A Clustering Method For Web Data With Multi-Type Interrelated Components

Levent Bolelli[1], Seyda Ertekin[1], Ding Zhou[1], C. Lee Giles[1,2]

[1]Department of Computer Science and Engineering
[2]School of Information Sciences and Technology
The Pennsylvania State University
University Park, PA 16802
{bolelli, sertekin, dzhou}@cse.psu.edu
giles@ist.psu.edu

ABSTRACT

Traditional clustering algorithms work on "flat" data, making the assumption that the data instances can only be represented by a set of homogeneous and uniform features. Many real world data, however, is heterogeneous in nature, comprising of multiple types of interrelated components. We present a clustering algorithm, K-SVMeans, that integrates the well known K-Means clustering with the highly popular Support Vector Machines(SVM) in order to utilize the richness of data. Our experimental results on authorship analysis of scientific publications show that K-SVMeans achieves better clustering performance than homogeneous data clustering.

Categories and Subject Descriptors

I.5.3 [**Pattern Recognition**]: Clustering, Algorithms

General Terms

Algorithms, Experimentation

Keywords

K-SVMeans, Multi-Type Data Clustering, Online SVM, K-Means

1. INTRODUCTION

Discovery of latent semantic groupings and identification of intrinsic structures in datasets is a crucial task for many data analysis needs. Most real-world data, especially data available on the web, possess rich structural relationships, such as web images and surrounding texts, web pages and hyperlinks, scientific publications and authors. In these examples, the secondary data types are often either neglected by traditional clustering algorithms, or individual clusterings on each dimension are mapped onto a combined clustering solution. The former approach under-utilizes the information available to the clusterer, whereas the latter neglects the structural relationships between the individual data types.

Copyright is held by the author/owner(s).
WWW 2007, May 8–12, 2007, Banff, Alberta, Canada.
ACM 978-1-59593-654-7/07/0005.

We present K-SVMeans clustering, which integrates two sources of information into a single clustering framework. The clustering along the main data type of interest is performed using the popular K-Means algorithm and relational similarity in the additional dimension is learned through Online Support Vector Machines [1]. The most significant advantage of online SVMs is that they are not batch learners and thus, can handle streaming data. The ability to use SVMs in an online setting enables us to efficiently integrate them with unsupervised learning algorithms, and as will be shown later, this combination does not require manually labeled data for SVM training.

2. K-SVMEANS CLUSTERING

The original formulation of K-Means algorithm first initializes p clusters with data objects and then assigns each object x_i, $1 \leq i \leq N$ to a cluster c_j, $1 \leq j \leq p$ where x_i's distance to the representative of its assigned cluster c_j 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 algorithm is independent of any of those variations, but for brevity, we will describe the algorithm for Spherical K-Means with random initialization where each cluster is represented by its centroid vector.

During K-SVMeans clustering process, an SVM is trained for each cluster on the additional(secondary) dimension of the data. For instance, in document clustering, the documents are clustered using K-Means and an SVM for each cluster is trained on the authors of the documents that belong to their respective clusters. The clustering decisions in K-SVMeans can be represented as follows: Let us denote the objects in the primal data type as $X = (x_1, x_2, \cdots, x_n)$ and the second data type as $U = (u_1, u_2, \cdots, u_m)$. Let u^i_j represent the relationship between x_i and u_j and let u^i denote the set of u's connected to x_i. Intermediate cluster assignment decisions in K-SVMeans are determined by two conditions. A data object x_i is moved from cluster c_j to c_k when 1) x_i is closer to c_k's centroid and c_j's SVM classifies u^i as negative and c_k's SVM classifies u^i as positive (both K-Means and SVM have to agree on the cluster assignment change), and 2) in case x_i's candidate cluster c_k's

Distance / Clus. Init.	K-Means	K-SVMeans(x1)	K-SVMeans(x2)	K-SVMeans(x3)
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**

Table 1: Experimental Results based on the F_1 scores of the clustering solutions.

SVM learner decides that u^i do not belong to that cluster (i.e. the decision values of the u^i are negative), then we apply a penalty term on the distance function 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 learners are not adversely effected by the incorrect clustering decisions of K-Means that result in mislabeling of the u^i. Only highly similar x_i are allowed a cluster change in case the SVM classification decision is not trusted. If x_i moves from c_j to c_k, u^i are added to the SVM of c_k as positive, and SVMs of $c_p, p \neq k$ as negative observations.

K-SVMeans can be run in multiple iterations where the initialization of the SVM learner 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 data type 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 $t+1$, K-SVMeans looks at each cluster π_i^t generated in the previous run and selects m objects closest to the centroid of π_i^t and use their associated u's for SVM initialization of c_i. 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.

3. EXPERIMENTS

We conducted experiments on a subset of CiteSeer's[1] repository of scientific literature to evaluate the clustering performance of K-SVMeans by comparing the predicted cluster of each document with the categorical labels from the document corpus. The CiteSeer dataset we used contains 7623 papers from 16 conferences, authored by 5623 distinct authors. The papers are grouped into 5 topical categories based on their publication venues.

Each author a_i is represented as a collection of the words in the documents that a_i has (co)authored. Since each document can potentially have multiple authors, each author is represented as

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

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. We initialize K-SVMeans(x1) with 50 authors from the clustering solution obtained from the K-Means iteration, and increase the number of initialization authors by %50 at each successive iteration of K-SVMeans. The penalty term that accounts for SVM misclassification of authors for the clustering distance function of the documents is set to 1.5 empirically. As the evaluation metric, we used the standard F_1 measure that measures the harmonic mean of precision(p) and recall(r). Our reported results are micro-averaged F_1 scores which gives equal weight to each document and is independent of cluster sizes. For the K-Means clustering section of K-SVMeans algorithm, we used the Gmeans clustering toolkit [2] and we integrated it with the LASVM package [1]. We report results for two clustering criterion functions of K-Means, averaged over ten runs. The first clustering algorithm is the Euclidean K-Means that makes clustering 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 initialization schemes. In the first scheme, each document is initially assigned a random cluster ID. 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.

Our experimental results show that K-SVMeans ourperforms K-Means 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.

4. CONCLUSIONS

Traditional clustering 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. We presented a novel clustering algorithm K-SVMeans which is designed to perform clustering on rich structured multivariate datasets. 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.

5. REFERENCES

[1] 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.

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

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