Learning Spatially Variant Dissimilarity (SVaD) Measures

Krishna Kummamuru
IBM India Research Lab
Block 1, IIT, Hauz Khas
New Delhi 110016 INDIA
kkummamu@in.ibm.com

Raghu Krishnapuram
IBM India Research Lab
Block 1, IIT, Hauz Khas
New Delhi 110016 INDIA
kraghura@in.ibm.com

Rakesh Agrawal
IBM Almaden Research
Center
San Jose, CA 95120, USA
ragrawal@almaden.ibm.com

ABSTRACT

Clustering algorithms typically operate on a feature vector representation of the data and find clusters that are compact with respect to an assumed (dis)similarity measure between the data points in feature space. This makes the type of clusters identified highly dependent on the assumed similarity measure. Building on recent work in this area, we formally define a class of spatially varying dissimilarity measures and propose algorithms to learn the dissimilarity measure automatically from the data. The idea is to identify clusters that are compact with respect to the unknown spatially varying dissimilarity measure. Our experiments show that the proposed algorithms are more stable and achieve better accuracy on various textual data sets when compared with similar algorithms proposed in the literature.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications—*Data Mining*

General Terms

Algorithms

Keywords

Clustering, Learning Dissimilarity Measures

1. INTRODUCTION

Clustering plays a major role in data mining as a tool to discover structure in data. Object clustering algorithms operate on a feature vector representation of the data and find clusters that are compact with respect to an assumed (dis)similarity measure between the data points in feature space. As a consequence, the nature of clusters identified by a clustering algorithm is highly dependent on the assumed similarity measure. The most commonly used dissimilarity measure, namely the Euclidean metric, assumes that the dissimilarity measure is isotropic and spatially invariant, and it is effective only when the clusters are roughly spherical and all of them have approximately the same size, which is rarely the case in practice [8]. The problem of finding non-spherical clusters is often addressed by utilizing a feature weighting technique. These techniques discover a single set of weights such that relevant features are given more importance than irrelevant features. However, in practice, each cluster may have a different set of relevant features. We consider Spatially Varying Dissimilarity (SVaD) measures to address this problem.

Diday et. al. [4] proposed the *adaptive distance dynamic clusters* (ADDC) algorithm in this vain. A fuzzified version of ADDC, popularly known as the Gustafson-Kessel (GK) algorithm [7] uses a dynamically updated covariance matrix so that each cluster can have its own norm matrix. These algorithms can deal with hyperelliposoidal clusters of various sizes and orientations. The EM algorithm [2] with Gaussian probability distributions can also be used to achieve similar results. However, the above algorithms are computationally expensive for high-dimensional data since they invert covariance matrices in every iteration. Moreover, matrix inversion can be unstable when the data is sparse in relation to the dimensionality.

One possible solution to the problems of high computation and instability arising out of using covariance matrices is to force the matrices to be diagonal, which amounts to weighting each feature differently in different clusters. While this restricts the dissimilarity measures to have axis parallel isometry, the weights also provide a simple interpretation of the clusters in terms of relevant features, which is important in knowledge discovery. Examples of such algorithms are SCAD and Fuzzy-SKWIC [5, 6], which perform fuzzy clustering of data while simultaneously finding feature weights in individual clusters.

In this paper, we generalize the idea of the feature weighting approach to define a class of spatially varying dissimilarity measures and propose algorithms that learn the dissimilarity measure automatically from the given data while performing the clustering. The idea is to identify clusters inherent in the data that are compact with respect to the unknown spatially varying dissimilarity measure. We compare the proposed algorithms with a diagonal version of GK (DGK) and a crisp version of SCAD (CSCAD) on a variety of data sets. Our algorithms perform better than DGK and CSCAD, and use more stable update equations for weights than CSCAD.

The rest of the paper is organized as follows. In the next section, we define a general class of dissimilarity measures

KDD'04, August 22–25, 2004, Seattle, Washington, USA.

and formulate two objective functions based on them. In Section 3, we derive learning algorithms that optimize the objective functions. We present an experimental study of the proposed algorithms in Section 4. We compare the performance of the proposed algorithms with that of DGK and CSCAD. These two algorithms are explained in Appendix A. Finally, we summarize our contributions and conclude with some future directions in Section 5.

2. SPATIALLY VARIANT DISSIMILARITY (SVAD) MEASURES

We first define a general class of dissimilarity measures and formulate a few objective functions in terms of the given data set. Optimization of the objective functions would result in learning the underlying dissimilarity measure.

2.1 SVaD Measures

In the following definition, we generalize the concept of dissimilarity measures in which the weights associated with features change over feature space.

DEFINITION 2.1 *We define the measure of dissimilarity of $\boldsymbol{x}$ from $\boldsymbol{y}$*[1] *to be a weighted sum of M dissimilarity measures between $\boldsymbol{x}$ and $\boldsymbol{y}$ where the values of the weights depend on the region from which the dissimilarity is being measured. Let $\mathcal{P} = \{R_1, \ldots, R_K\}$ be a collection of K regions that partition the feature space, and $\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots,$ and $\boldsymbol{w}_K$ be the weights associated with $R_1, R_2, \ldots,$ and R_K, respectively. Let $g_1, g_2, \ldots,$ and g_M be M dissimilarity measures. Then, each $\boldsymbol{w}_j, j = 1, \ldots, K$, is an M-dimensional vector where its l-th component, w_{jl} is associated with g_l. Let W denote the K-tuple $(\boldsymbol{w}_1, \ldots, \boldsymbol{w}_K)$ and let r be a real number. Then, the dissimilarity of $\boldsymbol{x}$ from $\boldsymbol{y}$ is given by:*

$$f_W(\boldsymbol{x}, \boldsymbol{y}) \triangleq \sum_{l=1}^{M} w_{jl}^r g_l(\boldsymbol{x}, \boldsymbol{y}), \;\; \textit{if } \boldsymbol{y} \in R_j. \tag{1}$$

We refer to f_W as a Spatially Variant Dissimilarity (SVaD) measure.

Note that f_W need not be symmetric even if g_i are symmetric. Hence, f_W is not a metric. Moreover, the behavior of f_W depends on the behavior of g_i. There are many ways to define g_i. We list two instances of f_W.

EXAMPLE 2.1 *(Minkowski) Let $\Re^d$ be the feature space and $M = d$. Let a point $\boldsymbol{x} \in \Re^d$ be represented as $(x_1, \ldots, x_d)$. Then, when $g_i(\boldsymbol{x}, \boldsymbol{y}) = |x_i - y_i|^p$ for $i = 1, \ldots, d$, and $p \geq 1$, the resulting SVaD measure, f_W^M is called Minkowski SVaD (MSVaD) measure. That is,*

$$f_W^M(\boldsymbol{x}, \boldsymbol{y}) \triangleq \sum_{l=1}^{d} w_{jl}^r |x_l - y_l|^p, \;\; \textit{if } \boldsymbol{y} \in R_j. \tag{2}$$

One may note that when $\boldsymbol{w}_1 = \cdots = \boldsymbol{w}_K$ and $p = 2$, f_W^M is the weighted Euclidean distance. When $p = 2$, we call f_W^M a Euclidean SVaD (ESVaD) measure and denote it by f_W^E.

EXAMPLE 2.2 *(Cosine) Let the feature space be the set of points with l_2 norm equal to one. That is, $\|\boldsymbol{x}\|_2 = 1$ for all points $\boldsymbol{x}$ in feature space. Then, when $g_l(\boldsymbol{x}, \boldsymbol{y}) = (1/d - x_l \cdot y_l)$ for $l = 1, \ldots, d$, the resulting SVaD measure f_W^C is called a Cosine SVaD (CSVaD) measure:*

$$f_W^C(\boldsymbol{x}, \boldsymbol{y}) \triangleq \sum_{i=1}^{d} w_{jl}^r (1/d - x_l \cdot y_l), \;\; \textit{if } \boldsymbol{y} \in R_j. \tag{3}$$

In the formulation of the objective function below, we use a set of parameters to represent the regions $R_1, R_2, \ldots,$ and R_K. Let $\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots,$ and $\boldsymbol{c}_K$ be K points in feature space. Then $\boldsymbol{y} \in R_j$ iff

$$f_W(\boldsymbol{y}, \boldsymbol{c}_j) < f_W(\boldsymbol{y}, \boldsymbol{c}_i) \text{ for } i \neq j. \tag{4}$$

In the case of ties, $\boldsymbol{y}$ is assigned to the region with the lowest index. Thus, the K-tuple of points $C = (\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_K)$ defines a partition in feature space. The partition induced by the points in C is similar in nature to a Voronoi tessellation. We use the notation $f_{W,C}$ whenever we use the set C to parameterize the regions used in the dissimilarity measure.

2.2 Objective Function for Clustering

The goal of the present work is to identify the spatially varying dissimilarity measure and the associated compact clusters simultaneously. It is worth mentioning here that, as in the case of any clustering algorithm, the underlying assumption in this paper is the existence of such a dissimilarity measure and clusters for a given data set.

Let $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots,$ and $\boldsymbol{x}_n$ be n given data points. Let K be a given positive integer. Assuming that C represents the cluster centers, let us assign each data point $\boldsymbol{x}_i$ to a cluster R_j with the closest $\boldsymbol{c}_j$ as the cluster center[2], i.e.,

$$j = \arg\min_l f_{W,C}(\boldsymbol{x}_i, \boldsymbol{c}_l). \tag{5}$$

Then, the within-cluster dissimilarity is given by

$$J(W, C) = \sum_{j=1}^{K} \sum_{\boldsymbol{x}_i \in R_j} \sum_{l=1}^{M} w_{jl}^r g_l(\boldsymbol{x}_i, \boldsymbol{c}_j). \tag{6}$$

$J(W, C)$ represents the sum of the dissimilarity measures of all the data points from their closest centroids. The objective is to find W and C that minimize $J(W, C)$. To avoid the trivial solution to $J(W, C)$, we consider a normalization condition on $\boldsymbol{w}_j$, viz.,

$$\sum_{l=1}^{M} w_{jl} = 1. \tag{7}$$

Note that even with this condition, $J(W, C)$ has a trivial solution: $w_{jp} = 1$ where $p = \arg\min_l \sum_{\boldsymbol{x}_i \in R_j} g_l(\boldsymbol{x}_i, \boldsymbol{c}_j)$, and the remaining weights are zero. One way to avoid convergence of $\boldsymbol{w}_j$ to unit vectors is to impose a regularization condition on $\boldsymbol{w}_j$. We consider the following two regularization measures in this paper: (1) *Entropy measure:* $\sum_{l=1}^{M} w_{jl} \log(w_{jl})$ and (2) *Gini measure:* $\sum_{l=1}^{M} w_{jl}^2$.

[1] We use the phrase "dissimilarity of $\boldsymbol{x}$ from $\boldsymbol{y}$" rather than "dissimilarity between $\boldsymbol{x}$ and $\boldsymbol{y}$" because we consider a general situation where the dissimilarity measure depends on the location of $\boldsymbol{y}$. As an example of this situation in text mining, when the dissimilarity is measured from a document on '*terrorism*' to a document $\boldsymbol{x}$, a particular set of keywords may be weighted heavily whereas when the dissimilarity is measured from a document on '*football*' to $\boldsymbol{x}$, a different set of keywords may be weighted heavily.

[2] We use $\mathcal{P} = \{R_1, R_2, \ldots, R_K\}$ to represent the corresponding partition of the data set as well. The intended interpretation (cluster or region) would be evident from the context.

3. ALGORITHMS TO LEARN SVAD MEASURES

The problem of determining the optimal W and C is similar to the traditional clustering problem that is solved by the K-Means Algorithm (KMA) except for the additional W matrix. We propose a class of iterative algorithms similar to KMA. These algorithms start with a random partition of the data set and iteratively update C, W and $\mathcal{P}$ so that $J(W, C)$ is minimized. These iterative algorithms are instances of Alternating Optimization (AO) algorithms. In [1], it is shown that AO algorithms converge to a local optimum under some conditions. We outline the algorithm below before actually describing how to update C, W and $\mathcal{P}$ in every iteration.

```
  Randomly assign the data points to $K$ clusters.
REPEAT
  Update $C$: Compute the centroid of each cluster $\boldsymbol{c}_j$.
  Update $W$: Compute the $w_{jl} \forall j, l$.
  Update $\mathcal{P}$: Reassign the data points to the clusters.
UNTIL (termination condition is reached).
```

In the above algorithm, the update of C depends on the definition of g_i, and the update of W on the regularization terms. The update of $\mathcal{P}$ is done by reassigning the data points according to (5). Before explaining the computation of C in every iteration for various g_i, we first derive update equations for W for various regularization measures.

3.1 Update of Weights

While updating weights, we need to find the values of weights that minimize the objective function for a given C and $\mathcal{P}$. As mentioned above, we consider the two regularization measures for w_{jl} and derive update equations. If we consider the entropy regularization with $r = 1$, the objective function becomes:

$$\begin{aligned} J_{ENT}(W, C) &= \sum_{j=1}^{K} \sum_{\boldsymbol{x}_i \in R_j} \sum_{l=1}^{M} w_{jl} g_l(\boldsymbol{x}_i, \boldsymbol{c}_j) \\ &+ \sum_{j=1}^{K} \delta_j \sum_{l=1}^{M} w_{jl} \log(w_{jl}) + \sum_{j=1}^{K} \lambda_j \left(\sum_{l=1}^{M} w_{jl} - 1 \right). \end{aligned} \quad (8)$$

Note that λ_j are the Lagrange multipliers corresponding to the normalization constraints in (7), and δ_j represent the relative importance given to the regularization term relative to the within-cluster dissimilarity. Differentiating $J_{ENT}(W, C)$ with respect to w_{jl} and equating it to zero, we obtain $w_{jl} = \exp\left(\frac{-(\lambda_j + \sum_{\boldsymbol{x}_i \in R_j} g_l(\boldsymbol{x}_i, \boldsymbol{c}_j))}{\delta_j} - 1 \right)$. Solving for λ_j by substituting the above value of w_{jl} in (7) and substituting the value of λ_j back in the above equation, we obtain

$$w_{jl} = \frac{\exp\left(- \sum_{\boldsymbol{x}_i \in R_j} g_l(\boldsymbol{x}_i, \boldsymbol{c}_j) / \delta_j \right)}{\sum_{n=1}^{M} \exp\left(- \sum_{\boldsymbol{x}_i \in R_j} g_n(\boldsymbol{x}_i, \boldsymbol{c}_j) / \delta_j \right)}. \quad (9)$$

If we consider the Gini measure for regularization with $r = 2$, the corresponding w_{jl} that minimizes the objective function can be shown to be

$$w_{jl} = \frac{1/(\delta_j + \sum_{\boldsymbol{x}_i \in R_j} g_l(\boldsymbol{x}_i, \boldsymbol{c}_j))}{\sum_{n=1}^{M} (1/(\delta_j + \sum_{\boldsymbol{x}_i \in R_j} g_n(\boldsymbol{x}_i, \boldsymbol{c}_j)))}. \quad (10)$$

In both cases, the updated value of w_{jl} is inversely related to $\sum_{\boldsymbol{x}_i \in R_j} g_l(\boldsymbol{x}_i, \boldsymbol{c}_j)$. This has various interpretations based on the nature of g_l. For example, when we consider the ESVaD measure, w_{jl} is inversely related to the variance of l-th element of the data vectors in the j-th cluster. In other words, when the variance along a particular dimension is high in a cluster, then the dimension is less important to the cluster. This popular heuristic has been used in various contexts (such as relevance feedback) in the literature [9]. Similarly, when we consider the CSVaD measure, w_{jl} is directly proportional to the correlation of the j-th dimension in the l-th cluster.

Algorithm	Update Equations		
Acronyms	$\mathcal{P}$	C	W
EEnt	(5)	(11)	(9)
EsGini	(5)	(11)	(10)
CEnt	(5)	(12)	(9)
CsGini	(5)	(12)	(10)

Table 1: Summary of algorithms.

3.2 Update of Centroids

Learning ESVaD Measures: Substituting the ESVaD measure in the objective function and solving the first order necessary conditions, we observe that

$$c_{jl} = \frac{1}{|R_j|} \sum_{\boldsymbol{x}_i \in R_j} x_{il} \quad (11)$$

minimizes $J_{ESVAD}(W, C)$.

Learning CSVaD Measures: Let $x'_{il} = w_{jl} x_{il}$, then using the Cauchy-Swartz inequality, it can be shown that

$$c_{jl} = \frac{1}{|R_j|} \sum_{\boldsymbol{x}_i \in R_j} x'_{il} \quad (12)$$

maximizes $\sum_{\boldsymbol{x}_i \in R_j} \sum_{l=1}^{d} w_{jl} x_{il} c_{jl}$. Hence, (12) also minimizes the objective function when CSVaD is used as the dissimilarity measure.

Table 1 summarizes the update equations used in various algorithms. We refer to this set of algorithms as SVaD learning algorithms.

4. EXPERIMENTS

In this section, we present an experimental study of the algorithms described in the previous sections. We applied the proposed algorithms on various text data sets and compared the performance of EEnt and EsGini with that of K-Means, CSCAD and DGK algorithms. The reason for choosing the K-Means algorithm (KMA) apart from CSCAD and DGK is that it provides a baseline for assessing the advantages of feature weighting. KMA is also a popular algorithm for text clustering. We have included a brief description of CSCAD and DGK algorithms in Appendix A.

Text data sets are sparse and high dimensional. We consider standard labeled document collections and test the proposed algorithms for their ability to discover dissimilarity measures that distinguish one class from another without actually considering the class labels of the documents. We measure the success of the algorithms by the purity of the regions that they discover.

4.1 Data Sets

We performed our experiments on three standard data sets: 20 News Group, Yahoo K1, and Classic 3. These data sets are described below.

20 News Group[3]: We considered different subsets of 20 News Group data that are known to contain clusters of varying degrees of separation [10]. As in [10], we considered three random samples of three subsets of the 20 News Group data. The subsets denoted by *Binary* has 250 documents each from talk.politics.mideast and talk.politics.misc. *Multi5* has 100 documents each from comp.graphics, rec.motorcycles, rec.sport.baseball, sci.space, and talk.politics.mideast. Finally, *Multi10* has 50 documents each from alt.atheism, comp. sys.mac.hardware, misc.forsale, rec.autos, rec.sport.hockey, sci.crypt, sci.electronics, sci.med, sci.space, and talk.politics. gun. It may be noted that *Binary* data sets have two highly overlapping classes. Each of *Multi5* data sets has samples from 5 distinct classes, whereas *Multi10* data sets have only a few samples from 10 different classes. The size of the vocabulary used to represent the documents in *Binary* data set is about 4000, *Multi5* about 3200 and *Multi10* about 2800. We observed that the relative performance of the algorithms on various samples of *Binary*, *Multi5* and *Multi10* data sets was similar. Hence, we report results on only one of them.

Yahoo K1[4]: This data set contains 2340 Reuters news articles downloaded from Yahoo in 1997. There are 494 from Health, 1389 from Entertainment, 141 from Sports, 114 from Politics, 60 from Technology and 142 from Business. After preprocessing, the documents from this data set are represented using 12015 words. Note that this data set has samples from 6 different classes. Here, the distribution of data points across the class is uneven, ranging from 60 to 1389.

Classic 3[5]: Classic 3 data set contains 1400 aerospace systems abstracts from the Cranfield collection, 1033 medical abstracts from the Medline collection and 1460 information retrieval abstracts from the Cisi collection, making up 3893 documents in all. After preprocessing, this data set has 4301 words. The points are almost equally distributed among the three distinct classes.

The data sets were preprocessed using two major steps. First, a set of words (vocabulary) is extracted and then each document is represented with respect to this vocabulary. Finding the vocabulary includes: (1) elimination of the standard list of stop words from the documents, (2) application of Porter stemming[6] for term normalization, and (3) keeping only the words which appear in at least 3 documents. We represent each document by the unitized frequency vector.

4.2 Evaluation of Algorithms

We use the accuracy measure to compare the performance of various algorithms. Let a_{ij} represent the number of data points from class i that are in cluster j. Then the accuracy of the partition is given by $\sum_j \max_i a_{ij}/n$ where n is the total number of data points.

It is to be noted that points coming from a single class need not form a single cluster. There could be multiple clusters in a class that represent sub-classes. We study the performance of SVaD learning algorithms for various values of K, i.e., the number of clusters.

Iteration	0	1	2	3	4	5
$J(W,C)$		334.7	329.5	328.3	328.1	327.8
Accuracy	73.8	80.2	81.4	81.6	82	82

Table 2: Evolution of $J(W,C)$ and Accuracies with iterations when EEnt applied on a Multi5 data.

4.3 Experimental Setup

In our implementations, we have observed that the proposed algorithms, if applied on randomly initialized centroids, show unstable behavior. One reason for this behavior is that the number of parameters that are estimated in feature-weighting clustering algorithms is twice as large as that estimated by the traditional KMA. We, therefore, first estimate the cluster centers giving equal weights to all the dimensions using KMA and then fine-tune the cluster centers and the weights using the feature-weighting clustering algorithms. In every iteration, the new sets of weights are updated as follows. Let $w_n(t+1)$ represent the weights computed using one of (9), (10), (14) or (15) in iteration $(t+1)$ and $w(t)$ the weights in iteration t. Then, the weights in iteration $(t+1)$ are

$$w(t+1) = (1 - \lambda(t))w(t) + \lambda(t)w_n(t+1), \qquad (13)$$

where $\lambda(t) \in [0,1]$ decreases with t. That is, $\lambda(t) = \alpha\lambda(t-1)$, for a given constant $\alpha \in [0,1]$. In our experiments, we observed that the variance of purity values for different initial values of $\lambda(0)$ and α above 0.5 is very small. Hence, we report the results for $\lambda(0) = 0.5$ and $\alpha = 0.5$. We set the value of $\delta_j = 1$.

It may be noted that when the documents are represented as unit vectors, KMA with the cosine dissimilarity measure and Euclidean distance measure would yield the same clusters. This is essentially the same as Spherical K-Means algorithms described in [3]. Therefore, we consider only the weighted Euclidean measure and restrict our comparisons to EEnt and EsGini in the experiments.

Since the clusters obtained by KMA are used to initialize all other algorithms considered here, and since the results of KMA are sensitive to initialization, the accuracy numbers reported in this section are averages over 10 random initializations of KMA.

4.4 Results and Observations

4.4.1 *Effect of SVaD Measures on Accuracies*

In Table 2, we show a sample run of EEnt algorithm on one of the Multi5 data sets. This table shows the evolution of $J(W,C)$ and the corresponding accuracies of the clusters with the iterations. The accuracy, shown at iteration 0, is that of the clusters obtained by KMA. The purity of clusters increases with decrease in the value of the objective function defined using SVaD measures. We have observed a similar behavior of EEnt and EsGini on other data sets also. This validates our hypothesis that SVaD measures capture the underlying structure in the data sets more accurately.

[3] http://www-2.cs.cmu.edu/afs/cs.cmu.edu/project/theo-20/www/data/news20.tar.gz

[4] ftp://ftp.cs.umn.edu/dept/users/boley/PDDPdata/doc-K

[5] ftp://ftp.cs.cornell.edu/pub/smart

[6] http://www.tartarus.org/˜martin/PorterStemmer/

4.4.2 Comparison with Other Algorithms

Figure 1 to Figure 5 show average accuracies of various algorithms on the 5 data sets for various number of clusters. The accuracies of KMA and DGK are very close to each other and hence, in the figures, the lines corresponding to these algorithms are indistinguishable. The lines corresponding to CSCAD are also close to that of KMA in all the cases except Class 3.

General observations: The accuracies of SVaD algorithms follow the trend of the accuracies of other algorithms. In all our experiments, both SVaD learning algorithms improve the accuracies of clusters obtained by KMA. It is observed in our experiments that the improvement could be as large as 8% in some instances. EEnt and EsGini consistently perform better than DGK on all data sets and for all values of K. EEnt and EsGini perform better than CSCAD on all data sets excepts in the case of Classic 3 and for a few values of K.

Note that the weight update equation of CSCAD (15) may result in negative values of w_{jl}. Our experience with CSCAD shows that it is quite sensitive to initialization and it may have convergence problems. In contrast, it may be observed that w_{jl} in (9) and (10) are always positive. Moreover, in our experience, these two versions are much less sensitive to the choice of δ_j.

Data specific observations: When $K = 2$, EEnt and EsGini could not further improve the results of KMA on the *Binary* data set. The reason is that the data set contains two highly overlapping classes. However, for other values of K, they marginally improve the accuracies.

In the case of *Multi5*, the accuracies of the algorithms are non-monotonic with K. The improvement of accuracies is large for intermediate values of K and small for extreme values of K. When $K = 5$, KMA finds relatively stable clusters. Hence, SVaD algorithms are unable to improve the accuracies as much as they did for intermediate values of K. For larger values of K, the clusters are closely spaced and hence there is little scope for improvement by the SVaD algorithms.

Multi10 data sets are the toughest to cluster because of the large number of classes present in the data. In this case, the accuracies of the algorithms are monotonically increasing with the number of clusters. The extent of improvement of accuracies of SVaD algorithms over KMA is almost constant over the entire range of K. This reflects the fact that the documents in *Multi10* data set are uniformly distributed over feature space.

The distribution of documents in Yahoo K1 data set is highly skewed. The extent of improvements that the SVaD algorithms could achieve decrease with K. For higher values of K, KMA is able to find almost pure sub-clusters, resulting in accuracies of about 90%. This leaves little scope for improvement.

The performance of CSCAD differs noticeably in the case of Classic 3. It performs better than the SVaD algorithms for $K = 3$ and better than EEnt for $K = 9$. However, for larger values of K, the SVaD algorithms perform better than the rest. As in the case of *Multi5*, the improvements of SVaD algorithms over others are significant and consistent. One may recall that *Multi5* and Classic 3 consist of documents from distinct classes. Therefore, this observation implies that when there are distinct clusters in the data set, KMA yields confusing clusters when the number of clusters is over-specified. In this scenario, EEnt and EsGini can fine-tune the clusters to improve their purity.

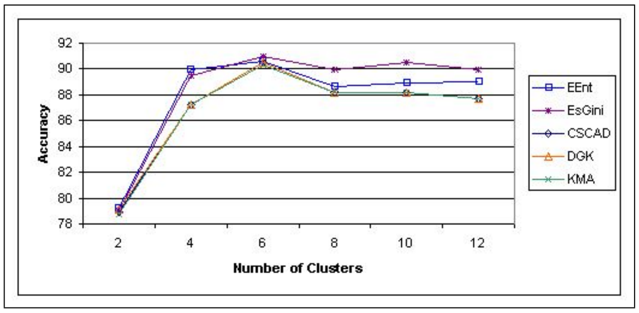

Figure 1: Accuracy results on *Binary* data.

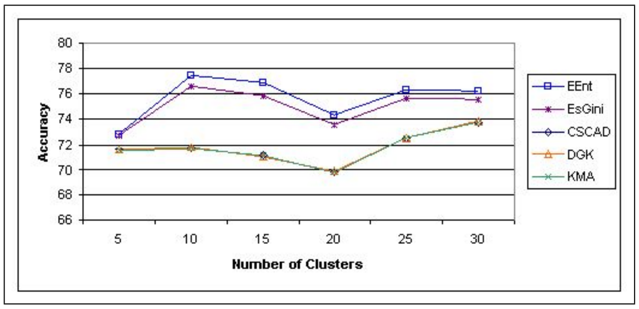

Figure 2: Accuracy results on *Multi5* data.

5. SUMMARY AND CONCLUSIONS

We have defined a general class of spatially variant dissimilarity measures and proposed algorithms to learn the measure underlying a given data set in an unsupervised learning framework. Through our experiments on various textual data sets, we have shown that such a formulation of dissimilarity measure can more accurately capture the hidden structure in the data than a standard Euclidean measure that does not vary over feature space. We have also shown that the proposed learning algorithms perform better than other similar algorithms in the literature, and have better stability properties.

Even though we have applied these algorithms only to text data sets, the algorithms derived here do not assume any specific characteristics of textual data sets. Hence, they

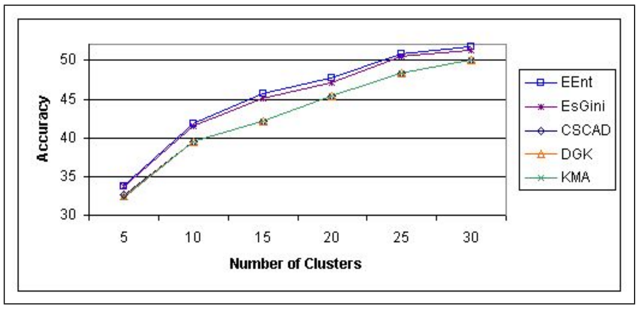

Figure 3: Accuracy results on *Multi10* data.

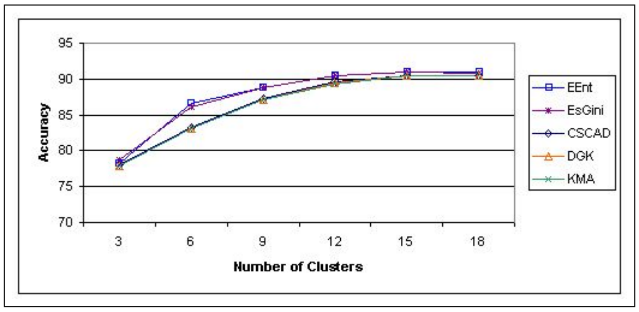

Figure 4: Accuracy results on Yahoo K1 data.

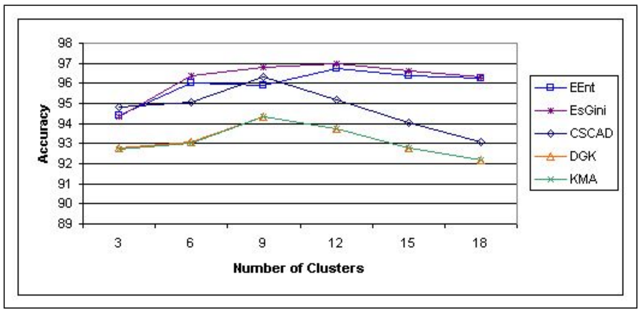

Figure 5: Accuracy results on Classic 3 data.

are applicable to general data sets. Since the algorithms perform better for larger K, it would be interesting to investigate whether they can be used to find subtopics of a topic. Finally, it will be interesting to learn SVaD measures for labeled data sets.

6. REFERENCES

[1] J. C. Bezdek and R. J. Hathaway. Some notes on alternating optimization. In *Proceedings of the 2002 AFSS International Conference on Fuzzy Systems. Calcutta*, pages 288–300. Springer-Verlag, 2002.

[2] A. P. Dempster, N. M. Laird, and Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal Royal Statistical Society B*, 39(2):1–38, 1977.

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

[4] E. Diday and J. C. Simon. Cluster analysis. In K. S. Fu, editor, *Pattern Recognition*, pages 47–94. Springer-Verlag, 1976.

[5] H. Frigui and O. Nasraoui. Simultaneous clustering and attribute discrimination. In *Proceedings of FUZZIEEE*, pages 158–163, San Antonio, 2000.

[6] H. Frigui and O. Nasraoui. Simultaneous categorization of text documents and identification of cluster-dependent keywords. In *Proceedings of FUZZIEEE*, pages 158–163, Honolulu, Hawaii, 2001.

[7] D. E. Gustafson and W. C. Kessel. Fuzzy clustering with the fuzzy covariance matrix. In *Proccedings of IEEE CDC*, pages 761–766, San Diego, California, 1979.

[8] R. Krishnapuram and J. Kim. A note on fuzzy clustering algorithms for Gaussian clusters. *IEEE Transactions on Fuzzy Systems*, 7(4):453–461, Aug 1999.

[9] Y. Rui, T. S. Huang, and S. Mehrotra. Relevance feedback techniques in interactive content-based image retrieval. In *Storage and Retrieval for Image and Video Databases (SPIE)*, pages 25–36, 1998.

[10] N. Slonim and N. Tishby. Document clustering using word clusters via the information bottleneck method. In *Proceedings of SIGIR*, pages 208–215, 2000.

APPENDIX

A. OTHER FEATURE WEIGHTING CLUSTERING TECHNIQUES

A.1 Diagonal Gustafson-Kessel (DGK)

Gustafson and Kessel [7] associate each cluster with a different norm matrix. Let $\mathcal{A} = (A_1, \ldots, A_k)$ be the set of k norm matrices associated with k clusters. Let u_{ji} is the fuzzy membership of $\boldsymbol{x}_i$ in cluster j and $U = [u_{ji}]$. By restricting A_js to be diagonal and $u_{ji} \in \{0, 1\}$, we can reformulate the original optimization problem in terms of SVaD measures as follows:

$$\min_{C,W} J_{DGK}(C, W) = \sum_{j=1}^{k} \sum_{\boldsymbol{x}_i \in R_j} \sum_{l=1}^{M} w_{jl} g_l(\boldsymbol{x}_i, \boldsymbol{c}_j),$$

subject to $\prod_l w_{jl} = \rho_j$. Note that this problem can be solved using the same AO algorithms described in Section 3. Here, the update for C and $\mathcal{P}$ would remain the same as that discussed in Section 3. It can be easily shown that when $\rho_j = 1, \forall j$,

$$w_{jl} = \frac{\left(\prod_{m=1}^{M} \sum_{\boldsymbol{x}_i \in R_j} g_m(\boldsymbol{x}_i, \boldsymbol{c}_j) \right)^{1/M}}{\sum_{\boldsymbol{x}_i \in R_j} g_l(\boldsymbol{x}_i, \boldsymbol{c}_j)} \tag{14}$$

minimize J_{DGK} for a given C.

A.2 Crisp Simultaneous Clustering and Attribute Discrimination (CSCAD)

Frigui et. al. in [5, 6], considered a fuzzy version of the feature-weighting based clustering problem (SCAD). To make a fair comparison of our algorithms with SCAD, we derive its crisp version and refer to it as Crisp SCAD (CSCAD). In [5, 6], the Gini measure is used for regularization. If the Gini measure is considered with $r = 1$, the weights w_{jl} that minimize the corresponding objective function for a given C and $\mathcal{P}$, are given by

$$w_{jl} = \frac{1}{M} + \frac{1}{2\delta_j} \left[\frac{1}{M} \sum_{n=1}^{M} \sum_{\boldsymbol{x}_i \in R_j} g_n(\boldsymbol{x}_i, \boldsymbol{c}_j) - \sum_{\boldsymbol{x}_i \in R_j} g_l(\boldsymbol{x}_i, \boldsymbol{c}_j) \right]. \tag{15}$$

Since SCAD uses the weighted Euclidean measure, the update equations of centroids in CSCAD remain the same as in (11). The update equation for w_{jl} in SCAD is quite similar to (15). One may note that, in (15), the value of w_{jl} can become negative. In [5], a heuristic is used to estimate the value δ_j in every iteration and set the negative values of w_{jl} to zero before normalizing the weights.