental results on text corpora and financial time series demonstrated that by leveraging higher-order co-occurrences between features, the proposed transformations achieved statistically significant improvements over the traditional methods. The experiments on financial time series also showed that pre-processing of real-valued data into boolean form may introduce additional noise and make the underlying subgroups of data instances more difficult to separate into clusters. Hence, one direction for future work lies in the extension of higher-order transformations beyond boolean data. Development of a rigorous theoretical framework encompassing and quantifying higher-order relations in the context of clustering and classification constitutes another future research direction.

Appendix A

Proof of Lemma 1

The proof of Lemma 1 follows from the method (2.11) of constructing polynomial clusterings. Suppose, $H = (h_1(x), \ldots, h_K(x)), H \in \mathcal{H}$, is an arbitrary clustering whose vector of the non-normalized cluster moments is

$$\mu(H) = \left(M_1^{(0)}, M_1^{(1)}, \ldots, M_1^{(r)}, \ldots, M_K^{(0)}, M_K^{(1)}, \ldots, M_K^{(r)}\right).$$

Let $c = \left(c_1^{(0)}, c_1^{(1)}, \ldots, c_1^{(r)}, \ldots, c_K^{(0)}, c_K^{(1)}, \ldots, c_K^{(r)}\right)$ denote an arbitrary vector of coefficients and let $H^c = (h_1^c(x), \ldots, h_K^c(x))$ denote the corresponding polynomial clustering. Further, let $\mu(H^c) = \left(\hat{M}_1^{(0)}, \hat{M}_1^{(1)}, \ldots, \hat{M}_1^{(r)}, \ldots, \hat{M}_K^{(0)}, \hat{M}_K^{(1)}, \ldots, \hat{M}_K^{(r)}\right)$ denote the vector of the non-normalized cluster moments under polynomial clustering H^c. Then, the scalar product $(c, \mu(H^c) - \mu(H))$ is

$$(c, \mu(H^c) - \mu(H)) = \sum_{\alpha=1}^{K} \sum_{l=0}^{r} \left(c_\alpha^{(l)}, \hat{M}_\alpha^{(l)} - M_\alpha^{(l)}\right).$$

By definition (2.9) of the non-normalized cluster moments, we obtain

$$\begin{aligned}
(c, \mu(H^c) - \mu(H)) &= \sum_{\alpha=1}^{K} \sum_{l=0}^{r} \left(c_\alpha^{(l)}, \int_{\mathcal{X}} x^l h_\alpha^c(x) dP(x) - \int_{\mathcal{X}} x^l h_\alpha(x) dP(x) \right) \\
&= \sum_{\alpha=1}^{K} \left[\int_{\mathcal{X}} \sum_{l=0}^{r} \left(c_\alpha^{(l)}, x^l\right) h_\alpha^c(x) dP(x) - \int_{\mathcal{X}} \sum_{l=0}^{r} \left(c_\alpha^{(l)}, x^l\right) h_\alpha(x) dP(x) \right] \\
&= \sum_{\alpha=1}^{K} \left[\int_{\mathcal{X}} f_\alpha(x) h_\alpha^c(x) dP(x) - \int_{\mathcal{X}} f_\alpha(x) h_\alpha(x) dP(x) \right] \\
&= \int_{\mathcal{X}} \left[\sum_{\alpha=1}^{K} f_\alpha(x) h_\alpha^c(x) - \sum_{\alpha=1}^{K} f_\alpha(x) h_\alpha(x) \right] dP(x).
\end{aligned}$$

From definition (2.11) of characteristic functions $h_\alpha^c(x)$ follows that

$$(c, \mu(H^c) - \mu(H)) \leq 0, \ \forall H \in \mathcal{H}. \quad \square$$

Appendix B

Proof of Theorem 2

Before proceeding to the proof of Theorem 2, we show that set $Z = \{\mu(H) : H \in \mathcal{H}\}$ of vectors of the non-normalized cluster moments is bounded, closed and convex. A point $\mu(H) = \left(M_1^{(0)}, M_1^{(1)}, \ldots, M_1^{(r)}, \ldots, M_K^{(0)}, M_K^{(1)}, \ldots, M_K^{(r)}\right)$ belongs to set Z if and only if the following equations are satisfied:

$$\sum_{\alpha=1}^{K} M_\alpha^{(l)} = \int_{\mathcal{X}} x^l dP(x),\ l = 0, \ldots, r. \tag{B.1}$$

Set Z is bounded since the probability density function $P(x)$ is zero outside of the bounded region R.

Equations (B.1) imply that set Z is closed, because its complement $\bar{Z}$ is defined by strict inequalities and is therefore open.

Convexity of set Z follows from the fact that for any two points $\mu(H), \mu(\hat{H}) \in Z$, and any $\epsilon \in [0, 1]$, the point $\mu^\epsilon = (1-\epsilon)\mu(H) + \epsilon\mu(\hat{H})$ also lies in set Z, i.e.,

$$\sum_{\alpha=1}^{K} \left[(1-\epsilon)M_\alpha^{(l)} - \epsilon\hat{M}_\alpha^{(l)}\right] = (1-\epsilon)\int_{\mathcal{X}} x^l dP(x) + \epsilon\int_{\mathcal{X}} x^l dP(x) = \int_{\mathcal{X}} x^l dP(x),$$

where $M_\alpha^{(l)}$ and $\hat{M}_\alpha^{(l)}$ are the non-normalized cluster moments under clusterings H and $\hat{H}$, respectively.

It follows that all local minima of a strictly concave functional $I(\mu(H))$ are attained on the boundary points of set Z. Lemma 1 establishes the fact that the boundary points of set Z correspond to polynomial clusterings. Now, we prove Theorem 2, which specifies the form of polynomial clusterings minimizing the strictly concave functional I.

First, we prove that local extremality of a strictly concave functional $I(\mu(H))$ on a clustering H^* implies that $\mu(H^*) = \mu(H^c)$, where H^c is the polynomial clustering specified using vector c determined as a supergradient of functional I at the point $\mu(H^*)$.

Suppose that clusterings H^* and H^c are not equivalent, i.e., $\mu(H^*) \neq \mu(H^c)$. Then, we can construct a point $\mu^\epsilon = (1-\epsilon)\mu(H^*) + \epsilon\mu(H^c),\ \epsilon \in (0,1),\ \mu^\epsilon \in Z$.

By concavity of functional I, the following inequality holds:

$$I(\mu^\epsilon) \leq I(\mu(H^*)) + (c, \mu^\epsilon - \mu(H^*))\,.$$

Due to strict concavity of functional I, the equality is attained if and only if $\mu^\epsilon = \mu(H^*)$, which contradicts the assumption. Therefore, it follows that

$$I(\mu^\epsilon) - I(\mu(H^*)) < \epsilon\,(c, \mu(H^c) - \mu(H^*))\,,$$

and from Lemma 1 follows that

$$\epsilon\,(c, \mu(H^c) - \mu(H^*)) \leq 0.$$

Thus,

$$I(\mu^\epsilon) - I(\mu(H^*)) < 0,$$

which, given that ϵ was chosen arbitrarily, contradicts local extremality of H^*.

We complete the proof by showing that the existence of a polynomial clustering H^c equivalent to a clustering H^* that provides the functional I with a local minimum implies that the vector c is a supergradient of functional I at the point $\mu(H^*)$.

Suppose that polynomial clusterings H^c and H^k are not equivalent, i.e., $\mu(H^c) \neq \mu(H^k)$, for any vector k determined as a supergradient of functional I at the point $\mu(H^*)$. Then, we can form a point $\mu^\epsilon = (1-\epsilon)\mu(H^c) + \epsilon\mu(H^k),\ \epsilon \in (0,1),\ \mu^\epsilon \in Z$.

From concavity of functional I and equivalence of clusterings H^* and H^c follows that

$$I(\mu^\epsilon) \leq I(\mu(H^c)) + (k, \mu^\epsilon - \mu(H^c))\,,$$

where, due to strict concavity of functional I, the equality is attained if and only if $\mu^{\epsilon} = \mu(H^c)$, which contradicts the assumption.

Therefore, it follows that

$$I(\mu^{\epsilon}) - I(\mu(H^c)) < \epsilon \left(k, \mu(H^k) - \mu(H^c)\right),$$

and from Lemma 1 follows that

$$\epsilon \left(k, \mu(H^k) - \mu(H^c)\right) \leq 0.$$

Thus,

$$I(\mu^{\epsilon}) - I(\mu(H^c)) < 0,$$

which, given that ϵ was chosen arbitrarily, contradicts local extremality of clustering H^*. $\square$

Appendix C

The K-means Criterion

The K-means criterion

$$I_1 = \sum_{\alpha=1}^{K} p_\alpha \sigma_\alpha^2 = \sum_{\alpha=1}^{K} p_\alpha \left(\mathcal{M}_\alpha^{(2)} - \left(\mathcal{M}_\alpha^{(1)} \right)^2 \right), \tag{C.1}$$

is a member of the large family of clustering criteria (2.8) and hence falls into the framework presented in Section 2.1. Below we prove that functional (C.1) is strictly concave, which ensures convergence of the BGD (Algorithm 1) to a locally optimal clustering for this criterion. It should be noted that functional (C.1) is independent of the second non-normalized cluster moments $M_\alpha^{(2)}$. This can be shown by rewriting functional (C.1) as

$$\begin{aligned} I_1 = \sum_{\alpha=1}^{K} p_\alpha \sigma_\alpha^2 &= \sum_{\alpha=1}^{K} p_\alpha \left(\mathcal{M}_\alpha^{(2)} - \left(\mathcal{M}_\alpha^{(1)} \right)^2 \right) \\ &= \sum_{\alpha=1}^{K} M_\alpha^{(2)} - \sum_{\alpha=1}^{K} p_\alpha \left(\mathcal{M}_\alpha^{(1)} \right)^2 \\ &= \sum_{\alpha=1}^{K} \int_{\mathcal{X}} x^2 h_\alpha^\pi(x) dP(x) - \sum_{\alpha=1}^{K} p_\alpha \left(\mathcal{M}_\alpha^{(1)} \right)^2 \\ &= \int_{\mathcal{X}} x^2 dP(x) - \sum_{\alpha=1}^{K} p_\alpha \left(\mathcal{M}_\alpha^{(1)} \right)^2, \\ &= C - \sum_{\alpha=1}^{K} p_\alpha \left(\mathcal{M}_\alpha^{(1)} \right)^2, \end{aligned} \tag{C.2}$$

where C is a constant independent of a clustering $H \in \mathcal{H}$. It follows that minimization of functional (C.1) corresponds to maximization of functional $I_1' = \sum_{\alpha=1}^{K} p_\alpha \left(\mathcal{M}_\alpha^{(1)} \right)^2$. As we will see shortly, independence of functional (C.1) from second non-normalized cluster moments results in independence from these moments of the cluster membership functions for functional (C.1).

Claim 2. Functional I_1 is strictly concave.

Proof. We prove the claim by showing that the α-th functional $I_{1\alpha} = p_\alpha \sigma_\alpha^2$ in summation (C.1) is strictly concave, from which it follows that functional I_1 is concave.

Computing the gradient $\nabla I_{1\alpha} = \left(c_\alpha^{(0)}, c_\alpha^{(1)}, c_\alpha^{(2)}\right)$ of functional $I_{1\alpha}$ yields

$$\begin{aligned} c_\alpha^{(0)} &= \frac{\partial I_1}{\partial p_\alpha} = \frac{\left(M_\alpha^{(1)}\right)^2}{p_\alpha^2} = \left(\mathcal{M}_\alpha^{(1)}\right)^2, \\ c_\alpha^{(1)} &= \frac{\partial I_1}{\partial M_\alpha^{(1)}} = -2\frac{M_\alpha^{(1)}}{p_\alpha} = -2\mathcal{M}_\alpha^{(1)}, \\ c_\alpha^{(2)} &= \frac{\partial I_1}{\partial M_\alpha^{(2)}} = 1. \end{aligned} \tag{C.3}$$

Let the non-normalized cluster moments of cluster α under a clustering $H \in \mathcal{H}$ be denoted by $\mu_\alpha(H) = \left(p_\alpha, M_\alpha^{(1)}, M_\alpha^{(2)}\right)$. For any two clusterings $H \in \mathcal{H}$ and $\hat{H} \in \mathcal{H}$ we have

$$\left(\nabla I_{1\alpha}(\mu_\alpha(H)), \mu_\alpha(\hat{H}) - \mu_\alpha(H)\right) = \left(\frac{M_\alpha^{(1)}}{p_\alpha}\right)^2 (\hat{p}_\alpha - p_\alpha) - \frac{2}{p_\alpha}\left(M_\alpha^{(1)}, \hat{M}_\alpha^{(1)} - M_\alpha^{(1)}\right) + \hat{M}_\alpha^{(2)} - M_\alpha^{(2)},$$

and

$$I_{1\alpha}(\mu_\alpha(\hat{H})) - I_{1\alpha}(\mu_\alpha(H)) = \hat{M}_\alpha^{(2)} - \frac{\left(\hat{M}_\alpha^{(1)}\right)^2}{\hat{p}_\alpha} - M_\alpha^{(2)} + \frac{\left(M_\alpha^{(1)}\right)^2}{p_\alpha}.$$

By subtracting the first equation from the second and simplifying, we have

$$\begin{aligned} & I_{1\alpha}(\mu_\alpha(\hat{H})) - I_{1\alpha}(\mu_\alpha(H)) - \left(\nabla I_{1\alpha}(\mu_\alpha(H)), \mu_\alpha(\hat{H}) - \mu_\alpha(H)\right) = \\ &= -\frac{\left(\hat{M}_\alpha^{(1)}\right)^2}{\hat{p}_\alpha} - \left(\frac{M_\alpha^{(1)}}{p_\alpha}\right)^2 \hat{p_\alpha} + \frac{2}{p_\alpha}\left(M_\alpha^{(1)}, \hat{M}_\alpha^{(1)}\right) \\ &= -\hat{p_\alpha}\left(\frac{M_\alpha^{(1)}}{p_\alpha} - \frac{\hat{M}_\alpha^{(1)}}{\hat{p}_\alpha}\right)^2 \\ &= -\hat{p_\alpha}\left(\mathcal{M}_\alpha^{(1)} - \hat{\mathcal{M}}_\alpha^{(1)}\right)^2 < 0,\ \mathcal{M}_\alpha^{(1)} \neq \hat{\mathcal{M}}_\alpha^{(1)} \end{aligned}$$

It follows that, by definition of a strictly concave function, functional $I_{1\alpha}$ is strictly concave, i.e.,

$$I_{1\alpha}(\mu_\alpha(\hat{H})) < I_{1\alpha}(\mu_\alpha(H)) + \left(\nabla I_{1\alpha}(\mu_\alpha(H)), \mu_\alpha(\hat{H}) - \mu_\alpha(H)\right),\ \mathcal{M}_\alpha^{(1)} \neq \hat{\mathcal{M}}_\alpha^{(1)}$$

Therefore, functional $I_1 = \sum_{\alpha=1}^{K} I_{1\alpha}$ is strictly concave. □

Using the gradient (C.3) for specifying membership functions (2.10) yields

$$\begin{aligned} f_\alpha(x) &= c_\alpha^{(0)} + \left(c_\alpha^{(1)}, x\right) + c_\alpha^{(2)} x^2 \\ &= \left(\mathcal{M}_\alpha^{(1)}\right)^2 - 2\left(\mathcal{M}_\alpha^{(1)}, x\right) + x^2 \\ &= \left(x - \mathcal{M}_\alpha^{(1)}\right)^2, \end{aligned} \tag{C.4}$$

where $\left(x - \mathcal{M}_\alpha^{(1)}\right)^2$ is the squared Euclidean distance between a point $x \in \mathcal{X}$ and the mean vector $\mathcal{M}_\alpha^{(1)}$ of cluster α. Note that membership functions (C.4) are independent of the second non-normalized cluster moments $M_\alpha^{(2)}$. From the definitions of characteristic functions (2.11) and membership functions (C.4) follows that under a polynomial clustering H^c, a point is assigned to a cluster whose mean vector is closest to that point, according to the squared Euclidean distance.

References

[1] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, February 1993.

[2] M. A. Aizerman, E. M. Braverman, and L. I. Rozonoer. *The Method of Potential Functions in Machine Training Theory*. Nauka, Moscow, 1970.

[3] L. D. Baker and A. K. McCallum. Distributional clustering of words for text classification. In *SIGIR '98: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 96–103, New York, NY, USA, 1998. ACM.

[4] Y. Bartal, M. Charikar, and D. Raz. Approximating min-sum k-clustering in metric spaces. In *STOC '01: Proceedings of the thirty-third annual ACM symposium on Theory of computing*, pages 11–20, New York, NY, USA, 2001. ACM.

[5] E. V. Bauman and A. A. Dorofeyuk. Recursive automatic classification algorithms. *Automation and Remote Control*, (3):345–355, 1982.

[6] M. W. Berry and M. Castellanos. *Survey of Text Mining II : Clustering, Classification and Retrieval*. Springer-Verlag, January 2008.

[7] S. M. Borodkin. Optimal grouping of interrelated ordered objects. *Automation and Remote Control*, (2):165–172, February 1980.

[8] S. M. Borodkin, A. M. Borodkin, and I. B. Muchnik. Optimal requantization of deep grayscale images and lloyd-max quantization. *IEEE Transactions on Image Processing*, 15(2):445–448, 2006.

[9] A. L. Bowley. Measurement of precision attained in sampling. *Bull. Inter. Statist. Inst.*, 22:1–26, 1926.

[10] E. M. Braverman, B. M. Litvakov, I. B. Muchnik, and S. G. Novikov. Stratified sampling in the organization of empirical data collection. *Automation and Remote Control*, 36(10):1629–1641, 1975.

[11] L. Cai, H. Huang, S. Blackshaw, J. Liu, C. Cepko, and W. Wong. Clustering analysis of sage data using a poisson approach. *Genome Biology*, 5(7):R51, 2004.

[12] S. Chakrabarti, B. Dom, and P. Indyk. Enhanced hypertext categorization using hyperlinks. *SIGMOD Rec.*, 27(2):307–318, 1998.

[13] Y. Chen, W. Trappe, and R. P. Martin. Detecting and localizing wireless spoofing attacks. *Sensor, Mesh and Ad Hoc Communications and Networks, 2007. SECON '07. 4th Annual IEEE Communications Society Conference on*, pages 193–202, 18-21 June 2007.

[14] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(5):603–619, 2002.

[15] D. Comaniciu, V. Ramesh, and P. Meer. Real-time tracking of non-rigid objects using mean shift. In *CVPR*, pages 2142–. IEEE Computer Society, 2000.

[16] D. Comaniciu, V. Ramesh, and P. Meer. The variable bandwidth mean shift and data-driven scale selection. In *ICCV*, pages 438–445, 2001.

[17] W. F. de la Vega, M. Karpinski, C. Kenyon, and Y. Rabani. Approximation schemes for clustering problems. In *STOC '03: Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*, pages 50–58, New York, NY, USA, 2003. ACM.

[18] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41:391–407, 1990.

[19] S. Déjean, P. G. Martin, A. Baccini, and P. Besse. Clustering time-series gene expression data using smoothing spl ine derivatives. *EURASIP Journal on Bioinformatics and Systems Biology*, 2007.

[20] C. Domeniconi, D. Gunopulos, S. Ma, B. Yan, M. Al-Razgan, and D. Papadopoulos. Locally adaptive metrics for clustering high dimensional data. *Data Min. Knowl. Discov.*, 14(1):63–97, 2007.

[21] P. Edmonds. Choosing the word most typical in context using a lexical co-occurrence network. In *In Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics*, pages 507–509, 1997.

[22] B. S. Everitt, S. Landau, and M. Leese. *Cluster Analysis*. Arnold Publishers, May 2001.

[23] K. Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press, San Diego, CA, 1990.

[24] M. C. Ganiz. *Higher-order path analysis for supervised machine learning*. PhD thesis, Lehigh University, Bethlehem, PA, USA, January 2008.

[25] M. C. Ganiz, S. Kanitkar, M. C. Chuah, and W. M. Pottenger. Detection of interdomain routing anomalies based on higher-order path analysis. In *ICDM '06: Proceedings of the Sixth International Conference on Data Mining*, pages 874–879, Washington, DC, USA, 2006. IEEE Computer Society.

[26] M. C. Ganiz, N. I. Lytkin, and W. M. Pottenger. Leveraging higher order dependencies between features for text classification. In *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD)*, Bled, Slovenia, 2009.

[27] Y. Gdalyahu, N. Shental, and D. Weinshall. Perceptual grouping and segmentation by stochastic clustering. *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, 1:367–374 vol.1, 2000.

[28] L. Getoor and C. P. Diehl. Link mining: a survey. *SIGKDD Explor. Newsl.*, 7(2):3–12, 2005.

[29] A. D. Gordon. *Classification, 2nd Edition (CRC Monographs on Statistics & Applied Probability)*. Chapman & Hall, 2 edition, June 1999.

[30] D. Greene and P. Cunningham. Practical solutions to the problem of diagonal dominance in kernel document clustering. In *Proc. 23rd International Conference on Machine learning (ICML'06)*, pages 377–384. ACM Press, 2006.

[31] J. A. Hartigan. *Clustering Algorithms*. John Wiley & Sons, Inc., New York, NY, USA, 1975.

[32] A. B. Hur, D. Horn, H. T. Siegelmann, and V. Vapnik. Support vector clustering. *J. Mach. Learn. Res.*, 2:125–137, 2002.

[33] D. J. J. Neville. Iterative classification in relational data. In *In Proc. AAAI*, pages 13–20. AAAI Press, 2000.

[34] A. K. Jain and R. C. Dubes. *Algorithms for clustering data*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1988.

[35] T. Joachims. *Learning to Classify Text Using Support Vector Machines: Methods, Theory and Algorithms*. Kluwer Academic Publishers, Norwell, MA, USA, 2002.

[36] L. Katz. A new status index derived from sociometric analysis. *Psychometrika*, 18(1):39–43, March 1953.

[37] N. E. Kiseleva, I. B. Muchnik, and S. G. Novikov. Stratified samples in the problem of representative types. *Automation and Remote Control*, 47(5):684–693, 1986.

[38] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *J. ACM*, 46(5):604–632, 1999.

[39] A. Kontostathis and W. M. Pottenger. A framework for understanding latent semantic indexing (lsi) performance. *Inf. Process. Manage.*, 42(1):56–73, 2006.

[40] U. H.-G. Kreßel. Pairwise classification and support vector machines. In *Advances in kernel methods: support vector learning*, pages 255–268, Cambridge, MA, USA, 1999. MIT Press.

[41] K. Lang. Newsweeder: Learning to filter netnews. In *Proceedings of the Twelfth International Conference on Machine Learning*, pages 331–339, 1995.

[42] T. V. Le, C. A. Kulikowski, and I. B. Muchnik. A graph-based approach for image segmentation. In G. Bebis, R. D. Boyle, B. Parvin, D. Koracin, P. Remagnino, F. M. Porikli, J. Peters, J. T. Klosowski, L. L. Arns, Y. K. Chun, T.-M. Rhyne, and L. Monroe, editors, *ISVC (1)*, volume 5358 of *Lecture Notes in Computer Science*, pages 278–287. Springer, 2008.

[43] S. Li, T. Wu, and W. M. Pottenger. Distributed higher order association rule mining using information extracted from textual data. *SIGKDD Explor. Newsl.*, 7(1):26–35, 2005.

[44] S. P. Lloyd. Least squares quantization in pcm. *IEEE Trans. Inf. Theory*, IT-28:129, March 1982.

[45] Q. Lu and L. Getoor. Link-based classification. In T. Fawcett and N. Mishra, editors, *ICML*, pages 496–503. AAAI Press, 2003.

[46] N. I. Lytkin, C. A. Kulikowski, and I. B. Muchnik. Variance-based criteria for clustering and their application to the analysis of management styles of mutual funds based on time series of daily returns. Technical Report 2008-01, DIMACS, Rutgers University, February 2008.

[47] Y. M. Marzouk and A. F. Ghoniem. K-means clustering for optimal partitioning and dynamic load balancing of parallel hierarchical n-body simulations. *J. Comput. Phys.*, 207(2):493–528, 2005.

[48] G. J. McLachlan and D. Peel. *Finite mixture models*. Wiley, New York, 2000.

[49] B. Mirkin. *Clustering For Data Mining: A Data Recovery Approach*. Chapman & Hall/CRC, 2005.

[50] J. Neville and D. Jensen. Dependency networks for relational data. *Data Mining, 2004. ICDM '04. Fourth IEEE International Conference on*, pages 170–177, Nov. 2004.

[51] J. Neyman. On the two different aspects of the representative method: the method of stratified sampling and the method of purposive selection. *Journal of the Royal Statistical Society*, 97:558–625, 1934.

[52] F. Pattarin, S. Paterlini, and T. Minerva. Clustering financial time series: an application to mutual funds style analysis. *Computational Statistics & Data Analysis*, 47(2):353–372, 2004.

[53] M. Rege, M. Dong, and F. Fotouhi. Co-clustering image features and semantic concepts. *Image Processing, 2006 IEEE International Conference on*, pages 137–140, 8-11 Oct. 2006.

[54] S. Roberts. Non-parametric unsupervised cluster analysis. *Pattern Recognition*, 30(2):261–272, 1997.

[55] B. Scholkopf and A. J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Cambridge, MA, USA, 2001.

[56] L. J. Schulman. Clustering for edge-cost minimization (extended abstract). In *STOC '00: Proceedings of the thirty-second annual ACM symposium on Theory of computing*, pages 547–555, New York, NY, USA, 2000. ACM.

[57] H. Schütze. Automatic word sense discrimination. *Comput. Linguist.*, 24(1):97–123, 1998.

[58] K. Sim, J. Li, V. Gopalkrishnan, and G. Liu. Mining maximal quasi-bicliques to co-cluster stocks and financial ratios for value investment. *Data Mining, 2006. ICDM '06. Sixth International Conference on*, pages 1059–1063, Dec. 2006.

[59] N. Slonim and N. Tishby. The power of word clusters for text classification. In *In 23rd European Colloquium on Information Retrieval Research*, 2001.

[60] B. Taskar, E. Segal, and D. Koller. Probabilistic classification and clustering in relational data. In *In Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence*, pages 870–878, 2001.

[61] S. Theodoridis and K. Koutroumbas. *Pattern Recognition, Third Edition.* Academic Press, February 2006.

[62] N. Tishby and N. Slonim. Data clustering by markovian relaxation and the information bottleneck method. In T. K. Leen, T. G. Dietterich, and V. Tresp, editors, *NIPS*, pages 640–646. MIT Press, 2000.

[63] V. Vapnik. *Statistical Learning Theory.* John Wiley, 1998.

[64] V. N. Vapnik. *The nature of statistical learning theory.* Springer-Verlag New York, Inc., New York, NY, USA, 1999.

[65] A. Vashist, C. A. Kulikowski, and I. B. Muchnik. Ortholog clustering on a multipartite graph. *IEEE/ACM Trans. Comput. Biol. Bioinformatics*, 4(1):17–27, 2007.

[66] J. Xu and W. B. Croft. Corpus-based stemming using cooccurrence of word variants. *ACM Trans. Inf. Syst.*, 16(1):61–81, 1998.

[67] X. Zhang, M. W. Berry, and P. Raghavan. Level search schemes for information filtering and retrieval. *Inf. Process. Manage.*, 37(2):313–334, 2001.

[68] Z. Zhao, A. Vashist, A. Elgammal, I. Muchnik, and C. Kulikowski. Combinatorial and statistical methods for part selection for object recognition. *Int. J. Comput. Math.*, 84(9):1285–1297, 2007.

Vita

Nikita I. Lytkin

Education

2003-2009 Ph. D. in Computer Science, Rutgers University

2003-2006 M. Sc. in Computer Science, Rutgers University

1999-2002 B. Sc. in Computer Science, Rutgers University

Principal Occupation

2006-2009 Graduate research assistant, Department of Computer Science, Rutgers University

2003-2005 Teaching assistant, Department of Computer Science, Rutgers University

Publications

M. C. Ganiz, N. I. Lytkin, and W. M. Pottenger. Leveraging Higher Order Dependencies between Features for Text Classification. *In Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD)*, Bled, Slovenia, 2009.

N. I. Lytkin and W. M. Pottenger. Information Theoretic Similarity Measures for Inter-domain Predicate Mapping. *In Proceedings of the Text Mining Workshop, SIAM Conference on Data Mining*, Atlanta, GA, 2008.