Ratio-Based Multiple Kernel Clustering

Grigorios Tzortzis and Aristidis Likas

Department of Computer Science & Engineering, University of Ioannina,
GR 45110, Ioannina, Greece
{gtzortzi,arly}@cs.uoi.gr

Abstract. Maximum margin clustering (MMC) approaches extend the large margin principle of SVM to unsupervised learning with considerable success. In this work, we utilize the ratio between the margin and the intra-cluster variance, to explicitly consider both the separation and the compactness of the clusters in the objective. Moreover, we employ multiple kernel learning (MKL) to jointly learn the kernel and a partitioning of the instances, thus overcoming the kernel selection problem of MMC. Importantly, the margin alone cannot reliably reflect the quality of the learned kernel, as it can be enlarged by a simple scaling of the kernel. In contrast, our ratio-based objective is scale invariant and also invariant to the type of norm constraints on the kernel parameters. Optimization of the objective is performed using an iterative gradient-based algorithm. Comparative clustering experiments on various datasets demonstrate the effectiveness of the proposed formulation.

Keywords: maximum margin clustering, unsupervised multiple kernel learning, kernel k-means.

1 Introduction

The success of large margin techniques in supervised learning, particularly that of support vector machines (SVM), has generated great interest in extending such techniques to the unsupervised setting, leading to the, so called, maximum margin clustering (MMC) problem [21]. Given a dataset $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^N$, $\mathbf{x}_i \in \Re^d$, MMC approaches attempt to find a labeling (clustering) $\mathbf{y} = [y_1, \ldots, y_N]^\top$, $y_i \in \{\pm 1\}$, of the instances, such that a subsequent training of a standard SVM would result in a margin that is maximal over all possible labellings. MMC is formulated as:

$$\min_{\mathbf{y}} \min_{\mathbf{w},b,\boldsymbol{\xi}} \ \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N}\xi_i, \tag{1}$$

$$s.t. \ \ -\ell \leq \sum_{i=1}^{N} y_i \leq \ell,\ \mathbf{y} \in \{\pm 1\}^N,\ y_i\left(\mathbf{w}^\top \phi(\mathbf{x}_i) + b\right) \geq 1 - \xi_i,\ \xi_i \geq 0,$$

where $\mathbf{w}$, b are the coefficients of the SVM hyperplane ($\|\mathbf{w}\|$ is the reciprocal of the margin), $\boldsymbol{\xi}$ the slack variables capturing the misclassification error and

T. Calders et al. (Eds.): ECML PKDD 2014, Part III, LNCS 8726, pp. 241–257, 2014.

$C > 0$ the regularizer. Instances are implicitly mapped through transformation ϕ to a higher dimensional feature space using the kernel trick ($\mathcal{K}(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^\top \phi(\mathbf{x}_j)$). Moreover, to prevent the trivially "optimal" solution of assigning all instances to the same cluster and thus obtaining an infinite margin ($\|\mathbf{w}\| = 0$), a cluster balance constraint ($-\ell \leq \sum_{i=1}^{N} y_i \leq \ell$) was introduced by Xu et al. [21], where $\ell \geq 0$ is a constant controlling the imbalance of the clusters. The MMC problem is non-convex with integer parameters $\mathbf{y}$, making the optimization much trickier than that of (convex) supervised SVM. To solve (1), some approaches employ semidefinite programing (SDP) [18, 21, 22], others exploit the cutting plane method [20, 25] and others rely on alternating between the outer and the inner minimization [24].

It is well-known that the performance of kernel-based approaches, like MMC, heavily depends on the choice of the kernel. However, it is often unclear which is the best kernel for a particular task. Multiple kernel learning (MKL) [9], which has been mainly studied under the SVM paradigm, attempts to simultaneously locate the hyperplane with the largest margin and also learn a suitable kernel. The kernel, $\widetilde{\mathcal{K}}(\mathbf{x}_i, \mathbf{x}_j) = \widetilde{\phi}(\mathbf{x}_i)^\top \widetilde{\phi}(\mathbf{x}_j)$, is usually parametrized by a vector $\boldsymbol{\theta} = [\theta_1, \ldots, \theta_V]^\top$ of parameters. Most existing MKL approaches focus on supervised learning and in principle derive from the following optimization (subject to some slight modifications) (e.g. [11, 12, 14, 23]):

$$\min_{\boldsymbol{\theta}, \mathbf{w}, b, \boldsymbol{\xi}} \ \frac{1}{2}\|\mathbf{w}\|^2 + C \sum_{i=1}^{N} \xi_i, \tag{2}$$

$$s.t. \ \ \theta_v \geq 0, \ \|\boldsymbol{\theta}\|_p^p \leq 1, \ y_i \left(\mathbf{w}^\top \widetilde{\phi}(\mathbf{x}_i) + b \right) \geq 1 - \xi_i, \ \xi_i \geq 0.$$

Kernel parameters θ_v are limited to nonnegative values to ensure the learned kernel is positive semidefinite and the p-norm constraint is employed to avoid overfitting. Usually the kernel is parametrized as a linear combination of some given basis kernels and either the 1-norm that promotes sparsity [14, 16, 26], or a more general p-norm, $p \geq 1$, [11, 12, 23], is chosen. There also exist a few studies that consider nonlinear combinations of basis kernels [3, 8], or even general types of parametric kernels [7, 19]. The optimization problem in (2) is non-convex due to $\boldsymbol{\theta}$. Depending on the form of the kernel parametrization and the choice of p-norm, various optimization strategies have been proposed, several of which alternate between updating $\boldsymbol{\theta}$ and solving a standard SVM to obtain $\mathbf{w}$, b and $\boldsymbol{\xi}$. For example, semi-infinite linear programming [11, 16, 26], gradient-based methods [7, 8, 14, 19] and closed-form methods [12, 23].

Extending MKL to the clustering domain, and in particular to MMC problems, is an interesting research direction, however, existing work is rather limited. The methods of [18, 25] seek to find a linear mixture of the basis kernels together with the cluster assignments, such that the margin is maximized, in essence combining (1) and (2). In this paper, we follow a similar path, but propose a novel objective that considers the ratio between the margin (a notion of cluster separability) and the intra-cluster variance criterion of kernel k-means [5] (a notion of cluster coherence). Hence, both the separation and the compactness

of the clusters are explicitly taken into account, which can possibly improve on the solutions returned by approaches utilizing either of the two. Importantly, the margin has been shown to suffer from a major deficiency when applied to supervised MKL [7]. It can become arbitrarily large by a simple scaling of the kernel, thus it is inappropriate for assessing the quality of the learned kernel. The same can be demonstrated to hold for unsupervised MKL and we prove that our ratio-based objective is invariant to kernel scaling, thus overcoming this deficiency. Moreover, its global optimum solution is invariant to the type of p-norm constraint on the kernel parameters $\boldsymbol{\theta}$ (when a linear combination of basis kernels is employed), making the selection of a suitable norm less crucial.

A simple gradient-based optimization procedure that alternates between updating the kernel parameters $\boldsymbol{\theta}$ and the cluster assignments $\mathbf{y}$ is devised, avoiding the invocation of complex optimizers, such as the SDP solvers [18] and the cutting plane method [25]. Experiments on several datasets, including two collections of handwritten numerals and two image collections, reveal the superiority of the proposed method over approaches that rely solely on the margin or the intra-cluster variance.

The rest of this paper is organized as follows. Section 2 introduces our ratio-based formulation and presents its invariance properties and optimization details. Experiments follow in Section 3, before the concluding remarks of Section 4.

2 The RMKC Algorithm

2.1 Problem Formulation

Consider a dataset $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^N$, $\mathbf{x}_i \in \Re^d$, for which we want to simultaneously infer the cluster labels and also perform kernel learning under the large margin framework. While presenting our method we shall restrict ourselves on a linear combination of basis kernels, which is the most common technique of parametrizing kernels for MKL [12, 14, 23]. Later we will show that our model can accommodate more general parametric forms of kernels.

Assume that V basis kernels, $\mathcal{K}^{(v)} : \mathcal{X} \times \mathcal{X} \to \Re$, are available, each implicitly inducing a transformation $\phi^{(v)} : \mathcal{X} \to \mathcal{H}^{(v)}$ on the instances to a feature space $\mathcal{H}^{(v)}$ through $\mathcal{K}^{(v)}(\mathbf{x}_i, \mathbf{x}_j) = \phi^{(v)}(\mathbf{x}_i)^\top \phi^{(v)}(\mathbf{x}_j)$. A linear mixture of kernels gives rise to a composite kernel $\widetilde{\mathcal{K}}$:

$$\widetilde{\mathcal{K}}(\mathbf{x}_i, \mathbf{x}_j) = \sum_{v=1}^{V} \theta_v \mathcal{K}^{(v)}(\mathbf{x}_i, \mathbf{x}_j),\ \theta_v \geq 0, \tag{3}$$

that is parametrized by $\boldsymbol{\theta} = [\theta_1, \ldots, \theta_V]^\top$. Since $\widetilde{\mathcal{K}}$ is a valid kernel it holds that $\widetilde{\mathcal{K}}(\mathbf{x}_i, \mathbf{x}_j) = \widetilde{\phi}(\mathbf{x}_i)^\top \widetilde{\phi}(\mathbf{x}_j)$, $\widetilde{\phi} : \mathcal{X} \to \widetilde{\mathcal{H}}$, and actually $\widetilde{\phi}(\mathbf{x}_i) = \left[\sqrt{\theta_1}\phi^{(1)}(\mathbf{x}_i)^\top, \ldots, \sqrt{\theta_V}\phi^{(V)}(\mathbf{x}_i)^\top\right]^\top$ due to the linear combination.

We propose a new formulation that does not depend only on the margin, like most existing MMC and MKL studies, but utilizes the ratio between the margin

and the intra-cluster variance objective of kernel k-means [5] in feature space $\widetilde{\mathcal{H}}$. Minimizing such a ratio can lead to superior partitionings as both compact and well-separated clusters are sought. Moreover, as it will be proved, it makes our formulation invariant to kernel scaling, an important property when kernel learning is involved [7]. Denoting by $\mathbf{y} = [y_1, \ldots, y_N]^\top$, $y_i \in \{\pm 1\}$, the vector of the instances' cluster labels, we consider the following optimization problem:

$$\min_{\boldsymbol{\theta}, \mathbf{y}} \mathcal{J}(\boldsymbol{\theta}, \mathbf{y}),\ s.t.\ \theta_v \geq 0,\ \|\boldsymbol{\theta}\|_p^p = 1,\ -\ell \leq \sum_{i=1}^{N} y_i \leq \ell,\ \mathbf{y} \in \{\pm 1\}^N, \tag{4}$$

$$\mathcal{J}(\boldsymbol{\theta}, \mathbf{y}) = \min_{\mathbf{w}, b, \boldsymbol{\xi}} \frac{1}{2}\mathcal{E}(\boldsymbol{\theta}, \mathbf{y})\|\mathbf{w}\|^2 + C\sum_{i=1}^{N} \xi_i, \tag{5}$$

$$s.t.\ \ y_i \left(\mathbf{w}^\top \widetilde{\phi}(\mathbf{x}_i) + b\right) \geq 1 - \xi_i,\ \xi_i \geq 0.$$

Here $\mathcal{E}(\boldsymbol{\theta}, \mathbf{y})$ is the kernel k-means criterion (6) describing the intra-cluster variance[1], where $\widetilde{\mathbf{m}}_k$ is the k-th cluster center and δ_{ik} is a cluster indicator variable with $\delta_{i1} = 1$ if $y_i = -1$ and $\delta_{i2} = 1$ if $y_i = 1$. Note that due to the SVM-like formulation we are limited to two-cluster solutions, i.e. $k \in \{1, 2\}$, which is the typical case for MMC methods.

$$\mathcal{E}(\boldsymbol{\theta}, \mathbf{y}) = \frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{2} \delta_{ik}\|\widetilde{\phi}(\mathbf{x}_i) - \widetilde{\mathbf{m}}_k\|^2, \tag{6}$$

$$\delta_{ik} = \begin{cases} 1, \ y_i = 2k - 3 \\ 0, \text{ otherwise} \end{cases},\ \widetilde{\mathbf{m}}_k = \frac{\sum_{i=1}^{N} \delta_{ik}\widetilde{\phi}(\mathbf{x}_i)}{\sum_{i=1}^{N} \delta_{ik}}$$

Note that the squared Euclidean distances in $\mathcal{E}(\boldsymbol{\theta}, \mathbf{y})$ can be posed solely in terms of the entries of the kernel matrix $\widetilde{K} \in \Re^{N \times N}$ corresponding to $\widetilde{\mathcal{K}}$, i.e. $\widetilde{K}_{ij} = \widetilde{\mathcal{K}}(\mathbf{x}_i, \mathbf{x}_j)$ [5]. Additionally, by using (3), this composite kernel matrix can be written as the sum of the basis kernel matrices $K^{(v)} \in \Re^{N \times N}$, i.e. $\widetilde{K} = \sum_{v=1}^{V} \theta_v K^{(v)}$, thus getting (7).

$$\mathcal{E}(\boldsymbol{\theta}, \mathbf{y}) = \frac{1}{N}\sum_{v=1}^{V}\theta_v \sum_{i=1}^{N}\sum_{k=1}^{2} \delta_{ik}\left(K_{ii}^{(v)} - \frac{2\sum_{j=1}^{N}\delta_{jk}K_{ij}^{(v)}}{\sum_{j=1}^{N}\delta_{jk}} + \frac{\sum_{j=1}^{N}\sum_{l=1}^{N}\delta_{jk}\delta_{lk}K_{jl}^{(v)}}{\sum_{j=1}^{N}\sum_{l=1}^{N}\delta_{jk}\delta_{lk}}\right) \tag{7}$$

For the above optimization problem (4), it is easy to verify that its objective function $\mathcal{J}(\boldsymbol{\theta}, \mathbf{y})$ at a given $\{\boldsymbol{\theta}, \mathbf{y}\}$ is defined as the optimal objective value of a problem (5) that closely resembles the standard SVM. The only difference is that the variance to margin ratio is employed in place of the margin. Similar to MMC methods [21,24], a cluster balance constraint ($-\ell \leq \sum_{i=1}^{N} y_i \leq \ell$) must be

[1] For simplicity, on the following, we shall refer to the intra-cluster variance as the variance of the clusters.

imposed to prevent meaningless solutions from arising. Finally, the composite kernel coefficients θ_v are required to be nonnegative so that $\widetilde{K}$ is a valid kernel and a p-norm constraint is introduced to avoid overfitting, as in (2).

Hence, the optimization in (4) searches for a pair of $\{\boldsymbol{\theta}, \mathbf{y}\}$ values that yields a small variance to margin ratio ($\mathcal{E}(\boldsymbol{\theta}, \mathbf{y})\|\mathbf{w}\|^2$) regularized by the misclassification error (captured by the slack variables $\boldsymbol{\xi}$). We shall call this approach Ratio-based Multiple Kernel Clustering, abbreviated as RMKC.

It should be clarified that the actual problem we are trying to solve is (s.t. the constraints in (4)-(5)):

$$\min_{\boldsymbol{\theta},\mathbf{y},\mathbf{w},b,\boldsymbol{\xi}} \frac{1}{2}\mathcal{E}(\boldsymbol{\theta}, \mathbf{y})\|\mathbf{w}\|^2 + C\sum_{i=1}^{N}\xi_i, \tag{8}$$

which is rather difficult to directly optimize, since it constitutes a non-convex problem with integer parameters $\mathbf{y}$. Reformulating it as in (4), analogously to Rakotomamonjy et al. [14], will enable us to devise an alternating optimization strategy, that benefits from differentiability w.r.t. $\boldsymbol{\theta}$ and does not demand the use of complex solvers.

2.2 Properties of RMKC

In this section, two properties of RMKC are presented, which highlight some important advantages of combining the margin with the variance of the clusters.

Suppose the composite kernel $\widetilde{\mathcal{K}}$ (3) is scaled by $\alpha > 0$, i.e. $\widetilde{\mathcal{K}}' = \alpha\widetilde{\mathcal{K}}$. Then the corresponding transformation becomes $\widetilde{\phi}' = \sqrt{\alpha}\widetilde{\phi}$. Moreover, as $\widetilde{\mathcal{K}}$ is a linear combination of basis kernels, its scaling can be equivalently posed as a scaling on its parameters, i.e. $\boldsymbol{\theta}' = \alpha\boldsymbol{\theta}$.

Proposition 1. (**Scale Invariance**) *If a kernel* $\widetilde{\mathcal{K}}$ *of the form defined in* (3) *is scaled by a scalar* $\alpha > 0$*, then* $\mathcal{J}(\alpha\boldsymbol{\theta}, \mathbf{y}) = \mathcal{J}(\boldsymbol{\theta}, \mathbf{y})$.

Proof. From (7) it is evident that $\mathcal{E}(\alpha\boldsymbol{\theta}, \mathbf{y}) = \alpha\mathcal{E}(\boldsymbol{\theta}, \mathbf{y})$, hence:

$$\mathcal{J}(\alpha\boldsymbol{\theta}, \mathbf{y}) = \min_{\mathbf{w},b,\boldsymbol{\xi}} \frac{1}{2}\alpha\mathcal{E}(\boldsymbol{\theta}, \mathbf{y})\|\mathbf{w}\|^2 + C\sum_{i=1}^{N}\xi_i,$$
$$s.t.\ \ y_i\left(\mathbf{w}^\top\left(\sqrt{\alpha}\widetilde{\phi}(\mathbf{x}_i)\right) + b\right) \geq 1 - \xi_i,\ \xi_i \geq 0.$$

Setting $\mathbf{w} = \mathbf{w}'/\sqrt{\alpha}$ and substituting in the above equation completes the proof, as (5) is recovered. □

Our quest for an objective that satisfies Proposition 1 was inspired by Gai et al. [7], where it was illustrated that relying solely on the margin is not sufficient to perform kernel learning in the supervised case. Analogously, if $\mathcal{J}(\boldsymbol{\theta}, \mathbf{y})$ in (4) is replaced with the more conventional margin-based objective:

$$\mathcal{J}'(\boldsymbol{\theta}, \mathbf{y}) = \min_{\mathbf{w},b,\boldsymbol{\xi}} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N}\xi_i,\ s.t.\ \ y_i\left(\mathbf{w}^\top\widetilde{\phi}(\mathbf{x}_i) + b\right) \geq 1 - \xi_i,\ \xi_i \geq 0, \tag{9}$$

it can be shown that an arbitrarily small $\mathcal{J}'(\boldsymbol{\theta}, \mathbf{y})$ value can be achieved by scaling the composite kernel, thus constituting the margin criterion unsuitable for evaluating the true quality of the kernel while learning $\{\boldsymbol{\theta}, \mathbf{y}\}$. Note that in the linear combination case (3), where scaling the composite kernel is equivalent to scaling its parameters, the scaling issue can be handled through the p-norm constraint on $\boldsymbol{\theta}$. However, this is not possible for nonlinear mixtures of basis kernels. On the contrary, our ratio-based objective (5) is scale invariant for arbitrary forms of composite kernels (the proof is analogous to Proposition 1) and also allows for norm invariance.

Proposition 2. (**Norm Invariance**) *Consider a kernel $\widetilde{\mathcal{K}}$ of the form defined in* (3) *as well as a) the optimization problem described by* (4) *without the p-norm constraint on $\boldsymbol{\theta}$ (p1) and b) the same problem* (4)*, but with the slightly more general p-norm constraint $\|\boldsymbol{\theta}\|_p^p = c$, $c > 0$, in place of $\|\boldsymbol{\theta}\|_p^p = 1$ (p2). If $\{\boldsymbol{\theta}_a^*, \mathbf{y}_a^*\}$ is a global optimal solution of p1 then $\left\{\frac{c^{1/p}}{\|\boldsymbol{\theta}_a^*\|_p}\boldsymbol{\theta}_a^*, \mathbf{y}_a^*\right\}$ is a global optimal solution of p2. Also, if $\{\boldsymbol{\theta}_b^*, \mathbf{y}_b^*\}$ is a global optimal solution of p2 then $\{\boldsymbol{\theta}_b^*, \mathbf{y}_b^*\}$ is a global optimal solution of p1.*

Proof. From the scale invariance property and since $\{\boldsymbol{\theta}_a^*, \mathbf{y}_a^*\}$ is a global optimum of p1 we get $\mathcal{J}\left(\frac{c^{1/p}}{\|\boldsymbol{\theta}_a^*\|_p}\boldsymbol{\theta}_a^*, \mathbf{y}_a^*\right) = \mathcal{J}(\boldsymbol{\theta}_a^*, \mathbf{y}_a^*) \leq \mathcal{J}(\boldsymbol{\theta}, \mathbf{y})$ for any $\{\boldsymbol{\theta}, \mathbf{y}\}$ satisfying the constraints of p1. Note that the admissible $\boldsymbol{\theta}$ values for problem p2 are a subset of those allowed in p1, hence the above inequality also holds for every $\{\boldsymbol{\theta}, \mathbf{y}\}$ adhering to the constraints of p2 (the constraints for $\mathbf{y}$ are identical in p1 and p2)). Together with the fact that $\left\|\frac{c^{1/p}}{\|\boldsymbol{\theta}_a^*\|_p}\boldsymbol{\theta}_a^*\right\|_p^p = c$ the first part of the proof is completed.
For any $\{\boldsymbol{\theta}, \mathbf{y}\}$ complying to the constraints of p1 it holds that $\left\{\frac{c^{1/p}}{\|\boldsymbol{\theta}\|_p}\boldsymbol{\theta}, \mathbf{y}\right\}$ is admissible for p2, since $\left\|\frac{c^{1/p}}{\|\boldsymbol{\theta}\|_p}\boldsymbol{\theta}\right\|_p^p = c$. The scale invariance property and the global optimality of $\{\boldsymbol{\theta}_b^*, \mathbf{y}_b^*\}$ w.r.t. p2 yields $\mathcal{J}(\boldsymbol{\theta}_b^*, \mathbf{y}_b^*) \leq \mathcal{J}\left(\frac{c^{1/p}}{\|\boldsymbol{\theta}\|_p}\boldsymbol{\theta}, \mathbf{y}\right) = \mathcal{J}(\boldsymbol{\theta}, \mathbf{y})$, thus completing the second part of the proof. □

Proposition 2 implies that the global optimal solution of the proposed formulation (4) is insensitive to the selected type of p-norm constraint, up to a scaling on the composite kernel parameters. The norm constraint can be even dropped from (4) without affecting its optimal solution. Of course, a solver that locates local optima of the ratio-based objective may produce different solutions when different p-norms are employed for the same problem, but at least the overall best will be the same, making the choice of the p-norm less crucial.

2.3 Optimizing the RMKC Objective

An iterative algorithm that alternates between updating the cluster labels $\mathbf{y}$ and reestimating the composite kernel coefficients $\boldsymbol{\theta}$, starting from some initial $\{\boldsymbol{\theta}, \mathbf{y}\}$ value, is presented and its main steps are summarized in Algorithms 1-2.

Evaluating the Objective Function. To compute the value of the objective function $\mathcal{J}(\boldsymbol{\theta}, \mathbf{y})$ for some fixed $\{\boldsymbol{\theta}, \mathbf{y}\}$, we need to solve the convex SVM-like optimization problem in (5). This can be facilitated by turning to its dual, which can be obtained by incorporating the constraints into the primal via Lagrange multipliers and setting the derivatives w.r.t. $\mathbf{w}$, b, and $\boldsymbol{\xi}$ to zero. After some manipulation the following dual emerges:

$$\max_{\boldsymbol{\alpha}} \sum_{i=1}^{N} \alpha_i - \frac{1}{2\mathcal{E}(\boldsymbol{\theta}, \mathbf{y})} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \widetilde{K}_{ij}, \ s.t. \ 0 \leq \alpha_i \leq C, \ \sum_{i=1}^{N} \alpha_i y_i = 0. \tag{10}$$

Since the cluster variance $\mathcal{E}(\boldsymbol{\theta}, \mathbf{y})$ is a constant for given $\{\boldsymbol{\theta}, \mathbf{y}\}$, it can be included in the kernel matrix and, thus, (10) actually coincides with the dual of the standard SVM, with $\frac{1}{\mathcal{E}(\boldsymbol{\theta}, \mathbf{y})} \widetilde{K}$ as the kernel matrix. Hence, the optimal solution for (10), denoted by $\boldsymbol{\alpha}^*$, can be located using any of the existing SVM solvers (the optimal values for $\mathbf{w}$, b, and $\boldsymbol{\xi}$ in (5) are calculated based on the solution of the dual). Moreover, due to strong duality, the value of $\mathcal{J}(\boldsymbol{\theta}, \mathbf{y})$ can be directly acquired from the dual:

$$\mathcal{J}(\boldsymbol{\theta}, \mathbf{y}) = \sum_{i=1}^{N} \alpha_i^* - \frac{1}{2\mathcal{E}(\boldsymbol{\theta}, \mathbf{y})} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i^* \alpha_j^* y_i y_j \widetilde{K}_{ij}. \tag{11}$$

Updating the Kernel Parameters. Changing the composite kernel coefficients so that the ratio-based objective $\mathcal{J}(\boldsymbol{\theta}, \mathbf{y})$ is reduced, while keeping the cluster labels $\mathbf{y}$ fixed, can be effectively performed by means of gradient descent. Due to strong duality between (5) and (10) (Section 2.3), we can exploit (11) to compute the gradient of $\mathcal{J}(\boldsymbol{\theta}, \mathbf{y})$ w.r.t. $\boldsymbol{\theta}$.

Proof for the differentiability of $\mathcal{J}(\boldsymbol{\theta}, \mathbf{y})$ comes from Danskin's theorem [4], similar to [14, 19]. To apply this theorem to our problem, two conditions must be satisfied. First, the optimal solution $\boldsymbol{\alpha}^*$ of (10) must be unique. This can be ensured by demanding the composite kernel matrix $\widetilde{K}$ to be strictly positive definite for every admissible $\boldsymbol{\theta}$. Second, the objective function optimized in the dual (10) must be continuously differentiable w.r.t. $\boldsymbol{\theta}$, which can be ensured by demanding $\widetilde{K}$ to be continuously differentiable w.r.t. $\boldsymbol{\theta}$. As $\widetilde{K}$ is a linear mixture of basis kernel matrices $K^{(v)}$, both requirements are fulfilled as long as every $K^{(v)}$ is strictly positive definite. The theorem also states that $\mathcal{J}(\boldsymbol{\theta}, \mathbf{y})$ can be differentiated as if $\boldsymbol{\alpha}^*$ does not depend on $\boldsymbol{\theta}$. Therefore, the derivatives can be obtained from (11) as:

$$\begin{aligned} \frac{\partial \mathcal{J}(\boldsymbol{\theta}, \mathbf{y})}{\partial \theta_v} =& \frac{1}{2\mathcal{E}(\boldsymbol{\theta}, \mathbf{y})^2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i^* \alpha_j^* y_i y_j \widetilde{K}_{ij} \frac{\partial \mathcal{E}(\boldsymbol{\theta}, \mathbf{y})}{\partial \theta_v} \\ & - \frac{1}{2\mathcal{E}(\boldsymbol{\theta}, \mathbf{y})} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i^* \alpha_j^* y_i y_j \frac{\partial \widetilde{K}_{ij}}{\partial \theta_v}, \end{aligned} \tag{12}$$

where $\frac{\partial \widetilde{K}_{ij}}{\partial \theta_v} = K_{ij}^{(v)}$ and $\frac{\partial \mathcal{E}(\boldsymbol{\theta},\mathbf{y})}{\partial \theta_v}$ follows directly from (7). Note that in order to calculate the derivatives, we must first obtain $\boldsymbol{\alpha}^*$ by solving (10) for the current $\{\boldsymbol{\theta}, \mathbf{y}\}$ values.

The procedure for updating $\boldsymbol{\theta}$ for given $\mathbf{y}$, begins by executing a standard gradient descent update on $\boldsymbol{\theta}$, using (12). Afterwards, $\boldsymbol{\theta}$ is projected back to its feasible set, so that the positivity and p-norm constraints (4) are enforced. In this work, we consider the values $p = 1, 2$ and execute the projections as shown in [6, 15]. Note that the gradient descent step size, η, is adjusted according to the Armijo rule, which may require additional optimizations of the dual.

Updating the Cluster Labels. Finding a new set of cluster assignments $\mathbf{y}'$ that will further decrease $\mathcal{J}(\boldsymbol{\theta}, \mathbf{y})$ (keeping the kernel parameters $\boldsymbol{\theta}$ fixed) is not straightforward, since the underlying optimization is a non-convex integer problem. Some single kernel MMC approaches relax $\mathbf{y}$ on the continuous domain to ease the optimization (e.g. [18, 21]), however, in the end the relaxed solution should be mapped back to the discrete space. Here, on the contrary, our aim is to work directly on the discrete cluster labels without any relaxations.

We have developed a practical search framework, where an improved cluster labeling $\mathbf{y}'$ is obtained by moving instances between the two clusters. One possible direction would be to change the cluster label of a single instance only and then proceed with reestimating $\boldsymbol{\theta}$. However, we have empirically found that such a minor modification on $\mathbf{y}$ results in premature convergence as the algorithm overcommits to the initial assignments. A better strategy is to change the labels of multiple instances before reestimating $\boldsymbol{\theta}$. The strategy we follow is motivated by several graph partitioning heuristics that have been applied to clustering, prominently the Kernighan-Lin algorithm [10]: an initial split of the graph is revamped by exchanging several nodes (specified in an incremental fashion) between partitions and selecting the best subset of these nodes. Based on this idea, we build a sequence of L candidate cluster label vectors, $\mathbf{y}^{(1)}, \ldots, \mathbf{y}^{(L)}$, ($L$ is user-defined) and select the one generating the greatest improvement on $\mathcal{J}(\boldsymbol{\theta}, \mathbf{y})$ in order to update $\mathbf{y}$. These L candidate label vectors are constructed incrementally (one after the other), such that compared to the previous candidate label vector, the next contains one more instance whose label has been changed (i.e. they differ in one element). Given $\mathbf{y}^{(l)}$, the $(l+1)$-th instance to change clusters is selected to be the one that is expected to produce the smallest objective value when added to the current l changes, thus constructing $\mathbf{y}^{(l+1)}$.

A meaningful approach for picking the $(l+1)$-th instance is to rank the contending instances based on the confidence about their labeling according to the current (after l cluster moves) separating hyperplane and select the one with the smallest $y_i(\mathbf{w}^\top \widetilde{\phi}(\mathbf{x}_i) + b)$ value. This way misclassified instances (if any exist) have a higher priority to change clusters, since $y_i(\mathbf{w}^\top \widetilde{\phi}(\mathbf{x}_i) + b) < 0$, followed by those falling inside the margin (if any exist), since $0 \leq y_i(\mathbf{w}^\top \widetilde{\phi}(\mathbf{x}_i) + b) < 1$, and finally those away from the margin, since $y_i(\mathbf{w}^\top \widetilde{\phi}(\mathbf{x}_i) + b) \geq 1$.

More specifically, let $\mathbf{y}^{(0)}$ to be the vector of the cluster labels before commencing the update process. Assume that $\mathbf{y}^{(l)}$ has already been generated, thus

at this point l instances have already changed clusters w.r.t $\mathbf{y}^{(0)}$. As mentioned, the $(l+1)$-th instance is selected to be the one we are the less confident about its labeling according to the separating hyperplane. However, when the labels change so does the hyperplane. Therefore, we must solve the dual (10) for the current assignments $\mathbf{y}^{(l)}$ to obtain the corresponding optimal hyperplane parameters ${\mathbf{w}^{(l)}}^*$ and ${b^{(l)}}^*$. Then, the index of the $(l+1)$-th instance is given by:

$$i^* = \underset{i:y_i^{(l)}=y_i^{(0)}}{\operatorname{argmin}} \; y_i^{(l)} \left({\mathbf{w}^{(l)}}^{*\top} \widetilde{\phi}(\mathbf{x}_i) + {b^{(l)}}^* \right), \tag{13}$$

and the $(l+1)$-th candidate label vector is defined as:

$$y_i^{(l+1)} = \begin{cases} y_i^{(l)}, & i \neq i^* \\ -y_i^{(l)}, & i = i^* \end{cases}. \tag{14}$$

From (13), it is obvious, that an instance $\mathbf{x}_i$ whose label has already changed is not considered again as a contender, since $y_i^{(l)} \neq y_i^{(0)}$, and the selected one flips its label (14). Moreover, observe that the label changes of all previous steps are retained when constructing $\mathbf{y}^{(l+1)}$, leading to an incremental reassignment of the instances. The above is repeated for $l = 0, 1, \ldots, L-1$.

The returned cluster assignments that are used to update $\mathbf{y}$ correspond to the cluster label vector $\mathbf{y}^{(l^*)}$ attaining the smallest objective value (i.e. $\mathbf{y}' = \mathbf{y}^{(l^*)}$):

$$l^* = \underset{0 \leq l \leq L}{\operatorname{argmin}} \mathcal{J}(\boldsymbol{\theta}, \mathbf{y}^{(l)}). \tag{15}$$

Note that if none of the candidate label vectors $\mathbf{y}^{(l)}$ reduces the objective, then $l^* = 0$ from (15), and no label change is accepted. This ensures that the ratio-based objective never increases after updating $\mathbf{y}$.

The procedure for modifying $\mathbf{y}$, as described up to this point, selects L instances belonging to either of the two clusters and flips their label to construct the candidate label vectors. Some trial experiments indicated that a better approach is to restrict all L instances that change clusters to originate from the same (i.e. a single) cluster. For this reason, our final procedure is divided into two phases. In the first phase the candidate vectors are formed by moving L instances from the cluster associated with the $+1$ label to the cluster associated with the -1 label, while in the second phase the opposite movement direction is considered. The two phases are independent from each other, both starting from $\mathbf{y}^{(0)}$. Hence, one phase does not take into account the cluster changes of the other. At the end, the best of the $2L$ candidate vectors is selected to update the cluster labels. To implement the above idea, in (13) we must, additionally to $y_i^{(l)} = y_i^{(0)}$, require that $y_i^{(l)} = +1$ ($y_i^{(l)} = -1$) for the first (second) phase contending instances. Our complete, two phase, framework is shown in Algorithm 2.

An issue we have yet to touch on is how to impose the cluster balance constraint (4). Fortunately, this is rather straightforward under our framework, since we can define an upper bound on the number L of candidate label vectors in each phase and, therefore, on the number of instances allowed to change

Algorithm 1. RMKC

Input: Basis kernel matrices $\{K^{(v)}\}_{v=1}^{V}$, Initial composite kernel coefficients $\boldsymbol{\theta}^{(0)}$ and cluster assignments $\mathbf{y}^{(0)}$

Output: Final kernel coefficients $\boldsymbol{\theta}$ and cluster assignments $\mathbf{y}$

Set $t = 0$
Set parameters L, ℓ and C
Set $\widetilde{K}^{(0)} = \sum_{v=1}^{V} \theta_v^{(0)} K^{(v)}$
repeat
  Solve the dual (10) for $\widetilde{K}^{(t)}$ (i.e. $\boldsymbol{\theta}^{(t)}$) and $\mathbf{y}^{(t)}$ to obtain $\boldsymbol{\alpha}^{(t)*}$
  for $v = 1$ to V **do** // Update $\boldsymbol{\theta}$.
    $\theta_v^{(t+1)} = \theta_v^{(t)} - \eta^{(t)} \left. \frac{\partial \mathcal{J}(\boldsymbol{\theta}, \mathbf{y})}{\partial \theta_v} \right|_{\boldsymbol{\theta}=\boldsymbol{\theta}^{(t)}, \mathbf{y}=\mathbf{y}^{(t)}, \boldsymbol{\alpha}^*=\boldsymbol{\alpha}^{(t)*}}$
  end for
  Project $\boldsymbol{\theta}^{(t+1)}$ to satisfy the constraints in (4)
  $\widetilde{K}^{(t+1)} = \sum_{v=1}^{V} \theta_v^{(t+1)} K^{(v)}$
  $\mathbf{y}^{(t+1)} = \textsc{Cluster\_upd}(\widetilde{K}^{(t+1)}, \mathbf{y}^{(t)})$ // Update $\mathbf{y}$.
  $t = t + 1$
until converged
return $\boldsymbol{\theta} = \boldsymbol{\theta}^{(t)}, \mathbf{y} = \mathbf{y}^{(t)}$

clusters, to guarantee that the constraint is never violated. For the first phase $L \leq (\ell + \sum_{i=1}^{N} y_i^{(0)})/2$, while for the second $L \leq (\ell - \sum_{i=1}^{N} y_i^{(0)})/2$. Note that $\sum_{i=1}^{N} y_i^{(0)}$ describes the initial imbalance before moving any instances (which, of course, satisfies the constraint) and $\ell \geq 0$ the maximum admissible imbalance.

2.4 Discussion

This section examines some additional aspects of the proposed RMKC method, starting with the convergence of the iterative algorithm used to optimize (4). In each iteration, the gradient descent update on $\boldsymbol{\theta}$ reduces the ratio-based objective value. Moreover, the subsequent update on $\mathbf{y}$ selects a candidate cluster label vector that further decreases the objective. Hence, the overall process is guaranteed to monotonically converge. The final solution, though, depends on the initial $\{\boldsymbol{\theta}, \mathbf{y}\}$ values, thus a local, and not the global, minimum of $\mathcal{J}(\boldsymbol{\theta}, \mathbf{y})$ is located. The solution also depends on the user-specified constants C, ℓ and L, as well as, on the selected p-norm for the composite kernel coefficients constraint.

An important advantage of RMKC is that it can be readily extended to learning general forms of parametric composite kernels $\widetilde{\mathcal{K}}$, such as a nonlinear mixture of basis kernels, without being restricted to just the linear combination case (3). The formulation itself remains unchanged (e.g. (4), (5), (6), (10), (11)) and the iterative algorithm is applicable out of the box, if the gradient of the ratio-based objective can be computed. This is possible when the composite kernel matrix is strictly positive definite and continuously differentiable w.r.t. its parameters $\boldsymbol{\theta}$ (see Section 2.3). Of course, $\frac{\partial \widetilde{K}_{ij}}{\partial \theta_v}$ and $\frac{\partial \mathcal{E}(\boldsymbol{\theta}, \mathbf{y})}{\partial \theta_v}$ in (12) depend on the specific form of the composite kernel. Moreover, the scale invariance of our objective

Algorithm 2. RMKC - cluster update

Input: Current composite kernel matrix $\widetilde{K}$ and cluster assignments $\mathbf{y}$
Output: Updated cluster assignments $\mathbf{y}'$

function CLUSTER_UPD($\widetilde{K}$, $\mathbf{y}$)
  // First phase.
  Set $\mathbf{y}^{(0)} = \mathbf{y}$
  for $l = 0$ to $L - 1$ **do**
    Solve the dual (10) for $\widetilde{K}$ and $\mathbf{y}^{(l)}$ to obtain $\mathbf{w}^{(l)*}$ and $b^{(l)*}$
    Calculate $\mathbf{y}^{(l+1)}$ (14) with the added constraint $y_i^{(l)} = +1$ in (13)
  end for
  // Second phase. This phase ignores the cluster moves of the first.
  Set $\mathbf{y}^{(L+1)} = \mathbf{y}$
  for $l = L + 1$ to $2L$ **do**
    Solve the dual (10) for $\widetilde{K}$ and $\mathbf{y}^{(l)}$ to obtain $\mathbf{w}^{(l)*}$ and $b^{(l)*}$
    Calculate $\mathbf{y}^{(l+1)}$ (14) with the added constraint $y_i^{(l)} = -1$ in (13)
  end for
  $l^* = \text{argmin}_{0 \leq l \leq 2L+1} \mathcal{J}(\boldsymbol{\theta}, \mathbf{y}^{(l)})$
  return $\mathbf{y}' = \mathbf{y}^{(l^*)}$
end function

(i.e. scaling $\widetilde{\mathcal{K}}$ by a scalar $\alpha > 0$) also holds in the general case (the proof is analogous to that in Proposition 1), but the same is not true for the norm invariance. Note that scaling $\widetilde{\mathcal{K}}$ is no more equivalent to scaling the parameters $\boldsymbol{\theta}$. The ability to accommodate general kernel forms broadness the applicability of RMKC and constitutes an advantage over existing MKL approaches that are usually limited to a particular type of composite kernel.

3 Empirical Evaluation

To investigate the potential of combining the margin with the variance in the clustering objective and perform kernel learning, the presented RMKC framework is compared to: a) kernel k-means, which serves as our baseline method, b) iterSVR [24], an iterative margin-based MMC approach that follows formulation (1), and c) two iterative variance-based MKL approaches that optimize (6), namely multi-view kernel k-means (MVKKM) and multi-view spectral clustering (MVSpec) [17]. The evaluation is made on various diverse datasets from the UCI repository[2] (Ionosphere, Letter, Satellite, Multiple Features and Optdigits), as well as on the COIL-20 image library of objects [13] and a subset of the Corel image collection[3]. Apart from Ionosphere, all other datasets contain instances of more than two categories. For this reason, we conduct experiments using pairs of the included categories. For Letter and Satellite we simply focus on the first two classes, i.e. A-B and C1(red soil)-C2(cotton crop), respectively, as in [24]. For

[2] http://archive.ics.uci.edu/ml
[3] http://www.cs.virginia.edu/~xj3a/research/CBIR/Download.htm

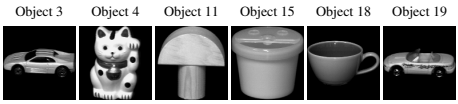

Fig. 1. The COIL-20 objects considered in the experiments

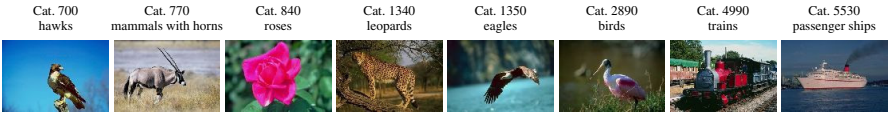

Fig. 2. Indicative images of the Corel categories considered in the experiments

the two databases of handwritten digits (i.e. Multiple Features and Optdigits) we try several pairs of the contained numerals (0-9), while for the two image collections we consider pairs of the classes depicted in Figures 1-2. The tested pairs are shown in Tables 3-4. Since ground truth information is available for every dataset, we employ the clustering accuracy metric to measure performance[4].

Multiple Features and Corel are multi-view datasets, hence, for the same instance multiple sets of attributes are available. Each attribute set naturally defines a basis kernel and the linear kernel is employed here to represent each view. For the other, single view, datasets, we follow [18,20] and construct 10 basis RBF kernels, where the kernel width σ varies from 10% to 100% of the range of distance between any two instances. Kernels are multiplicatively normalized [12].

Throughout the experiments, our algorithm is configured as follows: we fix the number of candidate label vectors in each phase to $L = 30$, the cluster imbalance parameter to $\ell = 0.5N$ (for the Corel images only, $\ell = 0.2N$) and conduct a grid search on the set $\{10^{-2}, 10^{-1}, \ldots, 10^{2}\}$ to locate the best performing value for the C regularizer in each dataset. The basis kernels are linearly combined (3) and their coefficients are uniformly initialized, i.e. $\theta_v = \frac{1}{V^{1/p}}$. To initialize the cluster assignments $\mathbf{y}$, we extract several pairs of instances (usually $0.25N$ pairs) using a k-means++-like procedure [1], where the first instance is chosen randomly and the second is picked with a probability that is proportional to its distance from the first. For each such pair, the remaining $N - 2$ instances are assigned to the closest of the two instances in the pair, thus producing a partitioning of the data. The partitioning $\mathbf{y}$ with the minimum $\mathcal{J}(\boldsymbol{\theta}, \mathbf{y})$ value is used to initialize a run of RMKC. Since the procedure for choosing the initial $\mathbf{y}$ is nondeterministic, the RMKC performance is averaged over 30 runs for each tried set of parameters (L, ℓ, C, p-norm). Finally, the LIBSVM toolbox [2] is utilized for solving (10).

3.1 Norm Invariance in Practice

In Proposition 2, it was proved that the global optimal solution of our formulation (4) is invariant to the p-norm applied on the composite kernel coefficients $\boldsymbol{\theta}$,

[4] To evaluate performance, we make the typical assumption that clusters correspond to classes and set their number equal to the number of classes (e.g. [18, 20, 22, 25]).

Table 1. RMKC clustering accuracy (%) (averaged over all pairs of categories considered in each dataset) for different p-norm constraints

Dataset	No-norm	1-norm	2-norm
Ionosphere	71.51 ± 0.00	71.51 ± 0.00	71.51 ± 0.00
Letter	94.47 ± 0.00	94.47 ± 0.00	94.47 ± 0.00
Satellite	96.17 ± 0.50	96.19 ± 0.52	96.16 ± 0.51
COIL-20	98.75 ± 2.60	98.61 ± 2.65	98.43 ± 2.73
Corel	94.55 ± 1.62	94.64 ± 1.58	94.69 ± 1.62
Multiple Features	99.58 ± 0.22	99.53 ± 0.37	99.59 ± 0.23
Optdigits	97.77 ± 2.45	97.65 ± 2.71	97.75 ± 2.50

Table 2. Clustering accuracy (%) of the compared methods on three popular UCI datasets

Dataset	RMKC (1-norm)	MVKKM	MVSpec	Kernel k-means	IterSVR (best)	IterSVR (average)
Ionosphere	71.51 ± 0.00	71.23	70.66	73.22 ± 2.90	$\mathbf{74.83 \pm 1.65}$	71.83 ± 1.99
Letter (A-B)	94.47 ± 0.00	93.50	88.68	93.63 ± 0.00	$\mathbf{94.51 \pm 1.70}$	92.29 ± 1.97
Satellite (C1-C2)	96.19 ± 0.52	94.19	96.24	94.15 ± 0.03	$\mathbf{96.42 \pm 0.00}$	91.53 ± 5.58

Table 3. Clustering accuracy (%) of the compared methods on image clustering

Dataset	RMKC (1-norm)	MVKKM	MVSpec	Kernel k-means	IterSVR (best)	IterSVR (average)
COIL-20						
3-19	$\mathbf{100.00 \pm 0.00}$	**100.00**	**100.00**	94.05 ± 10.27	$\mathbf{100.00 \pm 0.00}$	$\mathbf{100.00 \pm 0.00}$
4-11	$\mathbf{100.00 \pm 0.00}$	77.78	**100.00**	96.30 ± 10.41	98.47 ± 8.37	98.34 ± 8.34
15-18	$\mathbf{100.00 \pm 0.00}$	90.28	95.83	97.57 ± 3.74	99.72 ± 0.35	99.21 ± 0.21
15-19	$\mathbf{94.44 \pm 10.59}$	68.06	86.11	86.57 ± 14.84	93.43 ± 14.30	91.86 ± 14.52
Corel						
700-4990	$\mathbf{97.62 \pm 0.65}$	95.00	95.00	85.98 ± 9.58	96.43 ± 0.25	83.19 ± 1.85
700-5530	92.60 ± 1.42	**94.00**	**94.00**	85.50 ± 0.00	88.63 ± 6.40	68.03 ± 3.49
770-840	$\mathbf{97.55 \pm 0.91}$	94.50	90.00	90.47 ± 0.37	94.20 ± 3.04	87.85 ± 0.58
770-1350	$\mathbf{94.03 \pm 1.72}$	93.50	92.00	88.72 ± 0.96	92.67 ± 1.27	84.10 ± 1.89
1340-1350	$\mathbf{95.50 \pm 0.00}$	95.00	95.00	91.00 ± 0.00	92.50 ± 0.00	83.71 ± 0.00
2890-4990	$\mathbf{90.57 \pm 4.79}$	87.00	86.00	85.00 ± 0.00	90.00 ± 0.00	73.04 ± 5.68

if $\widetilde{\mathcal{K}}$ is a linear mixture of basis kernels (3). However, the RMKC method locates local optima of the ratio-based objective. Hence, it is of particular interest to explore how these local optima vary for different choices of p-norm constraints.

To demonstrate this, RMKC is executed (according to the above configuration) for $p = 1, 2$ and also for the case where no norm constraint is imposed on $\boldsymbol{\theta}$ and the results are illustrated in Table 1. It can be observed that the solutions obtained across the different norms are very similar, therefore, in practice, the uncovered local optima are not significantly influenced by the choice of p-norm, although this cannot be theoretically guaranteed. On the following, we shall focus on the 1-norm, when presenting the results of our approach.

Table 4. Clustering accuracy (%) of the compared methods on the task of handwritten digits recognition

Dataset	RMKC (1-norm)	MVKKM	MVSpec	Kernel k-means	IterSVR (best)	IterSVR (average)
Mult. Feat.						
1-7	99.62 ± 0.78	98.75	98.75	98.00 ± 0.00	$\mathbf{99.75 \pm 0.00}$	96.85 ± 0.00
2-7	$\mathbf{100.00 \pm 0.00}$	99.00	99.75	97.92 ± 0.24	99.75 ± 0.00	97.61 ± 1.73
2-3	$\mathbf{99.70 \pm 0.23}$	99.25	99.00	99.50 ± 0.00	99.50 ± 0.00	94.13 ± 7.16
3-8	99.28 ± 0.38	99.50	99.50	97.50 ± 0.00	$\mathbf{99.75 \pm 0.00}$	98.78 ± 0.04
5-6	$\mathbf{99.42 \pm 0.48}$	98.50	98.50	98.29 ± 0.09	98.75 ± 0.00	95.68 ± 2.37
6-8	$\mathbf{99.15 \pm 0.33}$	97.25	98.50	97.33 ± 0.16	99.00 ± 0.00	94.94 ± 6.47
Optdigits						
1-7	99.56 ± 1.41	**100.00**	**100.00**	89.38 ± 16.06	96.93 ± 9.83	94.26 ± 13.14
2-7	98.03 ± 1.31	96.35	92.42	95.03 ± 8.40	$\mathbf{99.32 \pm 0.16}$	98.88 ± 0.84
2-3	96.29 ± 5.44	90.56	88.89	89.92 ± 9.10	$\mathbf{96.50 \pm 0.82}$	95.59 ± 2.70
3-8	92.43 ± 8.00	94.12	93.28	92.56 ± 7.80	$\mathbf{96.20 \pm 0.16}$	95.01 ± 4.08
5-6	$\mathbf{99.72 \pm 0.00}$	99.45	99.45	99.57 ± 0.14	$\mathbf{99.72 \pm 0.00}$	99.33 ± 0.01
6-8	$\mathbf{99.89 \pm 0.14}$	99.15	98.87	99.32 ± 0.26	99.72 ± 0.00	99.45 ± 0.06

3.2 Comparative Results

We have conducted a comprehensive evaluation of RMKC, kernel k-means, iterSVR, MVKKM and MVSpec on all datasets. RMKC is set up as previously described. Kernel k-means is restarted 30 times, from randomly picked initial centers. For iterSVR we employ a similar setup to [24], i.e. the cluster imbalance parameter is fixed to $\ell = 0.03N$ for balanced and to $\ell = 0.3N$ for unbalanced datasets, while the initial cluster labels are obtained from the kernel k-means solution (iterSVR is, thus, repeated 30 times). For the C regularizer, the same grid search as for RMKC is implemented. Finally, the sparsity controlling parameter p for MVKKM and MVSpec is selected by a grid search on the values $\{1, 1.5, \ldots, 5\}$.

Performance is measured in terms of average clustering accuracy (and its deviation) over the 30 restarts (MVKKM and MVSpec are deterministically initialized [17], thus we have no restarts). Let us stress, that both kernel k-means and iterSVR are single kernel methods that do not implement kernel learning. For this reason, these algorithms are independently executed for each of the individual basis kernels in each data collection and the kernel attaining the highest accuracy is reported. Moreover, for iterSVR the average performance over all basis kernels is also shown. It is important to make clear that it is not possible to know a priori which is the best basis kernel for a given dataset.

In Table 2 we observe that iterSVR with the optimal basis kernel achieves the best accuracy, being closely matched by RMKC. Only for Ionosphere the difference is large, where, surprisingly, all three MKL approaches (RMKC, MVKKM and MVSpec) are even inferior to kernel k-means. However, this is a difficult dataset to cluster and all methods yield rather poor outcomes (accuracy does not exceed 75%).

Turning our attention to image clustering (Table 3), it is evident that our ratio-based objective constantly outperforms the other methods. For the

COIL-20 objects, whose images are taken from different angles in a neutral background, hence are easy to distinguish, our approach manages to find the correct clusters for 3/4 of subsets and iterSVR appears to be its closest competitor. Clustering the Corel images is a more difficult task, due to variations in the composition of the depicted scene within each class. Here the differences of RMKC to iterSVR are more distinct and its closest competitor is MVKKM, which clearly displays the benefits of combining information from multiple views under MKL.

For the task of handwritten digits recognition (Table 4) the best performance is equally shared between RMKC and iterSVR across the two datasets. Note that for Multiple Features, which, like Corel, is a multi-view dataset, RMKC is superior. MVKKM and MVSpec achieve the highest accuracy on a single case (Optdigits for the pair 1-7) and are superior to RMKC for only 3/12 of subsets.

Overall, the proposed RMKC algorithm obtains a higher clustering accuracy for the majority of the tested category pairs. The margin-based iterSVR approach seems to be close, or even better, for some cases, provided the optimal basis kernel is used (iterSVR(best)). However, in practice, the best kernel for a particular dataset is not a priori known. By looking at the Tables' last column, one can notice that iterSVR results degrade significantly if an inappropriate basis kernel is chosen. On the contrary, RMKC is able to automatically infer a meaningful kernel by combining the basis kernels.

4 Conclusions

We have proposed a novel MKL formulation that considers the ratio between the margin and the intra-cluster variance. Its objective is optimized by an iterative, gradient-based algorithm to get both the cluster assignments and the composite kernel parameters. Moreover, it is characterized by two important properties: it is invariant to scalings of the learned kernel and, when basis kernels are linearly mixed, is also invariant (on its global optimum) to the type of p-norm constraint on the composite kernel parameters. Our framework compares favorably to existing approaches that rely either on the margin or the intra-cluster variance.

Although multiple cluster problems can be tackled by iteratively solving a sequence of two-cluster problems, an interesting research direction would be to extend our formulation to directly handle multiple clusters, following the ideas in [22, 25, 26]. Moreover, evaluating different parametric forms for the composite kernel is in our plans.

References

1. Arthur, D., Vassilvitskii, S.: k-means++: the advantages of careful seeding. In: ACM-SIAM Symposium on Discrete Algorithms (SODA), pp. 1027–1035 (2007)
2. Chang, C.C., Lin, C.J.: LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology 2(3), 1–27 (2011), `http://www.csie.ntu.edu.tw/~cjlin/libsvm`

3. Cortes, C., Mohri, M., Rostamizadeh, A.: Learning non-linear combinations of kernels. In: Advances in Neural Information Processing Systems (NIPS), pp. 396–404 (2009)
4. Danskin, J.M.: The theory of max-min, with applications. SIAM Journal on Applied Mathematics 14(4), 641–664 (1966)
5. Dhillon, I.S., Guan, Y., Kulis, B.: Weighted graph cuts without eigenvectors a multilevel approach. IEEE Transactions on Pattern Analysis and Machine Intelligence 29(11), 1944–1957 (2007)
6. Duchi, J.C., Shalev-Shwartz, S., Singer, Y., Chandra, T.: Efficient projections onto the l_1-ball for learning in high dimensions. In: International Conference on Machine Learning (ICML), pp. 272–279 (2008)
7. Gai, K., Chen, G., Zhang, C.: Learning kernels with radiuses of minimum enclosing balls. In: Advances in Neural Information Processing Systems (NIPS), pp. 649–657 (2010)
8. Gönen, M., Alpaydin, E.: Localized multiple kernel learning. In: International Conference on Machine Learning (ICML), pp. 352–359 (2008)
9. Gönen, M., Alpaydin, E.: Multiple kernel learning algorithms. Journal of Machine Learning Research 12, 2211–2268 (2011)
10. Kernighan, B.W., Lin, S.: An efficient heuristic procedure for partitioning graphs. The Bell System Technical Journal 49(2), 291–308 (1970)
11. Kloft, M., Brefeld, U., Sonnenburg, S., Laskov, P., Müller, K.R., Zien, A.: Efficient and accurate lp-norm multiple kernel learning. In: Advances in Neural Information Processing Systems (NIPS), pp. 997–1005 (2009)
12. Kloft, M., Brefeld, U., Sonnenburg, S., Zien, A.: lp-norm multiple kernel learning. Journal of Machine Learning Research 12, 953–997 (2011)
13. Nene, S.A., Nayar, S.K., Murase, H.: Columbia Object Image Library (COIL-20). Tech. Rep. CUCS-005-96, Department of Computer Science, Columbia University (1996), `http://www.cs.columbia.edu/CAVE/software/softlib/coil-20.php`
14. Rakotomamonjy, A., Bach, F.R., Canu, S., Grandvalet, Y.: SimpleMKL. Journal of Machine Learning Research 9, 2491–2521 (2008)
15. Songsiri, J.: Projection onto an l_1-norm ball with application to identification of sparse autoregressive models. In: Asean Symposium on Automatic Control, ASAC (2011)
16. Sonnenburg, S., Rätsch, G., Schäfer, C., Schölkopf, B.: Large scale multiple kernel learning. Journal of Machine Learning Research 7, 1531–1565 (2006)
17. Tzortzis, G., Likas, A.: Kernel-based weighted multi-view clustering. In: International Conference on Data Mining (ICDM), pp. 675–684 (2012)
18. Valizadegan, H., Jin, R.: Generalized maximum margin clustering and unsupervised kernel learning. In: Advances in Neural Information Processing Systems (NIPS), pp. 1417–1424 (2006)
19. Varma, M., Babu, B.R.: More generality in efficient multiple kernel learning. In: International Conference on Machine Learning (ICML), pp. 1065–1072 (2009)
20. Wang, F., Zhao, B., Zhang, C.: Linear time maximum margin clustering. IEEE Transactions on Neural Networks 21(2), 319–332 (2010)
21. Xu, L., Neufeld, J., Larson, B., Schuurmans, D.: Maximum margin clustering. In: Advances in Neural Information Processing Systems (NIPS), pp. 1537–1544 (2004)
22. Xu, L., Schuurmans, D.: Unsupervised and semi-supervised multi-class support vector machines. In: AAAI Conference on Artificial Intelligence (AAAI). pp. 904–910 (2005)

23. Xu, Z., Jin, R., Yang, H., King, I., Lyu, M.R.: Simple and efficient multiple kernel learning by group lasso. In: International Conference on Machine Learning (ICML), pp. 1175–1182 (2010)
24. Zhang, K., Tsang, I.W., Kwok, J.T.: Maximum margin clustering made practical. In: International Conference on Machine Learning (ICML), pp. 1119–1126 (2007)
25. Zhao, B., Kwok, J.T., Zhang, C.: Multiple kernel clustering. In: SIAM International Conference on Data Mining (SDM), pp. 638–649 (2009)
26. Zien, A., Ong, C.S.: Multiclass multiple kernel learning. In: International Conference on Machine Learning (ICML), pp. 1191–1198 (2007)