Machine Learning, 52, 217–237, 2003

Feature Weighting in *k*-Means Clustering

DHARMENDRA S. MODHA dmodha@almaden.ibm.com
W. SCOTT SPANGLER spangles@almaden.ibm.com
IBM Almaden Research Center, 650 Harry Road, San Jose, CA 95120, USA

Editor: Douglas Fisher

Abstract. Data sets with multiple, heterogeneous feature spaces occur frequently. We present an abstract framework for integrating multiple feature spaces in the *k*-means clustering algorithm. Our main ideas are (i) to represent each data object as a tuple of multiple feature vectors, (ii) to assign a suitable (and possibly different) distortion measure to each feature space, (iii) to combine distortions on different feature spaces, in a convex fashion, by assigning (possibly) different relative weights to each, (iv) for a fixed weighting, to cluster using the proposed *convex k-means algorithm*, and (v) to determine the optimal feature weighting to be the one that yields the clustering that *simultaneously* minimizes the average within-cluster dispersion and maximizes the average between-cluster dispersion along *all* the feature spaces. Using precision/recall evaluations and known ground truth classifications, we empirically demonstrate the effectiveness of feature weighting in clustering on several different application domains.

Keywords: clustering, convexity, convex *k*-means algorithm, feature combination, feature selection, Fisher's discriminant analysis, text mining, unsupervised learning

1. Introduction

1.1. *Multiple, heterogeneous feature spaces*

The fundamental starting point for machine learning, multivariate statistics, or data mining is the assumption that a data object can be represented as a high-dimensional feature vector. In many traditional applications, all the features are essentially of the same "type." However, many emerging real-life data sets often consist of many different feature spaces, for example:

- Image indexing and searching systems use at least four different types of features: color, texture, shape, and location (Flickner et al., 1995).
- Hypertext documents contain at least three different types of features: the words, the out-links (forward links), and the in-links (backward links) (Modha & Spangler, 2000).
- XML data objects may have a number of different textual, referential, graphical, numerical, and categorical features.
- Profile of a typical Amazon.com customer may contain purchased books, music, toys, software, DVD/video, etc.
- For a publicly traded security, one may be interested in the time series' of its stock price, traded volume, fraction held short, and earnings history as well the current members of the board of directors, etc.

These examples illustrate that data sets with multiple, heterogeneous features are indeed natural and common. In addition, many data sets on the UCI Machine Learning and KDD repositories contain data sets with heterogeneous features (Blake & Merz, 1998; Bay, 1999).

1.2. The problem, our contributions, and outline

Clustering is a fundamental technique of unsupervised learning in machine learning and statistics (Hartigan, 1975; Duda & Hart, 1973). Clustering is generally used to find groups of similar items in a set of unlabeled data. We consider the problem of clustering data with multiple, heterogeneous feature spaces.

Given a data set with m feature spaces, we represent each data object as a tuple of m feature vectors. It is implicitly assumed that features within a feature space are "homogeneous" and that different feature spaces are "heterogeneous". A practical question is: For a given data set, how are the various feature spaces determined?

A necessary condition for a set of scalar features to be *homogeneous* is if an intuitively meaningful *symmetric distortion* can be defined on them. A sufficient condition for two sets of features to be considered *heterogeneous* is if, after clustering, we would like to interpret clusters along one set of features independently of the clusters along the other set of features or, conversely, if we would like to study "associations" or "causality" between clusters along different feature spaces. Finally, for computational reasons, if two features can be treated as homogeneous, we should only reluctantly treat them otherwise. In various applications, determination of appropriate feature spaces is often clear. For example, suppose we are interested in clustering hypertext documents using words, out-links, and in-links, then we may naturally group word, out-link, and in-link frequencies into three different feature vectors. As another example, suppose we are interested in clustering data with numerical and categorical features, then, at the very least, we may consider two feature spaces.

We now summarize our results and outline the organization of the paper.

- To cluster, we need a measure of "distortion" between two data objects. Since different types of features may have radically different statistical distributions, in general, it is unnatural to disregard fundamental differences between various different types of features and to impose a uniform, unweighted distortion measure across disparate feature spaces. In Section 2, as our *first contribution*, we define a distortion between two data objects as a weighted sum of suitable distortion measures on individual component feature vectors; where the distortions on individual components are allowed to be different.
- In Section 3, as our *second contribution*, using a convex optimization formulation, we generalize the classical Euclidean k-means algorithm to employ the weighted distortion measure. The generalized algorithm is termed the *convex k-means*, and it simultaneously extends the Euclidean k-means, the spherical k-means (Dhillon & Modha, 2001), and the toric k-means (Modha & Spangler, 2000).
- In Section 4, as our *third and main contribution*, by generalizing Fisher's discriminant analysis, we define the optimal feature weighting as one that leads to a clustering that *simultaneously* minimizes the average within-cluster dispersion and maximizes the average between-cluster dispersion along *all* the feature spaces. In general, optimal feature

weightings cannot be empirically computed, since (i) much like the Euclidean k-means, being a form of gradient-descent heuristic, our convex k-means is also susceptible to local minima, and (ii) for computational reasons, we search only over a coarse grid on the space of possible feature weightings. When discussing empirical results, we use the phrase *optimal feature weighting* to mean *optimal feature weighting within computational and heuristic constraints*.

- In Section 5, we briefly outline our evaluation strategy. In Sections 6 and 7, we demonstrate two concrete applications of our framework, respectively, clustering data sets with numerical and categorical features and clustering text data sets with words, 2-phrases, and 3-phrases. Using data sets with a known *ground truth* classification, we empirically demonstrate the *surprising and reassuring fact* that clusterings corresponding to optimal feature weightings deliver nearly the best precision/recall performance. We outline future work in Section 8, and present conclusions in Section 9.

1.3. Prior work

Wettschereck, Aha, and Mohri (1997) have pointed out that feature weighting may be thought of as a generalization of feature selection where the latter restricts attention to weights that are either 1 (retain the feature) or 0 (eliminate the feature). Feature selection in the context of supervised learning has a long history in machine learning, see, for example, Blum and Langley (1997), Caruana and Freitag (1994), John, Kohavi, and Pfleger (1994), and Koller and Sahami (1996). In contrast, feature selection in the context of unsupervised learning has only recently been systematically studied. For example, Devaney and Ram (1997) and Talavera (1999) have focussed on adding feature selection to conceptual clustering of Fisher (1987) and Vaithyanathan and Dom (1999) have considered the joint problem of feature and model selection for clustering. Recently, we considered the problem of clustering hypertext documents using words, out-links, and in-links where each of the three feature spaces was adaptively weighted (Modha & Spangler, 2000). This last work provided the starting point for the developments reported in this paper.

2. Data model and a distortion measure

2.1. Data model

Assume that we are given a set of data objects where each object is a tuple of m component feature vectors. We write a data object as

$$\mathbf{x} = (\mathbf{F}_1, \mathbf{F}_2, \ldots, \mathbf{F}_m),$$

where the l-th component feature vector $\mathbf{F}_l$, $1 \leq l \leq m$, is a column vector and lies in some (abstract) feature space $\mathcal{F}_l$. The data object $\mathbf{x}$ lies on the m-fold product feature space $\mathcal{F} = \mathcal{F}_1 \times \mathcal{F}_2 \times \cdots \mathcal{F}_m$. The feature spaces $\{\mathcal{F}_l\}_{l=1}^m$ can possess different dimensions and topologies, hence, our data model accommodates heterogeneous types of features. We present two generic examples of feature spaces that are used in this paper.

Euclidean case: $\mathcal{F}_l$ is either $\mathbb{R}^{f_l}$, $f_l \geq 1$, or some compact submanifold thereof.

Spherical case: $\mathcal{F}_l$ is the intersection of the f_l-dimensional, $f_l \geq 1$, unit sphere with the set of vectors in $\mathbb{R}^{f_l}$ with only non-negative components, namely, the non-negative orthant of $\mathbb{R}^{f_l}$.

2.2. *A weighted distortion measure*

Suppose we are interested in measuring distortion between two given two data objects $\mathbf{x} = (\mathbf{F}_1, \mathbf{F}_2, \ldots, \mathbf{F}_m)$ and $\tilde{\mathbf{x}} = (\tilde{\mathbf{F}}_1, \tilde{\mathbf{F}}_2, \ldots, \tilde{\mathbf{F}}_m)$. For $1 \leq l \leq m$, let D_l denote a *distortion measure* between the corresponding component feature vectors $\mathbf{F}_l$ and $\tilde{\mathbf{F}}_l$. Mathematically, we only demand two properties of the distortion function, namely, non-negativity and convexity, respectively:

- $D_l : \mathcal{F}_l \times \mathcal{F}_l \to [0, \infty)$.
- For a fixed $\mathbf{F}_l$, D_l is convex in $\tilde{\mathbf{F}}_l$.

In addition, for methodological reasons, we demand that D_l also be *symmetric*, that is, for any permutation σ of $\{1, 2, \ldots, f_l\}$:

$$D_l\big((g_1, \ldots, g_{f_l}), (g'_1, \ldots, g'_{f_l})\big) = D_l\big((g_{\sigma(1)}, \ldots, g_{\sigma(f_l)}), (g'_{\sigma(1)}, \ldots, g'_{\sigma(f_l)})\big), \quad (1)$$

where $(g_1, \ldots, g_{f_l})$ and $(g'_1, \ldots, g'_{f_l})$ are vectors in $\mathcal{F}_l$.

Euclidean case: The squared-Euclidean distance

$$D_l(\mathbf{F}_l, \tilde{\mathbf{F}}_l) = (\mathbf{F}_l - \tilde{\mathbf{F}}_l)^T(\mathbf{F}_l - \tilde{\mathbf{F}}_l)$$

trivially satisfies the non-negativity and, for $\lambda \in [0, 1]$, the convexity follows from

$$D_l(\mathbf{F}_l, \lambda\tilde{\mathbf{F}}'_l + (1 - \lambda)\tilde{\mathbf{F}}''_l) \leq \lambda D_l(\mathbf{F}_l, \tilde{\mathbf{F}}'_l) + (1 - \lambda)D_l(\mathbf{F}_l, \tilde{\mathbf{F}}''_l).$$

Spherical case: The cosine distance

$$D_l(\mathbf{F}_l, \tilde{\mathbf{F}}_l) = 2\big(1 - \mathbf{F}_l^T\tilde{\mathbf{F}}_l\big)$$

trivially satisfies the non-negativity and, for $\lambda \in [0, 1]$, the convexity follows from

$$D_l\left(\mathbf{F}_l, \frac{\lambda\tilde{\mathbf{F}}'_l + (1 - \lambda)\tilde{\mathbf{F}}''_l}{\|\lambda\tilde{\mathbf{F}}'_l + (1 - \lambda)\tilde{\mathbf{F}}''_l\|}\right) \leq \lambda D_l(\mathbf{F}_l, \tilde{\mathbf{F}}'_l) + (1 - \lambda)D_l(\mathbf{F}_l, \tilde{\mathbf{F}}''_l),$$

where $\| \cdot \|$ denotes the Euclidean-norm. The division by $\|\lambda\tilde{\mathbf{F}}'_l + (1 - \lambda)\tilde{\mathbf{F}}''_l\|$ ensures that the second argument of D_l is a unit vector. Geometrically, we define the convexity along the geodesic arc connecting the two unit vectors $\tilde{\mathbf{F}}'_l$ and $\tilde{\mathbf{F}}''_l$ and not along the chord connecting the two (Kendall, 1991).

Given m valid distortion measures $\{D_l\}_{l=1}^m$ (some of which may be derived from the same base distortion) between the corresponding m component feature vectors of $\mathbf{x}$ and $\tilde{\mathbf{x}}$, we define a *weighted distortion measure* between $\mathbf{x}$ and $\tilde{\mathbf{x}}$ as

$$D^\alpha(\mathbf{x}, \tilde{\mathbf{x}}) = \sum_{l=1}^{m} \alpha_l D_l(\mathbf{F}_l, \tilde{\mathbf{F}}_l), \tag{2}$$

where the*feature weights* $\{\alpha_l\}_{l=1}^m$ are non-negative and sum to 1 and $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_m)$. The weighted distortion D^α is a convex combination of convex distortion measures, and, hence, for a fixed $\mathbf{x}$, D^α is convex in $\tilde{\mathbf{x}}$.

We refer to the vector of weights $\boldsymbol{\alpha}$ as a*feature weighting*. Feature weighting $\boldsymbol{\alpha}$ is tunable in our framework, and is used to assign different relative importance to component feature vectors. In Section 4, we will discuss a scheme for selecting a suitable feature weighting.

3. k-Means with weighted distortion

3.1. *The problem*

Suppose that we are given n data objects such that

$$\mathbf{x}_i = \left(\mathbf{F}_{(i,1)}, \mathbf{F}_{(i,2)}, \ldots, \mathbf{F}_{(i,m)}\right), \quad 1 \leq i \leq n,$$

where the l-th, $1 \leq l \leq m$, component feature vector of every data object is in the feature space $\mathcal{F}_l$. We are interested in partitioning the data set $\{\mathbf{x}_i\}_{i=1}^n$ into k *disjoint* clusters $\{\pi_u\}_{u=1}^k$.

3.2. *Generalized centroids*

Given a partitioning $\{\pi_u\}_{u=1}^k$, for each partition π_u, write the corresponding *generalized centroid* as

$$\mathbf{c}_u = \left(\mathbf{c}_{(u,1)}, \mathbf{c}_{(u,2)}, \ldots, \mathbf{c}_{(u,m)}\right),$$

where, for $1 \leq l \leq m$, the l-th component $\mathbf{c}_{(u,l)}$ is in $\mathcal{F}_l$. We define $\mathbf{c}_u$ as follows:

$$\mathbf{c}_u = \arg\min_{\tilde{\mathbf{x}} \in \mathcal{F}} \left(\sum_{\mathbf{x} \in \pi_u} D^\alpha(\mathbf{x}, \tilde{\mathbf{x}}) \right). \tag{3}$$

As an empirical average, the generalized centroid may be thought of as being the closest in D^α to all the data objects in the cluster π_u.

The key to solving (3) is to observe that D^α is component-wise-convex, and, hence, we can solve (3) by *separately* solving for each of its m components $\mathbf{c}_{(u,l)}$, $1 \leq l \leq m$. In other words, we need to solve the following m independent convex optimization problems:

$$\mathbf{c}_{(u,l)} = \underset{\tilde{\mathbf{F}}_l \in \mathcal{F}_l}{\arg\min} \left(\sum_{\mathbf{x} \in \pi_u} D_l(\mathbf{F}_l, \tilde{\mathbf{F}}_l) \right), \quad 1 \leq l \leq m. \tag{4}$$

For the two feature spaces of interest (and for many others as well), we can write the solution of (4) in a closed form:

Euclidean case:

$$\mathbf{c}_{(u,l)} = \frac{1}{\sum_{\mathbf{x} \in \pi_u} 1} \sum_{\mathbf{x} \in \pi_u} \mathbf{F}_l, \quad \text{where } \mathbf{x} = (\mathbf{F}_1, \mathbf{F}_2, \ldots, \mathbf{F}_m).$$

Spherical case:

$$\mathbf{c}_{(u,l)} = \frac{\sum_{\mathbf{x} \in \pi_u} \mathbf{F}_l}{\left\| \sum_{\mathbf{x} \in \pi_u} \mathbf{F}_l \right\|}, \quad \text{where } \mathbf{x} = (\mathbf{F}_1, \mathbf{F}_2, \ldots, \mathbf{F}_m).$$

3.3. *The convex k-means algorithm*

Motivated by (3), we measure the distortion of each individual cluster π_u, $1 \leq u \leq k$, as

$$\sum_{\mathbf{x} \in \pi_u} D^{\alpha}(\mathbf{x}, \mathbf{c}_u),$$

and the quality of the entire partitioning $\{\pi_u\}_{u=1}^k$ as the combined distortion of all the k clusters:

$$\sum_{u=1}^{k} \sum_{\mathbf{x} \in \pi_u} D^{\alpha}(\mathbf{x}, \mathbf{c}_u).$$

We would like to find k disjoint clusters $\pi_1^\dagger, \pi_2^\dagger, \ldots, \pi_k^\dagger$ such that the following is minimized:

$$\{\pi_u^\dagger\}_{u=1}^k = \underset{\{\pi_u\}_{u=1}^k}{\arg\min} \left(\sum_{u=1}^{k} \sum_{\mathbf{x} \in \pi_u} D^{\alpha}(\mathbf{x}, \mathbf{c}_{\mathbf{u}}) \right), \tag{5}$$

where the feature weighting $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_m)$ is fixed. Even when only one of the weights $\{\alpha_l\}_{l=1}^m$ is nonzero, the minimization problem (5) is known to be NP-complete (Kleinberg, Papadimitriou, & Raghavan, 1998). We now present an adaptation of the classical *k-means* algorithm (Hartigan, 1975) to work with our notion of the weighted distortion; we refer to the adapted algorithm as the *convex k-means* algorithm. The convex k-means algorithm generalizes the classical Euclidean k-means algorithm, the spherical k-means algorithm (Dhillon & Modha, 2001), and the toric k-means algorithm (Modha & Spangler, 2000). Using results in Sabin and Gray (1986) and Dhillon and Modha (2001), Lemma 3.1 it can be thought of as a *gradient descent* method, and, hence, never increases the objective function (5) and eventually converges to a local minima.

Step 1: Start with an arbitrary partitioning of the data objects, say, $\{\pi_u^{(0)}\}_{u=1}^k$. Let $\{\mathbf{c}_u^{(0)}\}_{u=1}^k$ denote the generalized centroids associated with the given partitioning. Set the index of iteration $t = 0$. The choice of the initial partitioning is quite crucial to finding a "good" local minima (Bradley & Fayyad, 1998). In our experiments, we used the heuristics described in Dhillon and Modha (2001), Section 3.4.

Step 2: For each data object $\mathbf{x}_i$, $1 \leq i \leq n$, find the generalized centroid that is closest to $\mathbf{x}_i$. Now, for $1 \leq u \leq k$, compute the new partitioning $\{\pi_u^{(t+1)}\}_{u=1}^k$ induced by the old generalized centroids $\{\mathbf{c}_u^{(t)}\}_{u=1}^k$:

$$\pi_u^{(t+1)} = \left\{\mathbf{x} \in \{\mathbf{x}_i\}_{i=1}^n : D^\alpha\left(\mathbf{x}, \mathbf{c}_u^{(t)}\right) \leq D^\alpha\left(\mathbf{x}, \mathbf{c}_v^{(t)}\right), 1 \leq v \leq k\right\}. \tag{6}$$

In words, $\pi_u^{(t+1)}$ is the set of all data objects that are closest to the generalized centroid $\mathbf{c}_u^{(t)}$. If some data object is simultaneously closest to more than one generalized centroid, then it is randomly assigned to one of the clusters. Clusters defined using (6) are known as *Voronoi* or *Dirichlet* partitions.

Step 3: Compute the new generalized centroids $\{\mathbf{c}_u^{(t+1)}\}_{u=1}^k$ corresponding to the partitioning computed in (6) by using (3) and (4) where instead of π_u we use $\pi_u^{(t+1)}$.

Step 4: If some "stopping criterion" is met, then set $\pi_u^\dagger = \pi_u^{(t+1)}$ and set $\mathbf{c}_u^\dagger = \mathbf{c}_u^{(t+1)}$ for $1 \leq u \leq k$, and exit. Otherwise, increment t by 1, and go to Step 2 above. An example of a stopping criterion is: Stop if the change in the objective function between two successive iterations, is less than some specified threshold.

The convex k-means algorithm described above works for a fixed feature weighting. We now turn to the crucial question of how to select the "best" feature weighting.

4. The optimal feature weighting

Throughout this section, fix the number of clusters $k \geq 2$ and fix the initial partitioning used by the k-means algorithm in Step 1. Also, we let

$$\bar{\mathbf{c}} \equiv (\bar{\mathbf{c}}_1, \bar{\mathbf{c}}_2, \ldots, \bar{\mathbf{c}}_m)$$

denote the generalized centroid for the entire data set, that is, for $1 \leq l \leq m$,

$$\bar{\mathbf{c}}_l = \underset{\tilde{\mathbf{c}} \in \mathcal{F}_l}{\arg\min} \left(\sum_{i=1}^n D_l(\mathbf{F}_{(i,l)}, \tilde{\mathbf{c}}) \right).$$

Write the set of all possible feature weightings as:

$$\Delta = \left\{ \boldsymbol{\alpha} : \sum_{l=1}^{m} \alpha_l = 1, \alpha_l \geq 0, 1 \leq l \leq m \right\}.$$

Given a feature weighting $\boldsymbol{\alpha} \in \Delta$, let

$$\Pi(\boldsymbol{\alpha}) = \{\pi_u^\dagger(\boldsymbol{\alpha}), \mathbf{c}_u^\dagger(\boldsymbol{\alpha})\}_{u=1}^{k}$$

denote the partitioning and the corresponding centroids obtained by running the convex k-means algorithm with the fixed initial partitioning and the given feature weighting. For later use, we write

$$\mathbf{c}_u^\dagger(\boldsymbol{\alpha}) \equiv \big(\mathbf{c}_{(u,1)}^\dagger(\boldsymbol{\alpha}), \mathbf{c}_{(u,2)}^\dagger(\boldsymbol{\alpha}), \ldots, \mathbf{c}_{(u,m)}^\dagger(\boldsymbol{\alpha})\big), \quad 1 \leq u \leq k.$$

Observe that the algorithm uses the feature weighting $\boldsymbol{\alpha}$ in (6), and, hence, the final partitioning depends on $\boldsymbol{\alpha}$. We make this implicit dependence explicit by writing Π as a function of $\boldsymbol{\alpha}$. Hypothetically, suppose that we can run the convex k-means algorithm for every possible feature weighting in Δ. From the resulting set of all possible clusterings

$$\{\Pi(\boldsymbol{\alpha}) : \boldsymbol{\alpha} \in \Delta\}$$

we would like to select a clustering that is in some sense the *best*. Towards this end, we now introduce a figure-of-merit to compare various clusterings. Note that each feature weighting $\boldsymbol{\alpha}$ leads to a different distortion measure D^{α}. Hence, one cannot meaningfully compare the k-means errors

$$\sum_{u=1}^{k} \sum_{\mathbf{x} \in \pi_u^\dagger(\boldsymbol{\alpha})} D^{\alpha}(\mathbf{x}, \mathbf{c}^\dagger(\boldsymbol{\alpha}))$$

achieved by the convex k-means algorithm for different feature weightings. We now present an elegant and natural generalization of Fisher's discriminant analysis to overcome this problem.

Fix a partitioning $\Pi(\boldsymbol{\alpha})$. Let's focus on how well this partitioning clusters along the l-th, $1 \leq l \leq m$, component feature vector. Define the *average within-cluster distortion* and *average between-cluster distortion* along the l-th component feature vector, respectively, as

$$\Gamma_l(\boldsymbol{\alpha}) \equiv \Gamma_l(\Pi(\boldsymbol{\alpha})) = \sum_{u=1}^{k} \sum_{\mathbf{x} \in \pi_u^\dagger(\boldsymbol{\alpha})} D_l\big(\mathbf{F}_l, \mathbf{c}_{(u,l)}^\dagger(\boldsymbol{\alpha})\big),$$

$$\Lambda_l(\boldsymbol{\alpha}) \equiv \Lambda_l(\Pi(\boldsymbol{\alpha})) = \left[\sum_{i=1}^{n} D_l\big(\mathbf{F}_{(i,l)}, \bar{\mathbf{c}}_l\big) - \Gamma_l(\boldsymbol{\alpha}) \right],$$

where $\mathbf{x} = (\mathbf{F}_1, \mathbf{F}_2, \ldots, \mathbf{F}_m)$. Intuitively, we would like to minimize $\Gamma_l(\boldsymbol{\alpha})$ and to maximize $\Lambda_l(\boldsymbol{\alpha})$, that is, we like coherent clusters that are well-separated from each other. Hence, we minimize

$$\mathcal{Q}_l(\boldsymbol{\alpha}) \equiv \mathcal{Q}_l(\Pi(\boldsymbol{\alpha})) = \left(\frac{\Gamma_l(\boldsymbol{\alpha})}{\Lambda_l(\boldsymbol{\alpha})}\right)^{n_l/n} \tag{7}$$

where n_l denotes the number of data objects that have a non-zero feature vector along the l-th component. The quantity n_l is introduced to accommodate sparse data sets. Observe that while the quantities Γ_l, Λ_l, and $\mathcal{Q}_l$ depend on the feature weighting $\boldsymbol{\alpha}$, this dependence is implicit in the fact they are functions of $\Pi(\boldsymbol{\alpha})$ which in turn depends on $\boldsymbol{\alpha}$ through (6).

Minimizing $\mathcal{Q}_l(\boldsymbol{\alpha})$ leads to a good discrimination along the l-th component feature space. Since we would like to *simultaneously* attain good discrimination along *all* the m feature spaces, we select the optimal feature weighting $\boldsymbol{\alpha}^\dagger$ as

$$\boldsymbol{\alpha}^\dagger = \arg\min_{\boldsymbol{\alpha}\in\Delta}[\mathcal{Q}(\boldsymbol{\alpha})], \tag{8}$$

where

$$\mathcal{Q}(\boldsymbol{\alpha}) \equiv \mathcal{Q}_1(\boldsymbol{\alpha}) \times \mathcal{Q}_2(\boldsymbol{\alpha}) \times \cdots \mathcal{Q}_m(\boldsymbol{\alpha}).$$

By taking the product of $\mathcal{Q}_1, \mathcal{Q}_2, \ldots, \mathcal{Q}_m$, in essence, we create a *dimensionless* quantity that can be meaningfully compared across clusterings.

Remark 4.1. If the l-th feature space is $\mathbb{R}^{f_l}$ and D_l is the squared-Euclidean distance, then we can provide an intuitive interpretation of the quantities $\Gamma_l(\boldsymbol{\alpha})$ and $\Lambda_l(\boldsymbol{\alpha})$. Write

$$\Gamma_l(\boldsymbol{\alpha}) = \sum_{u=1}^{k} \sum_{\mathbf{x}\in\pi_u^\dagger(\boldsymbol{\alpha})} \left(\mathbf{F}_l - \mathbf{c}_{(u,l)}^\dagger(\boldsymbol{\alpha})\right)^T \left(\mathbf{F}_l - \mathbf{c}_{(u,l)}^\dagger(\boldsymbol{\alpha})\right),$$

where $\mathbf{x} = (\mathbf{F}_1, \mathbf{F}_2, \ldots, \mathbf{F}_m)$. Observe that $\Gamma_l(\boldsymbol{\alpha})$ is simply the trace of the within-class covariance matrix. Now, the quantity

$$\sum_{i=1}^{n} D_l\left(\mathbf{F}_{(i,l)}, \bar{\mathbf{c}}_l\right) = \sum_{i=1}^{n} \left(\mathbf{F}_{(i,l)} - \bar{\mathbf{c}}_l\right)^T \left(\mathbf{F}_{(i,l)} - \bar{\mathbf{c}}_l\right)$$

is simply the trace of the covariance matrix, and, hence, it follows from Duda and Hart (1973) that $\Lambda_l(\boldsymbol{\alpha})$ is the trace of the between-class covariance matrix. In this case, assuming that $n_l = n$, $\mathcal{Q}_l(\boldsymbol{\alpha})$ is the ratio used by Fisher in determining the quality of a given classification, and as the objective function underlying his multiple discriminant analysis. In this light, we refer to the quantity in (7) as the *generalized Fisher ratio*.

Remark 4.2. Observe that the formulation in Section 2 continues to hold even when we scale distortion on each of the feature spaces by a possibly different constant. To be precise, we may write (2) as

$$D^{\alpha}(\mathbf{x}, \tilde{\mathbf{x}}) = \sum_{l=1}^{m} \alpha_l \beta_l D_l(\mathbf{F}_l, \tilde{\mathbf{F}}_l),$$

where $\beta_1, \beta_2, \ldots, \beta_l$ are positive constants. Clearly, such scaling will affect the numerical value of the optimal feature weighting. Hence, we should always interpret the optimal feature weighting with respect to any explicit or implicit scaling of the distortions on individual feature spaces.

5. Evaluation methodology

Suppose that we have selected the optimal feature weighting $\boldsymbol{\alpha}^{\dagger}$ by minimizing the objective function in (8). A natural question is: *How good is the clustering corresponding to the optimal feature weighting?* In other words, *Can we empirically verify that the optimal feature weighting indeed leads to desirable clustering in some precise sense?* To answer these questions, we assume that we are given pre-classified data and benchmark the precision/recall performance of various clusterings against the given *ground truth*. Precision/recall numbers measure the "overlap" between a given clustering and the ground truth classification. We stress that the precision/recall numbers are not used in selecting the optimal feature weighting, and are only intended to provide an independent, *postfactum* verification of the utility of feature weighting. We shall use the precision/recall numbers to only compare partitionings with a fixed number of clusters k, that is, partitionings with the same "model complexity".

To meaningfully define precision/recall, we convert the clusterings into classification using the following simple rule: identify each cluster with the class that has the largest overlap with the cluster, and assign every element in that cluster to the found class. The rule allows multiple clusters to be assigned to a single class, but never assigns a single cluster to multiple classes.

Suppose there are c classes $\{\omega_t\}_{t=1}^{c}$ in the ground truth classification. For a given clustering, by using the above rule, let a_t denote the number of data objects that are correctly assigned to the class ω_t, let b_t denote the data objects that are incorrectly assigned to the class ω_t, and let c_t denote the data objects that are incorrectly rejected from the class ω_t. We define *precision* and *recall* as

$$p_t = \frac{a_t}{a_t + b_t} \quad \text{and} \quad r_t = \frac{a_t}{a_t + c_t}, \quad 1 \le t \le c,$$

The precision and recall are defined per class. Following Yang (1999), we capture the performance averages across classes using macro-precision (macro-p), macro-recall (macro-r), micro-precision (micro-p), and micro-recall (micro-r):

$$\text{macro-}p = \frac{1}{c}\sum_{t=1}^{c} p_t \quad \text{and} \quad \text{macro-}r = \frac{1}{c}\sum_{t=1}^{c} r_t$$

$$\text{micro-}p \overset{(a)}{=} \text{micro-}r = \frac{1}{n}\sum_{t=1}^{c} a_t,$$

where (a) follows since, in our case, $\sum_{t=1}^{c}(a_t + b_t) = \sum_{t=1}^{c}(a_t + c_t) = n$.

6. Clustering data sets with numerical and categorical attributes

6.1. Data model

Suppose we are given a data set with both numerical and categorical features. Standard k-means is designed to work with numerical data, and does not work well with categorical data. Hence, in our setting, at the very least, we would like to have two feature spaces. It is possible to further break-up numerical and categorical features into smaller feature spaces. However, we linearly scale each numerical feature (that is, we subtract the mean and divide by the square-root of the variance) and use a 1-in-q representation for each q-ary categorical feature. This makes all numerical features roughly homogeneous to each other, and all categorical features roughly homogeneous to each other; thus obviating any need for further division. For the numerical feature space, we use the squared-Euclidean distance. Assuming no missing values, all the categorical feature vectors have the same norm. We only retain the "direction" of the categorical feature vectors, that is, we normalize each categorical feature vector to have an unit Euclidean norm, and use the cosine distance. Essentially, we represent each data object $\mathbf{x}$ as a m-tuple, $m = 2$, of feature vectors $(\mathbf{F}_1, \mathbf{F}_2)$.

6.2. HEART, ADULT, and AUSTRALIAN data sets

All data sets described below can be obtained from the UCI Machine Learning Repository (Blake & Merz, 1998).

The HEART data set consists of $n = 270$ instances. Every instance consists of 13 features from which we treated 5 as numerical (age, blood pressure, serum cholesterol, heart rate, and oldpeak) and the remaining 8 as categorical. The data set has two classes: absence and presence of heart disease; 55.56% (resp. 44.44%) instances were in the former (resp. later) class.

The ADULT data set consists of $n = 32561$ instances that were extracted from the 1994 Census database. Every instance consists of 6 numerical and 8 categorical features. The data set has two classes: those with income less than or equal to \$50,000 and those with income more than \$50, 000; 75.22% (resp. 24.78%) instances were in the former (resp. later) class.

The AUSTRALIAN credit data set consists of $n = 690$ instances. Every instance consists of 6 numerical and 8 categorical features. The data set has two classes: 55.5% instances were in one class and the rest in the other.

6.3. *Optimal feature weighting*

In this case, the set of feasible feature weightings is

$$\Delta = \{(\alpha_1, \alpha_2) : \alpha_1 + \alpha_2 = 1, \alpha_1, \alpha_2 \geq 0\},$$

where α_1 and α_2 represent the weights of the numerical and categorical feature spaces, respectively. We select the number of clusters $k = 2, 4, 6, 8, 16$, and do an exhaustive search over a fine grid on the interval [0, 1] to determine (heuristic approximations to) optimal feature weightings which minimize the objective function in (8).

For all the three data sets, Table 1 shows micro-p values achieved by (the clusterings corresponding to) optimal feature weightings. In each case, we compare micro-p numbers

Table 1. The top, middle, and bottom tables correspond, respectively, to HEART, ADULT, and AUSTRALIAN data sets. For each data set, we show results for $k = 2, 4, 6, 8, 16$ clusters. We write optimal and uniform feature weighting, respectively, as $\alpha^\dagger$ and $\tilde{\alpha}$.

	Optimal feature weighting			Uniform feature weighting			Best	Worst
k	$\alpha^\dagger$	$Q(\alpha^\dagger)$	micro-p	$\tilde{\alpha}$	$Q(\tilde{\alpha})$	micro-p	micro-p	micro-p
				HEART data set				
2	(.09, .91)	20.49	.804	(.5, .5)	34.64	.741	.830	.711
4	(.08, .92)	6.35	.815	(.5, .5)	11.21	.733	.826	.718
6	(.08, .92)	3.77	.803	(.5, .5)	7.69	.711	.819	.707
8	(.10, .90)	2.77	.800	(.5, .5)	5.20	.711	.811	.678
16	(.12, .88)	1.15	.793	(.5, .5)	2.38	.767	.822	.744
				ADULT data set				
2	(.14, .86)	68.69	.759	(.5, .5)	154.44	.759	.759	.759
4	(.10, .90)	9.90	.761	(.5, .5)	24.80	.760	.761	.759
6	(.09, .91)	5.08	.812	(.5, .5)	13.68	.761	.818	.761
8	(.11, .89)	2.75	.820	(.5, .5)	10.49	.770	.822	.770
16	(.09, .91)	1.17	.819	(.5, .5)	2.68	.800	.820	.777
				AUSTRALIAN data set				
2	(.09, .91)	38.68	.829	(.5, .5)	107.29	.646	.833	.643
4	(.09, .91)	10.31	.762	(.5, .5)	30.16	.648	.805	.640
6	(.08, .92)	5.63	.832	(.5, .5)	13.35	.686	.835	.670
8	(.10, .90)	3.89	.836	(.5, .5)	10.75	.690	.842	.677
16	(.08, .92)	1.17	.829	(.5, .5)	2.75	.738	.855	.691

The "best micro-p" and "worst micro-p" columns refer, respectively, to the best and worst micro-p achieved by any feature weighting over a fine grid in [0, 1]. It can be seen the micro-p values for optimal feature weighting are consistently better than micro-p values for uniform feature weighting–for every data set and for every k. Furthermore, micro-p values for optimal feature weighting are close to the best micro-p values and are quite far from the worst micro-p values.

for optimal feature weighting to those for uniform feature weighting. It can be clearly seen that optimal feature weighting consistently outperforms uniform feature weighting on all data sets and for every k. Furthermore, micro-p values for optimal feature weighting are close to (resp. far from) the best (resp. worst) micro-p values achieved by any feature weighting (in the fine grid over [0, 1] that we used). While macro-p and macro-r numbers are not shown, they lead to identical conclusions.

We now present fine-grain analysis of the behavior of the objective function $\mathcal{Q}(\boldsymbol{\alpha})$ and micro-p values over the space of feature weightings. For the HEART (resp. ADULT and AUSTRALIAN) data, the top panel in figure 1 (resp. figures 2 and 3) shows a plot of the objective function $\mathcal{Q}(\boldsymbol{\alpha})$ in (8) versus the weight α_1 for $k = 8$. In all cases, the bottom panel shows a plot of micro-p values versus the weight α_1. For all three figures, by comparing the top panel with the bottom panel, it can be seen that, roughly, micro-p values are *negatively correlated* with the objective function $\mathcal{Q}(\boldsymbol{\alpha})$ and that, in fact, *the optimal feature weightings achieve nearly the best micro-p.* A similar negative correlation also holds between the objective function and both macro-p and macro-r values. This surprising and reassuring finding underscores the usefulness of our feature weighting scheme in clustering data sets with multiple, heterogeneous feature spaces. To put it another way, from the class of all clusterings parameterized by the set of feasible weightings Δ, our feature weighting scheme identifies essentially the best clustering—when compared to a known ground truth classification. The results of the next section also testify to the same end, but on an altogether different application.

7. Clustering text data with words and phrases

7.1. Phrases in information retrieval

Vector space models (Salton & McGill, 1983) represent each *document* as a vector of certain (possibly weighted and normalized) *term* frequencies. Typically, terms are single words. However, to capture word ordering, it is intuitively desirable to also include multi-word sequences, namely, *phrases*, as terms.

The use of phrases as terms in vector space models has been well studied. For example Salton, Yang, and Yu (1975) and Mitra et al. (1997) showed that statistical phrases enhance precision of information retrieval systems. Recently, for supervised learning using a naïve-Bayesian classifier (Mladenić & Grobelnik, 1998) have used words and phrases as terms. Unlike these studies, we do not group the phrases along with the words in a *single* vector space model. This direction was anticipated and suggested by Smeaton and Kelledy (1998) who noted that "Phrases and single word terms have different frequency distributions and this should really be taken into account in applying 2 different term weighting functions as part of retrieval instead of bundling phrase and single word representations into one as we have done."

For information retrieval, when single words are also simultaneously used (Mitra et al., 1997) found that *natural language phrases* do not perform significantly better than *statistical phrases*. Hence, we shall focus on statistical phrases which are simpler to extract, see, for example Agrawal and Srikant (1995) and Ahonen-Myka (1999). Also, Mladenić and

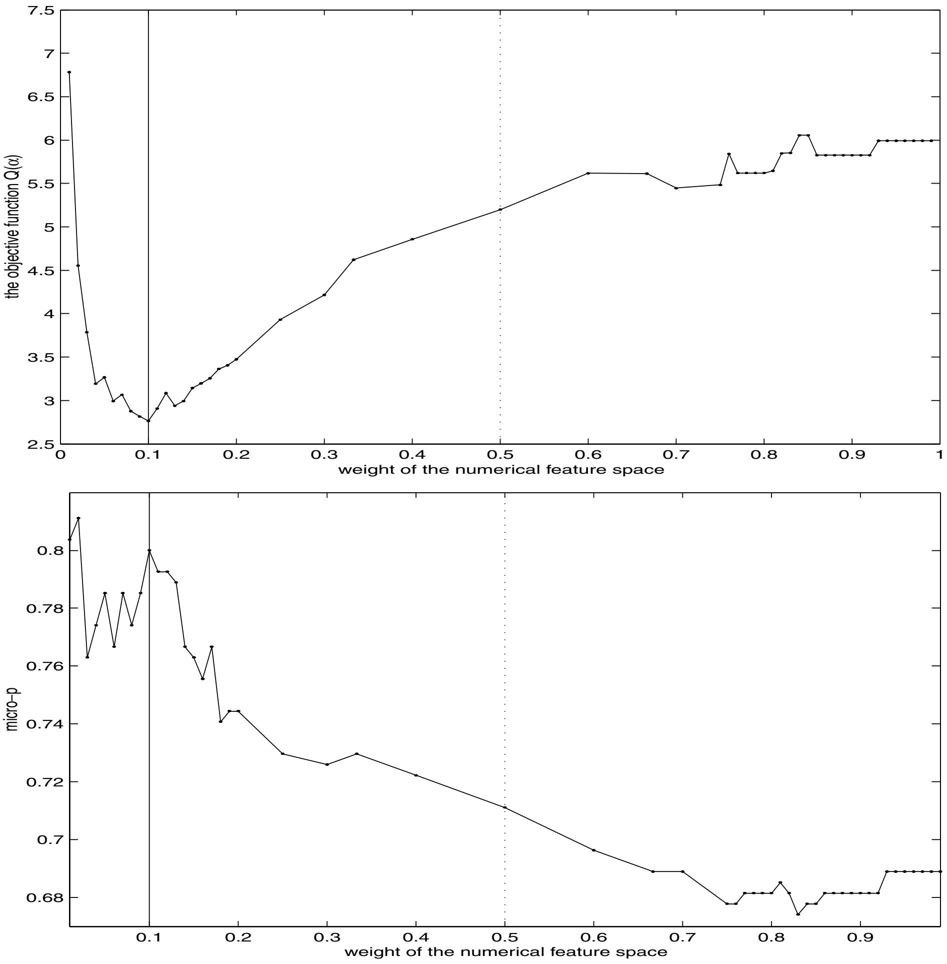

Figure 1. For the HEART data and for $k = 8$, the top panel (resp. bottom panel) shows a plot of the objective function (resp. micro-p values) versus the weight of the numerical feature space. The solid (resp. dotted) vertical line indicate the position of the optimal feature weighting (resp. the uniform feature weighting). The "negative correlation" between the objective function and micro-p values is clearly evident from the plots. Finally, superiority of the optimal feature weighting over the uniform feature weighting is also evident.

Grobelnik (1998) found that while adding 2- and 3-word phrases improved the classifier performance, longer phrases did not. Hence, we shall restrict attention to single words, 2-word phrases, and 3-word phrases.

7.2. Data model

We represent each document as a triplet of three vectors: a vector of word frequencies, a vector of 2-word phrase frequencies, and a vector of 3-word phrase frequencies, that is,

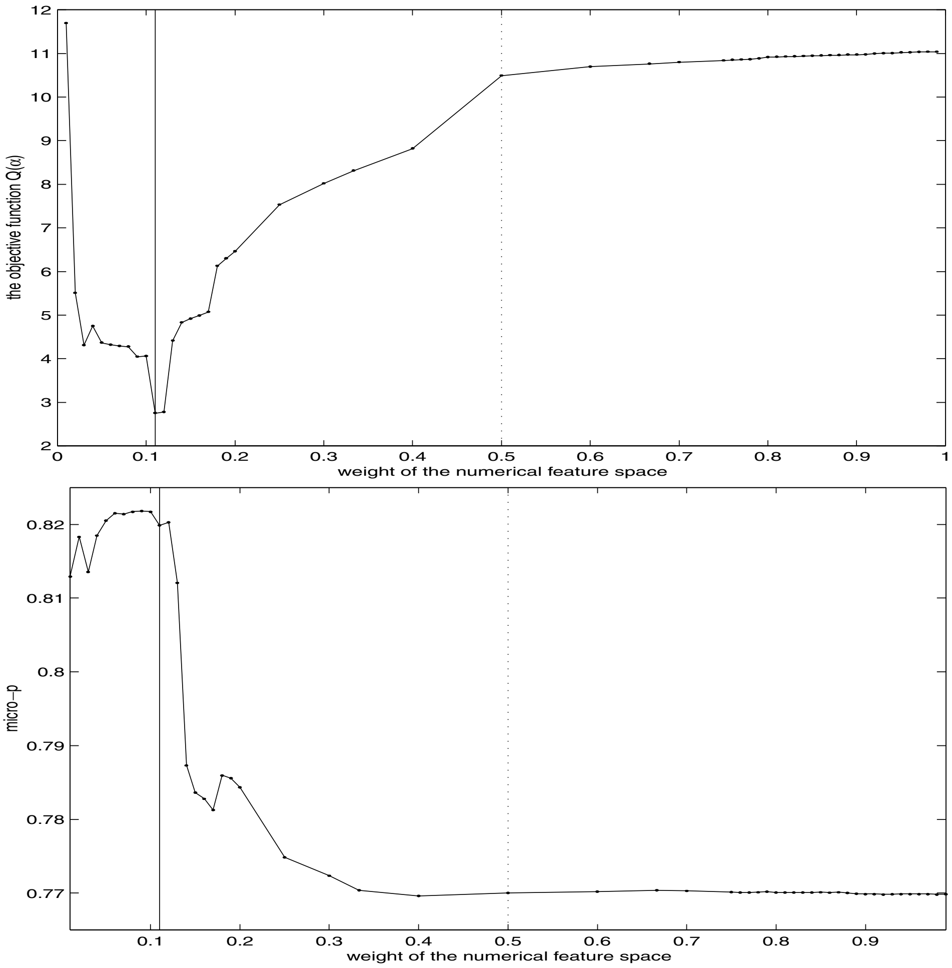

Figure 2. For the ADULT data and for $k = 8$, the top panel (resp. bottom panel) shows a plot of the objective function (resp. micro-p values) versus the weight of the numerical feature space. The solid (resp. dotted) vertical line indicate the position of the optimal feature weighting (resp. the uniform feature weighting). The "negative correlation" between the objective function and micro-p values is clearly evident from the plots. Finally, superiority of the optimal feature weighting over the uniform feature weighting is also evident.

$\mathbf{x} = (\mathbf{F}_1, \mathbf{F}_2, \mathbf{F}_3)$. We now show how to compute such representations for every document in a given corpus.

The creation of the first feature vector is a standard exercise in information retrieval (Salton & McGill, 1983). The basic idea is to construct a *word dictionary* of all the words that appear in any of the documents in the corpus, and to prune or eliminate *stopwords* words from this dictionary; for a list of standard stopwords, see, Frakes and Baeza-Yates (1992). Sometimes, the size of the word dictionary is further reduced using stemming (Frakes & Baeza-Yates, 1992). For the present application, we also eliminated those low-frequency

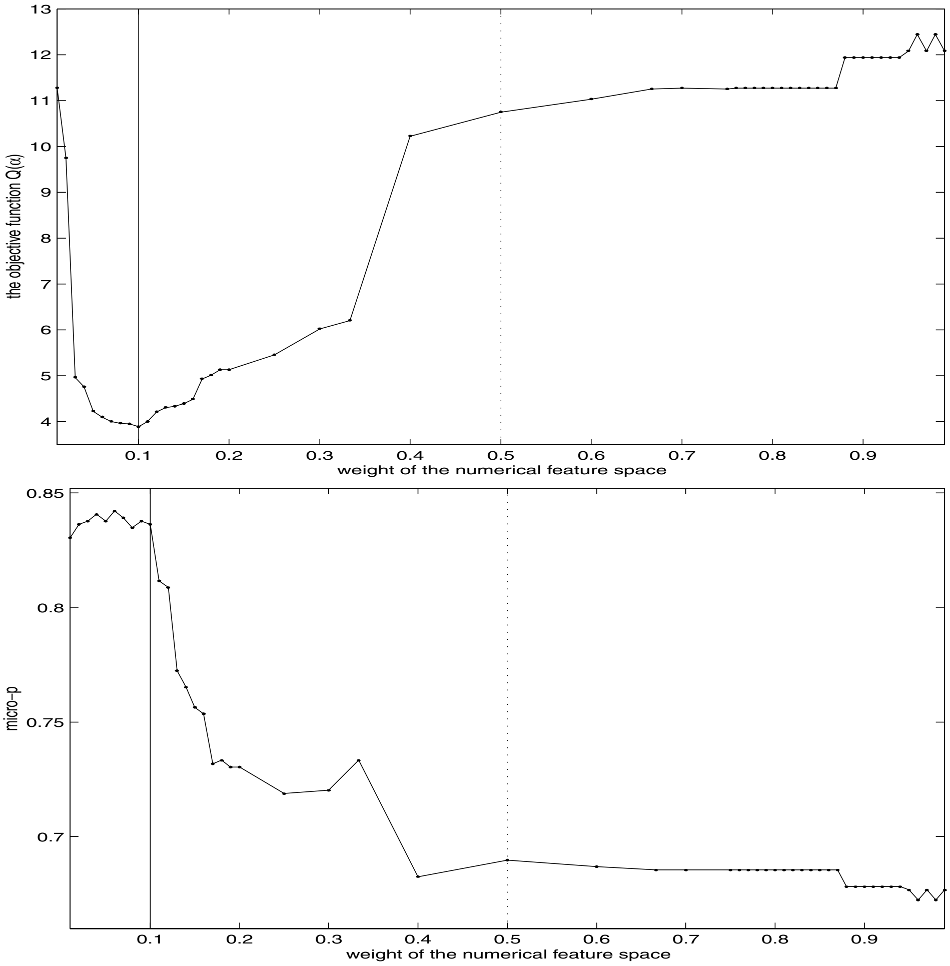

Figure 3. For the AUSTRALIAN data and for $k = 8$, the top panel (resp. bottom panel) shows a plot of the objective function (resp. micro-p values) versus the weight of the numerical feature space. The solid (resp. dotted) vertical line indicate the position of the optimal feature weighting (resp. the uniform feature weighting). The "negative correlation" between the objective function and micro-p values is clearly evident from the plots. Finally, superiority of the optimal feature weighting over the uniform feature weighting is also evident.

words which appeared in less than 0.64% of the documents. Suppose f_1 unique words remain in the dictionary after such elimination. Assign an unique identifier from 1 to f_1 to each of these words. Now, for each document $\mathbf{x}$ in the corpus, the first vector $\mathbf{F}_1$ in the triplet will be a f_1-dimensional vector. The jth column entry, $1 \leq j \leq f_1$, of $\mathbf{F}_1$ is the number of occurrences of the jth word in the document $\mathbf{x}$.

Creation of the second (resp. third) feature vector is essentially the same as the first, except we set the low-frequency 2-word (resp. 3-word) phrase elimination threshold to one-half (resp. one-quarter) that for the words. The lower threshold somewhat compensates for the fact that a 2-word (resp. 3-word) phrase is generally less likely than a single word. Let f_2 (resp. f_3) denote the dimensionalities of the second (resp. third) feature vector.

Finally, each of the three components $\mathbf{F}_1, \mathbf{F}_2$, and $\mathbf{F}_3$ is normalized to have a unit Euclidean norm, that is, their directions are retained and their lengths are discarded (Singhal et al., 1996). There are a large number of term-weighting schemes in information retrieval for assigning different relative importance to various terms in the feature vectors (Salton & Buckley, 1988). Our feature vectors correspond to a popular scheme known as *normalized term frequency*.

We let the distortion measures D_1, D_2, and D_3 to be the cosine distances.

7.3. Newsgroups data

We picked out the following 10 newsgroups from the "Newsgroups data" (Bay, 1999; Joachims, 1997) to illustrate our results.

sci.crypt
rec.autos
sci.space
misc.forsale
comp.windows.x
rec.sport.baseball
talk.politics.guns
comp.sys.mac.hardware
soc.religion.christian
talk.politics.mideast

Each newsgroup contains 1000 documents; after removing empty documents, we had a total of $n = 9961$ documents.

For this data set, the unpruned word (resp. 2-word phrase and 3-word phrase) dictionary had size 72586 (resp. 429604 and 461132) out of which we retained $f_1 = 2583$ (resp. $f_2 = 2144$ and $f_3 = 2268$) elements that appeared in at least 64 (resp. 32 and 16) documents. All the three features spaces were highly sparse; on an average, after pruning, each document had only 50 (resp. 7.19 and 8.34) words (resp. 2-word and 3-word phrases). Finally, $n_1 = n = 9961$ (resp. $n_2 = 8639$ and $n_3 = 4664$) documents had at least one word (resp. 2-word phrase and 3-word phrase). Observe that the numbers n_1, n_2, and n_3 are used in (7).

7.4. Optimal feature weighting

In this case, the set of feasible feature weightings is the triangular region shown in figure 4. For each $k = 10, 15, 20$, to determine (heuristic approximations to) the optimal feature weightings, we ran the convex k-means algorithm with 31 different feature weightings that are shown using the symbol • in figure 4. Table 2 shows optimal feature weighting $\alpha^\dagger$, the corresponding objective function $Q(\alpha^\dagger)$, and corresponding micro-p value for $k = 10, 15, 20$. For $\tilde{\alpha} = (.33, .33, .33)$ which corresponds to an uniform feature weighting, we also display the corresponding objective function $Q(\tilde{\alpha})$ and micro-p value. Finally, we show the best and the worst micro-p numbers achieved by one of the 31 feature weightings in figure 4. It can be clearly seen that micro-p values for optimal feature weighting are consistently better than those for uniform feature weighting-for every k. Furthermore, micro-p values for optimal feature weighting are close to the best micro-p values and are quite far from the worst micro-p values. While macro-p and macro-r numbers are not shown, they lead to identical conclusions.

Table 2. For the Newsgroups data set, we show results for $k = 10, 15, 20$ clusters. We write optimal and uniform feature weighting, respectively, as $\alpha^\dagger$ and $\tilde{\alpha}$.

	Optimal feature weighting			Uniform feature weighting			Best	Worst
k	$\alpha^\dagger$	$Q(\alpha^\dagger)$	micro-p	$\tilde{\alpha}$	$Q(\tilde{\alpha})$	micro-p	micro-p	micro-p
10	(.50, .25, .25)	21.06	.686	(.33, .33, .33)	24.01	.593	.700	.320
15	(.75, .00, .25)	14.38	.656	(.33, .33, .33)	16.20	.616	.690	.378
20	(.75, .00, .25)	11.03	.664	(.33, .33, .33)	13.39	.602	.693	.394

The "best micro-p" and "worst micro-p" columns refer, respectively, to the best and worst micro-p achieved by any feature weighting over a fine grid (see figure 4) in Δ. It can be seen that micro-p values for optimal feature weighting are consistently better than micro-p values for uniform feature weighting–for every k. Furthermore, micro-p values for optimal feature weighting are close to the best micro-p values and are quite far from the worst micro-p values.

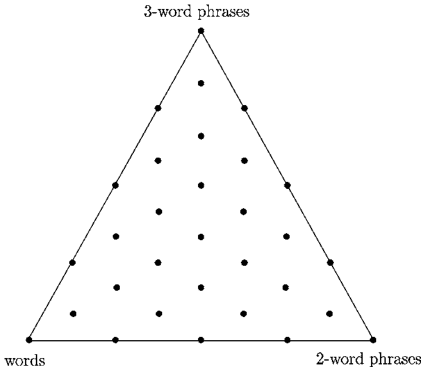

Figure 4. When $m = 3$, Δ is the triangular region formed by the intersection of the plane $\alpha_1 + \alpha_2 + \alpha_3 = 1$ with the nonnegative orthant of $\mathbb{R}^3$. The left-vertex, the right-vertex, and the top-vertex of the triangle corresponds to the points (1, 0, 0), (0, 1, 0), and (0, 0, 1), respectively.

In figure 5, for $k = 10$, we plot the objective function $\mathcal{Q}(\boldsymbol{\alpha})$ versus micro-p values for all the 31 feature weightings. It can be seen that, roughly, as the objective function decreases, micro-p increases. The optimal feature weighting is distinguished by putting the symbol □ around it; by definition, this is the lowest point on the plot. Surprisingly, it is also almost the right-most point on the plot, that is, it has almost the highest micro-p value. Although macro-p and macro-r results are not shown, they lead to the same conclusions. Similar conclusions also hold for $k = 15, 20$.

This finding reiterates the conclusion outlined at the end of Section 6 that: *optimal feature weighting achieves nearly the best precision and recall.*

8. Future work

The most important problem that we leave open is: how to efficiently determine the optimal feature weighting. Ideally, instead of the two step process offered in this paper, we would like

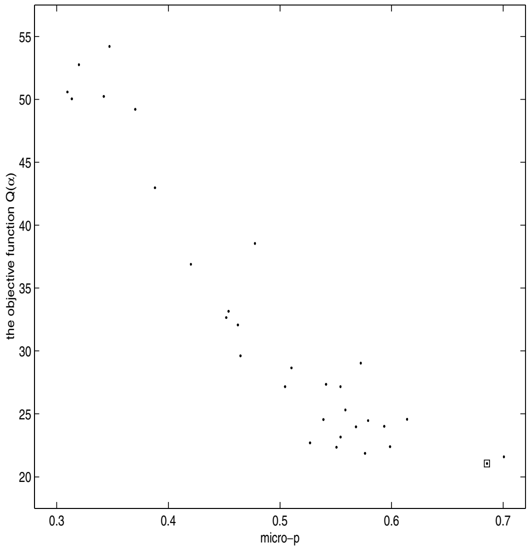

Figure 5. For the Newsgroups data set and for $k = 10$, we plot the objective function $\mathcal{Q}(\boldsymbol{\alpha})$ versus micro-p values for all the feature weightings in figure 4. The micro-p value corresponding to optimal feature weighting is distinguished by putting the symbol □ around it. The "negative correlation" between the objective function and micro-p values is clearly evident from the plot.

a single algorithm that jointly finds the optimal feature weighting $\boldsymbol{\alpha}^\dagger$ and the corresponding clustering $\Pi(\boldsymbol{\alpha}^\dagger) = \{\pi_u^\dagger(\boldsymbol{\alpha}^\dagger), \mathbf{c}_u^\dagger(\boldsymbol{\alpha}^\dagger)\}_{u=1}^k$. Observe that (i) the objective function $\mathcal{Q}(\boldsymbol{\alpha})$ is an implicit function of $\boldsymbol{\alpha}$, and, hence, analytic computation of its derivatives is not possible and numerical computation may be expensive and (perhaps) unstable; (ii) figures 1–3 lead us to conjecture that that the objective function $\mathcal{Q}(\boldsymbol{\alpha})$ is generally well-behaved. This observations suggest the tantalizing possibility of combining the downhill simplex method (Nelder & Mead, 1965; Press et al., 1993) with the convex k-means algorithm. We now offer a tentative scheme for this marriage. The idea is to start the convex k-means algorithm with an initial feature weighting (which may simply be the uniform feature weighting) and Δ as the starting simplex. Now, the moves of the downhill simplex algorithm (consisting of either reflection, contraction, or reflection with expansion of the simplex) are interspersed with the moves of the convex k-means algorithm. In particular, one may insert a downhill simplex step between Steps 3 and 4 of the convex k-means algorithm. Finally, the algorithm is said to converge when the stopping criteria for both the algorithms are met.

In this paper, we have employed the new weighted distortion measure D^α in the k-means algorithm; it may also be possible to use it with a graph-based algorithm such as the complete link method or with hierarchical agglomerative clustering algorithms.

Throughout this paper, we assumed that the number of clusters k is given; however, an important future problem is to automatically determine the number of clusters in an adaptive or data-driven fashion using information-theoretic criteria such as the MDL principle.

9. Conclusion

We have presented a framework for integrating multiple, heterogeneous feature spaces in the k-means clustering algorithm. Our methodology adaptively selects, in an unsupervised fashion, the relative weights assigned to various feature spaces with the objective of *simultaneously* attaining good separation along *all* the feature spaces. Using precision/recall evaluations, we have empirically demonstrated that optimal feature weighting is extremely effective in locating essentially the best clustering when compared against known ground truth classifications.

References

Agrawal, R., & Srikant, R. (1995). Mining sequential patterns. In *Proc. Int. Conf. Data Eng.* (pp. 3–14).

Ahonen-Myka, H. (1999). Finding all maximal frequent sequences in text. In D. Mladenic & M. Grobelnik (eds.), *ICML-99 Workshop: Machine Learning in Text Data Analysis* (pp. 11–17).

Bay, S. D. (1999). The UCI KDD archive. Dept. Inform. and Comput. Sci., Univ. California, Irvine, CA. Available at http://kdd.ics.uci.edu.

Blake, C., & Merz, C. (1998). UCI repository of machine learning databases. Dept. Inform. and Comput. Sci., Univ. California, Irvine, CA. Available at http://www.ics.uci.edu/∼mlearn/MLRepository.html.

Blum, A. L., & Langley, P. (1997). Selection of relevant features and examples in machine learning. *Artificial Intelligence, 97*, 245–271.

Bradley, P., & Fayyad, U. (1998). Refining initial points for k-means clustering. In *Proc. 16th Int. Machine Learning Conf.,* (pp. 91–99). Bled, Slovenia.

Caruana, R., & Freitag, D. (1994). Greedy attribute selection. In *Proc. 11th Int. Machine Learning Conf.* (pp. 28–36).

Devaney, M., & Ram, A. (1997). Efficient feature selection in conceptual clustering. In *Proc. 14th Int. Machine Learning Conf.* (pp. 92–97). Nashville, TN.

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

Duda, R. O., & Hart, P. E. (1973). *Pattern Classification and Scene Analysis*. Wiley.

Fisher, D. H. (1987). Knowledge acquisition via incremental conceptual clustering. *Machine Learning*, 139–172.

Flickner, M., Sawhney, H., Niblack, W., Ashley, J., Huang, Q., Dom, B., Gorkani, M., Hafner, J., Lee, D., Petkovic, D., Steele, D., & Yanker, P. (1995). Query by image and video content: The QBIC system. *IEEE Computer, 28:9*, 23–32.

Frakes, W. B., & Baeza-Yates, R. (1992). *Information Retrieval: Data Structures and Algorithms*. New Jersey: Prentice Hall, Englewood Cliffs.

Hartigan, J. A. (1975). *Clustering Algorithms*. Wiley.

Joachims, T. (1997). A probabilistic analysis of the Rocchio Algorithm with TFIDF for text categorization. In *Proc. 14th Int. Conf. Machine Learning*. (pp. 143–151).

John, G. H., Kohavi, R., & Pfleger, K. (1994). Irrelevant features and the subset selection problem. In *Proc. 11th Int. Machine Learning Conf.* (pp. 121–129).

Kendall, W. S. (1991). Convexity and the hemisphere. *J. London Math. Soc. 43*, 567–576.

Kleinberg, J., Papadimitriou, C. H., & Raghavan, P. (1998). A microeconomic view of data mining. *Data Mining and Knowledge Discovery, 2/4*, 311–324.

Koller, D., & Sahami, M. (1996). Towards optimal feature selection. In *Proc. 13th Int. Conf. Machine Learning*. (pp. 284–292). Bari, Italy.

Mitra, M., Buckley, C., Singhal, A., & Cardie, C. (1997). An analysis of statistical and syntactic phrases. In *Proc. RIAO97: Computer-Assisted Inform. Searching on the Internet* (pp. 200–214). Montreal, Canada.

Mladenić, D., & Grobelnik, M. (1998). Word sequences as features in text-learning. In *Proc. 7th Electrotech. Comput. Sci. Conf. ERK'98* (pp. 145–148). Ljubljana, Slovenia.

Modha, D. S., & Spangler, W. S. (2000). Clustering hypertext with applications to web searching. In *Proc. ACM Hypertext Conf.* (pp. 143–152). San Antonio, TX.

Nelder, J., & Mead, R. (1965). A simplex method for function minimization. *Computer Journal, 7*, 308.

Press, W. H., Teukolsky, S. A., Vetterling, W. T., & Flannery, B. P. (1993). *Numerical Recipes in C*. New York: Cambridge University Press.

Sabin, M. J., & Gray, R. M. (1986). Global convergence and empirical consistency of the generalized Lloyd algorithm. *IEEE Trans. Inform. Theory, 32:2*, 148–155.

Salton, G., & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Inform. Proc. & Management* (pp. 513–523).

Salton, G., & McGill, M. J. (1983). *Introduction to Modern Retrieval*. McGraw-Hill Book Company.

Salton, G., Yang, C. S., & Yu, C. T. (1975). A theory of term importance in automatic text analysis. *J. Amer. Soc. Inform. Sci., 26:1*, 33–44.

Singhal, A., Buckley, C., Mitra, M., & Salton, G. (1996). Pivoted document length normalization. In *Proc. ACM SIGIR* (pp. 21–29).

Smeaton, A. F., & Kelledy, F. (1998). User-chosen phrases in interactive query formulation for information retrieval. In *Proc. 20th BCS-IRSG Colloquium, Springer-Verlag Electronic Workshops in Comput.*, Grenoble, France.

Talavera, L. (1999). Feature selection as a preprocessing step for hierarchical clustering. In *Proc. 16th Int. Machine Learning Conf.* (pp. 389–397). Bled, Slovenia.

Vaithyanathan, S., & Dom, B. (1999). Model selection in unsupervised learning with applications to document clustering. In *Proc. 16th Int. Machine Learning Conf.* Bled, Slovenia.

Wettschereck, D., Aha, D. W., & Mohri, T. (1997). A review and empirical evaluation of feature weighting methods for a class of lazy learning algorithms. *Artificial Intelligence Review, 11*, 273–314.

Yang, Y. (1999). An evaluation of statistical approaches to text categorization. Inform. *Retrieval J., 1:1/2*, 67–88.

Received October 2, 2000
Revised February 27, 2002
Accepted February 27, 2002
Final manuscript March 1, 2002