Kernel-based Weighted Multi-view Clustering

Grigorios Tzortzis and Aristidis Likas
Department of Computer Science
University of Ioannina
Ioannina 45110, Greece
Email: {gtzortzi, arly}@cs.uoi.gr

***Abstract*—Exploiting multiple representations, or views, for the same set of instances within a clustering framework is a popular practice for boosting clustering accuracy. However, some of the available sources may be misleading (due to noise, errors in measurement etc.) in revealing the true structure of the data, thus, their inclusion in the clustering process may have negative influence. This aspect seems to be overlooked in the multi-view literature where all representations are equally considered. In this work, views are expressed in terms of given kernel matrices and a weighted combination of the kernels is learned in parallel to the partitioning. Weights assigned to kernels are indicative of the quality of the corresponding views' information. Additionally, the combination scheme incorporates a parameter that controls the admissible sparsity of the weights to avoid extremes and tailor them to the data. Two efficient iterative algorithms are proposed that alternate between updating the view weights and recomputing the clusters to optimize the intra-cluster variance from different perspectives. The conducted experiments reveal the effectiveness of our methodology compared to other multi-view methods.**

***Keywords*-multi-view clustering; multiple kernel learning; kernel *k*-means**

I. Introduction

Multi-modal datasets are very common in practice due to the use of different measuring methods (e.g. infrared and visual cameras), or of different media, like text, video and audio. Each instance in these datasets has multiple representations, called *views*, from various feature spaces. Typical examples include web pages, represented by both text and hyperlinks, and images, where color and texture information can be utilized. The existence of such data has raised interest in the so called *multi-view learning*, which has been extensively studied under the semi-supervised classification setting [1], [2], [3]. Our work focuses on *multi-view clustering* [4], [5], where the absence of a ground-truth to guide the learning process makes the underlining task much harder. The main challenge that arises is to find a suitable way of simultaneously exploiting the, possibly, complementary information of all available views in order to derive a robust partitioning, considering the diversity (e.g. different statistical properties) and the disagreement (i.e. different views produce different partitionings) among the views.

Surprisingly, most multi-view methods *rely equally on every view*, something that may lead to performance degradation in the case of degenerate views (e.g. noisy or irrelevant views). Identifying and appropriately handling such views is difficult though. The proposed approach tackles this problem from the kernel perspective, i.e. data points are mapped to a nonlinear high-dimensional space through a kernel function [6]. Each view is represented by a kernel matrix and views are combined using a weighted sum of the kernel matrices, accompanied by an appropriate constraint on the weights. *The weights express the quality* (*importance in clustering*) *of the views* and determine their *degree of contribution* to the final solution accordingly. They are *learned automatically*, together with the inference of the cluster labels, through *closed form expressions*, by minimizing the typical intra-cluster variance objective of *k*-means in the space induced by combining the individual kernels. Two iterative optimization strategies are developed, one based on kernel *k*-means [7], [8] and the other on spectral techniques [9].

Our strategy of mixing the kernels is inspired by *unsupervised multiple kernel learning* [10], [11], [12], [13], where for a singly represented dataset a, usually linear, combination of base kernels is sought together with the partitioning, to solve the kernel selection problem. In our case those kernels are derived from the views. Thus a connection between these popular machine learning problems emerges. There appears to be some dispute over the sparsity[1] of the combination weights, with some authors favoring high sparsity [12] and others a more uniform solution [10]. We believe that a good choice lies somewhere between the two ends, such that an algorithm is flexible enough to allow the data to harness the kernel weights, without being too prone to either end. For this reason, the proposed methodology *incorporates a parameter controlling this flexibility that must be specified prior to execution*. Experiments on synthetic and real world datasets support the above claim and indicate that view weighting under our framework is successful in reflecting the underlying properties of the studied data. The main contributions of this work can be summarized in:

1) The estimation of view weights, a subject generally overlooked in multi-view clustering.
2) The inclusion of a parameter that controls the sparsity

[1]Sparsity is defined relative to the number of kernels in the solution that carry significant weights.

of the weights.

3) The use of kernels to represent the views and the way they are combined, which connects multi-view clustering to multiple kernel learning.

The rest of this paper is organized as follows. The next section reviews related work, while Section III presents the foundations of our multi-view method which is detailed in Section IV. The experiments follow in Section V, before the concluding remarks of Section VI.

II. Related Work

Most of the existing work in multi-view clustering extends well-known algorithms to the multi-view setting by explicitly or implicitly exploiting the "minimizing-disagreement" idea [4]. Bickel and Scheffer [4] developed a two-view EM and a two-view *k*-means algorithm. They also studied the problem of mixture model estimation with more than two views [14]. De Sa [15] proposed a two-view spectral clustering technique that creates a bipartite graph of the views. A framework that generalizes the normalized cut to the multi-view case was introduced by Zhou and Burges [16]. This model contains a parameter that determines the relative importance of each view, but, unlike ours, it is not learned during training and must be fixed a priori. The top directions obtained by canonical correlation analysis across the views are utilized in [17], [18] to first project the data and then cluster the projections. A convex mixture distribution models each view in [19] and a weighted combination that reflects the views' importance is automatically learned through EM. In [20] the spectral embedding from one view is used to bootstrap the clustering of the other view, by modifying its similarity matrix in a co-training [3] like fashion. A drawback of the co-training idea is that convergence is not guaranteed.

According to the model of Long *et al.* [5] the views are independently clustered and a final partitioning of the data is derived by minimizing an objective function that measures how close the final split, based on all views, is to the split of each single view with the help of a mapping function. In a similar fashion, a matrix factorization approach was adopted in [21] to reconcile the groups arising from the individual views. Overall, despite the wide variety of multi-view clustering methods, most of them treat equally all views, *regardless of the conveyed information* (except [16], [19]). The proposed method explores this neglected aspect by introducing weights to the views which are learned automatically.

From the viewpoint of how the view kernels are combined under our framework, unsupervised multiple kernel learning can also be considered as related work. In most cases a linear weighting is applied in conjunction with a constraint on the weight values that also acts as a sparsity manipulator. This constraint is usually either the ℓ_1-norm regularizer [11], [12], which promotes a very sparse solution, or the ℓ_2-norm regularizer [13], which favors less sparse ones. Lange and Buhmann [10] learned a linear mixture of similarity matrices using an entropy criterion to control sparsity. In the supervised scenario the more general ℓ_p-norm regularizer, $p \geq 1$, was introduced [22] with higher p values producing a more uniform weighting. In our method, a parameter that can be set beforehand, allows for a similar kind of influence on the weights as the ℓ_p-norm regularizer. Finally, a nonlinear kernel combination is proposed in [23] and is applied to regression problems.

III. Kernel-based Clustering

Two kernel-oriented methods for optimizing the intra-cluster variance are described in this section, which are both considered under our framework.

A. Kernel k-means

Kernel *k*-means [7] is a generalization of the standard *k*-means algorithm where the dataset $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^N$, $\mathbf{x}_i \in \Re^d$ is mapped from input space to a higher dimensional reproducing kernel Hilbert space $\mathcal{H}$, a.k.a. feature space, via a nonlinear transformation $\phi : \mathcal{X} \to \mathcal{H}$.

To partition dataset $\mathcal{X}$ into M disjoint clusters, $\{\mathcal{C}_k\}_{k=1}^M$, the intra-cluster variance in feature space (1) is minimized over clusters $\{\mathcal{C}_k\}_{k=1}^M$, where $\mathbf{m}_k$ is the k-th cluster center and δ_{ik} is an indicator variable with $\delta_{ik} = 1$ if $\mathbf{x}_i \in \mathcal{C}_k$ and 0 otherwise.

$$\mathcal{E}_{\mathcal{H}} = \sum_{i=1}^{N}\sum_{k=1}^{M} \delta_{ik}\|\phi(\mathbf{x}_i) - \mathbf{m}_k\|^2 \,, \mathbf{m}_k = \frac{\sum_{i=1}^N \delta_{ik}\phi(\mathbf{x}_i)}{\sum_{i=1}^N \delta_{ik}} \tag{1}$$

Usually a kernel function $\mathcal{K} : \mathcal{X} \times \mathcal{X} \to \Re$ [6] is applied to directly provide the inner products in feature space without explicitly defining transformation ϕ (for certain kernel functions the corresponding transformation is intractable). This gives rise to the kernel matrix $K \in \Re^{N\times N}$, $K_{ij} = \mathcal{K}(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^\top\phi(\mathbf{x}_j)$, which is the most common way of representing data in feature space. The squared Euclidean distances in (1) can now be computed using solely the kernel matrix entries (2) (centers $\mathbf{m}_k$ cannot be analytically calculated).

$$\begin{aligned}\|\phi(\mathbf{x}_i) - \mathbf{m}_k\|^2 =& K_{ii} - \frac{2\sum_{j=1}^N \delta_{jk}K_{ij}}{\sum_{j=1}^N \delta_{jk}} \\ &+ \frac{\sum_{j=1}^N\sum_{l=1}^N \delta_{jk}\delta_{lk}K_{jl}}{\sum_{j=1}^N\sum_{l=1}^N \delta_{jk}\delta_{lk}}\end{aligned} \tag{2}$$

By iteratively updating the partitioning through assignments of the instances to their closest center in feature space, kernel *k*-means monotonically converges to a local minimum, which heavily depends on the initial cluster assignments. To avoid poor minima, multiple restarts or, even better,

deterministic-incremental approaches such as the global kernel k-means algorithm [8] could be applied.

B. Spectral Clustering

According to [9], the intra-cluster variance (1) can be equivalently posed as a trace difference:

$$\mathcal{E}_{\mathcal{H}} = tr(K) - tr(Y^{\top}KY)\ , \tag{3}$$

$$\text{where}\ \ Y \in \Re^{N\times M}\ , Y_{ik} = \frac{\delta_{ik}}{\sqrt{\sum_{j=1}^{N}\delta_{jk}}}\ .$$

The first term on the above equation is a constant, therefore the minimization of (3) is equivalent to the maximization of $tr(Y^{\top}KY)$ *w.r.t.* the indicator matrix Y. Due to the discrete nature of Y this becomes a hard optimization problem, but if Y is *relaxed* to be an arbitrary orthonormal matrix (i.e. $Y^{\top}Y = I$), a standard result in linear algebra states that the *optimal* Y is composed of the top M eigenvectors of the kernel matrix K. Therefore, spectral methods which calculate the top eigenvectors of an appropriate matrix and then perform post-processing on these eigenvectors to recover a partitioning can substitute kernel k-means. A popular spectral technique is that of [24].

IV. Multi-view Kernel k-means and Multi-view Spectral Clustering

Motivated by the absence of multi-view clustering methods that differentiate the contribution of the views according to the conveyed information, we present a simple and effective kernel-based scheme which embeds in the clustering process an automatic "ranking" of the views. This "ranking" should wipe out a completely uninformative view, but also allow a less informative one to contribute, with a smaller degree, to the clustering solution.

A. Model Description

Consider a dataset $\mathcal{X}$ with N instances and V views: $\mathcal{X} = \{\boldsymbol{x}_i\}_{i=1}^{N}$, where $\boldsymbol{x}_i = \left\{\mathbf{x}_i^{(v)}\right\}_{v=1}^{V}$ and $\mathbf{x}_i^{(v)} \in \Re^{d^{(v)}}$ are the view vectors for instance $\boldsymbol{x}_i$. As already discussed in Section III, to apply kernel methods, the dataset is implicitly mapped to a feature space and is represented through a kernel matrix. Here it is assumed that V kernel matrices are available, $\left\{K^{(v)}\right\}_{v=1}^{V}$, to which (unknown) transformations $\left\{\phi^{(v)}\right\}_{v=1}^{V}$ and feature spaces $\left\{\mathcal{H}^{(v)}\right\}_{v=1}^{V}$ correspond. To take advantage of all views, we propose the following kernel combination, where w_v are the view weights and p is an exponent:

$$\widetilde{K} = \sum_{v=1}^{V} w_v^p K^{(v)}\ , w_v \geq 0\ , \sum_{v=1}^{V} w_v = 1\ , p \geq 1\ . \tag{4}$$

It is easy to verify that the composite matrix $\widetilde{K}$ is a valid kernel matrix, i.e. a positive semidefinite matrix, to which a transformation $\widetilde{\phi}(\boldsymbol{x}_i) = \left[\sqrt{w_1^p}\phi^{(1)}(\mathbf{x}_i^{(1)})^{\top}, \right.$ $\left., \ldots, \sqrt{w_V^p}\phi^{(V)}(\mathbf{x}_i^{(V)})^{\top}\right]^{\top}$ corresponds, i.e. $\widetilde{K}_{ij} = \widetilde{\phi}(\boldsymbol{x}_i)^{\top}\widetilde{\phi}(\boldsymbol{x}_j)$, that maps the instances to feature space $\widetilde{\mathcal{H}} = \mathcal{H}^{(1)} \times \ldots \times \mathcal{H}^{(V)}$. The weight values, w_v, of the combination (the w_v^p values to be precise) *represent the relevance* of each kernel (view) to the clustering task.

This technique of kernel mixing is widespread in multiple kernel learning, where usually the ℓ_p-norm regularizer is applied, i.e. $\widetilde{K} = \sum_{v=1}^{V} w_v K^{(v)}, w_v \geq 0, \sum_{v=1}^{V} w_v^p \leq 1, p \geq 1$. Different norms allow for different levels of sparsity on the weights, with the ℓ_1-norm [11], [12], [22] favoring very sparse weights and the ℓ_∞-norm [22] reducing to the unweighted case, i.e. $\widetilde{K} = \sum_{v=1}^{V} K^{(v)}$. Norms for $p > 1$ provide a tradeoff between these two extremes [13], [22]. We shall shortly discuss how the exponent p in the above kernel mixture (4) affects sparsity likewise. However, it must be clarified that this work does not focus, by any means, on kernel learning, but exploits kernels as a tool for representing and combining views in multi-view learning.

In order to partition the dataset into M disjoint clusters, $\{\mathcal{C}_k\}_{k=1}^{M}$, and simultaneously exploit all views by learning a suitable kernel $\widetilde{K}$ of the form (4), the intra-cluster variance in space $\widetilde{\mathcal{H}}$ (5) is minimized over the clusters and the weights, *w.r.t.* the constraints in (6). Note that we do not optimize *w.r.t.* p, which must be fixed a priori.

$$\mathcal{E}_{\widetilde{\mathcal{H}}} = \sum_{i=1}^{N}\sum_{k=1}^{M}\delta_{ik}\|\widetilde{\phi}(\boldsymbol{x}_i) - \widetilde{\mathbf{m}}_k\|^2\ , \widetilde{\mathbf{m}}_k = \frac{\sum_{i=1}^{N}\delta_{ik}\widetilde{\phi}(\boldsymbol{x}_i)}{\sum_{i=1}^{N}\delta_{ik}} \tag{5}$$

$$\min_{\{\mathcal{C}_k\}_{k=1}^{M}\ ,\{w_v\}_{v=1}^{V}} \mathcal{E}_{\widetilde{\mathcal{H}}}\ ,\ \textit{s.t.}\ w_v \geq 0\ , \sum_{v=1}^{V} w_v = 1\ , p \geq 1 \tag{6}$$

Using (2) and (4) the objective is rewritten as:

$$\mathcal{E}_{\widetilde{\mathcal{H}}} = \sum_{i=1}^{N}\sum_{k=1}^{M}\delta_{ik}\left(\widetilde{K}_{ii} - \frac{2\sum_{j=1}^{N}\delta_{jk}\widetilde{K}_{ij}}{\sum_{j=1}^{N}\delta_{jk}} + \frac{\sum_{j=1}^{N}\sum_{l=1}^{N}\delta_{jk}\delta_{lk}\widetilde{K}_{jl}}{\sum_{j=1}^{N}\sum_{l=1}^{N}\delta_{jk}\delta_{lk}}\right) \Rightarrow$$

$$\mathcal{E}_{\widetilde{\mathcal{H}}} = \sum_{v=1}^{V} w_v^p \sum_{i=1}^{N}\sum_{k=1}^{M}\delta_{ik}\left(K_{ii}^{(v)} - \frac{2\sum_{j=1}^{N}\delta_{jk}K_{ij}^{(v)}}{\sum_{j=1}^{N}\delta_{jk}} + \frac{\sum_{j=1}^{N}\sum_{l=1}^{N}\delta_{jk}\delta_{lk}K_{jl}^{(v)}}{\sum_{j=1}^{N}\sum_{l=1}^{N}\delta_{jk}\delta_{lk}}\right) \Rightarrow$$

$$\mathcal{E}_{\widetilde{\mathcal{H}}} = \sum_{v=1}^{V} w_v^p \sum_{i=1}^{N}\sum_{k=1}^{M}\delta_{ik}\|\phi^{(v)}(\mathbf{x}_i^{(v)}) - \mathbf{m}_k^{(v)}\|^2\ , \tag{7}$$

$$\text{where} \quad \mathbf{m}_k^{(v)} = \frac{\sum_{i=1}^{N} \delta_{ik} \phi^{(v)}(\mathbf{x}_i^{(v)})}{\sum_{i=1}^{N} \delta_{ik}} \ .$$

Under the spectral perspective, (5) can also be stated in terms of matrix traces, where Y is defined as in (3):

$$\begin{aligned} \mathcal{E}_{\widetilde{\mathcal{H}}} &= tr(\widetilde{K}) - tr(Y^\top \widetilde{K} Y) \\ &= \sum_{v=1}^{V} w_v^p \left(tr(K^{(v)}) - tr(Y^\top K^{(v)} Y) \right) \ . \end{aligned} \tag{8}$$

From (7) and (8) it is obvious that the intra-cluster variance in feature space $\widetilde{\mathcal{H}}$ is the weighted sum of the intra-cluster variances of the individual views' feature spaces, $\mathcal{H}^{(v)}$, *under a common clustering*. Minimizing the view disagreement is the basic principle over which multi-view approaches are built [4].

B. Model Training

Two iterative algorithms that in each iteration alternate between updating the clusters and reestimating the weights are proposed. One follows the distance-based formulation of $\mathcal{E}_{\widetilde{\mathcal{H}}}$ (5) and the other the trace-based spectral formulation (8). They are called multi-view kernel *k*-means (MVKKM) and multi-view spectral clustering (MVSpec) respectively.

1) Updating the clusters for given weights - MVKKM algorithm: When the weights w_v are known, the cluster assignments that minimize the intra-cluster variance can be found in the same way as when only a single kernel is available. The composite kernel, $\widetilde{K} = \sum_{v=1}^{V} w_v^p K^{(v)}$, is first calculated and then kernel *k*-means is applied in space $\widetilde{\mathcal{H}}$. Note that kernel *k*-means requires an initial set of clusters as input. The partitioning returned by the previous MVKKM iteration is used for initializing kernel *k*-means for the current iteration.

2) Updating the clusters for given weights - MVSpec algorithm: Like MVKKM, the composite kernel is first calculated and then the relaxed version of (8) (i.e. Y is allowed to be an arbitrary orthonormal matrix) is considered to compute Y. The optimal solution is composed of the M largest eigenvectors of $\widetilde{K}$, according to the discussion in Section III-B. Note that Y should not be discretized during the iterative process. Otherwise, the monotonic convergence of MVSpec cannot be guaranteed.

3) Updating the weights for given clusters - MVKKM algorithm: For ease of computation, the form of the objective described in (7) is considered together with the constraints from (6). It is easy to verify that the constrained objective is convex *w.r.t.* the weights when $p > 1$, hence their optimal values that minimize $\mathcal{E}_{\widetilde{\mathcal{H}}}$ for the current partitioning can be determined. After some manipulation the following closed form solution emerges (the analytical proof is provided in the appendix):

$$w_v = 1 / \sum_{v'=1}^{V} \left(\frac{\mathcal{D}_v}{\mathcal{D}_{v'}} \right)^{\frac{1}{p-1}} \ \text{if} \ \ p > 1 \ , \tag{9}$$

$$\text{where} \quad \mathcal{D}_v = \sum_{i=1}^{N} \sum_{k=1}^{M} \delta_{ik} \| \phi^{(v)}(\mathbf{x}_i^{(v)}) - \mathbf{m}_k^{(v)} \|^2 \ .$$

For $p = 1$ the optimization problem (6) becomes a linear program. Its solutions lie on the corners of the simplex in the positive orthant spanned by the constraints, which results in a completely sparse outcome:

$$w_v = \begin{cases} 1 \ , & v = \text{argmin}_{v'} \mathcal{D}_{v'} \\ 0 \ , & \text{otherwise} \end{cases} \quad \text{if} \ \ p = 1 \ . \tag{10}$$

4) Updating the weights for given clusters - MVSpec algorithm: We follow an analogous procedure to that of MVKKM with the only difference being that the relaxed formulation of (8) is used instead of (7). All the above remarks regarding the convexity of the objective and the optimality of the weights carry over to MVSpec. Thus, if $p > 1$ the weights are updated through (9), while if $p = 1$ through (10), where now $\mathcal{D}_v = tr(K^{(v)}) - tr(Y^\top K^{(v)} Y)$.

5) Initialization and post-processing: In order to apply both algorithms, initial values for the view weights are required. A uniform weighting ($w_v = 1/V$) of the kernels can be used, which is a reasonable choice, unless prior knowledge regarding the quality of the views is available. Additionally, MVKKM requires an initial set of clusters. To locate a meaningful initial partitioning before executing MVKKM, which is very important in avoiding poor minima during the subsequent iterations, the *global kernel k-means algorithm* [8] is applied that yields near-optimal solutions in a deterministic-incremental fashion. Finally, in the MVSpec method, the eigenvectors are discretized after convergence using *k*-means, as in [24], to get the disjoint clusters.

C. Discussion

In this section, some aspects of the proposed methods are analyzed, starting with the effect of the p exponent. As can be seen from (9), the less the intra-cluster variance $\mathcal{D}_v$ of a view the larger its weight. For $p = 1$ a completely sparse solution emerges (10), regardless of the relative differences in $\mathcal{D}_v$ among the views. Hence, $p = 1$ may discard useful views and thus is effective when a single view is of good quality. For $p > 1$ it is easy to see (9) that the greater (smaller) the p value the less (more) sparse the weights w_v become, i.e. the relative differences in $\mathcal{D}_v$ among the views are suppressed (enhanced). Therefore, a very large p value is useful when kernels of similar quality are available. In practice, intermediate p values are a more reasonable choice, since the most common scenario is that views with complementary information and also degenerate ones exist for the same problem. The above remarks also hold for the w_v^p values, which are the actual coefficients used to combine the kernels (4). Hence, as p increases the w_v^p values become more uniform.

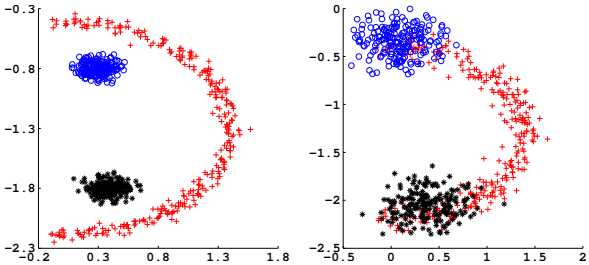
To demonstrate the above a bit more formally, the ratio between any two weights, $w_v / w_{v'}$, can be considered as

an indicator for the sparsity of the solution. The more this ratio tends to 1 the less sparse the outcome. Assume a fixed clustering, i.e. a fixed $\mathcal{D}_v$ and $\mathcal{D}_{v'}$. From (9), $\frac{w_v}{w_{v'}} = \left(\frac{\mathcal{D}_{v'}}{\mathcal{D}_v}\right)^{\frac{1}{p-1}}$ and $\frac{w_v^p}{w_{v'}^p} = \left(\frac{\mathcal{D}_{v'}}{\mathcal{D}_v}\right)^{\frac{p}{p-1}}$, $p > 1$. As p increases, the exponents $1/(p-1)$ and $p/(p-1)$ decrease, therefore both ratios get closer to 1. Hence, the distribution of the w_v and w_v^p values becomes less sparse as p increases. Finally, note that $0 < p < 1$ is not permitted, as in this case the constrained optimized objectives (7), (8) become concave *w.r.t.* the weights, thus the updates, which take the same form as in (9), will increase $\mathcal{E}_{\widetilde{\mathcal{H}}}$.

Regarding the computational complexity, during each of the τ' iterations three main calculations take place; these of the weights, the composite kernel and kernel *k*-means (or the eigenvectors of $\widetilde{K}$). Thus, the overall cost for MVKKM, including the single execution of global kernel *k*-means, is $O(N^2(V+\tau)\tau' + N^3M\tau)$ (τ are the kernel *k*-means iterations), while for MVSpec is $O(N^2MV\tau' + N^3\tau')$. Practically these complexities are comparable and dominated by the N^3 term, as usually $V, M << N$. A reduction by an order of magnitude is possible if global kernel *k*-means is accelerated [8] and specialized numerical packages (e.g. ARPACK) are used to compute the top eigenvectors only.

It is known that kernel *k*-means monotonically decreases the intra-cluster variance. The update on the weights further reduces the objective value. Hence, the distance-based iterative scheme is guaranteed to monotonically converge to a local minimum of $\mathcal{E}_{\widetilde{\mathcal{H}}}$. Moreover, we anticipate this to be a good local mode, since the iterative process starts with a high quality set of clusters, due to the global kernel *k*-means initialization, which is refined after estimating new values for the weights. As previously mentioned, the spectral approach provides a matrix Y that is *optimal* for the current weights *w.r.t.* a *relaxed version* of the considered problem (8), where Y is allowed to be an arbitrary orthonormal matrix. The subsequent update on the weights further reduces the objective, leading to a monotonic convergence to a local minimum as well. Note that a discrete partitioning is obtained only after the MVSpec method has converged. Therefore, it remains to be seen if the decision to relax Y and thus locate the optimal Y in each iteration is effective, compared to the MVKKM case which at each iteration operates with discrete cluster assignments.

We decided to apply the intra-cluster variance function for multi-view clustering as this is one of the most popular clustering criteria and is well posed for kernel-based learning. Moreover, it fits well to the task of automatically constructing a "ranking" of the views, through the kernel combination of (4), and it gives rise to two iterative schemes where the update of the weights and the corresponding partitioning are calculated very easily. In contrast, the maximum margin criterion, which is the most studied for multiple kernel learning [11], [13], [22], [25], results in cumbersome



(a) View 1 (b) View 2 - Noisy

Figure 1. The two synthetic views. Different symbols represent the three sought clusters.

computations even in the supervised case [22], [25]. Iterative frameworks are constantly gaining ground in multiple kernel learning [12], [23], [25], and are proving to be quite efficient.

Finally, it is crucial for the application of both methods that views have comparable intra-cluster variances in feature space. Hence, views must be normalized, for example, as in the experiments, by dividing each view's kernel entries $K_{ij}^{(v)}$ by the average of the pairwise square distances of the view's instances in feature space:$\sum_{i=1}^{N}\sum_{j=1}^{N}(K_{ii}^{(v)} - 2K_{ij}^{(v)} + K_{jj}^{(v)})/N^2$.

V. Empirical Evaluation

The performance of MVKKM and MVSpec is studied on synthetic data as well as on a collection of images and a set of handwritten digits, where multiple views occur naturally. The aim of the experimental evaluation is twofold. First to investigate the p parameters's impact on the returned clusters and the kernel combination coefficients w_v^p, and second to inspect how effective view weighting under our framework is, compared to other multi-view algorithms.

To achieve these goals the two proposed algorithms are executed for various p values, $p > 1$. Moreover, two trivial kernel combinations, $p = 1$ and uniform, are considered. $p = 1$ corresponds to selecting the best kernel, through the weight update process, and splitting the dataset using the information of this kernel only, i.e. it is the best single view case. The uniform combination evenly considers all kernels to obtain a split of the data, i.e. we fix $w_v = 1/V$ in our algorithms and no weight updates are performed (the uniform combination does not depend on the p value). In addition, they are compared to correlational spectral clustering (CSC) [17] and weighted multi-view convex mixture models (MVCMM) [19].

CSC projects the views, which are all thought of as being of the same quality (i.e. no view weights are available), through kernel canonical correlation analysis (KCCA) and then clusters these projections with *k*-means. As in [17], the number of projection axes is set equal to the number of clusters, the KCCA regularization parameters are determined using grid search and *k*-means is restarted 30 times with

Table I
NMI SCORE AND KERNEL COEFFICIENTS DISTRIBUTION ($w_v^p / \sum_{v'=1}^{V} w_{v'}^p$) OF MVKKM AND MVSPEC ON THE SYNTHETIC DATASET, FOR SEVERAL p VALUES AND FOR THE UNIFORM CASE

	MVKKM			MVSpec		
	NMI	Coefficients		NMI	Coefficients	
		View 1	View 2		View 1	View 2
$p = 1$	1	1	0	0.681	1	0
$p = 1.3$	1	0.85	0.15	0.671	0.84	0.16
$p = 1.5$	1	0.77	0.23	0.663	0.74	0.26
$p = 2$	0.769	0.64	0.36	0.632	0.66	0.34
$p = 4$	0.749	0.58	0.42	0.593	0.62	0.38
$p = 6$	0.747	0.56	0.44	0.593	0.62	0.38
Unif.	0.701	0.5	0.5	0.552	0.5	0.5

random initializations and the run with the smallest *k*-means objective is kept.

In MVCMM each view is modeled by a convex mixture model (CMM) [26] and an automatically tuned weight is associated with each view. The only parameter that must be determined in advance is a β parameter which controls the sharpness of the components of the CMMs. In the experiments we search around a reference value β_0, as discussed in [19], to locate a good β and report the best performance.

For all datasets the ground-truth labels are given and are only used to assess the quality of the returned solution with the NMI criterion [8]. Higher NMI values indicate a better match between cluster labels and class labels. The number of clusters is set equal to the true number of classes and linear kernels are employed for MVKKM, MVSpec and CSC to represent the views, unless stated otherwise. For MVCMM, Gaussian convex mixture models are adopted (see [19]). Note that the global kernel *k*-means algorithm (Section IV-B) is utilized to locate initial clusters for MVKKM, thus avoiding the need for multiple restarts.

A. Synthetic Data

To outline the basic properties of the proposed algorithms, a three cluster toy example was created, consisting of two views where the second view is a noisy version of the first (Figure 1). Due to the nonlinearly separable nature of the dataset, an rbf kernel is adopted for each view and its parameter is determined through exhaustive search (here $\sigma = 0.2$ for both views).

From Table I it is evident that as p increases the coefficients w_v^p become more uniform and clustering degrades. This is anticipated since the first view contains all the necessary information to correctly split the data, while the second mixes the clusters. Thus, as the contribution of the second "noisy" view increases, it becomes less probable to recover the true assignments. For small p values, which admit sparser outcomes, the weighting is consistent with the noise level present on the views and MVKKM manages to correctly cluster the data points. Note that even the noisy view contains structural information, hence it is expected to receive nonzero weight even for small p ($p = 1.3, 1.5$). MVSpec, although its coefficients match those of MVKKM, achieves low NMI. We observed that spectral clustering on the first view alone fails to recover the clusters (we executed the popular normalized cut method of [24] for several σ values), giving similar results to MVSpec for $p = 1$ and explaining the deficit of MVSpec.

Figure 2. Examples of handwritten digits contained in the multiple features dataset.

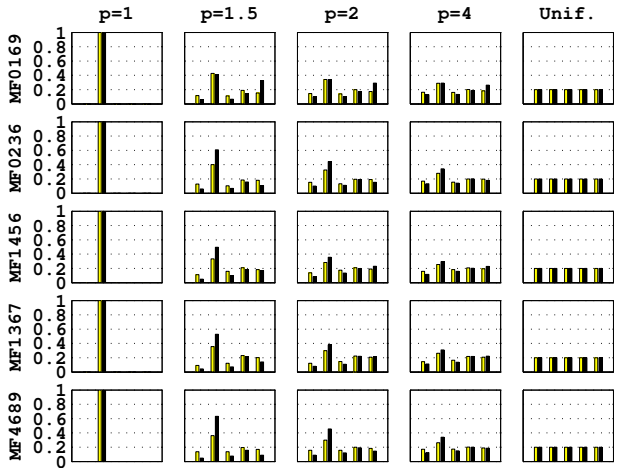
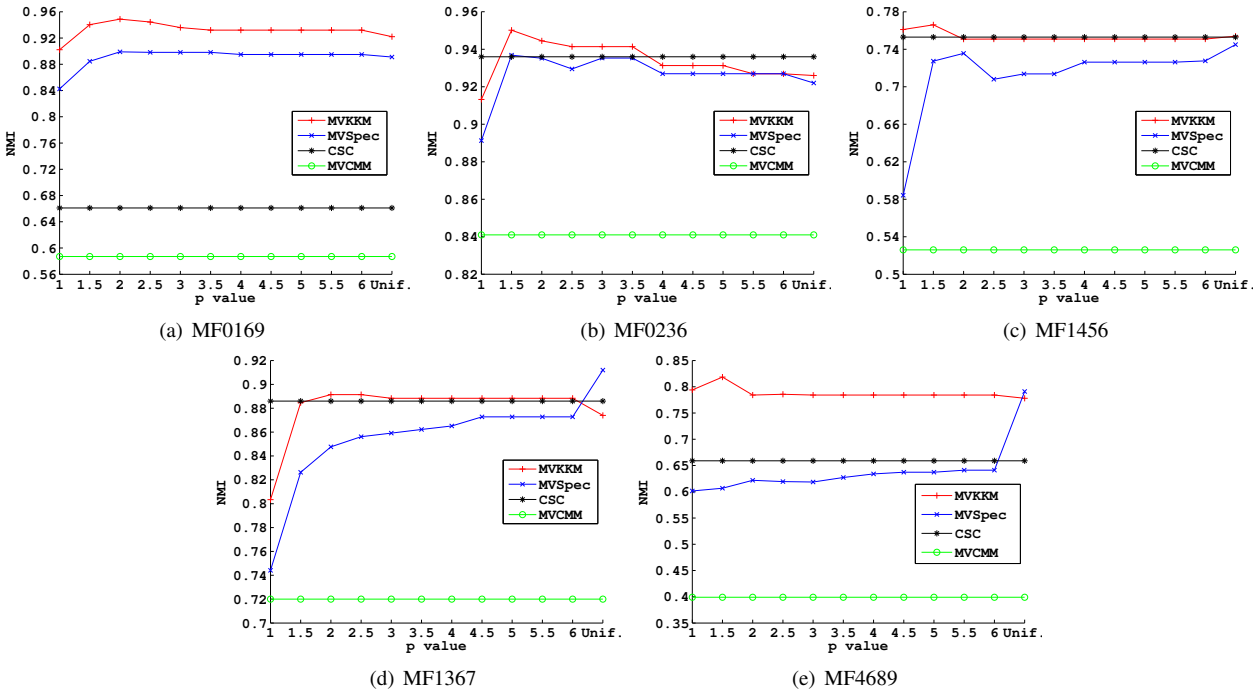


Figure 3. MVKKM (yellow) and MVSpec (black) kernel coefficients distribution ($w_v^p / \sum_{v'=1}^{V} w_{v'}^p$) on the multiple features dataset, for several p values and for the uniform case.

B. Multiple Features Dataset

Multiple features is a database of handwritten digits (0-9)[2]. The digits (200 per class) are represented by several attribute sets (i.e. views), namely Fourier coefficients, profile correlations, Karhunen-Love coefficients, pixel averages and Zernike moments (note that this is the order of the views in Figure 3). From the original dataset several four class subsets were created and the most representative ones are presented here. As attributes within the same view exhibit significantly different scales, all views' attributes were normalized to unit variance. Moreover, kernel entries were divided by the average pairwise square distance of the corresponding view, as discussed in Section IV-C. This preprocessing was also

[2]http://www.ics.uci.edu/~mlearn/MLRepository.html



(a) MF0169 (b) MF0236 (c) MF1456

(d) MF1367 (e) MF4689

Figure 4. NMI score of the compared methods on the multiple features dataset, for several p values and for the uniform case.

Figure 5. Examples of images contained in the corel collection.

applied to CSC and MVCMM[3].

The comparison of the four algorithms is provided in Figure 4, where the subsets are named according to the included numerals. Note that CSC and MVCMM do not depend on p. MVKKM is superior to MVSpec for almost all p values, indicating that the distance-based formulation of the objective is more appropriate. MVCMM, despite using weights, always yields the least NMI, thus highlighting the potential of our clustering technique. CSC is quite competitive, being slightly (except for MF0169 and MF4689) inferior to MVKKM and MVSpec for the best p. Moreover, the single view case ($p = 1$) proves to be the worst, while the uniform (Unif.) is close in accuracy to that for the best p. This fact together with i) the effectiveness of the unweighted CSC method for most subsets and ii) the minor, only, drop in NMI as p increases (for MVSpec even an increase is observed for MF1367 and MF4689), hence the kernel coefficients, w_v^p, distribution evolves towards uniformity (Figure 3), lead us to conclude that all views contribute significantly in the multiple features dataset. Still though, a p value that admits some sparsity on the solution can enhance performance, particularly for MVKKM where $p = 1.5$ or $p = 2$ is always the best choice.

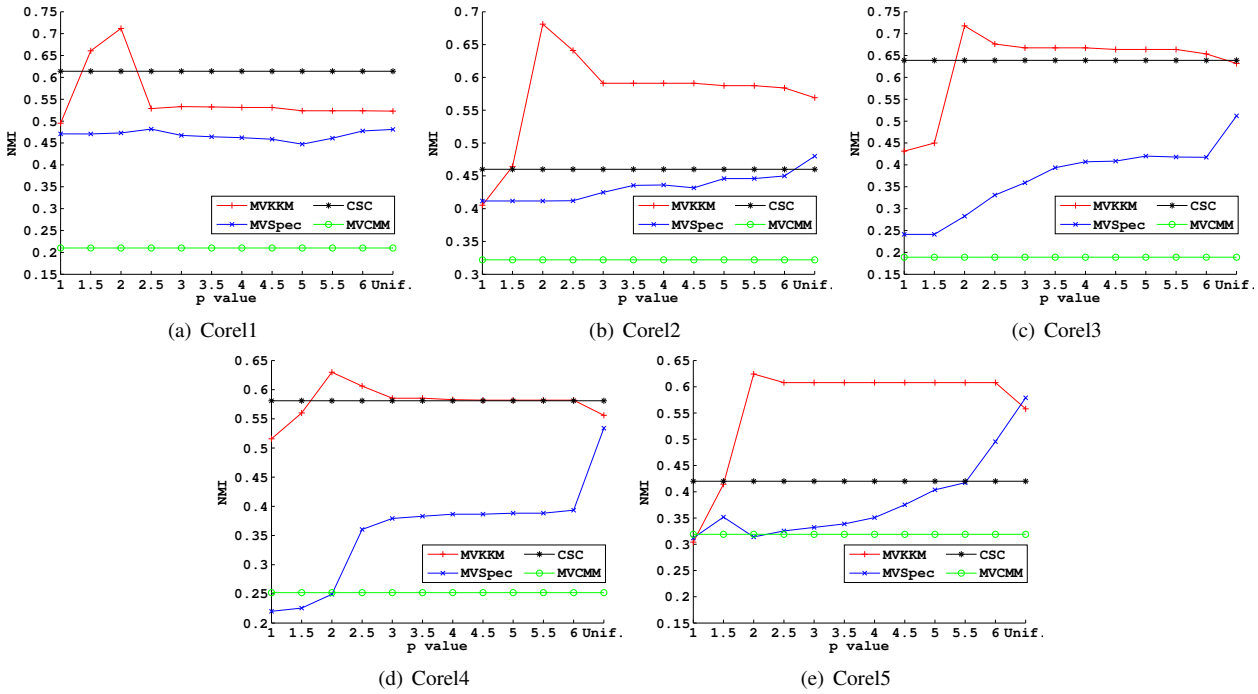
[3]MVCMM is not kernel-based, therefore the distances in the Gaussian components were instead normalized.

Table II
CATEGORIES CONTAINED IN THE TESTED COREL SUBSETS

Subset	Categories			
Corel1	*owls*	*wildlife*	*trains*	*cargo ships*
Corel2	*buses*	*leopards*	*trains*	*cargo ships*
Corel3	*buses*	*leopards*	*cars*	*passenger ships*
Corel4	*owls*	*wildlife*	*hawks*	*roses*
Corel5	*eagles*	*elephants*	*trains*	*passenger ships*

C. Corel Images Dataset

A part of the popular corel collection consisting of 34 categories, each with 100 images, serves as our second real multi-modal paradigm. Images consist of a salient foreground object, but within each class there is great variance in terms of distance and angle of the object, color, lighting, and background composition, making this dataset difficult for unsupervised learning. Attribute vectors that represent the images in terms of seven views, three color-related views (color histogram, moment and coherence) and four texture-related views (coarseness and directionality of



(a) Corel1 (b) Corel2 (c) Corel3

(d) Corel4 (e) Corel5

Figure 6. NMI score of the compared methods on the corel dataset, for several p values and for the uniform case.

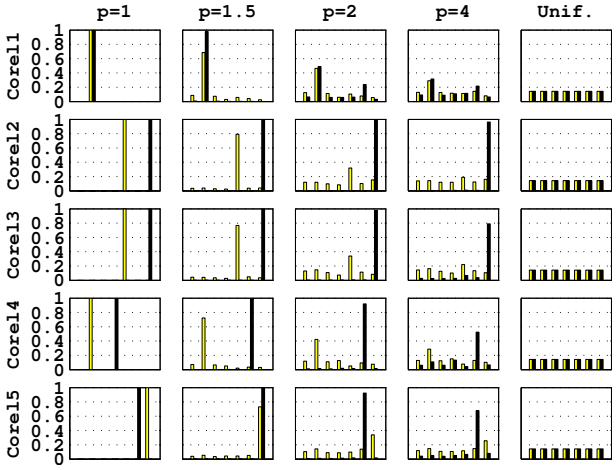


Figure 7. MVKKM (yellow) and MVSpec (black) kernel coefficients distribution ($w_v^p / \sum_{v'=1}^{V} w_{v'}^p$) on the corel dataset, for several p values and for the uniform case.

tamura texture, wavelet and mrsar texture) are available for this collection[4] (note that this is the order of the views in Figure 7). Many four class subsets were extracted and the most representative ones are included in the paper (Table II). The kernels were normalized as for multiple features.

Results are depicted in Figure 6. MVKKM for $p = 2$ considerably outperforms the other three algorithms and its kernel coefficients, w_v^p, distribution (Figure 7) indicates that a nonuniform mixture is suited to this dataset, thus explaining the deficit of larger p values and CSC. Moreover, its advantage over MVSpec, for which the NMI increases as p increases and a uniform solution is preferable, is significant for all p values. The difference between the two clustering schemes can be explained from Figure 7, where a disagreement is observed regarding which view should acquire the highest weight (except for corel1) and a more peaked coefficient distribution for MVSpec. It seems that MVSpec selects inappropriate views, indicating that the relaxed problem becomes detached from the actual objective (8) during the iterative process (it even yields worse results than CSC). It is worth noting that both MVKKM and MVSpec underperform for very small p ($p = 1$, $p = 1.5$), i.e. for very sparse combinations, thus exploiting information from all views is necessary for the tested real data. Finally note that MVCMM is not performing well on this or the previous dataset, despite automatically estimating view weights. This emanates from the very sparse solution recovered by the method, that assigns zero weights to most views.

D. Discussion

The empirical evaluation has shown that the distance-based formulation of the objective provides better results than the spectral. There is dual reason for this behavior. First, MVSpec provides at each iteration a continuous solution Y which at the end is discretized to obtain the final partitioning. The continuous solution runs the risk of deviating from the original non-relaxed objective, especially in iterative

[4]http://www.cs.virginia.edu/~xj3a/research/CBIR/Download.htm

algorithms, such as MVSpec, where the weights get updated based on the relaxed objective. On the contrary, MVKKM provides a discrete partition in every iteration, thus following "closely" the intra-cluster variance objective. Hence, the relaxation can lead to the selection of suboptimal views, whose influence is enhanced for sparser solutions (i.e. for smaller p). This case arose for the corel dataset (Figure 7) and explains why MVSpec usually attains its highest NMI for the uniform case.

Second, for the initialization of MVKKM the global kernel *k*-means procedure was employed, which is deterministic and very effective [8]. As the experiments with the synthetic and multiple features datasets indicate, a properly initialized kernel *k*-means can locate better clusters than spectral techniques, since MVKKM outperforms MVSpec despite both techniques resulting in similar w_v^p values. These two reasons also elucidate why CSC performs better than MVSpec for most p values on the real data.

Further ground for the above remarks was provided when we executed for the corel subsets i) a run of kernel *k*-means using the MVSpec-derived composite kernel and ii) spectral analysis over the MVKKM-derived composite kernel. The results were always inferior to those reported for MVKKM in Section V-C, demonstrating that the distance-based formulation infers both better cluster structures and view weights.

Furthermore, for MVKKM, which is always the best of the tested methods, selecting either the best view or equally all views proves to be inadequate, highlighting the importance of allowing the clustering algorithm to mix views more robustly and finding a balance between sparsity and uniformity. This is also reported in many multi-view and multiple kernel learning studies [4], [10], [16], [19], [22], [25]. The appropriate p value is, of course, dataset dependent.

VI. Conclusions

We have proposed two multi-modal approaches that represent modalities through kernel matrices and optimize the intra-cluster variance function. A weighted combination of the kernels that resembles the ℓ_p-norm regularizer [22] and reflects the views' relevance to the clustering task is automatically learned using closed form updates. The new methods, particularly MVKKM, compare favorable to existing ones, underlying the strength of our framework and that view weighting can boost the quality of the partitioning, if the sparsity of the weights is appropriately moderated.

In future work we plan to explore possible ways of determining p automatically. Moreover, investigating the connections between multi-view clustering and multiple kernel learning could provide interesting directions in developing and improving multi-modal algorithms. Also, the ideas of view weighting could be adapted to kernel-based unsupervised attribute weighting.

References

[1] S. Dasgupta, M. L. Littman, and D. A. McAllester, "PAC generalization bounds for co-training," in *Adv. Neural Inf. Process. Syst. 14*, 2001, pp. 375–382.

[2] K. Nigam and R. Ghani, "Analyzing the effectiveness and applicability of co-training," in *Proc. 9th Int. Conf. Inf. Knowl. Manage.*, 2000, pp. 86–93.

[3] A. Blum and T. Mitchell, "Combining labeled and unlabeled data with co-training," in *Proc. 11th Annu. Conf. Comput. Learn. Theory*, 1998, pp. 92–100.

[4] S. Bickel and T. Scheffer, "Multi-view clustering," in *Proc. 4th IEEE Int. Conf. Data Mining*, 2004, pp. 19–26.

[5] B. Long, P. S. Yu, and Z. M. Zhang, "A general model for multiple view unsupervised learning," in *Proc. 8th SIAM Int. Conf. Data Mining*, 2008, pp. 822–833.

[6] M. Filippone, F. Camastra, F. Masulli, and S. Rovetta, "A survey of kernel and spectral methods for clustering," *Pattern Recognit.*, vol. 41, no. 1, pp. 176–190, 2008.

[7] B. Schölkopf, A. J. Smola, and K.-R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Comput.*, vol. 10, no. 5, pp. 1299–1319, 1998.

[8] G. Tzortzis and A. Likas, "The global kernel k-means algorithm for clustering in feature space," *IEEE Trans. Neural Netw.*, vol. 20, no. 7, pp. 1181–1194, 2009.

[9] I. S. Dhillon, Y. Guan, and B. Kulis, "Weighted graph cuts without eigenvectors a multilevel approach," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 11, pp. 1944–1957, 2007.

[10] T. Lange and J. M. Buhmann, "Fusion of similarity data in clustering," in *Adv. Neural Inf. Process. Syst. 18*, 2005, pp. 723–730.

[11] H. Valizadegan and R. Jin, "Generalized maximum margin clustering and unsupervised kernel learning," in *Adv. Neural Inf. Process. Syst. 19*, 2006, pp. 1417–1424.

[12] H. Zeng and Y. ming Cheung, "Kernel learning for local learning based clustering," in *Proc. Int. Conf. Artif. Neural Netw. Part I*, 2009, pp. 10–19.

[13] B. Zhao, J. T. Kwok, and C. Zhang, "Multiple kernel clustering," in *Proc. 9th SIAM Int. Conf. Data Mining*, 2009, pp. 638–649.

[14] S. Bickel and T. Scheffer, "Estimation of mixture models using co-em," in *Proc. 16th Eur. Conf. Mach. Learn.*, 2005, pp. 35–46.

[15] V. R. de Sa, "Spectral clustering with two views," in *Proc. 22nd Int. Conf. Mach. Learn. Workshop Learn. Multiple Views*, 2005, pp. 20–27.

[16] D. Zhou and C. J. C. Burges, "Spectral clustering and transductive learning with multiple views," in *Proc. 24th Int. Conf. Mach. Learn.*, 2007, pp. 1159–1166.

[17] M. B. Blaschko and C. H. Lampert, "Correlational spectral clustering," in *Proc. 21st IEEE Conf. Comput. Vis. Pattern Recognit.*, 2008, pp. 1–8.

[18] K. Chaudhuri, S. M. Kakade, K. Livescu, and K. Sridharan, "Multi-view clustering via canonical correlation analysis," in *Proc. 26th Int. Conf. Mach. Learn.*, 2009, pp. 129–136.

[19] G. Tzortzis and A. Likas, "Multiple view clustering using a weighted combination of exemplar-based mixture models," *IEEE Trans. Neural Netw.*, vol. 21, no. 12, pp. 1925–1938, 2010.

[20] A. Kumar and H. Daumé III, "A co-training approach for multiview spectral clustering," in *Proc. 28th Int. Conf. Mach. Learn.*, 2011, pp. 393–400.

[21] D. Greene and P. Cunningham, "A matrix factorization approach for integrating multiple data views," in *Proc. Eur. Conf. Mach. Learn. Knowl. Discovery Databases*, 2009, pp. 423–438.

[22] M. Kloft, U. Brefeld, S. Sonnenburg, P. Laskov, K.-R. Müller, and A. Zien, "Efficient and accurate lp-norm multiple kernel learning," in *Adv. Neural Inf. Process. Syst. 22*, 2009, pp. 997–1005.

[23] C. Cortes, M. Mohri, and A. Rostamizadeh, "Learning non-linear combinations of kernels," in *Adv. Neural Inf. Process. Syst. 22*, 2009, pp. 396–404.

[24] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Adv. Neural Inf. Process. Syst. 14*, 2001, pp. 849–856.

[25] Z. Xu, R. Jin, H. Yang, I. King, and M. R. Lyu, "Simple and efficient multiple kernel learning by group lasso," in *Proc. 27th Int. Conf. Mach. Learn.*, 2010, pp. 1175–1182.

[26] D. Lashkari and P. Golland, "Convex clustering with exemplar-based models," in *Adv. Neural Inf. Process. Syst. 20*, 2008, pp. 825–832.

APPENDIX A.
PROOF OF THE WEIGHT UPDATE FORMULA FOR MVKKM

For convenience, let us rewrite the optimization problem for given clusters, using the form of the objective in (7):

$$\min_{\{w_v\}_{v=1}^V} \sum_{v=1}^{V} w_v^p \sum_{i=1}^{N}\sum_{k=1}^{M} \delta_{ik} \|\phi^{(v)}(\mathbf{x}_i^{(v)}) - \mathbf{m}_k^{(v)}\|^2 \ ,$$

$$s.t.\ w_v \geq 0 \ , \sum_{v=1}^{V} w_v = 1 \ . \tag{11}$$

Consider the case for $p > 1$ and denote by $\mathcal{D}_v$ the intra-cluster variance of the v-th view feature space $\mathcal{H}^{(v)}$, i.e. $\mathcal{D}_v = \sum_{i=1}^{N}\sum_{k=1}^{M} \delta_{ik} \|\phi^{(v)}(\mathbf{x}_i^{(v)}) - \mathbf{m}_k^{(v)}\|^2$. By incorporating into the objective the sum-to-unity constraint, the Lagrangian becomes:

$$\mathcal{L} = \sum_{v=1}^{V} w_v^p \mathcal{D}_v + \lambda \left(\sum_{v=1}^{V} w_v - 1 \right) \ . \tag{12}$$

Setting the derivative of the Lagrangian to zero yields

$$\frac{\partial \mathcal{L}}{\partial w_v} = 0 \Rightarrow p w_v^{(p-1)} \mathcal{D}_v + \lambda = 0 \Rightarrow w_v = \left(\frac{-\lambda}{p D_v} \right)^{\frac{1}{p-1}} \ . \tag{13}$$

By summing over all views, together with the constraint $\sum_{v=1}^{V} w_v = 1$, we get

$$\sum_{v'=1}^{V} \left(\frac{-\lambda}{p D_{v'}} \right)^{\frac{1}{p-1}} = 1 \Rightarrow (-\lambda)^{\frac{1}{p-1}} = 1 / \sum_{v'=1}^{V} \left(\frac{1}{p \mathcal{D}_{v'}} \right)^{\frac{1}{p-1}} . \tag{14}$$

Finally, substituting (14) into (13) completes the proof:

$$w_v = 1 / \sum_{v'=1}^{V} \left(\frac{\mathcal{D}_v}{\mathcal{D}_{v'}} \right)^{\frac{1}{p-1}} \text{ if } \ p > 1 \ . \tag{15}$$

Note that the non-negativity of the weights was not enforced into (12), since it is verified by the solution (15), as $\mathcal{D}_v \geq 0$.

For the $p = 1$ case, it easy to see that for any weight values $w_{v'}$ obeying the constraints of (11) and corresponding $\mathcal{D}_{v'} \geq 0$, the following holds:

$$\mathcal{D}_{v^*} \leq \sum_{v'=1}^{V} w_{v'} \mathcal{D}_{v'} \ , v^* = \operatorname*{argmin}_{v'} \mathcal{D}_{v'} \ , \tag{16}$$

from which directly follows that (11) is minimized for

$$w_v = \begin{cases} 1 \ , & v = \operatorname{argmin}_v \mathcal{D}_v \\ 0 \ , & \text{otherwise} \end{cases} \text{ if } \ p = 1 \ . \tag{17}$$

□

APPENDIX B.
PROOF OF THE WEIGHT UPDATE FORMULA FOR MVSPEC

Using the form of the objective in (8), the optimization problem for given clusters can be written as:

$$\min_{\{w_v\}_{v=1}^V} \sum_{v=1}^{V} w_v^p \left(tr(K^{(v)}) - tr(Y^\top K^{(v)} Y) \right) \ ,$$

$$s.t.\ w_v \geq 0 \ , \sum_{v=1}^{V} w_v = 1 \ . \tag{18}$$

The similarity to the MVKKM optimization problem is evident, with the only difference being that $\mathcal{D}_v = tr(K^{(v)}) - tr(Y^\top K^{(v)} Y)$. Since $K^{(v)}$ is a positive semidefinite matrix and $Y^\top Y = I$, $Y \in \Re^{N \times M}$, by the Ky-Fan theorem [5] we have $\mathcal{D}_v \geq 0$. Therefore, the derivations are analogous to the MVKKM case. □