Information Fusion 44 (2018) 46–56

Contents lists available at ScienceDirect

Information Fusion

journal homepage: www.elsevier.com/locate/inffus

Multi-View Kernel Spectral Clustering

Lynn Houthuys*, Rocco Langone, Johan A.K. Suykens

Department of Electrical Engineering ESAT-STADIUS, KU Leuven Kasteelpark Arenberg 10 B-3001 Leuven, Belgium

ARTICLE INFO

Keywords:
Multi-view learning
Clustering
Out-of-sample extension
Kernel CCA

ABSTRACT

In multi-view clustering, datasets are comprised of different representations of the data, or views. Although each view could individually be used, exploiting information from all views together could improve the cluster quality. In this paper a new model Multi-View Kernel Spectral Clustering (MVKSC) is proposed that performs clustering when two or more views are available. This model is formulated as a weighted kernel canonical correlation analysis in a primal-dual optimization setting typical of Least Squares Support Vector Machines (LS-SVM). The primal model includes, in particular, a coupling term, which enforces the clustering scores corresponding to the different views to align. Because of the out-of-sample extension, this model is easily applied to large-scale datasets. The performance of the proposed model is shown on synthetic and real-world datasets, as well as on some large-scale datasets. Experimental comparisons with a number of other methods show that using multiple views improves the clustering results and that the proposed method is competitive with other state-of-the-art algorithms in terms of clustering accuracy and runtime. Especially on the large-scale datasets the advantage of the proposed method is clearly shown, as it is able to handle larger datasets than the other state-of-the-art algorithms.

1. Introduction

In various application domains, data from different sources or *views* are available. Many real-world datasets have representations in the form of multiple views [1]. For example, web pages can be classified based on both the page content (text) and hyperlink information [2], for social networks one could use the user profile but also the friend links [3], images can be classified based on the colors as well as the texture [4], and so on. Although each of the views by itself might already be sufficient for a given learning task, additional views often provide complementary information which can lead to an improved performance [5]. For an extensive overview of recent multi-view learning methods we refer to the work of Zhao et al. [6].

The information from multiple views can be fused in different ways as well as in different stages of the training process. In early fusion techniques, the views are fused before the training process starts, e.g. by means of feature concatenation [7] or in a more complex way like the work done by e.g. Yu et al. [8] and Lin et al. [9]. In this way the information from all views is taken into account early on in the training process. In late fusion techniques the models are usually trained separately and a combination of the individual results is taken to determine the final result. This combination can be formed in many ways, like for example by taking a weighted average, e.g. as done by Bekker et al. [10] for classification, or selective voting, e.g. as done by Xie et al. [11] for clustering.

The *clustering* problem [12] refers to the task of finding a partition of a given dataset based on some similarity measure between the examples. While there are various clustering algorithms available (e.g. the work by Sharma et al. [13,14] and Elhamifar et al. [15]), Spectral Clustering methods are increasingly popular due to the well-defined mathematical framework and its strong performance on arbitrary shaped clusters [16]. Spectral clustering methods make use of the eigenvectors of a rescaled affinity matrix derived from the data (i.e. the Laplacian) to divide a dataset into natural groups, such that points within the same group are similar and points in different groups are dissimilar to each other [17–19]. *Kernel Spectral Clustering* (KSC) [20] is a well-known clustering technique that represents a spectral clustering formulation as a weighted kernel PCA problem, cast in the LS-SVM framework [21].

In this paper a new model is introduced, called *Multi-View Kernel Spectral Clustering* (MVKSC)[1], which is an extension to KSC that allows to deal with multiple data-sources. This is done by integrating two or more KSC models in the joint MVKSC approach and adding a coupling term which maximizes the correlation of the score variables. This coupling can be thought of as a combination of early and late fusion, where the information of all views is already exploited during the

* Corresponding author.
E-mail addresses: lynn.houthuys@esat.kuleuven.be (L. Houthuys), rocco.langone@esat.kuleuven.be (R. Langone), johan.suykens@esat.kuleuven.be (J.A.K. Suykens).

[1] The Matlab implementation of the MVKSC algorithm is available for downloading from http://www.esat.kuleuven.be/stadius/ADB/software.php.

https://doi.org/10.1016/j.inffus.2017.12.002
Received 20 April 2017; Received in revised form 20 September 2017; Accepted 16 December 2017
Available online 18 December 2017
1566-2535/ © 2017 Elsevier B.V. All rights reserved.

training phase while still allowing for some degree of freedom to model the data from the different views differently.

Furthermore, the proposed model is also closely related to *Kernel Canonical Correlation Analysis* (KCCA) [21], which is a method for determining nonlinear relations among several variables. Although the KCCA learning task is essentially different from clustering, the two formulations are similar.

Expanding spectral clustering techniques to multi-view learning has been done in the past, for example by Cai et al. [22], Kumar et al. [23], Xie et al. [11] and Xia et al. [24]. Although these methods have achieved good accuracy, they are usually computationally expensive and not suitable for large-scale data. Li et al. [25] designed a method to deal with large-scale data by forming a bipartite graph for each view and running spectral clustering on the fusion of all graphs.

Similar to KSC, MVKSC has a natural out-of-sample extension to deal with new test data. Due to this extension the method is able to deal with large-scale data by training on only a small randomly chosen subset. This approach was used for KSC on large-scale network data by Mall et al. [26], although the authors did not simply pick the subset at random but used an algorithm that preserves the overall community structure. There are more complex extensions to KSC to deal with large-scale data, for example the fixed-size approach done by Langone & Suykens [27], but we show here that even this simple approach achieves good performance.

This paper shows how the clustering performance achieved by KSC on one view can be improved by exploiting information from multiple different views. The paper further shows that the out-of-sample extension can be used to deal with large-scale data in a natural way and shows the performance of MVKSC on a real-world large-scale dataset.

We will denote matrices as bold uppercase letters and vectors as bold lowercase letters. The superscript $^{[v]}$ will denote the vth set of variables for KCCA or the vth view for the multi-view method. Whereas the superscript $^{(l)}$ will denote the lth binary clustering problem in case there are more than two clusters.

The rest of this paper is organized as follows: Section 2.1 and Section 2.2 give a summary of the KCCA and the KSC model respectively. Section 3 discusses the proposed model MVKSC. It shows the mathematical formulation, explains the cluster assignment for the training data as well as for the out-of-sample test data and describes the model selection process. Section 4 discusses the experiments done with MVKSC and compares it to other state-of-the-art methods, and to KSC on the separate views alone. Section 4 further discusses the obtained results. Section 5 shows the performance of MVKSC when handling large-scale data. Finally, in Section 6 some conclusions are drawn.

2. Background

This section introduces the concepts of Kernel Canonical Correlation Analysis (KCCA) and Kernel Spectral Clustering (KSC).

2.1. Kernel Canonical Correlation Analysis

Canonical Correlation Analysis (CCA) was originally studied by Hotelling [28] and is a statistical method for determining linear relations among several variables. A nonlinear extension of CCA was introduced by Lai and Fyfe [29], Bach and Jordan [30] and by Van Gestel et al. [31] as *kernel CCA* or KCCA. To determine nonlinear relations, the input space is mapped to a high-dimensional feature space where classical CCA is applied.

A formulation in the LS-SVM framework was proposed by Suykens et al. [21]. Given data $\mathscr{D}^{[1]} = \{\mathbf{x}_i^{[1]}\}_{i=1}^N \subset \mathbb{R}^{d^{[1]}}$ and $\mathscr{D}^{[2]} = \{\mathbf{x}_i^{[2]}\}_{i=1}^N \subset \mathbb{R}^{d^{[2]}}$, the primal model of KCCA is formulated as follows:

$$\max_{\substack{\mathbf{w}^{[1]},\mathbf{w}^{[2]},\\ \mathbf{e}^{[1]},\mathbf{e}^{[2]}}} -\frac{1}{2}\mathbf{w}^{[1]T}\mathbf{w}^{[1]} - \frac{1}{2}\mathbf{w}^{[2]T}\mathbf{w}^{[2]} - \gamma^{[1]}\frac{1}{2}\mathbf{e}^{[1]T}\mathbf{e}^{[1]} - \gamma^{[2]}\frac{1}{2}\mathbf{e}^{[2]T}\mathbf{e}^{[2]} + \rho\mathbf{e}^{[1]T}\mathbf{e}^{[2]}$$

$$\text{s.t.}\quad \mathbf{e}^{[1]} = (\boldsymbol{\Phi}^{[1]} - \mathbf{1}_N\hat{\boldsymbol{\mu}}^{[1]T})\mathbf{w}^{[1]},$$
$$\mathbf{e}^{[2]} = (\boldsymbol{\Phi}^{[2]} - \mathbf{1}_N\hat{\boldsymbol{\mu}}^{[2]T})\mathbf{w}^{[2]} \tag{1}$$

where $\mathbf{e}^{[1]} \in \mathbb{R}^N$ and $\mathbf{e}^{[2]} \in \mathbb{R}^N$ are the score variables indicating the nonlinear relations. $\boldsymbol{\Phi}^{[1]} \in \mathbb{R}^{N\times d_h^{[1]}}$ and $\boldsymbol{\Phi}^{[2]} \in \mathbb{R}^{N\times d_h^{[2]}}$ are feature matrices with $\boldsymbol{\Phi}^{[1]} = [\varphi^{[1]}(\mathbf{x}_1^{[1]})^T;\cdots;\varphi^{[1]}(\mathbf{x}_N^{[1]})^T]$ and $\boldsymbol{\Phi}^{[2]} = [\varphi^{[2]}(\mathbf{x}_1^{[2]})^T;\cdots;\varphi^{[2]}(\mathbf{x}_N^{[2]})^T]$ where $\varphi^{[1]}$: $\mathbb{R}^{d^{[1]}} \to \mathbb{R}^{d_h^{[1]}}$ and $\varphi^{[2]}$: $\mathbb{R}^{d^{[2]}} \to \mathbb{R}^{d_h^{[2]}}$ are the mappings to high-dimensional feature spaces. $\hat{\mu}^{[1]} = (1/N)\sum_{i=1}^N \varphi^{[1]}(\mathbf{x}_i^{[1]}) = (1/N)\boldsymbol{\Phi}^{[1]T}\mathbf{1}_N$ and $\hat{\boldsymbol{\mu}}^{[2]} = (1/N)\sum_{i=1}^N \varphi^{[2]}(\mathbf{x}_i^{[2]}) = (1/N)\boldsymbol{\Phi}^{[2]T}\mathbf{1}_N$ are used to center the data and $\gamma^{[1]} \in \mathbb{R}^+$ and $\gamma^{[2]} \in \mathbb{R}^+$ are regularization parameters.

The dual problem related to this primal formulation is:

$$\begin{bmatrix} \mathbf{0}_N & \boldsymbol{\Omega}_c^{[2]} \\ \boldsymbol{\Omega}_c^{[1]} & \mathbf{0}_N \end{bmatrix}\begin{bmatrix} \boldsymbol{\alpha}^{[1]} \\ \boldsymbol{\alpha}^{[2]} \end{bmatrix} = \frac{1}{\rho}\begin{bmatrix} \gamma^{[1]}\boldsymbol{\Omega}_c^{[1]} + I & \mathbf{0}_N \\ \mathbf{0}_N & \gamma^{[2]}\boldsymbol{\Omega}_c^{[2]} + I \end{bmatrix}\begin{bmatrix} \boldsymbol{\alpha}^{[1]} \\ \boldsymbol{\alpha}^{[2]} \end{bmatrix} \tag{2}$$

where $\boldsymbol{\Omega}_c^{[1]} = (\boldsymbol{\Phi}^{[1]} - \mathbf{1}_N\hat{\boldsymbol{\mu}}^{[1]T})(\boldsymbol{\Phi}^{[1]} - \mathbf{1}_N\hat{\boldsymbol{\mu}}^{[1]T})^T$ and $\boldsymbol{\Omega}_c^{[2]} = (\boldsymbol{\Phi}^{[2]} - \mathbf{1}_N\hat{\boldsymbol{\mu}}^{[2]T})(\boldsymbol{\Phi}^{[2]} - \mathbf{1}_N\hat{\boldsymbol{\mu}}^{[2]T})^T$ are the centered kernel matrices and where

$$\Omega_{c_{kl}}^{[1]} = (\varphi^{[1]}(\mathbf{x}_k^{[1]}) - \hat{\boldsymbol{\mu}}^{[1]})^T(\varphi^{[1]}(\mathbf{x}_l^{[1]}) - \hat{\boldsymbol{\mu}}^{[1]})$$
$$\Omega_{c_{kl}}^{[2]} = (\varphi^{[2]}(\mathbf{x}_k^{[2]}) - \hat{\boldsymbol{\mu}}^{[2]})^T(\varphi^{[2]}(\mathbf{x}_l^{[2]}) - \hat{\boldsymbol{\mu}}^{[2]}) \tag{3}$$

are the elements of these centered kernel matrices for $k, l = 1, \ldots, N$. In practice they can be computed by $\boldsymbol{\Omega}_c^{[1]} = \mathbf{M}_c\boldsymbol{\Omega}^{[1]}\mathbf{M}_c$ and $\boldsymbol{\Omega}_c^{[2]} = \mathbf{M}_c\boldsymbol{\Omega}^{[2]}\mathbf{M}_c$ where $\boldsymbol{\Omega}^{[1]}$ and $\boldsymbol{\Omega}^{[2]}$ are the kernel matrices with $\boldsymbol{\Omega}^{[1]} = \boldsymbol{\Phi}^{[1]}\boldsymbol{\Phi}^{[1]T}$ and $\boldsymbol{\Omega}^{[2]} = \boldsymbol{\Phi}^{[2]}\boldsymbol{\Phi}^{[2]T}$ and where $\Omega_{ij}^{[1]} = K^{[1]}(\mathbf{x}_i^{[1]}, \mathbf{x}_j^{[1]}) = \varphi^{[1]}(\mathbf{x}_i^{[1]})^T\varphi^{[1]}(\mathbf{x}_j^{[1]})$ and $\Omega_{ij}^{[2]} = K^{[2]}(\mathbf{x}_i^{[2]}, \mathbf{x}_j^{[2]}) = \varphi^{[2]}(\mathbf{x}_i^{[2]})^T\varphi^{[2]}(\mathbf{x}_j^{[2]})$ and $\mathbf{M}_c = \mathbf{I}_N - (1/N)\mathbf{1}_N\mathbf{1}_N^T$ is a centering matrix. $\boldsymbol{\alpha}^{[1]}$ and $\boldsymbol{\alpha}^{[2]}$ are the Lagrange multipliers relayed to the constraints in Eq. (1), also called the dual variables. The kernel functions $K^{[1]}$: $\mathbb{R}^{d^{[1]}} \times \mathbb{R}^{d^{[1]}} \to \mathbb{R}$ and $K^{[2]}$: $\mathbb{R}^{d^{[2]}} \times \mathbb{R}^{d^{[2]}} \to \mathbb{R}$ are similarity functions and have to be positive definite.

The eigenvalues and eigenvectors that give an optimal correlation coefficient value are selected. The score variables on the training data can be computed by:

$$\mathbf{e}^{[1]} = \boldsymbol{\Omega}_c^{[1]}\boldsymbol{\alpha}^{[1]}$$
$$\mathbf{e}^{[2]} = \boldsymbol{\Omega}_c^{[2]}\boldsymbol{\alpha}^{[2]}. \tag{4}$$

Since the KCCA method is used to find interesting relations between variables it could be applied to do input selection. It is however important to make a good choice of the regularization constants $\gamma^{[1]}$ and $\gamma^{[2]}$ and of the kernels and their tuning parameters. For this purpose an additional validation set can be used to ensure meaningful generalization of the method.

2.2. Kernel Spectral Clustering

This section summarizes the Kernel Spectral Clustering (KSC) model as introduced by Alzate & Suykens [20]. KSC represents a spectral clustering formulation as a weighted kernel PCA problem, cast in the LS-SVM framework [21].

Given training data $\mathscr{D} = \{\mathbf{x}_i\}_{i=1}^N \subset \mathbb{R}^d$ and the number of clusters k, the primal model of KSC is formulated as follows:

$$\min_{\mathbf{w}^{(l)},\mathbf{e}^{(l)},b^{(l)}} \frac{1}{2}\sum_{l=1}^{k-1}\mathbf{w}^{(l)T}\mathbf{w}^{(l)} - \frac{1}{2N}\sum_{l=1}^{k-1}\gamma^{(l)}\mathbf{e}^{(l)T}\mathbf{D}^{-1}\mathbf{e}^{(l)}$$
$$\text{s.t.}\quad \mathbf{e}^{(l)} = \boldsymbol{\Phi}\mathbf{w}^{(l)} + b^{(l)}\mathbf{1}_N,\ l = 1,\cdots,k-1 \tag{5}$$

where $\mathbf{e}^{(l)} = [e_1^{(l)},\ldots,e_N^{(l)}]^T$ are the clustering scores or projections, $l = 1, \ldots, k-1$ indicate the score variables needed to encode k clusters, $\boldsymbol{\Phi} \in \mathbb{R}^{N\times d_h}$ is the feature matrix with $\boldsymbol{\Phi} = [\varphi(\mathbf{x}_1)^T;\cdots;\varphi(\mathbf{x}_N)^T]$ where φ: $\mathbb{R}^d \to \mathbb{R}^{d_h}$ is the mapping to a high-dimensional feature space, $b^{(l)}$ are bias terms, $\mathbf{D}^{-1} \in \mathbb{R}^{N\times N}$ is the inverse of the degree matrix $\mathbf{D}$ with

$D_{ii} = \sum_j \varphi(\mathbf{x}_i)^T\varphi(\mathbf{x}_j)$ and $\gamma^{(l)} \in \mathbb{R}^+$ are regularization constants.

The dual problem related to this primal formulation is:

$$\mathbf{D}^{-1}\mathbf{M_D}\boldsymbol{\Omega}\boldsymbol{\alpha}^{(l)} = \lambda^{(l)}\boldsymbol{\alpha}^{(l)} \tag{6}$$

where $\lambda^{(l)} = N/\gamma^{(l)}$, $\boldsymbol{\Omega}$ is the kernel matrix with $\boldsymbol{\Omega} = \boldsymbol{\Phi}\boldsymbol{\Phi}^T$ and

$$\begin{aligned}\Omega_{ij} &= K(\mathbf{x}_i, \mathbf{x}_j)\\ &= \varphi(\mathbf{x}_i)^T\varphi(\mathbf{x}_j).\end{aligned} \tag{7}$$

$\mathbf{M_D} = \mathbf{I}_N - \frac{1}{\mathbf{1}_N^T\mathbf{D}^{-1}\mathbf{1}_N}\mathbf{1}_N\mathbf{1}_N^T\mathbf{D}^{-1}$ is a centering matrix and $\boldsymbol{\alpha}^{(l)}$ are dual variables that we seek. The kernel function K: $\mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ is a similarity function and has to be positive definite.

The projections $e_i^{(l)}$ represent the latent variables of a set of $k-1$ binary clustering indicators given by $\text{sign}(e_i^{(l)})$. Since the first eigenvector $\boldsymbol{\alpha}^{(1)}$ already provides a binary clustering, only $k-1$ score variables are needed to encode k clusters [20]. To do cluster assignment in the training phase a codebook $\mathscr{C} = \{c_p\}_{p=1}^k$ is constructed where each codeword is a binary string of length $k-1$ representing a cluster. For each data point the corresponding clustering indicators $\text{sign}(e_i^{(1)})$, ...,$\text{sign}(e_i^{(k-1)})$ are compared against the codebook and the nearest codeword in terms of Hamming distance is selected.

The choice of the weight matrix $\mathbf{D}^{-1}$ is motivated by the random walks model [32] and the piecewise constant property of the eigenvectors when the clusters are well formed. More precisely, by choosing $\mathbf{D}^{-1}$, the dual problem in Eq. (6) is equivalent to spectral clustering with random walk Laplacian. This weight matrix is important since when omitted, the model results in kernel PCA which is known to lack discriminatory features for clustering.

For out-of-sample test data $\mathscr{D}_{\text{test}} = \{\mathbf{x}_{\text{test}_r}\}_{r=1}^{N_{\text{test}}}$ the projections, and hence the clustering indicators, can be calculated as follows:

$$\mathbf{e}_{\text{test}}^{(l)} = \boldsymbol{\Omega}_{\text{test}}\boldsymbol{\alpha}^{(l)} + b^{(l)}\mathbf{1}_{N_{\text{test}}} \tag{8}$$

where $l = 1, ...,k-1$ and $\boldsymbol{\Omega}_{\text{test}} \in \mathbb{R}^{N_{\text{test}}\times N}$ is the kernel matrix evaluated using the test data with $\Omega_{\text{test}ri} = K(\mathbf{x}_{\text{test}_r}, \mathbf{x}_i)$, $r = 1, ...,N_{\text{test}}$, $i = 1, ...,N$. The same codebook that was constructed during the training phase can be used to do the cluster assignment of the test data.

In this KSC framework a clustering model can be trained, validated and tested similarly to a standard classification learning scheme.

3. Multi-View Kernel Spectral Clustering

In this section the model *Multi-View Kernel Spectral Clustering* (MVKSC) is introduced. This is an extension to KSC and it is closely related to the KCCA formulation. Up to our knowledge, the proposed multi-view formulation of the spectral clustering problem as a weighted KCCA problem is new in the literature. In MVKSC data comes from two or more different views. When training on one view, the other views are taken into account by introducing a coupling term in the primal model.

3.1. Model

Given a number of V views, training data $\mathscr{D}^{[v]} = \{\mathbf{x}_i^{[v]}\}_{i=1}^N \subset \mathbb{R}^{d^{[v]}}$ for $v = 1, ...,V$ and the number of clusters k, the primal formulation of the MVKSC model is stated as follows:

$$\begin{aligned}\min_{\mathbf{w}^{[v](l)},\mathbf{e}^{[v](l)}} \quad & \frac{1}{2}\sum_{v=1}^{V}\sum_{l=1}^{k-1}\mathbf{w}^{[v](l)T}\mathbf{w}^{[v](l)} - \frac{1}{2N}\sum_{v=1}^{V}\sum_{l=1}^{k-1}\gamma^{[v](l)}\mathbf{e}^{[v](l)T}\mathbf{D}^{[v]-1}\mathbf{e}^{[v](l)}\\ & - \frac{1}{2}\sum_{v,u=1,v\neq u}^{V}\sum_{l=1}^{k-1}\rho^{(l)}\,\mathbf{e}^{[v](l)T}\mathbf{S}^{[v,u]}\mathbf{e}^{[u](l)}\end{aligned} \tag{9}$$

$$\text{s.t.}\begin{cases}\mathbf{e}^{[1](l)} = (\boldsymbol{\Phi}^{[1]} - \mathbf{1}_N\hat{\boldsymbol{\mu}}^{[1]T})\mathbf{w}^{[1](l)}\\ \vdots\\ \mathbf{e}^{[V](l)} = (\boldsymbol{\Phi}^{[V]} - \mathbf{1}_N\hat{\boldsymbol{\mu}}^{[V]T})\mathbf{w}^{[V](l)} \quad l = 1,\cdots,k-1\end{cases} \tag{10}$$

where similarly to the KSC notation, $\mathbf{e}^{[v](l)} \in \mathbb{R}^{N\times 1}$ are the clustering scores or projections related to the vth view, $l = 1, ...,k-1$ indicate the score variables needed to encode k clusters, $\boldsymbol{\Phi}^{[v]} \in \mathbb{R}^{N\times d_h^{[v]}}$ are the feature matrices with $\boldsymbol{\Phi}^{[v]} = [\varphi^{[v]}(\mathbf{x}_1^{[v]})^T;\cdots;\varphi^{[v]}(\mathbf{x}_N^{[v]})^T]$ where $\varphi^{[v]}$: $\mathbb{R}^{d^{[v]}} \to \mathbb{R}^{d_h^{[v]}}$ are the mappings to a high-dimensional feature space and $\gamma^{[v](l)} \in \mathbb{R}^+$ are regularization variables. $\mathbf{D}^{[v]^{-1}} \in \mathbb{R}^{N\times N}$ is the inverse of the degree matrix $\mathbf{D}^{[v]}$ with

$$D_{ii}^{[v]} = \sum_j \varphi^{[v]}(\mathbf{x}_i^{[v]})^T\varphi^{[v]}(\mathbf{x}_j^{[v]}). \tag{11}$$

As for KCCA the data is centered by means of the terms $\hat{\boldsymbol{\mu}}^{[v]}$ where

$$\begin{aligned}\hat{\boldsymbol{\mu}}^{[v]} &= \frac{1}{\mathbf{1}_N^T\mathbf{D}^{[v]^{-1}}\mathbf{1}_N}\sum_{i=1}^{N}\varphi^{[v]}(\mathbf{x}_i^{[v]})D_{ii}^{[v]^{-1}}\\ &= \frac{1}{\mathbf{1}_N^T\mathbf{D}^{[v]^{-1}}\mathbf{1}_N}\boldsymbol{\Phi}^{[v]^T}\mathbf{D}^{[v]^{-1}}\mathbf{1}_N.\end{aligned} \tag{12}$$

The primal optimization function is a sum of V different KSC objectives (one for each view) coupled by means of the *coupling term*, $-\sum_{v,u=1;v\neq u}^{V}\sum_{l=1}^{k-1}\rho^{(l)}\,\mathbf{e}^{[v](l)T}\mathbf{S}^{[v,u]}\mathbf{e}^{[u](l)}$, where $\rho^{(l)}$ are additional regularization constants and will be called the *coupling variables*. The entire coupling term describes the correlation between the score variables of the different views, which is maximized. A key feature of KSC is the addition of the weighting matrix consisting of the inverse of the degrees. Since the multi-view model aims at maximizing the variance of the weighted score variables, intuitively it follows that it should also aim to maximize the correlation between the weighted score variables of each view. This is achieved by setting $\mathbf{S}^{[v,u]} = \mathbf{D}^{[v]^{-\frac{1}{2}}}\mathbf{D}^{[u]^{-\frac{1}{2}}}$, for $v, u = 1, ...,V$ and $v \neq u$.

The Lagrangian of the primal problem is:

$$\begin{aligned}\mathscr{L}(\mathbf{w}^{[v](l)}, \mathbf{e}^{[v](l)}; \boldsymbol{\alpha}^{[v](l)}) \;=\; & \frac{1}{2}\sum_{v=1}^{V}\sum_{l=1}^{k-1}\mathbf{w}^{[v](l)T}\mathbf{w}^{[v](l)}\\ & - \frac{1}{2N}\sum_{v=1}^{V}\sum_{l=1}^{k-1}\gamma^{[v](l)}\mathbf{e}^{[v](l)T}\mathbf{D}^{[v]-1}\mathbf{e}^{[v](l)}\\ & - \sum_{v,u=1;v\neq u}^{V}\sum_{l=1}^{k-1}\rho^{(l)}\,\mathbf{e}^{[v](l)T}\mathbf{S}^{[v,u]}\mathbf{e}^{[u](l)}\\ & + \frac{1}{2}\sum_{v=1}^{V}\sum_{l=1}^{k-1}\boldsymbol{\alpha}^{[v](l)T}(\mathbf{e}^{[v](l)} - (\boldsymbol{\Phi}^{[v]} - \mathbf{1}_N\hat{\boldsymbol{\mu}}^{[v]T})\mathbf{w}^{[v](l)})\end{aligned} \tag{13}$$

with the Lagrange multipliers $\boldsymbol{\alpha}^{[v](l)}$ for $v = 1, ...,V$ and $l = 1, ...,k-1$.

The KKT necessary optimality conditions are:

$$\begin{cases}\dfrac{\partial\mathscr{L}}{\partial\mathbf{w}^{[v](l)}} = 0 \;\to \mathbf{w}^{[v](l)} = (\boldsymbol{\Phi}^{[v]} - \mathbf{1}_N\hat{\boldsymbol{\mu}}^{[v]T})^T\boldsymbol{\alpha}^{[v](l)},\\ \dfrac{\partial\mathscr{L}}{\partial\mathbf{e}^{[v](l)}} = 0 \;\to \boldsymbol{\alpha}^{[v](l)} = \dfrac{\gamma^{[v](l)}}{N}\mathbf{D}^{[v]^{-1}}\mathbf{e}^{[v](l)} + \rho^{(l)}\sum_{u=1;u\neq v}^{V}\mathbf{S}^{[v,u]}\mathbf{e}^{[u](l)},\\ \dfrac{\partial\mathscr{L}}{\partial\boldsymbol{\alpha}^{[v](l)}} = 0 \;\to \mathbf{e}^{[v](l)} = (\boldsymbol{\Phi}^{[v]} - \mathbf{1}_N\hat{\boldsymbol{\mu}}^{[v]T})\mathbf{w}^{[v](l)},\\ \qquad\text{where } v = 1,...,V \text{ and } l = 1,...,k-1.\end{cases} \tag{14}$$

Eliminating the primal variables $\mathbf{w}^{[v](l)}$ and $\mathbf{e}^{[v](l)}$ leads to the following generalized eigenvalue problem:

$$\begin{aligned}&\begin{bmatrix}\mathbf{0}_N & \mathbf{S}^{[1,2]}\boldsymbol{\Omega}_c^{[2]} & \cdots & \mathbf{S}^{[1,V]}\boldsymbol{\Omega}_c^{[V]}\\ \mathbf{S}^{[2,1]}\boldsymbol{\Omega}_c^{[1]} & \mathbf{0}_N & \cdots & \mathbf{S}^{[2,V]}\boldsymbol{\Omega}_c^{[V]}\\ \vdots & \vdots & \ddots & \vdots\\ \mathbf{S}^{[V,1]}\boldsymbol{\Omega}_c^{[1]} & \mathbf{S}^{[V,2]}\boldsymbol{\Omega}_c^{[2]} & \cdots & \mathbf{0}_N\end{bmatrix}\begin{bmatrix}\boldsymbol{\alpha}^{[1](l)}\\ \boldsymbol{\alpha}^{[2](l)}\\ \vdots\\ \boldsymbol{\alpha}^{[V](l)}\end{bmatrix}\\ &= \frac{1}{\rho^{(l)}}\begin{bmatrix}\mathbf{B}^{[1]} & \cdots & \mathbf{0}_N\\ \vdots & \ddots & \vdots\\ \mathbf{0}_N & \cdots & \mathbf{B}^{[V]}\end{bmatrix}\begin{bmatrix}\boldsymbol{\alpha}^{[1](l)}\\ \vdots\\ \boldsymbol{\alpha}^{[V](l)}\end{bmatrix}\end{aligned} \tag{15}$$

where $\mathbf{B}^{[v]} = \mathbf{I}_N - \frac{\gamma^{[v](l)}}{N}\mathbf{D}^{[v]^{-1}}\boldsymbol{\Omega_c}^{[v]}$ and $\boldsymbol{\alpha}^{[v](l)}$ are dual variables. $\boldsymbol{\Omega_c}^{[v]} = (\boldsymbol{\Phi}^{[v]} - \mathbf{1}_N\widehat{\boldsymbol{\mu}}^{[v]^T})(\boldsymbol{\Phi}^{[v]} - \mathbf{1}_N\widehat{\boldsymbol{\mu}}^{[v]^T})^T$ are the centered kernel matrices that capture the similarity between data of the view v. Similarly to KCCA, these centered kernel matrices can be computed by

$$\boldsymbol{\Omega_c}^{[v]} = \mathbf{M}_\mathbf{D}^{[v]}\boldsymbol{\Omega}^{[v]}\mathbf{L}_\mathbf{D}^{[v]} \tag{16}$$

where

$$\mathbf{M}_\mathbf{D}^{[v]} = \mathbf{I}_N - \frac{1}{\mathbf{1}_N^T\mathbf{D}^{[v]^{-1}}\mathbf{1}_N}\mathbf{1}_N\mathbf{1}_N^T\mathbf{D}^{[v]^{-1}}$$

and

$$\mathbf{L}_\mathbf{D}^{[v]} = \mathbf{I}_N - \frac{1}{\mathbf{1}_N^T\mathbf{D}^{[v]^{-1}}\mathbf{1}_N}\mathbf{D}^{[v]^{-1}}\mathbf{1}_N\mathbf{1}_N^T$$

are centering matrices and where $\boldsymbol{\Omega}^{[v]} = \boldsymbol{\Phi}^{[v]}\boldsymbol{\Phi}^{[v]^T}$ are the kernel matrices. In practice, we will not explicitly define the (possibly infinite) feature maps and instead compute the kernel matrices as

$$\begin{aligned}\Omega_{ij}^{[v]} &= \varphi^{[v]}(\mathbf{x}_i^{[v]})^T\varphi^{[v]}(\mathbf{x}_j^{[v]}) \\ &= K^{[v]}(\mathbf{x}_i^{[v]}, \mathbf{x}_j^{[v]}).\end{aligned} \tag{17}$$

As for KCCA and KSC, the kernel functions $K^{[v]}: \mathbb{R}^{d^{[v]}} \times \mathbb{R}^{d^{[v]}} \to \mathbb{R}$ are similarity functions and have to be positive definite. The degree matrix $\mathbf{D}^{[v]}$ will also be computed through the kernel matrix as

$$D_{ii}^{[v]} = \sum_j K^{[v]}(\mathbf{x}_i^{[v]}, \mathbf{x}_j^{[v]}) \tag{18}$$

which is equivalent to Eq. (11). The degree matrix can hence be interpreted as the similarity degree of each point with regard to all other points. Since each kernel function $K^{[v]}$ is defined only on one view v, it is possible to choose a different kernel function for each view. The eigenvalues associated with this generalized eigenvalue problem are $\frac{1}{\gamma^{[v](l)}}$, and $\gamma^{[v](l)}$ are the parameters to be tuned.

3.2. Decision rule

The cluster indicators $\text{sign}\left(e_i^{[v](1)}\right), \ldots, \text{sign}\left(e_i^{[v](k-1)}\right)$ for a certain training sample $\{\mathbf{x}_i^{[1]}, \ldots, \mathbf{x}_i^{[V]}\}$ for each view $v = 1, \ldots, V$ form the *encoding vector* for this sample. The score variables can be calculated as

$$\mathbf{e}^{[v](l)} = \boldsymbol{\Omega_c}^{[v]}\boldsymbol{\alpha}^{[v](l)}. \tag{19}$$

As for KSC, these encodings are then used to form the codebook (consisting of the k most occurring encoding vectors) to assign the training and test points to a certain cluster.

The cluster assignment can be done in two ways:

1. The cluster assignment is done separately for each view. Hence V codebooks $\mathscr{C}^{[v]} = \{c_p^{[v]}\}_{p=1}^k$ for each $v = 1, \ldots, V$ are created and the result will be a separate cluster assignment for each view and these can differ from each other.
2. The cluster assignment is done together on all views. A set of new score variables is defined as follows:

$$\mathbf{e}_{\text{total}}^{(l)} = \sum_{v=1}^{V} \beta^{[v]}\mathbf{e}^{[v](l)} \tag{20}$$

where $\sum_{v=1}^{V}\beta^{[v]} = 1$ and $\beta^{[v]} \in [0, 1]$ for $v = 1\ldots,V$, thus Eq. (20) is a convex combination of the vectors $\mathbf{e}^{[v](l)}$. Only one codebook $\mathscr{C} = \{c_p\}_{p=1}^k$ is created and the cluster assignments for all views is performed using these new score variables. The value of $\beta^{[v]}$ for each $v = 1, \ldots, V$ can be $1/V$ to take the average, or can be calculated based on the error covariance matrix where the error is computed in an unsupervised manner as one minus the mean silhouette value [33] rescaled between 0 and 1. The value of $\beta^{[v]}$ for each $v = 1, \ldots, V$ may then be chosen so that it minimizes the error, similarly to how it is done for committee networks [34]. Since in our experiments we noticed that taking the average produced overall good results, we will use this throughout the reminder of the paper.

Algorithm 1 summarizes the algorithm used to cluster the training data. The notations $^{[1:\,V]}$ and $^{(1:k-1)}$ are shorthand for 'for all views $v = 1, \ldots, V$' and 'for all binary subproblems $l = 1, \ldots, k-1$', respectively. Notice that this algorithm assigns the clusters separately. To use the second version of the model (cluster assignment on all views together) the score variables defined in Eq. (20) should be binarized and used as encoding vectors in line 4.

3.3. Out-of-sample extension

Finally following from the KKT conditions (see Eq. (14)) for out-of-sample test data the projections can be calculated as follows:

$$\mathbf{e}_{\text{test}}^{[v](l)} = \boldsymbol{\Omega}_{\mathbf{c}_{\text{test}}}^{[v]}\boldsymbol{\alpha}^{[v](l)} \tag{21}$$

where $l = 1, \ldots, k-1$ and $v = 1, \ldots, V$. $\boldsymbol{\Omega}_{\mathbf{c}_{\text{test}}}^{[v]} \in \mathbb{R}^{N_{\text{test}} \times N}$ are the centered kernel matrices evaluated using the test data with

$$\boldsymbol{\Omega}_{\mathbf{c}_{\text{test}}}^{[v]} = \mathbf{M}_{\mathbf{D}_{\text{test}}}^{[v]}\boldsymbol{\Omega}_{\text{test}}^{[v]}\mathbf{L}_\mathbf{D}^{[v]}. \tag{22}$$

The test kernel matrix is defined as $\boldsymbol{\Omega}_{\text{test}}^{[v]} = \boldsymbol{\Phi}_{\text{test}}^{[v]}\boldsymbol{\Phi}^{[v]^T}$ and can practically be computed as

---

Input: Training sets $\mathcal{D}^{[1:V]} = \{\mathbf{x}_i^{[1:V]}\}_{i=1}^N$, kernel function K with kernel parameters θ (if any), regularization parameters $\gamma^{[1:V]}$ and a number of clusters k.

Output: Cluster assignment $q_i^{[v]}$ for each point $\mathbf{x}_i^{[v]}$.

1: Compute centered kernel matrix $\boldsymbol{\Omega_c}^{[1:V]}$ and degree matrix $\mathbf{D}^{[1:V]}$ based on Eq. (16) and Eq. (18) using $(\mathcal{D}^{[1:V]}, K, \theta)$.
2: Compute the solution vectors $\boldsymbol{\alpha}^{[1:V]^{(1:k-1)}}$ of the generalized eigenvalue problem stated in Eq. (15) using $(\boldsymbol{\Omega_c}^{[1:V]}, \mathbf{D}^{[1:V]}, \gamma^{[1:V]})$.
3: Compute the score variables $\mathbf{e}^{[1:V]^{(1:k-1)}}$ by means of Eq. (19) using $(\boldsymbol{\alpha}^{[1:V]^{(1:k-1)}}, \boldsymbol{\Omega_c}^{[1:V]})$.
4: Binarize the score variables: $\text{sign}(\mathbf{e}^{[1:V]^{(1:k-1)}})$ and let $\text{sign}(\mathbf{e}_i^{[v]^{(1:k-1)}}) \in \{-1, 1\}^{k-1}$ be the encoding vector for the training data point $\mathbf{x}_i^{[v]}$ belonging to view v.
5: Count the occurrences of the different encodings and find the k encodings which occur most. Let the codebook be formed by these k encodings: $C^{[1:V]} = \{c_p^{[1:V]}\}_{p=1}^k, c_p^{[1:V]} \in \{-1, 1\}^{k-1}$.
6: For each view v: assign each training point $\mathbf{x}_i^{[v]}$ to cluster $q_i^{[v]}$ by means of applying the codebook $C^{[v]}$ on the encoding vector: $q_i^{[v]} = \text{argmin}_p d_H(\text{sign}(\mathbf{e}_i^{[v]^{(1:k-1)}}), c_p^{[v]})$ and where $d_H(\cdot,\cdot)$ is the Hamming distance.

Algorithm 1. MVKSC.

Input: Training sets $\mathcal{D}^{[1:V]} = \{\mathbf{x}_i^{[1:V]}\}_{i=1}^{N}$, independent test sets $\mathcal{D}_{\text{test}}^{[1:V]} = \{\mathbf{x}_{\text{test}_i}^{[1:V]}\}_{i=1}^{N_{\text{test}}}$, kernel function K with kernel parameters θ (if any), dual variables $\boldsymbol{\alpha}^{[1:V]^{(1:k-1)}}$ and codebooks $C^{[1:V]}$.
Output: Cluster assignment $q_{\text{test}_i}^{[v]}$ for each test point $\mathbf{x}_{\text{test}_i}^{[v]}$.

1: Compute centered kernel matrix $\boldsymbol{\Omega}_{\text{test}}^{[1:V]}$ based on Eq. (21) using $(\mathcal{D}_{\text{test}}^{[1:V]}, \mathcal{D}^{[1:V]}, K, \theta)$.
2: Compute the score variables $\mathbf{e}_{\text{test}}^{[v]^{(1:k-1)}}$ by means of Eq. (21) using $(\boldsymbol{\alpha}^{[1:V]^{(1:k-1)}}, \boldsymbol{\Omega}_{\text{test}}^{[1:V]})$.
3: Binarize the score variables: $\text{sign}(\mathbf{e}_{\text{test}}^{[1:V]^{(1:k-1)}})$ and let $\text{sign}(\mathbf{e}_{\text{test}}^{[v]^{(1:k-1)}}) \in \{-1, 1\}^{k-1}$ be the encoding vector for the test data point $\mathbf{x}_{\text{test}_i}^{[v]}$ belonging to view v.
4: For each view v: assign each test point $\mathbf{x}_{\text{test}_i}^{[v]}$ to cluster $q_{\text{test}_i}^{[v]}$ by means of applying the codebook $C^{[v]}$ on the encoding vector: $q_{\text{test}_i}^{[v]} = \text{argmin}_p d_H(\text{sign}(\mathbf{e}_{\text{test}_i}^{[v]^{(1:k-1)}}), c_p^{[v]})$ and where $d_H(\cdot,\cdot)$ is the Hamming distance.

Algorithm 2. MVKSC Out-of-sample extension.

Input: Training sets $\mathcal{D}^{[1:V]} = \{\mathbf{x}_i^{[1:V]}\}_{i=1}^{N}$, independent test sets $\mathcal{D}_{\text{test}}^{[1:V]} = \{\mathbf{x}_{\text{test}_i}^{[1:V]}\}_{i=1}^{N_{\text{test}}}$, kernel function K and a tuning criteria.
Output: Cluster assignment $q_{\text{test}_i}^{[v]}$ for each test point $\mathbf{x}_{\text{test}_i}^{[v]}$.

1: Perform Simulated Annealing with given criteria on Algorithm 1, using $(\mathcal{D}^{[1:V]}, K)$ to obtain tuned parameters θ (if any), $\gamma^{[1:V]}$ and k.
2: Apply Algorithm 1 $(\mathcal{D}^{[1:V]}, K, \theta, \gamma^{[1:V]}, k)$ to compute $\boldsymbol{\alpha}^{[1:V]^{(1:k-1)}}, C^{[1:V]}$.
3: Use Algorithm 2 $(\mathcal{D}^{[1:V]}, \mathcal{D}_{\text{test}}^{[1:V]}, K, \theta, \boldsymbol{\alpha}^{[1:V]^{(1:k-1)}}, C^{[1:V]})$ to obtain the cluster assignments on all of the test data points.

Algorithm 3. MVKSC Model selection.

$$\Omega_{\text{test}_{ij}}^{[v]} = K^{[v]}\left(\mathbf{x}_{\text{test}_i}^{[v]}, \mathbf{x}_j^{[v]}\right). \tag{23}$$

The Out-of-sample extension method is given by Algorithm 2. The same codebook(s) that was constructed during the training phase are used to do the cluster assignment of the test data. The cluster assignment is done in the same manner as for the training phase, so either separately or together.

3.4. Model selection

The results obtained by MVKSC depend on the choice of the kernel function and its parameters and on the choice of the view-specific regularization parameters $\gamma^{[1]}, ..., \gamma^{[V]}$.

In these experiments it is chosen to take a different regularization parameter $\gamma^{[v]}$ for each view $v = 1, ..., V$ but to take the same parameter for each binary cluster problem, hence $\gamma^{[v]} = \gamma^{[v]^{(1)}} = ... = \gamma^{[v]^{(k-1)}}$, in order to reduce the tuning complexity. To decrease the tuning complexity even further one could also choose to tune only one regularization parameter γ and set $\gamma = \gamma^{[1]} = ... = \gamma^{[V]}$. Since the MVKSC model allows for a different feature map for each view, V different kernels could be chosen, one for each view specific. In these experiments however this is not considered.

The tuning of the kernel and regularization parameters is done by simulated annealing. The model selection procedure is described by Algorithm 3. Notice that since the method is unsupervised, model selection is performed without knowing the true underlying clustering (labels). Hence, the tuning criteria have to be unsupervised as well. Three criteria were considered to measure the performance of the model with a certain set of parameters: *Silhouette* (Sil), *Balanced Line Fit* (BLF) and *Balanced Angular Fit* (BAF)[2] . Silhouette [33] is a widely used internal criterion that measures how tightly grouped all the data in the obtained clusters are. BLF [20] expresses how validation points belonging to the same cluster are collinear in the space of the projections. The values of BLF lie in the range [0, 1]. A higher value means that the clusters are better separated. For BAF [26] the sum of the cosine similarity between the validation points and the cluster prototypes is calculated and divided by the cardinality of that cluster. The similarity values are then summed up and divided by the total number of clusters to become the BAF value. The criteria are evaluated for each view and the total performance of the model is the mean of all.

[2] For BLF and BAF five values of η are considered namely: 0.75, 0.80, 0.85, 0.90 and 0.95. Only the best obtained results will be reported.

4. Experiments

In this section the results of MVKSC are shown and compared to other multi-view clustering methods. The results will be discussed on two toy problems and three real-world datasets.

4.1. Datasets

A brief description of each dataset used is given here. The important statistics of them are summarized in Table 1.

- **Synthetic dataset 1:** The first synthetic dataset consists of two views where each view is generated similar to Yi et al. [35], where for each cluster a sample $(\mathbf{x}_i^{[1]}, \mathbf{x}_i^{[2]})$ is generated form a two-component Gaussian mixture model. The cluster means for view 1 are (1 1) and (2 2), and for view 2 are (2 2) and (1 1). The covariances

Table 1
Details of the datasets used in the experiments.

Dataset	# Data points	# Views	Dimensions
Synth data 1	1000	2	$d^{[1]} = 2$
			$d^{[2]} = 2$
Synth data 2	1000	3	$d^{[1]} = 2$
			$d^{[2]} = 2$
			$d^{[3]} = 2$
Reuters 1	1200	2	$d^{[1]} = 21531$
			$d^{[2]} = 24892$
Reuters 2	600	3	$d^{[1]} = 9749$
			$d^{[2]} = 9109$
			$d^{[3]} = 7774$
3-Sources	169	3	$d^{[1]} = 3560$
			$d^{[2]} = 3631$
			$d^{[3]} = 3068$

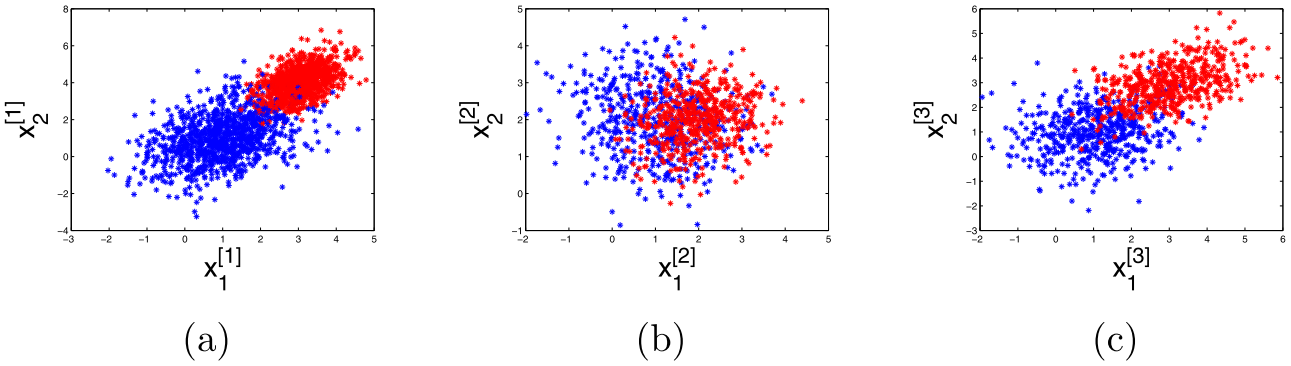


Fig. 1. (a) First, (b) second and (c) third view of synthetic dataset 2.

for the two views are

$$\boldsymbol{\Sigma}_1^{[1]} = \begin{pmatrix} 1 & 0.5 \\ 0.5 & 1.5 \end{pmatrix}, \boldsymbol{\Sigma}_2^{[1]} = \begin{pmatrix} 0.3 & 0 \\ 0 & 0.6 \end{pmatrix},$$

$$\boldsymbol{\Sigma}_1^{[2]} = \begin{pmatrix} 0.3 & 0 \\ 0 & 0.6 \end{pmatrix}, \boldsymbol{\Sigma}_2^{[2]} = \begin{pmatrix} 1 & 0.5 \\ 0.5 & 1.5 \end{pmatrix}.$$

For each data source 1000 points are sampled.

- **Synthetic dataset 2:** The second synthetic dataset is depicted in Fig. 1 and consist of three views where again each sample $(\mathbf{x}_i^{[1]}, \mathbf{x}_i^{[2]}, \mathbf{x}_i^{[3]})$ is generated for each of the two clusters by a two component Gaussian mixture model. The cluster means for view 1 are (1 1) and (3 4), and for view 2 are (1 2) and (2 2), and finally for view 3 the cluster means are (1 1) and (3 3). The covariances for the three views are

$$\boldsymbol{\Sigma}_1^{[1]} = \begin{pmatrix} 1 & 0.5 \\ 0.5 & 1.5 \end{pmatrix}, \boldsymbol{\Sigma}_2^{[1]} = \begin{pmatrix} 0.3 & 0.2 \\ 0.2 & 0.6 \end{pmatrix},$$

$$\boldsymbol{\Sigma}_1^{[2]} = \begin{pmatrix} 1 & -0.2 \\ -0.2 & 1 \end{pmatrix}, \boldsymbol{\Sigma}_2^{[2]} = \begin{pmatrix} 0.6 & 0.1 \\ 0.1 & 0.5 \end{pmatrix},$$

$$\boldsymbol{\Sigma}_1^{[3]} = \begin{pmatrix} 1.2 & 0.2 \\ 0.2 & 1 \end{pmatrix}, \boldsymbol{\Sigma}_2^{[3]} = \begin{pmatrix} 1 & 0.4 \\ 0.4 & 0.7 \end{pmatrix}.$$

For each data source 1000 points are sampled.

- **Reuters Multilingual dataset:** The first real-world dataset is taken from the UCI repository and is a subset of the Reuters Multilingual dataset described by Amini et al. [36]. The dataset consist of documents originally written in five different languages (English, French, German, Spanish and Italian) and their translations, over a common set of six categories. To fairly compare with the work of Kumar er al. [23] and Liu et al. [37] we use the same two subsets as described in these papers, called Reuters 1 and Reuters 2.
  - **Reuters 1:** This dataset is also used by Kumar et al. [23] and consist of two views. The first view consist of documents originally written in English. Their French translations are used for the second view. For this subset 1200 documents have been randomly selected in such a way that from each of the six clusters 200 documents are sampled.
  - **Reuters 2:** This dataset is also used by Liu et al. [37] and consist of three views. The first view again consist of documents originally written in English. The second and third view are their French and German translations, respectively. Although this subset contains more views than the first subset, it is considerably smaller because here only 600 documents are randomly chosen. These documents are again chosen in a balanced way so that from each of the six clusters 100 documents are sampled.
    In both subsets the documents are represented by a bag-of-words representation and hence the features are very sparse and high-dimensional.
- **3-Sources Text dataset:** This real-world dataset is collected from three online news sources: BBC, Reuters and The Guardian described by Greene and Cunningham [38]. The dataset contains 948 news articles covering 416 distinct news stories from the periods February to April 2009. Of these stories, 169 were reported in all three sources and hence only these 169 stories are used for the experiments. Each story was manually annotated with one of the six topical labels: business, entertainment, health, politics, sport and technology.

For the two synthetic datasets the radial basis function (RBF) kernel is chosen, so the corresponding kernel function is $K(\mathbf{x}_i^{[v]}, \mathbf{x}_j^{[v]}) = \exp\left(-\frac{\left\|\mathbf{x}_i^{[v]} - \mathbf{x}_j^{[v]}\right\|^2}{2\sigma^2}\right)$ for $v = 1, \ldots, V$ and where σ is a kernel parameter to be tuned. Since the Reuters and 3-Sources datasets are very high-dimensional using an RBF kernel, hence bringing the data to a even higher-dimensional feature space, is not recommended [39]. Therefore a normalized polynomial kernel of degree 1 (linear) and 2 were considered for these datasets. So the proposed kernel function for these datasets is $K(\mathbf{x}_i^{[v]}, \mathbf{x}_j^{[v]}) = \frac{\left(\mathbf{x}_i^{[v]T}\mathbf{x}_j^{[v]} + t^2\right)^d}{\sqrt{\left(\mathbf{x}_i^{[v]T}\mathbf{x}_i^{[v]} + t^2\right)^d \left(\mathbf{x}_j^{[v]T}\mathbf{x}_j^{[v]} + t^2\right)^d}}$ for $v = 1, \ldots, V$, $d = 1, 2$ and where t is a kernel parameter to be tuned. Other appropriate kernel functions for text-data such as Chi-square kernels [40] and String kernels [41] were not considered.

4.2. Baseline algorithms

The performances of the proposed method MVKSC on the different datasets are compared with the following baseline algorithms:

- **Best single view:** The results of applying KSC on the most informative view, i.e., the one on which KSC achieves the best performance.
- **Feature concatenation:** The features of all views are concatenated and KSC is used to do clustering on this concatenated view representation.
- **Kernel addition:** The separate kernels are combined by adding them, and KSC is used to do clustering with this combined kernel.
- **Kernel product:** The separate kernels are combined by taking the element-wise product, and KSC is used to do clustering with this combined kernel.
- **Optimized Kernel *k*-means clustering (OKKC):** This method, by

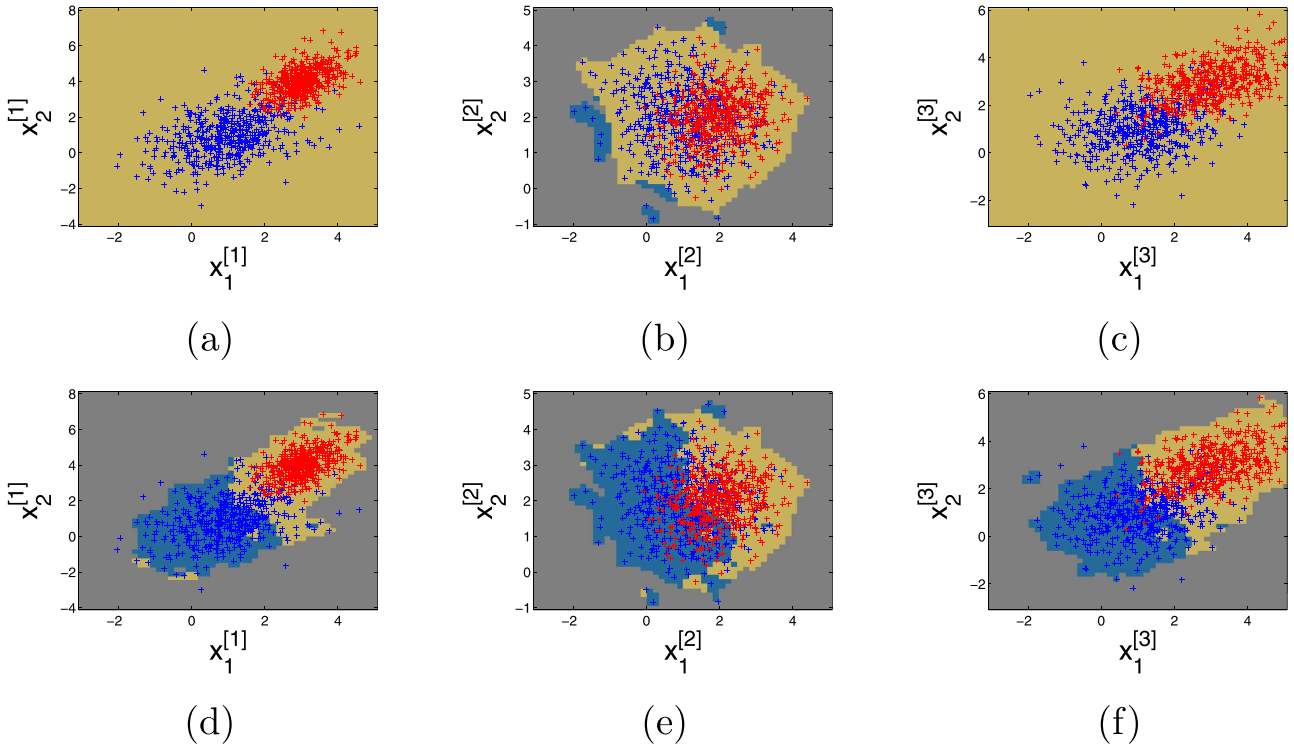


Fig. 2. Cluster boundaries for the second synthetic dataset. Fig. (a), (b) and (c) show the cluster boundaries when applying KSC on respectively the first, second and third view separately. The RBF kernel parameter is set to $\sigma^2 = 0.1$ and the eigenvalue γ equals 1 for all views. Fig. (d), (e) and (f) shows the cluster boundaries for each view when applying MVKSC on the entire dataset. The RBF kernel parameter equals $\sigma^2 = 0.1$ and the regularization parameters are set to $\gamma^{[1]} = \gamma^{[2]} = \gamma^{[3]} = 1$. The coloring of the datapoints indicate the true underlying clustering. The areas colored in yellow and blue indicate the cluster partitioning, the zero clusters are indicated in gray. For this example MVKSC is capable of improving the cluster quality. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Yu et al. [8], combines multiple data sources for *k*-means clustering. The optimal combination of the kernel matrices for each source is found by applying an alternating minimization method that optimizes the cluster memberships and the kernel coefficients iteratively.

- **Co-regularized spectral clustering (Co-reg):** This method is suggested by Kumar et al. [23] and applies the co-regularization framework to spectral clustering. It introduces a joint objective function by adding up the standard spectral clustering objectives of each view and adding an unweighted coupling of the score variables. It further uses an iterative alternating maximization framework to reduce the problem to the standard spectral clustering objective. They propose two versions of this co-regularization scheme, named pairwise (P) and centroid-based (C). The co-regularization parameters (one for each view) are varied from 0.01 to 0.05.
- **Multi-View NMF (MVNMF):** This method is suggested by Liu et al. [37] and performs a joint matrix factorization process with an additional constraint that pushes clustering solutions of each view to a common consensus. The method has one regularization parameter corresponding to each view.

The baseline algorithms are tuned in the same way as MVKSC. For Best Single View, Feature Concatenation, Kernel Addition, Kernel Product, OKKC and Co-reguSC the same type of kernels are chosen and the kernel parameters are tuned in the same way. For Co-reguSC and for MultiNMF the (co-)regularization parameters are tuned and can differ for each view.

4.3. Results

First MVKSC is compared to KSC on the second synthetic dataset. An RBF kernel is used and the kernel parameter $\sigma^2 = 0.1$ is chosen for all views. The regularization variables are fixed to $\gamma^{[1]} = \gamma^{[2]} = \gamma^{[3]} = 1$. For KSC the parameter γ is also equal to 1 for all three views, which results in that the only difference in the models is the multi-view property of MVKSC. KSC is applied on all three views separately and MVKSC on all views together. The resulting clustering boundaries are shown in Fig. 2. These figures show that MVKSC is clearly better at finding the underlying cluster boundaries, even when the clusters are overlapping. Fig. 3 show the first two projections when applying KSC on the three views separately and when applying MVKSC on the entire dataset. For each point $(e_i^{(1)}, e_j^{(2)})$ in this plot the obtained cluster is indicated by means of color. The plots show that KSC is not able to differentiate well between the two clusters, they suggest that there is one dominant cluster although the dataset contains the same amount of samples for each cluster. The plot corresponding to the MVKSC model clearly shows two separate clusters of almost equal size. This example shows that MVKSC is capable of improving the cluster quality by taking into account the information from multiple views.

For the next experiments all baseline algorithms are applied to the above discussed datasets. To evaluate the cluster quality the criteria *Normalized Mutual Information* (NMI) and *Adjusted Rand Index* (ARI) are used. NMI [42] gives the mutual information between the obtained clustering and the true underlying clustering, normalized by the cluster entropies. It takes values in the range [0, 1] where a higher value indicates a closer match to the true underlying clustering. ARI [43] computes a similarity measure between the obtained clustering and the true underlying clustering by considering all pairs of samples and counting those that are assigned to the same or to different clusters. The ARI is then adjusted for chance, which ensures that it will have a value close to 0 for random labeling independently of the number of clusters and samples and exactly 1 when the clustering equals the true underlying clustering. The true underlying clustering is available for all datasets but is only used to calculate the NMI and ARI of the trained model and not during the training/validation phase.

As explained before the cluster assignment can be done in two ways;

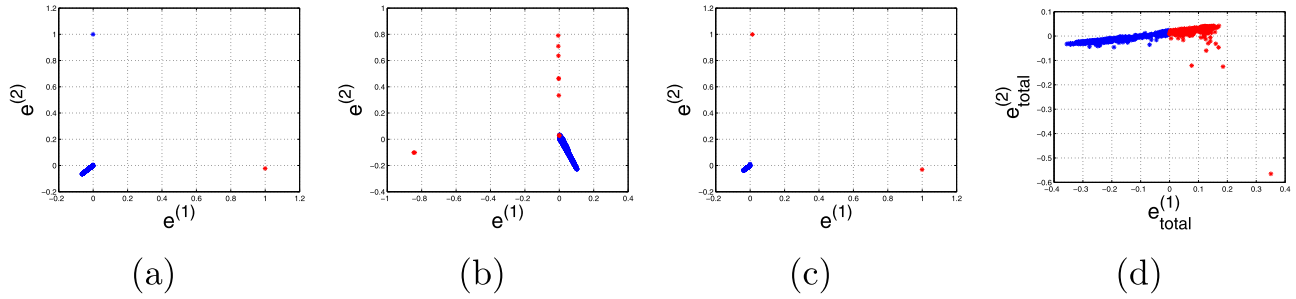


Fig. 3. Projections for the second synthetic dataset. Fig. (a), (b) and (c) show the projections **e** of KSC applied respectively on the first, second and third view. Fig. (d) shows the projections $\mathbf{e}_{total}$ (see Eq. (20)) of MVKSC applied on all views. The RBF kernel parameter is set to $\sigma^2 = 0.1$ for all models, the eigenvalue γ of the KSC models equals 1 for all views and the regularization parameters of MVKSC are analogously set to $\gamma^{[1]} = \gamma^{[2]} = \gamma^{[3]} = 1$. The coloring indicates the obtained clustering. For this example MVKSC is better at finding the two clusters than KSC.

Table 2
NMI results on two synthetic datasets, for three tuning criteria, with the proposed methods. The highest NMI values, and hence the best performing methods, for each dataset are indicated in bold.

Method	Synth 1			Synth 2		
	Sil	BLF	BAF	Sil	BLF	BAF
Best Single KSC	0.0207	0.268	0.268	0.912	0.901	0.872
Feature Concat	0.0401	0.348	0.348	0.936	0.928	0.912
Kernel Addition	0.366	0.348	0.390	0.832	0.912	0.917
Kernel Product	0.156	0.348	0.350	0.832	0.928	0.917
OKKC [8]	0.149	0.250	0.250	0.428	0.952	0.952
Co-reg(P) [23]	0.324	0.294	0.319	0.931	0.936	0.973
Co-reg(C) [23]	0.320	0.309	0.320	**0.989**	**0.989**	0.963
MVNMF [37]	0.00635	0.00635	0.00635	0.0117	0.0117	0.0117
MVKSC ($\mathbf{e}^{[v]}$)	0.301	0.305	0.300	0.764	0.764	0.764
MVKSC ($\mathbf{e}_{total}$)	0.365	**0.404**	0.361	**0.989**	0.936	0.917

Table 3
ARI results on two synthetic datasets, for three tuning criteria, with the proposed methods. The highest ARI values, and hence the best performing methods, for each dataset are indicated in bold.

Method	Synth 1			Synth 2		
	Sil	BLF	BAF	Sil	BLF	BAF
Best Single KSC	5.60e-05	0.335	0.335	0.949	0.941	0.918
Feature Concat	3.69e-05	0.442	0.442	0.968	0.960	0.949
Kernel Addition	0.462	0.442	0.490	0.883	0.949	0.952
Kernel Product	0.169	0.442	0.445	0.883	0.960	0.952
OKKC [8]	0.084	0.079	0.288	0.306	0.956	0.976
Co-reg(P) [23]	0.437	0.308	0.430	0.960	0.968	0.988
Co-reg(C) [23]	0.429	0.447	0.441	**0.996**	**0.996**	0.984
MVNMF [37]	1.20e-07	1.20e-07	1.20e-07	7.06e-04	7.06e-04	7.06e-04
MVKSC ($\mathbf{e}^{[v]}$)	0.332	0.333	0.328	0.732	0.736	0.737
MVKSC ($\mathbf{e}_{total}$)	0.455	**0.505**	0.457	**0.996**	0.968	0.952

separately for each view (denoted as MVKSC($\mathbf{e}^{[v]}$)) and together on all views (denoted as MVKSC($\mathbf{e}_{total}$)) by means of Eq. (20) where for these experiments $\beta^{[v]} = 1/V$ for $v = 1, ..., V$ so that the average score variables are used.

These results are depicted in Tables 2 and 3 for the two synthetic datasets and in Tables 4 and 5 for the real-world datasets. The results show that MVKSC is able to improve on KSC for all datasets, due to the fact that it is capable of exploiting information from multiple views. Secondly the results show that MVKSC performs better than feature concatenation, kernel addition and kernel product for all datasets. This suggests that adding an extra coupling term to the primal model is a better way to combine multiple views than to simply concatenate the features or do a simple combination of the kernels. The results also show that MVKSC is better or at least as good as the other state-of-the-art methods in four out of the five datasets. The poor results of MVNMF on the synthetic datasets can be explained by the nature of the datasets and the clustering assumption of MVNMF. MVNMF is an extension to Non-Negative Matrix Factorization (NMF) which, as explained by Kuang et al. [44], does not perform well when the cluster centers are along the same direction. Tables 2, 3, 4 and 5 report the results for the three tuning criteria. For the second synthetic dataset and the first Reuters dataset, tuning MVKSC with Silhouette resulted in the most optimal parameters. Whereas for the first synthetic dataset and the second Reuters dataset BLF, and for the 3-sources dataset BAF, proved to be more successful. This shows that when tuning MVKSC it is best to consider multiple tuning criteria.

Another way to evaluate the models is by looking at the runtime on test data. These results are depicted in Table 6. For these timing experiments all methods were run in Matlab (R2014a). As to be expected, the results clearly show that for all methods and datasets it takes more time to do multi-view clustering compared to clustering on one view. One can also notice that although MVKSC is slower than the three simple coupling methods feature concatenation, kernel addition and kernel product, it is considerably faster than the other state-of-the-art methods from Yu et al. [8], Kumar et al. [23] and Liu et al. [37].

5. Large-scale experiments

Because of the out-of-sample extension of MVKSC, the method is suitable for clustering large-scale datasets. This can easily be achieved by randomly selecting a subset of the data to do the training and using the out-of-sample extension to cluster the entire large dataset. In particular, if the dataset consists of N datapoints, we randomly choose $m \ll N$ points to solve the generalized eigenvalue problem in Eq. (15). This way, the kernel matrices $\boldsymbol{\Omega}_c^{[v]}$, for all views $v = 1, ..., V$, to be stored have dimension $m \times m$. The largest matrix to be stored is the $(mV \times mV)$-dimensional matrix on the left hand side of the generalized eigenvalue problem. The entire dataset could than be clustered by means of Eq. (21). The vector $\boldsymbol{\alpha}^{[v](l)}$ was obtained during training so the eigenvalue problem does not need to be computed anymore. The test kernel matrix $\boldsymbol{\Omega}_{\mathbf{c}_{\text{test}}}^{[v]}$ needs to be stored but this matrix is at most (if all datapoints are computed simultaneously) $N \times m$-dimensional. Of course, because of the out-of-sample nature, this could even be avoided by clustering the test points in smaller groups or even point-by-point. This approach is summarized by Algorithm 4. Note that if the cluster assignment is done together on all views (see Eq. (20)) then $\mathbf{q}^{[1]} = ... = \mathbf{q}^{[V]}$.

To show the performance of MVKSC on large-scale data we first considered a synthetic case. For this purpose we generated the first synthetic dataset (as described in the previous section) again but with increasing number of datapoints N sampled. We considered $N \in \{10^2, 0.5 \cdot 10^2, 10^3, 0.5 \cdot 10^3, 10^4, 0.5 \cdot 10^4, 10^5, 0.5 \cdot 10^5, 10^6\}$ and $m = N$ for $N < 10^3$ and $m = 10^3$ for $N \geq 10^3$. The results are depicted in Fig. 4.

Langone et al. [45] note that the computational complexity of KSC

Table 4
NMI results on three real-world datasets, for three tuning criteria, with the proposed methods. The highest NMI values, and hence the best performing methods, for each dataset are indicated in bold.

Method	Reuters 1			Reuters 2			3-Sources		
	Sil	BLF	BAF	Sil	BLF	BAF	Sil	BLF	BAF
Best Single KSC	0.230	0.225	0.250	0.204	0.258	0.291	0.357	0.437	0.555
Feature Concat	0.348	0.260	0.440	0.179	0.147	0.271	0.503	0.270	0.584
Kernel Addition	0.410	0.315	0.407	0.301	0.194	0.297	0.374	0.331	0.538
Kernel Product	0.102	0.277	0.313	0.126	0.219	0.221	0.602	0.546	0.614
OKKC [8]	0.386	0.378	0.419	0.0606	0.340	0.370	0.465	0.568	0.549
Co-reg(P) [23]	0.410	0.450	0.409	**0.403**	0.381	0.395	0.566	0.596	0.639
Co-reg(C) [23]	0.451	0.467	0.467	0.331	0.330	0.378	0.633	0.640	0.631
MVNMF [37]	0.379	0.459	0.442	0.313	0.319	0.323	0.459	0.415	0.411
MVKSC ($\mathbf{e}^{[v]}$)	0.428	0.428	**0.479**	0.259	0.315	0.305	0.620	0.599	0.690
MVKSC ($\mathbf{e}_{total}$)	0.386	0.386	0.409	0.161	0.319	0.311	**0.697**	0.599	0.627

Table 5
ARI results on three real-world datasets, for three tuning criteria, with the proposed methods. The highest ARI values, and hence the best performing methods, for each dataset are indicated in bold.

Method	Reuters 1			Reuters 2			3-Sources		
	Sil	BLF	BAF	Sil	BLF	BAF	Sil	BLF	BAF
Best Single KSC	0.142	0.155	0.209	0.171	0.188	0.195	0.162	0.373	0.609
Feature Concat	0.199	0.154	0.315	0.127	0.092	0.145	0.490	0.148	0.577
Kernel Addition	0.351	0.205	0.350	0.179	0.130	0.199	0.280	0.150	0.527
Kernel Product	0.025	0.191	0.251	0.045	0.114	0.112	0.383	0.483	0.514
OKKC [8]	0.145	0.226	0.223	0.004	0.186	0.146	0.340	0.448	0.227
Co-reg(P) [23]	0.292	0.315	0.288	0.294	**0.302**	0.294	0.460	0.442	0.567
Co-reg(C) [23]	0.337	0.366	0.354	0.158	0.142	0.302	0.419	0.535	0.489
MVNMF [37]	0.202	0.355	0.342	0.173	0.214	0.227	0.338	0.330	0.329
MVKSC ($\mathbf{e}^{[v]}$)	0.365	0.365	**0.394**	0.177	0.209	0.171	0.471	0.406	0.675
MVKSC ($\mathbf{e}_{total}$)	0.327	0.327	0.220	0.095	0.214	0.183	**0.658**	0.475	0.633

Table 6
Runtime (in seconds) on test data for five datasets with the proposed methods.

Method	Synth 1	Synth 2	Reuters 1	Reuters 2	3-Sources
Best Single KSC	0.15	0.188	4.79	0.213	0.0789
Feature Concat	0.11	0.233	10.5	0.270	0.107
Kernel Addition	0.25	0.263	8.73	0.410	0.105
Kernel Product	0.21	0.262	8.49	0.369	0.197
OKKC [8]	657	1.61e+03	1.48e+04	161	1.74
Co-reg(P) [23]	24.5	20.1	51.9	13.3	5.88
Co-reg(C) [23]	17.6	15.3	51.6	8.00	3.12
MVNMF [37]	75.86	176	476	137	51.6
MVKSC ($\mathbf{e}^{[v]}$)	1.36	2.02	17.7	2.45	0.205
MVKSC ($\mathbf{e}_{total}$)	1.13	2.06	13.7	2.55	0.250

depend on solving the eigenvalue problem in Eq. (6) for the training phase and computing Eq. (8) for the testing phase. This complexity is given by $O(m^2) + O(mN)$. For MVKSC the computational complexity depend on solving the generalized eigenvalue problem in Eq. (15) for the training and solving Eq. (21) (line 4 in Algorithm 4) for the test phase. This results in a complexity given by $O(Vm^2) + O(VmN)$, where the number of views V is usually small. This agrees with the experimental findings in Fig. 4(a) which shows an almost linear correlation between the runtime of MVKSC and the number of datapoints N in the dataset. Notice that for the other multi-view methods the runtime could only be timed up to $N = 10^4$. Because of the lack of out-of-sample extension, running them with a higher number of datapoints resulted in memory problems. The figure further shows that MVKSC has a substantial lower runtime than the other multi-view methods, which is in line with the time results found in the previous section.

Fig. 4(b) shows that the training time of MVKSC increases up till $N = 10^3$ and stays more or less the same after this point. This is to be expected since m is at most equal to 10^3 and hence the size of the training dataset does not increase anymore after this point. Another observation is that from $N = 10^3$ on the training time does not seem to visibly influence the total time anymore.

Finally, Fig. 4(c) shows that, although only a subset of the data is taken into account for training, the clustering performance does not seem to suffer from it. The NMI value of the clustering done by MVKSC stays more or less the same from $N = 10^3$ on. It also shows that it is able to outperform all other multi-view methods, even though for $N = 0.5{\cdot}10^4$ and $N = 10^4$ they take more datapoints into account to build up their model.

Next, we ran MVKSC on a real-world large-scale dataset. In the previous section we already looked at results for two subsets of the

Input: Training data $\mathcal{D}^{[v]} = \{\mathbf{x}_i^{[v]}\}_{i=1}^N$, training set size m, kernel function K and a tuning criteria.
Output: Cluster assignment $q_i^{[v]}$ for each point $\mathbf{x}_i^{[v]}$ in the total dataset.
1: Randomly select the same m points from $\mathcal{D}^{[v]}$ for each $v = 1, \ldots, V$ to form the training set $\mathcal{D}_{\text{tr}}^{[v]} = \{\mathbf{x}_i^{[v]}\}_{i=1}^m$.
2: Apply model selection Algorithm 3 ($\mathcal{D}_{\text{tr}}^{[v]}$, $\mathcal{D}^{[v]}$, K,criteria) with the newly formed training set and the total dataset as test set to obtain the cluster assignment of all points.

Algorithm 4. Large-scale MVKSC.

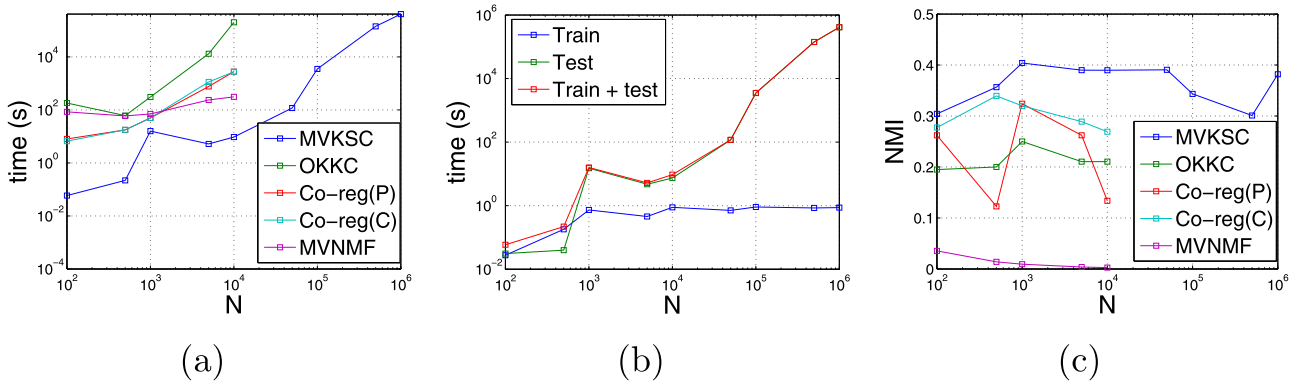


Fig. 4. Results of MVKSC, OKKC, Co-reg(P), Co-reg(C), MVNMF and OKKC for the first synthetic dataset where the number of datapoints is increased from $N = 10^2$ to $N = 10^6$. Fig. (a) shows the time (in seconds) it takes to cluster the entire dataset (training and test time) with regard to the number of datapoints N. Fig. (b) shows the training time ($m = 1000$), test time (for the total N datapoints) and total (training + test) time of MVKSC with regard to N. Fig. (c) shows the NMI value for the clustering done by all 4 methods on the total dataset.

Table 7
Performance results on the large-scale Reuters dataset for three tuning criteria, with the proposed methods. The highest NMI value, and hence the best performing method, is indicated in bold.

Method	Large-scale Reuters		
	Sil	BLF	BAF
Best Single KSC	0.332	0.339	0.331
Feature Concat	0.226	0.231	0.263
Kernel Addition	0.205	0.275	0.255
Kernel Product	0.153	0.244	0.227
MVKSC ($\mathbf{e}^{[v]}$)	0.307	0.340	0.375
MVKSC ($\mathbf{e}_{total}$)	0.370	0.343	**0.385**

Reuters Multilingual dataset. For this purpose we took the largest possible Reuters set, which consists of documents written in German for one view and translation of them in English, French, Spanish and Italian for the other four views. This dataset contains $V = 5$ views with $N = 29,953$ documents each. The dimension of the data over the views range from 11, 547 to 34, 279. The training was done with $m = 1000$ randomly chosen datapoints, where the performance was averaged over three randomizations. Model selection was performed as described in the previous section. Since the multi-view methods OKKC, CoreguSC and MultiNMF do not include an out-of-sample extension we could not run them on this dataset as this resulted in memory problems. Since the three simple coupling mechanisms Feature Concatenation, Kernel Addition and Kernel Product are based on KSC, they include the out-of-sample extension and were hence used to compare our method with. The NMI results on the large-scale dataset Reuters dataset are given in Table 7.

The table shows that MVKSC is able to obtain the best clustering with all tuning criteria. It shows again that MVKSC is able to achieve a better performance than just using the most informative view and that it outperforms the simple coupling schemes. Since the other state-of-the-art multi-view method considered here are not able to do clustering on the full dataset it shows the importance of the out-of-sample extension of MVKSC for large datasets.

6. Conclusion

In this paper Multi-View Kernel Spectral Clustering that exploits information from two or more views when performing clustering is proposed. The formulation is based on a weighted KCCA and is cast in the LS-SVM framework. The coupling of the different views is obtained by an additional coupling term in the primal model which couples the projected values corresponding to the different views. In this way, the information from all views together is already into account early on in the training process, while still allowing for a degree of freedom to model the different views differently and hence combining the advantages of early and late fusion. The aim of this new model is to improve the clustering performance of KSC by incorporating information from multiple views. The model is tested on several datasets and the performance in terms of NMI, ARI and runtime are compared to KSC, some simple coupling mechanisms and three state-of-the-art multi-view clustering methods. The obtained results show the improvement of using multiple views. Furthermore this paper has shown that MVKSC is suitable to handle large-scale datasets because of the out-of-sample extension. The model was tested on a large-scale synthetic and real-world example and was shown to outperform the other considered state-of-the-art multi-view clustering methods in terms of clustering accuracy and runtime.

Acknowledgments

The research leading to these results has received funding from the European Research Council under the European Union Seventh Framework Programme (FP7/2007-2013) / ERC AdG A-DATADRIVE- B (290923). This paper reflects only the authors views and the Union is not liable for any use that may be made of the contained information. Research Council KUL: CoE PFV/10/002 (OPTEC), BIL12/11T; PhD/Postdoc grants Flemish Government: FWO: projects: G.0377.12 (Structured systems), G.088114N (Tensor based data similarity), G0A4917N (Deep restricted kernel machines); PhD/Postdoc grant iMinds Medical Information Technologies SBO 2015 IWT: POM II SBO 100031 Belgian Federal Science Policy Office: IUAP P7/19 (DYSCO, Dynamical systems, control and optimization, 2012–2017).

References

[1] K. Chaudhuri, S.M. Kakade, K. Livescu, K. Sridharan, Multi-view clustering via canonical correlation analysis, International Conference on Machine Learning, (2009), pp. 129–136.
[2] A. Blum, T. Mitchell, Combining labeled and unlabeled data with co-training, Conference on Learning Theory(1998) 92–100.
[3] Y. Yang, C. Lan, X. Li, J. Huan, B. Luo, Automatic social circle detection using multi-view clustering, ACM Conference on Information and Knowledge Management (CIKM), (2014), pp. 1019–1028.
[4] Z.-H. Zhou, K.-J. Chen, Y. Jiang, Exploiting unlabeled data in content-based image retrieval, European Conference on Machine Learning, (2004), pp. 525–536.
[5] J. Du, C.X. Ling, Z.-H. Zhou, When does co-training work in real data? IEEE Trans. Knowl. Data Eng. (TKDE) 23 (2010) 788–799.
[6] J. Zhao, X. Xie, X. Xu, S. Sun, Multi-view learning overview: recent progress and new challenges, Inf. Fusion 38 (2017) 43–54, http://dx.doi.org/10.1016/j.inffus.2017.02.007.
[7] R.D. Zilca, Y. Bistritz, Feature concatenation for speaker identification, 2000 10th European Signal Processing Conference, (2000), pp. 1–4.
[8] S. Yu, L.-C. Tranchevent, X. Liu, W. Glanzel, J.A.K. Suykens, B. De Moor, Y. Moreau, Optimized data fusion for kernel k-means clustering, IEEE Trans. Pattern Anal. Mach. Intell. 34 (5) (2012) 1031–1039.

[9] G. Lin, H. Zhu, X. Kang, C. Fan, E. Zhang, Feature structure fusion and its application, Inf. Fusion 20 (Supplement C) (2014) 146–154, http://dx.doi.org/10.1016/j.inffus.2014.01.002.
[10] A.J. Bekker, M. Shalhon, H. Greenspan, J. Goldberger, Multi-view probabilistic classification of breast microcalcifications, T-MI 35 (2) (2016) 645–653, http://dx.doi.org/10.1109/TMI.2015.2488019.
[11] X. Xie, S. Sun, Multi-view clustering ensembles, Proceedings of the 2013 International Conference on Machine Learning and Cybernetics, (2013), pp. 51–56.
[12] A.K. Jain, Data clustering: 50 years beyond K-means, Pattern Recognit. Lett. 31 (8) (2010) 651–666, http://dx.doi.org/10.1016/j.patrec.2009.09.011.
[13] A. Sharma, K.a. Boroevich, D. Shigemizu, Y. Kamatani, M. Kubo, T. Tsunoda, Hierarchical maximum likelihood clustering approach, IEEE Trans. Biomed. Eng. 64 (1) (2017) 112–122.
[14] A. Sharma, D. Shigemizu, K.a. Boroevich, Y. López, Y. Kamatani, M. Kubo, T. Tsunoda, Stepwise iterative maximum likelihood clustering approach, BMC Bioinformat. 17 (2016) 319–333, http://dx.doi.org/10.1186/s12859-016-1184-5.
[15] E. Elhamifar, R. Vidal, Sparse subspace clustering: algorithm, theory, and applications, IEEE Trans. Pattern Anal. Mach. Intell. 35 (11) (2013) 2765–2781, http://dx.doi.org/10.1109/TPAMI.2013.57.
[16] U. von Luxburg, A tutorial on spectral clustering, Stat. Comput. 17 (4) (2007) 395–416, http://dx.doi.org/10.1007/s11222-007-9033-z.
[17] A.Y. Ng, A.Y. Jordan, Y. Weiss, On spectral clustering: analysis and an algorithm, Adv. Neural Inf. Process. Syst. 2 (2002) 849;–856.
[18] U. von Luxburg, A tutorial on spectral clustering, Stat. Comput. 17(4) (2007) 395;–416.
[19] F.R. Chung, Spectral Graph Theory, Vol. 92 Providence, RI, USA: AMS, 1997.
[20] C. Alzate, J.A.K. Suykens, Multiway spectral clustering with out-of-sample extensions through weighted kernel PCA, IEEE Trans. Pattern Anal. Mach Intell. 32 (2) (2010) 335–347.
[21] J.A.K. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor, J. Vandewalle, Least Squares Support Vector Machines, World Scientific, 2002.
[22] X. Cai, F. Nie, H. Huang, F. Kamangar, Heterogeneous image feature integration via multi-modal spectral clustering, Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), (2011), pp. 1977–1984.
[23] A. Kumar, P. Rai, H. Daume, Co-regularized multi-view spectral clustering, Neural Information Processing Systems 2011(2011) 1413–1421.
[24] R. Xia, Y. Pan, L. Du, J. Yin, Robust multi-view spectral clustering via low-rank and sparse decomposition, AAAI Conference on Artificial Intelligence, (2014), pp. 2149–2155.
[25] Y. Li, F. Nie, H. Huang, J. Huang, Large-scale multi-view spectral clustering via Bipartite graph, Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, (2011), pp. 2750–2756.
[26] R. Mall, R. Langone, J.A.K. Suykens, Kernel spectral clustering for big data networks, Entropy 15 (2013) 1567–1586.
[27] R. Langone, J.A.K. Suykens, Fast kernel spectral clustering, Neurocomputing 268 (2017) 27–33.
[28] H. Hotelling, Relations between two sets of variates, Biometrica 28 (1936) 321–377.
[29] P.L. Lai, C. Fyfe, Kernel and nonlinear canonical correlation analysis, Int. J. Neural Syst. 10(5) (2000) 365–;377.
[30] F.R. Bach, M.I. Jordan, Kernel independent component analysis, J. Mach. Learn. Res. 3 (2002) 1;–48.
[31] T. Van Gestel, J.A.K. Suykens, J. De Brabanter, B. De Moor, J. Vandewalle, Kernel canonical correlation analysis and least squares support vector machines, Proc. of the International Conference on Artificial Neural Networks (ICANN 2001), (2001), pp. 381–386.
[32] M. Meila, J. Shi, A random walks view of spectral segmentation, Proceedings of the Eighth International Workshop on Artificial Intelligence and Statistics (AISTATS), (2001).
[33] P.J. Rousseeuw, Silhouettes: a graphical aid to the interpretation and validation of cluster analysis, J. Comput. Appl Math 20 (1987) 53–65.
[34] C.M. Bishop, Neural Networks for Pattern Recognition, Oxford University Press, 1995.
[35] X. Yi, Y. Xu, C. Zhang, Multi-view EM algorithm for finite mixture models, International Conference on Advances in Pattern Recognition, 3686 (2005), pp. 420–425.
[36] M.-R. Amini, N. Usunier, C. Goutte, Learning from multiple partially observed views - an application to multilingual text categorization, Adv. Neural Inf. Process. Syst. (2009) 28–36.
[37] J. Liu, C. Wang, J. Gao, J. Han, Multi-view clustering via joint nonnegative matrix factorization, Proc. SIAM Data Mining Conference (SDM'13), (2013), pp. 252–260.
[38] D. Greene, P. Cunningham, A matrix factorization approach for integrating multiple data views, European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, (2009), pp. 423;–438.
[39] C.-W. Hsu, C.-C. Chang, C.-J. Lin, A practical guide to support vector classification, Technical report, University of National Taiwan, Department of Computer Science and Information Engineering(2003) 1–12.
[40] P. Li, G. Samorodnitsky, J. Hopcroft, Sign cauchy projections and chi-square kernel, Adv. Neural Inf. Process. Syst. 26 (2013) 2571–2579.
[41] H. Lodhi, C. Saunders, J. Shawe-Taylor, N. Cristianini, C. Watkins, Text classification using string kernels, J. Mach. Learn. Res. 2 (2002) 419–444.
[42] A. Strehl, J. Ghosh, Cluster ensembles - a knowledge reuse framework for combining multiple partitions, J. Mach. Learn. Res. 3 (2002) 583;–617.
[43] L. Hubert, P. Arabie, Comparing partitions, J. Classification 2 (1) (1985) 193–218, http://dx.doi.org/10.1007/BF01908075.
[44] D. Kuang, S. Yun, H. Park, SymNMF: nonnegative low-rank approximation of a similarity matrix for graph clustering, J. Global Optim. 62 (2015) 545–574, http://dx.doi.org/10.1007/s10898-014-0247-2.
[45] R. Langone, R. Mall, C. Alzate, J.A.K. Suykens, Kernel spectral clustering and applications, in: M.E. Celebi, K. Aydin (Eds.), Unsupervised Learning Algorithms, Springer International Publishing, 2016, pp. 135–161.