2007 IEEE/WIC/ACM International Conference on Web Intelligence

K-SVMeans: A Hybrid Clustering Algorithm for Multi-Type Interrelated Datasets

Levent Bolelli[1], Seyda Ertekin[1], Ding Zhou[1], C. Lee Giles[1,2]
[1]Department of Computer Science and Engineering
[2]'College of Information Sciences and Technology
The Pennsylvania State University
University Park, PA, USA

Abstract

Identification of distinct clusters of documents in text collections has traditionally been addressed by making the assumption that the data instances can only be represented by homogeneous and uniform features. Many real-world data, on the other hand, comprise of multiple types of heterogeneous interrelated components, such as web pages and hyperlinks, online scientific publications and authors and publication venues to name a few. In this paper, we present K-SVMeans, a clustering algorithm for multi-type interrelated datasets that integrates the well known K-Means clustering with the highly popular Support Vector Machines. The experimental results on authorship analysis of two real world web-based datasets show that K-SVMeans can successfully discover topical clusters of documents and achieve better clustering solutions than homogeneous data clustering.

1. Introduction

Discovery of latent semantic groupings and identification of intrinsic structures in datasets is a crucial task for many data analysis needs. This task falls in the field of clustering, where the goal is to find distinct groups of instances within data collections. In general terms, clustering is an optimization problem that tries to find a partition of the data collection such that the items that belong to the same cluster are as similar as possible (cluster compactness) and the discovered clusters are as separate as possible (cluster distinctness) based on a specified (dis)similarity metric within the high dimensional space that the data objects exist. Although research in the field of clustering predates the vast popularity of the world wide web, the characteristics of web-based data requires us to tailor traditional clustering algorithms to utilize the heterogeneous relationships between the data objects. Most real-world data, especially data available on the web, possess rich structural relationships. One of the most pronounced characteristics of web based data is that the data contains *heterogeneous* components; that is, they have multiple types of information that describe the entities that we are interested in clustering, such as authorship and citation graph of scientific documents [2], hyperlinks in web connectivity graph [11] and surrounding text around images in web pages [7].

Traditionally, research in clustering has mainly focused on "flat" data clustering where the data instances are represented as a vector of homogeneous and uniform set of features, like words in text documents or visual features in image collections. For instance, scientific papers, email messages, blog and newsgroup posts can be directly clustered solely based on the textual content of the documents using traditional clustering algorithms. One problem with this approach is that it discards the *global view* of the authorship of documents. Figure 1 depicts the document and authorship spaces of a given corpus. In addition to the textual content of the documents, authors can be represented by the collection of words of the documents they have authored. Since documents authored by the same person tend to be topically similar, we can use this information as an additional dimension of similarity in the clustering process. Combining these two dimensions has the potential to yield better clustering solutions than investigating a single source of similarity in isolation.

We present K-SVMeans that clusters datasets with heterogeneous similarity characteristics. K-SVMeans simultaneously clusters along one dimension of the data while learning a classifier in another dimension, which, in turn effects the intermediate cluster assignment decisions in the original dimension. K-SVMeans clustering is a hybrid clustering solution that merges the well known K-Means clustering algorithm with Support Vector Machines (SVM), a highly popular supervised learning algorithm that has been shown to be highly effective, especially for text classifica-

0-7695-3026-5/07 $25.00 © 2007 IEEE
DOI 10.1109/WI.2007.61

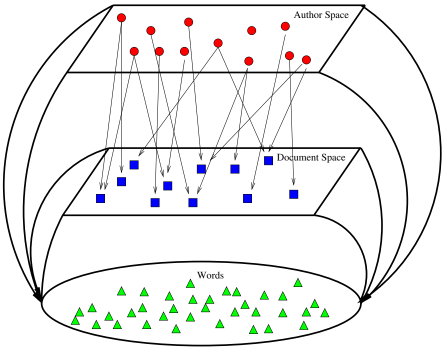


Figure 1. Author space and Document space. In addition to the textual representation of documents, each author can be represented as the collection of the words of the documents they have (co)authored.

tion tasks. The hybrid nature of K-SVMeans comes from the fact that it merges an unsupervised learner with a supervised learning algorithm, while eliminating the need for labeled training instances for SVM learning. The cluster assignments of K-Means are used to train an Online SVM in the secondary data type, and the SVM effects the clustering decisions of K-Means in the primary clustering space. This ping-pong style clustering of heterogeneous datasets effectively increases the clustering performance compared to clustering using a single homogeneous data source.

2 Related Work

Although clustering is a decades old problem, research in multivariate data clustering where the data can be represented by multiple interrelated components has gained momentum only in the past couple of years due to its applicability in many domains. The initial directions towards multivariate clustering started with the simultaneous clustering of both rows and columns of contingency tables, also known as coclustering, biclustering, or block clustering. [4] proposed a spectral graph bipartitioning algorithm that clusters documents based on words, and words based on documents by finding the normalized cut of the bipartite graph. The same problem has been addressed in [5] by taking an information-theoretic approach. The proposed solution attempts to minimize the loss in mutual information between the original and the clustered contingency tables. In [12], a partial singular value decomposition of the edge-weight matrix of the bipartite graph is computed to cocluster words and documents. Although these works have laid the foundations of multivariate data clustering, these algorithms can not handle multi-type interrelated data objects.

A multi-type extension of the bipartite spectral graph partitioning has been proposed in [8] for textual datasets and then for images and surrounding texts in web environment [7]. The data objects form a tripartite graph, and the tripartite graph is treated as two separate bipartite graphs. The spectral partitioning of the bipartite graphs is obtained by minimizing the cuts of both bipartite graphs using semidefinite programming in $m+n+t$ dimensional space where each dimension represents the dimension of a separate data type. The high-dimensionality of the problem space is prohibitive and prevents its applicability to real-world datasets of big sizes. A multi-way clustering framework is proposed in [1] that maximizes the mutual information between the clusters of multiple data types based on representation of the interaction between each pair of data types as a contingency table of co-occurence counts. The generation of clusters is performed by a combination of agglomerative (bottom-up) and partitional (top-down) clusterings of different data types. The decision as to which types will be clustered agglomeratively or partitional, and the order of their clusterings is determined by a clustering schedule determined beforehand of the clustering process, and an optimal clustering schedule needs to be provided by the user.

3 Background on Support Vector Learning

A key feature of K-SVMeans is the integration of Support Vector Machines with K-Means clustering. In this section, we provide background on SVMs and Online Support Vector Learning, which is a key component in clustering decisions made by K-SVMeans. Section 4 will tie K-Means with Online SVM and provide details of K-SVMeans algorithm.

Support Vector Machines are well known for their generalization performance and ability to handle high dimensional data which is a common case in document classification problems. Considering the binary classification case, let $((x_1, y_1) \cdots (x_n, y_n))$ be the training dataset where x_i are the feature vectors that represent the observations and $y_i \in (-1, +1)$ be the two labels that each observation can be assigned to. From these observations, SVM builds an optimum hyperplane – a linear discriminant in the kernel transformed higher dimensional feature space – that maximally separates the two classes by the widest margin by minimizing the following objective function

$$\min_{\mathbf{w},\mathbf{b},\xi_\mathbf{i}} \frac{1}{2}\mathbf{w} \cdot \mathbf{w}^T + C\sum_{i=1}^{N} \xi_i \tag{1}$$

where $\mathbf{w}$ is the norm of the hyperplane, b is the offset, $y(x_i)$ are the labels and ξ_i are the slack variables that permit the non-separable case by allowing misclassification of training instances. The convex quadratic programming (QP) problem in equation 1 can be solved analytically by Sequential Minimal Optimization(SMO) [10]. It solves the QP problem by optimizing the dual form of equation 1, given as:

$$\max L_D \equiv \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x_i} \cdot \mathbf{x_j} \tag{2}$$

$$\forall i \ , \ 0 \leq \alpha_i \leq C$$

where the α_i are the Lagrange multipliers introduced in the dual representation of equation 1. SMO optimizes two α's at a time while holding the remaining α's fixed. The most significant benefits of being able to solve the QP problem analytically is increased efficiency of SVMs, the ability to handle massive datasets and thus make it practical for real-world applications, and lead way to *online* SVM algorithms, such as LASVM [3]. LASVM does not require all the labeled training instances be presented to the learner before the SVM training phase, unlike traditional batch learners. Thus, it can incrementally build a learner by updating its model when new observations become available. Once a training instance becomes available, LASVM searches a pairing support vector in its set of existing support vectors that maximizes the dual function, and adjusts both α's by the maximal step size. If no such pair can be found, the new observation does not become a support vector. It has been shown that the classification accuracies of Online SVM is competitive to the batch SVM learning while the performance of online learning is much faster with the freedom of adding new labeled observations during training phase [3]. The ability to use SVMs in an online setting enables us to efficiently integrate it with unsupervised learners, and as will be shown later, this combination does not require labeled training data for SVM.

4 K-SVMeans Algorithm

K-SVMeans is a K-Means based clustering algorithm for heterogeneous datasets where clustering along one data type learns a classifier in another, and the classifiers effect the clustering decisions made by the clusterer.

The original formulation of K-Means algorithm first initializes k clusters with data objects and then assigns each object $d_i, \ 1 \leq i \leq N$ to a cluster $c_i, \ 1 \leq i \leq k$ where d_i's distance to the representative of its assigned cluster c_i is minimum. Variants of K-Means algorithm differ in the initialization of clusters (e.g. random or maximum cluster distance initialization), the definition of similarity (e.g. Euclidean or Kullback-Leibler Divergence), or the definition of cluster representativeness (e.g. mean, median or weighted centroid vector). K-SVMeans is independent of any of those variations, but for brevity, we describe the algorithm for Spherical K-Means with random initialization that represents each cluster by its centroid vector.

We start with a brief overview of traditional K-Means. Given n data objects $x_1, x_2 \cdots x_n$, $\forall \mathbf{x_i} \in \mathbf{R^w}$ where $\mathbf{w}$ is the size of the feature space and each x_i is normalized such that $||x_i|| = 1$, K-Means partitions the x_i into k disjoint clusters $\pi_1, \pi_2, \cdots, \pi_k$, so that

$$\bigcup_{i=1}^{k} \pi_i = \{x_1, x_2, \cdots x_n\} \quad \textit{where} \quad \pi_i \cap \pi_j = \emptyset, \ \ i \neq j \tag{3}$$

where the centroid c_i of each cluster π_i is defined as

$$c_i = \frac{\sum_{x_k \in \pi_i} \mathbf{x_j}}{|| \sum_{x_j \in \pi_i} \mathbf{x_j} ||} \tag{4}$$

The goal of the clusterer is to maximize the similarity between the data objects and their assigned clusters, hence, the objective function becomes

$$\max Q = \sum_{j=1}^{k} \sum_{\mathbf{x_i} \in \pi_\mathbf{j}} \mathbf{x_i^T} \cdot \mathbf{c_j} \quad \forall \pi_i \ \ 1 \leq i \leq k \tag{5}$$

K-Means optimizes the objective function iteratively by following two steps: A cluster assignment step, where each data object is assigned to a cluster with the closest centroid, followed by a cluster centroid update step. The algorithm terminates when the change in the objective function value between two successive iterations is below a given threshold. Upon the termination of the algorithm, each data object belongs to one of the k clusters. This partitioning, however, is done on a single dimension.

Consider that the instances in the set $X = (x_1, x_2, \cdots, x_n)$, which we want to obtain a clustering solution, are related to another set $U = (u_1, u_2, \cdots, u_m)$ in some way. Each x_i can be related to one or multiple u_j's in a $X \rightarrow U$ mapping where objects in U denote a unique property of x_i. The reverse map $U \rightarrow X$ lets us represent each u as a mixture of the x_i's that are connected to it. Let T denote the relationship matrix where $T_{ij} = 1$ if x_i is related to u_j, and zero otherwise.

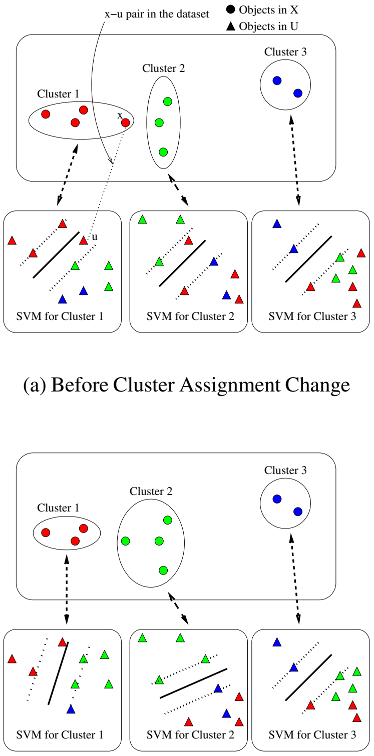
During the clustering process, the intermediate cluster assignments in K-SVMeans are determined by two conditions. In the first condition, a data object x_i is reassigned from a cluster π_i to π_j if x_i is closer to π_j's centroid than π_i's centroid and the u's of x_i are classified into the positive class by π_j's SVM and into the negative class by π_i's SVM.

K-SVMeans Cluster Assignment

Definitions:

x_i:*Objects to be clustered*
d_{ij}: *distance of object x_i to cluster π_j*
$m(i)$: *assigned cluster of x_i*
$l(\pi_i)$: *SVM learner of cluster π_i*
$\hat{y}(u,\pi) = \sum_{z=1}^{n} \alpha_z^{\pi} \mathbf{K}^{\pi}(u, u_z^{\pi}) + b^{\pi}$ *SVM decision value* **u** *for cluster π*
λ: Penalty term

```
For each $x_i \in X$
  $d = x_i \cdot c_{m(i)}$ ,   $\pi_i \leftarrow m(i)$
  $s = \sum_{\forall u_k, T_{ik}=1} \hat{y}(u_k, \pi_i)$
  For each cluster $\pi_j$,   $i \neq j$
    $\hat{d} = x_i \cdot c_j$
    $\hat{s} = \sum_{\forall u_k, T_{ik}=1} \hat{y}(u_k, \pi_j)$
    If ($\hat{d} < d$ and $s < 0$ and $\hat{s} > 0$) or
       ($\hat{d} \cdot \lambda < d$ and $\hat{s} < 0$)
      Remove $u$'s related with $x_i$ from $l(\pi_i)$
      Insert $u$'s related with $x_i$ to $l(\pi_j)$ as +1
      Insert $u$'s related with $x_i$ to $l(\pi_p)$ as -1, $j \neq p$
      $m(x_i) \leftarrow \pi_j$
    End
  End
End
```



(a) Before Cluster Assignment Change

(b) Document assigned to Cluster 2 and it's author added as a positive observation to Cluster 2, and negative observation to the rest of the clusters.

Figure 2. Cluster assignment and SVM update in K-SVMeans for three clusters

Second condition is as follows: In case the candidate cluster's SVM learner decides that the objects in U that are connected to x_i do not belong to that cluster (i.e. the decision values of the u's are negative), then we apply a penalty term ($\lambda > 1$) on the distance metric of K-Means so that the similarity between x_i and the candidate cluster centroid must be strong enough to warrant a cluster assignment change of x_i. The penalty term also ensures us that the SVM classifiers are not adversely effected by incorrect clustering decisions of K-Means that result in mislabeling of the u_j. Only highly similar x_i are allowed a cluster change in case the SVM classification decision is not trusted. A representation of the clustering in K-SVMeans is given in Figure 2. Each cluster has an associated SVM learner that is trained during the clustering process. In the graphical example, the object x is closer to Cluster 2, and its mapping u is misclassified in the SVM of Cluster 1, and correctly classified in the SVM of Cluster 2. The algorithm, therefore, assigns x to Cluster 2, updates the learners of all learners to reflect the cluster change of x. Depending on the model characteristics of each cluster's SVM learner, the change of status of u (moving from one cluster to another) may or may not effect the learners. Note that the label change of u can only potentially effect either the old class of u (where it's label changed from +1 to -1), or it's new class (where the label changed from -1 to +1). The learners of rest of the clusters are not effected by this cluster reassignment.

K-SVMeans can be run in multiple iterations where the SVM learner initialization is performed by using the clustering solution generated in the previous run. In the first iteration, we run standard K-Means algorithm to yield a clustering based on the primary space X. This iteration has two purposes. First, we use the clustering result from this step as a baseline for comparison. Second, and more importantly, it generates the labeled initialization set for the SVM learners of K-SVMeans. In the beginning of an iteration $t+1$, we look at each cluster π_i^t generated in the previous run and select m objects closest to the centroid of π_i^t and use their associated u_i for SVM initialization. We use one-against-rest classification in the SVMs, so the u's become positive observations for their respective clusters, and negative observations for the rest of the clusters. Although it is possible that the previous iteration may have assigned some of the x_i's to incorrect clusters, the x_i's that are closest to the centroids are more likely to be correctly assigned to their

CITESEER DATASET									
Cluster 1		Cluster 2		Cluster 3		Cluster 4		Cluster 5	
Venue	Samples	Venue	Samples	Venue	Samples	Venue	Samples	Venue	Samples
AAAI	606	SIGCOMM	680	EUROCRYPT	379	POPL	803	KDD	607
IJCAI	961	INFOCOM	1109	CRYPTO	265	ASPLOS	300	PKDD	134
ICTAI	207			ASIACRYPT	145	ECOOP	316	CIKM	392
						ICLP	296	SIGIR	423
total	1774	*total*	1789	*total*	789	*total*	1715	*total*	1556

20 NEWSGROUP DATASET				
Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5
Religion	**Hardware**	**Politics**	**Software**	**Sports**
talk.religion.misc	sci.electronics	talk.politics.guns	comp.os.ms-windows.misc	rec.sport.hockey
soc.religion.christian	comp.sys.mac.hardware	talk.politics.mideast	comp.graphics	rec.sport.baseball
alt.atheism	comp.sys.ibm.pc.hardware	talk.politics.misc	comp.windows.x	
2424 posts	2924 posts	2625 posts	2938 posts	1993 posts

Table 1. CiteSeer Dataset Venue Distribution and 20 NG Topic Distribution among five topical clusters

correct clusters whereas the incorrect assignments tend to appear towards the boundaries of the clusters. One thing to note about K-SVMeans is that the objective function in equation 5 is guaranteed to converge to a local maxima since K-SVMeans, as in the case of K-Means, reassigns an object from an old cluster π_i to a new cluster π_j only if the object is more similar to π_j's centroid than π_i's centroid.

5 Data Sets and Evaluation

We ran experiments on a subset of CiteSeer's[1] paper collection and on the 20 Newsgroup dataset to evaluate the clustering performance of K-SVMeans by comparing the predicted cluster of each document with the categorical labels from the document corpus. In both datasets, we are interested in the effect of the authorship of documents for the topical clustering of documents. We use the standard F_1 measure as our evaluation criteria. F_1 measure combines precision(p) and recall(r) with equal weight in the form of $F_1(p,r) = \frac{2.p.r}{p+r}$. The reported results are micro-averaged F_1 scores which gives equal weight to each document, regardless of the size of individual clusters. The characteristics of the datasets and the results are presented in the following subsections. We selected the RBF kernel for the online SVM and ran the experiments with the SVM parameters $C = 100$ and $\gamma = 0.001$ after 10-fold cross validation. For the K-Means clustering section of K-SVMeans algorithm, we used the Gmeans clustering toolkit [6], which we integrated it with the LASVM package [3].

5.1 CiteSeer Dataset

The first dataset we used is a collection of scientific papers extracted from CiteSeer's [9] repository. The categorical distribution of the subset of papers from CiteSeer's collection we used in our experiments is given in Table 1. From each paper, we extracted the title, abstract and keyword sections, and removed stop words. We also removed the words that appear less than three times in the overall collection. In the corpus that we used, there are a total of 7623 papers that have been authored by 5623 researchers. Each author is represented as a collection of the words in the documents that he/she has (co)authored. Since there is a one to many relationship between the documents and the authors, we integrated the effect of order of authorship in the representation of authors in vector form. The weight of feature f_i of author vector $\vec{a_i}$ is

$$\vec{a_i}^{f_j} = \sum_{a_i \in d_k} \frac{1}{Rank(a_i, d_k)} \cdot w(f_j, d_k) \tag{6}$$

where $Rank(a_i, d_k)$ is the rank of authorship of author a_i in document d_k and $w(f_j, d_k)$ is the TF-IDF score of feature f_j in d_k. The author vectors are L_2 normalized to eliminate the effects of different document lengths and number of authored documents. The documents are clustered based on their topics, where the topics of documents are determined from their publication venues.

5.2 20 Newsgroup Dataset

The second collection we used is the 20 Newsgroup dataset, a collection of approximately 20.000 postings to Usenet newsgroups. Each message is authored by one author and is about a single topic. We used the reduced version of the dataset where the cross-posts in the collection are removed and the messages only contain the From and Subject fields in addition to the message body. We combined 14 of the newsgroups in 5 categories for a total of

[1] http://citeseer.ist.psu.edu

Dataset	Distance / Clus. Init.	K-Means	K-SVMeans(x1)	K-SVMeans(x2)	K-SVMeans(x3)
CiteSeer	Spherical / Random	68.418	73.318	76.102	**76.194**
	Spherical / Well Sep.	69.306	75.243	77.713	**80.596**
	Euclidean / Random	55.945	60.284	61.575	**62.082**
	Euclidean / Well Sep.	58.712	64.392	65.941	**66.746**
20 Newsgroup	Spherical / Random	70.792	75.486	76.918	**77.315**
	Spherical / Well Sep.	72.314	77.263	78.368	**78.764**
	Euclidean / Random	52.623	54.978	55.711	**56.013**
	Euclidean / Well Sep.	53.747	55.549	56.292	**56.426**

Table 2. Experimental Results based on the F_1 scores of the clusterings.

12904 messages submitted by 5992 users. The list of newsgroup topics we used and their categorical groupings are given in Table 1. Each unique person is identified from the email address found in the From fields of the messages. Since each message has only one sender, each author has a straightforward representation of the cumulative collection of words found in all of that person's messages. Each message is cleaned from stop words, infrequent words less than three occurences are removed and vectors normalized to unit length. The author vectors are constructed from their corresponding document vectors and the lengths of author vectors are also normalized.

6 Experimental Results

We report results on each dataset for two clustering criterion functions of K-Means, averaged over ten runs. The first clustering algorithm is the Euclidean K-Means that makes the cluster assignment decisions based on the euclidean distances between the document vectors. The second algorithm we used is the Spherical K-Means that uses the cosine distances between documents as the similarity metric.

For both clusterings, we experimented with two separate initialization schemes. In the first scheme, each document is assigned a random cluster ID upon the initialization of K-Means. The second scheme chooses one of the cluster centroids as the farthest point from the center of the whole data set, and all cluster centroids are well separated.

In each experiment, following the completion of K-Means, K-SVMeans initializes each cluster's SVM learner with the authors of 50 documents that are closest to the cluster centroids in the first run. In each successive iteration, we increase our confidence in the clustering achieved in the previous K-SVMeans run, and we increase the number of authors that are used for SVM initialization by %50 of the previous run. The penalty term that accounts for SVM misclassification of authors for the clustering distance function of the documents is empirically set to 1.5.

From Table 2, it can be seen that we were able to outperform K-Means clustering results significantly, regardless of the clustering criterion function, or the initialization scheme of K-Means. K-SVMeans(x2) and K-SVMeans(x3) are the second and third iterations of the clustering, respectively. The inclusion of more and more authors to the SVM initialization set in each successive iteration enables the learners to build accurate models earlier in the clustering solution, and thus, increases the clustering accuracies.

It can be observed that we have obtained higher improvement in clustering accuracies for CiteSeer dataset than the Newsgroup collection. In scientific publications, researchers generally target the venues that lie within their research interests. Therefore, it is easier for the SVM learner to predict the category that a particular author is interested in. On the other hand, since the newsgroup messages are comparably more random, and are based on personal interests, a person's messages may be more distributed across topics, hence making it difficult for the classifier to make

K-Means					
	AI	**COMM**	**CRYPT**	**PL**	**DM**
Cluster 1	**681**	15	11	23	667
Cluster 2	13	**1697**	3	200	23
Cluster 3	428	20	**762**	263	33
Cluster 4	86	23	0	**1103**	43
Cluster 5	566	34	13	126	**790**

K-SVMeans(x3)					
	AI	**COMM**	**CRYPT**	**PL**	**DM**
Cluster 1	**1271**	8	10	70	69
Cluster 2	17	**1658**	1	113	22
Cluster 3	25	56	**770**	42	8
Cluster 4	132	23	1	**1444**	43
Cluster 5	329	44	7	46	**1414**

Table 3. CiteSeer confusion matrix for K-Means and K-SVMeans(x3) for a sample run. The topical clusters are AI : Artificial Intelligence, COMM: Communications, CRYPT: Cryptography, PL: Programming Languages, DM: Data Mining

K-Means					
	REL	**HW**	**POL**	**SW**	**SP**
Cluster 1	**2121**	10	153	11	9
Cluster 2	33	**2165**	114	2041	36
Cluster 3	175	691	**1132**	165	20
Cluster 4	93	41	1206	**711**	1
Cluster 5	2	17	20	10	**1927**

K-SVMeans(x3)					
	REL	**HW**	**POL**	**SW**	**SP**
Cluster 1	**2144**	10	205	21	10
Cluster 2	29	**2478**	39	314	23
Cluster 3	228	50	**2367**	27	4
Cluster 4	17	371	5	**2572**	28
Cluster 5	6	15	9	4	**1928**

Table 4. Newsgroup confusion matrix for K-Means and K-SVMeans(x3) for a sample run. The topical clusters are REL : Religion, HW: Hardware, POL: Politics, SW: Software, SP: Sports

accurate predictions. Even in that case, the SVM classification of authors was able to assist the clustering decisions made by K-Means significantly. In Tables 3 and 4, we show the confusion matrices of CiteSeer and Newsgroup datasets where K-SVMeans has outperformed K-Means by a wide margin. In the CiteSeer dataset, clusters Artificial Intelligence and Data Mining contain many documents that have been incorrectly assigned to each other. The same problem can be observed with the Hardware and Software categories of the Newsgroup dataset. The problem stems from the narrow focus of K-Means which only looks at each document in isolation. The presence of common terms between those categories misleads the K-Means clusterer to make incorrect clustering decisions. A global view that considers the main interests of an author by looking at all of the content generated by that person helps us correctly model his/her interests and enables us to gain better understanding of the nature of the information produced by that author.

7 Conclusions

Traditional clustering algorithms do not handle rich structured data well by either focusing on a single homogeneous type or by disarding the interrelationships between the multiple aspects of the data. Hence, those algorithms are not sufficient to deal with the existing (and emerging) data that is heterogeneous in nature, where relationships between objects can be represented through multiple layers of connectivity and similarity. In this paper, we presented a novel clustering algorithm K-SVMeans which is designed to perform clustering on rich structured multivariate datasets. We have shown that the applicability of Support Vector Machines are not limited to classification problems and SVM classification can greatly effect the performance of clustering algorithms for multivariate datasets. Even in the absence of labeled training instances for SVM, generating labels on-the-fly effectively increases clustering performance. Our experimental results on the integration of authorship analysis with topical clustering of documents show significant improvements over traditional K-Means and confirms that there is great benefit in incorporating additional dimensions of similarity into a unified clustering solution.

References

[1] R. Bekkerman, R. El-Yaniv, and A. McCallum. Multi-way distributional clustering via pairwise interactions. In *Proceedings of ICML'05*, pages 41–48, 2005.

[2] L. Bolelli, S. Ertekin, and C. L. Giles. Clustering scientific literature using sparse citation graph analysis. In *PKDD 2006, 10th European Conference on Principles and Practice of Knowledge Discovery in Databases*, pages 30–41, 2006.

[3] A. Bordes, S. Ertekin, J. Weston, and L. Bottou. Fast kernel classifiers with online and active learning. *Journal of Machine Learning Research*, 6:1579–1619, September 2005.

[4] I. S. Dhillon. Co-clustering documents and words using bipartite spectral graph partitioning. In *Proc. of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 269–274, 2001.

[5] I. S. Dhillon, S. Mallela, and D. S. Modha. Information-theoretic co-clustering. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 89–98, 2003.

[6] I. S. Dhillon and D. S. Modha. Concept decompositions for large sparse text data using clustering. *Machine Learning*, 42(1):143–175, Jan 2001.

[7] B. Gao, T.-Y. Liu, T. Qin, X. Zheng, Q.-S. Cheng, and W.-Y. Ma. Web image clustering by consistent utilization of visual features and surrounding texts. In *MULTIMEDIA '05*, pages 112–121, 2005.

[8] B. Gao, T.-Y. Liu, X. Zheng, Q.-S. Cheng, and W.-Y. Ma. Consistent bipartite graph co-partitioning for star-structured high-order heterogeneous data co-clustering. In *Proceeding of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 41–50, 2005.

[9] C. L. Giles, K. Bollacker, and S. Lawrence. CiteSeer: An automatic citation indexing system. In *The 3rd ACM Conf. on Digital Libraries*, pages 89–98, June 23–26 1998.

[10] J. C. Platt. *Fast training of support vector machines using sequential minimal optimization.* MIT Press, Cambridge, MA, USA, 1999.

[11] C. D. X. He, H. Zha and H. Simon. Web document clustering using hyperlink structures. *Computational Statistics and Data Analysis*, 41:19–45, 2002.

[12] H. Zha, X. He, C. Ding, H. Simon, and M. Gu. Bipartite graph partitioning and data clustering. In *CIKM '01: Proceedings of the tenth international conference on Information and knowledge management*, pages 25–32, New York, NY, USA, 2001. ACM Press.