Unpaired Multi-View Kernel Spectral Clustering

Lynn Houthuys and Johan A. K. Suykens

Department of Electrical Engineering ESAT-STADIUS, KU Leuven
Kasteelpark Arenberg 10 B-3001 Leuven, Belgium
Email: {lynn.houthuys, johan.suykens}@esat.kuleuven.be

***Abstract*—In multi-view learning, data is described through multiple representations or views. Multi-view learning methods aim to improve the performance of using only one view, by incorporating the information of all available views. A common assumption is that the data is paired, which could be over-rigorous in certain applications. This paper introduces an unpaired multi-view clustering model called Unpaired Multi-View Kernel Spectral Clustering (UPMVKSC) which performs multi-view clustering when there is no information about which points in the different views represent the same object. The information that is included, is in the form of pairwise inter- and intra-view constraints. The proposed model is tested on four different datasets and the experimental results demonstrate the effectiveness of our model and the behavior with respect to the number of constraints used.**

I. Introduction

Many real-world applications in machine learning involve datasets that are comprised of different sources of features or *views*. The different views each represent the dataset in a different way. Think for example of classifying images based on the pixel values and the associated captions [1] or learning from videos based on some text, audio and video features [2], and others. Instead of only looking at one representation of the data, existing multi-view methods exploit the information from all views to increase the learning performance.

A common assumption among these methods is that the data is *paired*, which means that for every datapoint in one view, the corresponding datapoints in all other views which represent the same object are known. So for example, when classifying figures based on the pixel arrays and captions, it is known which caption corresponds to which pixel array.

In some real-world problems this assumption is not met. For example Gu et al. [3] proposes learning from data coming from a wireless sensor network, where the data could be missing or polluted during data transmission. Hence there is always a part of the data that is unpaired. A number of the classical multi-view techniques are adjusted to deal with this kind of *semi-paired* data. For example Blaschko et al. [4], Kimura et al. [5], Zhang et al. [6] and Zhou et al. [7] have extended Canonical Correlation Analysis (CCA) to perform semi-paired multi-view clustering.

In some applications, data might even be completely *unpaired*. Consider for example a learning task based on web page data coming from English and French routers, where it might not be known which English web page corresponds to which French one [8]. In these applications there might however be some other side information available based on pairwise inter- and intra-view constraints. Pairwise intra-view constraints are constraints between two points belonging to the same view. The constraint can indicate that both datapoints belong to the same class or cluster (*must-link* constraint) or that they belong to a different class or cluster (*cannot-link* constraint). Pairwise inter-view constraints are constraints between two points which belong to a different view. Notice that a must-link inter-view constraint between two points is not the same as them being paired. Two points who belong to the same class or cluster do not necessarily represent the same object.

Pairwise constraints have been used for single-view learning, for example by Yan et al. [9] for classification and by Alzate and Suykens [10] for clustering. Eaton et al. [11] perform semi-paired multi-view clustering where the intra-view constraints of each view are propagated to the other views. Qian et al. [8] perform unpaired semi-supervised multi-view classification using pairwise inter-view constraints. The model is only defined for two views and two classes. Lu and Peng [12] propose a model to do unpaired semi-supervised multi-view classification and cross-view retrieval with pairwise inter- and intra-view constraints. The model is again only defined for two views.

In this paper a novel unpaired multi-view clustering method with pairwise inter- and intra-view constraints is proposed. The model is called Unpaired Multi-View Kernel Spectral Clustering (UPMVKSC) and it can be seen as an extension of Multi-View Kernel Spectral Clustering (MVKSC) [13]. In contrast to the previous discussed unpaired multi-view models, this model is defined for two or more views and the learning task (clustering) is unsupervised. UPMVKSC is set in the primal-dual setting typical to Least Squares Support Vector Machines [14], where the primal formulations includes a coupling term which maximizes the correlation of the score variables of two points for which a must-link constraint is defined, and minimizes the correlation of the score variables of two points on which a cannot-link is defined.

This paper shows the behavior of UPMVKSC with respect to the number of constraints and it shows that unpaired multi-view clustering can outperform single-view clustering with a relatively small number of constraints.

We will denote matrices as bold uppercase letters and vectors as bold lowercase letters. The superscript $^{[v]}$ will denote 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 II overviews the necessary background namely, the paired multi-view model MVKSC. Section III introduces the novel unpaired multi-view UPMVKSC model. Section IV discusses the experiments done with UPMVKSC for a different number of constraints. The section further describes the four datasets used and discusses the results. Finally, Section V concludes this work.

II. Background: Multi-View Kernel Spectral Clustering

This section summarizes the Multi-View Kernel Spectral Clustering (MVKSC) [13] model. This model can be described as a weighted form of Kernel Canonical Correlation Analysis (KCCA) [14], where the weights are related to the Kernel Spectral Clustering (KSC) [15] model. The model is used to perform multi-view clustering with paired information. The model is formulated in the primal-dual setting, typical of Least Squares Support Vector Machines (LS-SVM) [14].

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

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

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

where $\mathbf{e}^{[v]^{(l)}} \in \mathbb{R}^{N\times 1}$ are the clustering scores or projections related to the vth view, $l = 1, \ldots, k-1$ indicate the score variables needed to encode k clusters 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 $D_{ii}^{[v]} = \sum_j \boldsymbol{\varphi}^{[v]}(\mathbf{x}_i^{[v]})^T\boldsymbol{\varphi}^{[v]}(\mathbf{x}_j^{[v]})$. $\boldsymbol{\Phi}^{[v]} \in \mathbb{R}^{N\times d_h^{[v]}}$ are the feature matrices with $\boldsymbol{\Phi}^{[v]} = [\boldsymbol{\varphi}^{[v]}(\mathbf{x}_1^{[v]})^T; \ldots; \boldsymbol{\varphi}^{[v]}(\mathbf{x}_N^{[v]})^T]$ where $\boldsymbol{\varphi}^{[v]} : \mathbb{R}^{d^{[v]}} \to \mathbb{R}^{d_h^{[v]}}$ are the mappings to a high (possibly infinite) dimensional feature space

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

$$\hat{\boldsymbol{\mu}}^{[v]} = \frac{1}{\mathbf{1}_N^T\mathbf{D}^{[v]^{-1}}\mathbf{1}_N}\boldsymbol{\Phi}^{[v]^T}\mathbf{D}^{[v]^{-1}}\mathbf{1}_N. \tag{2}$$

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. The coupling term maximizes the correlation between the score variables $e_i^{[v]^{(l)}}$ and $e_i^{[u]^{(l)}}$ for all $l = 1, \ldots, k-1$, $i = 1, \ldots, N$ and different views $v, u = 1, \ldots, V$ with $v \neq u$, since it is known that $e_i^{[v]^{(l)}}$ and $e_i^{[u]^{(l)}}$ represent the same sample. In correspondence to the weighting matrix used in KSC, the coupling matrix also aims 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 $v, u = 1, \ldots, V$ and $v \neq u$.

The dual problem related to this primal formulation is:

$$\begin{bmatrix}\mathbf{0}_N & \mathbf{S}^{[1,2]}\boldsymbol{\Omega_c}^{[2]} & \ldots & \mathbf{S}^{[1,V]}\boldsymbol{\Omega_c}^{[V]} \\ \mathbf{S}^{[2,1]}\boldsymbol{\Omega_c}^{[1]} & \mathbf{0}_N & \ldots & \mathbf{S}^{[2,V]}\boldsymbol{\Omega_c}^{[V]} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{S}^{[V,1]}\boldsymbol{\Omega_c}^{[1]} & \mathbf{S}^{[V,2]}\boldsymbol{\Omega_c}^{[2]} & \ldots & \mathbf{0}_N\end{bmatrix}\begin{bmatrix}\boldsymbol{\alpha}^{[1]^{(l)}} \\ \boldsymbol{\alpha}^{[2]^{(l)}} \\ \vdots \\ \boldsymbol{\alpha}^{[V]^{(l)}}\end{bmatrix}$$

$$= \frac{1}{\rho^{(l)}}\begin{bmatrix}\mathbf{B}^{[1]} & \ldots & \mathbf{0}_N \\ \vdots & \ddots & \vdots \\ \mathbf{0}_N & \ldots & \mathbf{B}^{[V]}\end{bmatrix}\begin{bmatrix}\boldsymbol{\alpha}^{[1]^{(l)}} \\ \vdots \\ \boldsymbol{\alpha}^{[V]^{(l)}}\end{bmatrix} \tag{3}$$

with $\mathbf{B}^{[v]} = \mathbf{I}_N - \frac{\gamma^{[v]^{(l)}}}{N}\mathbf{D}^{[v]^{-1}}\boldsymbol{\Omega_c}^{[v]}$ and $\boldsymbol{\alpha}^{[v]^{(l)}}$ being the dual variables, i.e. the Langrange multipliers related to the constraints in Eq.(1). $\boldsymbol{\Omega_c}^{[v]} = (\boldsymbol{\Phi}^{[v]} - \mathbf{1}_N\hat{\boldsymbol{\mu}}^{[v]^T})(\boldsymbol{\Phi}^{[v]} - \mathbf{1}_N\hat{\boldsymbol{\mu}}^{[v]^T})^T$ are the centered kernel matrices that capture the similarity between data of view v. These centered kernel matrices can be computed by

$$\boldsymbol{\Omega_c}^{[v]} = \mathbf{M_D}^{[v]}\boldsymbol{\Omega}^{[v]}\mathbf{L_D}^{[v]} \tag{4}$$

where $\mathbf{M_D}^{[v]} = \mathbf{I}_N - \frac{1}{\mathbf{1}_N^T\mathbf{D}^{[v]^{-1}}\mathbf{1}_N}\mathbf{1}_N\mathbf{1}_N^T\mathbf{D}^{[v]^{-1}}$ and $\mathbf{L_D}^{[v]} = \mathbf{I}_N - \frac{1}{\mathbf{1}_N^T\mathbf{D}^{[v]^{-1}}\mathbf{1}_N}\mathbf{D}^{[v]^{-1}}\mathbf{1}_N\mathbf{1}_N^T$ are centering matrices and where $\boldsymbol{\Omega}^{[v]} = \boldsymbol{\Phi}^{[v]}\boldsymbol{\Phi}^{[v]^T}$ are the kernel matrices which are defined as

$$\begin{aligned}\Omega_{ij}^{[v]} &= \boldsymbol{\varphi}^{[v]}(\mathbf{x}_i^{[v]})^T\boldsymbol{\varphi}^{[v]}(\mathbf{x}_j^{[v]}) \\ &= K^{[v]}(\mathbf{x}_i^{[v]}, \mathbf{x}_j^{[v]}).\end{aligned} \tag{5}$$

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 projections are used to determine the clustering. The authors in [13] argue that for MVKSC this can be done in two ways, either together on all views where they use a (weighted) average of the score variables to determine the clustering, or for each view separately which entails that the clustering can be slightly different for each view. To determine the clustering the score variables are binarized, such that $[\text{sign}(e_i^{[v]^{(1)}})), \ldots, \text{sign}(e_i^{[v]^{(k-1)}})]$ forms the encoding vector for datapoint $\mathbf{x}_i^{[v]}$ belonging to view v. The k most occurring encoding vectors then form the codebook $\mathcal{C}^{[v]} = \{c_p^{[v]}\}_{p=1}^k$ where each codeword $c_p^{[v]} \in \{-1, 1\}^{k-1}$ represent a cluster.

For each datapoint the binarized score variables are compared with the codebook and the nearest codeword in terms of Hamming distance is selected. When it is chosen to use a (weighted) average of the score variables each datapoint has one encoding vector over all views and there is hence only one codebook for all views.

A major advantage of MVKSC over other popular multi-view clustering methods is the out-of-sample extension. The score variables of test datapoints can be calculated as follows:

$$\mathbf{e}_{\text{test}}^{[v]^{(l)}} = \mathbf{\Omega}_{\mathbf{c}_{\text{test}}}^{[v]} \boldsymbol{\alpha}^{[v]^{(l)}} \tag{6}$$

where $l = 1, \ldots, k-1$ and $v = 1, \ldots, V$. $\mathbf{\Omega}_{\mathbf{c}_{\text{test}}}^{[v]} \in \mathbb{R}^{N_{\text{test}} \times N}$ are the centered kernel matrices evaluated using the test data with $\mathbf{\Omega}_{\mathbf{c}_{\text{test}}}^{[v]} = \mathbf{M}_{\mathbf{D}}^{[v]}{}_{\text{test}} \mathbf{\Omega}_{\text{test}}^{[v]} \mathbf{L}_{\mathbf{D}}{}^{[v]}$. The test kernel matrix is computed as $\mathbf{\Omega}_{\text{test}}^{[v]} = K^{[v]}(\mathbf{x}_{\text{test}_i}^{[v]}, \mathbf{x}_j^{[v]})$. So the eigenvalue problem in Eq.(3) does not need to be computed for test datapoints.

III. Unpaired Multi-View Kernel Spectral Clustering

This section introduces the Unpaired Multi-View Kernel Spectral Clustering (UPMVKSC) model with pairwise inter- and intra-view constraints. The constraint can be of type must-link or cannot-link, where a must-link constraint between two points means that they should belong to the same cluster, and a cannot-link constraint restricts the two points to be in different clusters. In MVKSC, the correlation between the weighted score variables of paired datapoints are maximized in the coupling term. We apply this same idea to the unpaired version by introducing a coupling term that maximizes the correlation between the weighted score variables of two points if there is a must-link constraint between them and minimizes it if there is a cannot-link between them.

A. Model formulation

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

$$\begin{aligned} \min_{\mathbf{w}^{[v]^{(l)}}, \mathbf{e}^{[v]^{(l)}}} & \frac{1}{2} \sum_{v=1}^{V} \sum_{l=1}^{k-1} \mathbf{w}^{[v]^{(l)T}} \mathbf{w}^{[v]^{(l)}} \\ & - \sum_{v=1}^{V} \frac{1}{2N^{[v]}} \sum_{l=1}^{k-1} \gamma^{[v]^{(l)}} \mathbf{e}^{[v]^{(l)T}} \mathbf{D}^{[v]^{-1}} \mathbf{e}^{[v]^{(l)}} \\ & - \frac{1}{2} \sum_{v,u=1}^{V} \sum_{l=1}^{k-1} \rho^{(l)} \, \mathbf{e}^{[v]^{(l)T}} \mathbf{S}^{*[v,u]} \mathbf{e}^{[u]^{(l)}} \end{aligned} \tag{7}$$

$$\text{s.t.} \begin{cases} \mathbf{e}^{[1]^{(l)}} = (\mathbf{\Phi}^{[1]} - \mathbf{1}_{N^{[1]}} \hat{\boldsymbol{\mu}}^{[1]^T}) \mathbf{w}^{[1]^{(l)}} \\ \qquad \vdots \\ \mathbf{e}^{[V]^{(l)}} = (\mathbf{\Phi}^{[V]} - \mathbf{1}_{N^{[V]}} \hat{\boldsymbol{\mu}}^{[V]^T}) \mathbf{w}^{[V]^{(l)}} \\ \qquad \text{where } l = 1, \ldots, k-1 \end{cases}$$

At first glance, this formulation looks very similar to the primal MVKSC formulation in Eq.(1), however, the coupling term holds some key differences. The coupling matrix $\mathbf{S}^{*[v,u]} \in \mathbb{R}^{N^{[v]} \times N^{[u]}}$ for views $v, u = 1, \ldots, V$ is defined as

$$\mathbf{S}^{*[v,u]} = \mathbf{D}^{[v]^{-\frac{1}{2}}} \Upsilon^{[v,u]} \mathbf{D}^{[u]^{-\frac{1}{2}}} \tag{8}$$

where

$$\Upsilon_{ij}^{[v,u]} = \begin{cases} 1 & \text{if } \mathbf{x}_i^{[v]} \text{ and } \mathbf{x}_j^{[u]} \text{ must-link,} \\ -1 & \text{if } \mathbf{x}_i^{[v]} \text{ and } \mathbf{x}_j^{[u]} \text{ cannot-link,} \\ 0 & \text{if } \mathbf{x}_i^{[v]} \text{ and } \mathbf{x}_j^{[u]} \text{ unknown link.} \end{cases} \tag{9}$$

By introducing the matrix $\Upsilon^{[v,u]}$ in the coupling term, we enforce that the correlation between the weighted score variables of two points is maximized if there is a must-link constraint between them, hence if they belong to the same cluster and that the correlation between the weighted score variables of two points is minimized if there is a cannot-link constraint between them, hence if they belong to a different cluster.

This formulation entails two other key differences with the MVKSC model, namely the possibility for a different dataset size $N^{[v]}$ for each view and the introduction of coupling between points in the same view.

Since the views are not coupled, it is clearly possible that the number of available datapoints in each view is different. Therefore the coupling matrix $\mathbf{S}^{*[v,u]}$ is not necessarily square. This matrix is hence also not always symmetric, in contrast to the coupling matrix in the MVKSC model, but it does hold that $\mathbf{S}^{*[v,u]T} = \mathbf{S}^{*[u,v]}$ and that $\mathbf{e}^{[v]^{(l)T}} \mathbf{S}^{*[v,u]} \mathbf{e}^{[u]^{(l)}} = \mathbf{e}^{[u]^{(l)T}} \mathbf{S}^{*[u,v]} \mathbf{e}^{[v]^{(l)}}$ for every two views v, u. Notice that the intra-view coupling matrix $\mathbf{S}^{*[v,v]}$ is square and symmetric.

While the MVKSC model only allows for coupling between different views, the coupling term in the UPMVKSC model allows for inter-, as well as intra-, view coupling. So if there exist some prior knowledge about whether two points in the same view belong to the same cluster or not, this is straightforward incorporated in the model. This also entails the default constraints $\mathbf{S}^{*[v,v]}_{ii} = 1$ for $v = 1, \ldots, V$ and $i = 1, \ldots, N^{[v]}$.

The Lagrangian of the primal problem is:

$$\begin{aligned} \mathcal{L}(\mathbf{w}^{[v]^{(l)}}, \mathbf{e}^{[v]^{(l)}}; \boldsymbol{\alpha}^{[v]^{(l)}}) &= \frac{1}{2} \sum_{v=1}^{V} \sum_{l=1}^{k-1} \mathbf{w}^{[v]^{(l)T}} \mathbf{w}^{[v]^{(l)}} \\ & - \sum_{v=1}^{V} \frac{1}{2N^{[v]}} \sum_{l=1}^{k-1} \gamma^{[v]^{(l)}} \mathbf{e}^{[v]^{(l)T}} \mathbf{D}^{[v]^{-1}} \mathbf{e}^{[v]^{(l)}} \\ & - \frac{1}{2} \sum_{v,u=1}^{V} \sum_{l=1}^{k-1} \rho^{(l)} \, \mathbf{e}^{[v]^{(l)T}} \mathbf{S}^{*[v,u]} \mathbf{e}^{[u]^{(l)}} \\ & + \sum_{v=1}^{V} \sum_{l=1}^{k-1} \boldsymbol{\alpha}^{[v]^{(l)T}} (\mathbf{e}^{[v]^{(l)}} - (\mathbf{\Phi}^{[v]} - \mathbf{1}_{N^{[v]}} \hat{\boldsymbol{\mu}}^{[v]^T}) \mathbf{w}^{[v]^{(l)}}). \end{aligned} \tag{10}$$

The KKT optimality conditions are:

$$\begin{cases} \frac{\partial \mathcal{L}}{\partial \mathbf{w}^{[v]^{(l)}}} = 0 \rightarrow \mathbf{w}^{[v]^{(l)}} = (\mathbf{\Phi}^{[v]} - \mathbf{1}_{N^{[v]}} \hat{\boldsymbol{\mu}}^{[v]^T})^T \boldsymbol{\alpha}^{[v]^{(l)}}, \\ \frac{\partial \mathcal{L}}{\partial \mathbf{e}^{[v]^{(l)}}} = 0 \rightarrow \boldsymbol{\alpha}^{[v]^{(l)}} = \frac{\gamma^{[v]^{(l)}}}{N^{[v]}} \mathbf{D}^{[v]^{-1}} \mathbf{e}^{[v]^{(l)}} \\ \qquad\qquad + \rho^{(l)} \sum_{u=1}^{V} \mathbf{S}^{*[v,u]} \mathbf{e}^{[u]^{(l)}}, \\ \frac{\partial \mathcal{L}}{\partial \boldsymbol{\alpha}^{[v]^{(l)}}} = 0 \rightarrow \mathbf{e}^{[v]^{(l)}} = (\mathbf{\Phi}^{[v]} - \mathbf{1}_{N^{[v]}} \hat{\boldsymbol{\mu}}^{[v]^T}) \mathbf{w}^{[v]^{(l)}}, \\ \qquad\qquad \text{where } v = 1, \ldots, V \text{ and } l = 1, \ldots, k-1. \end{cases} \tag{11}$$

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

$$\begin{bmatrix} \mathbf{S}^{*[1,1]} \mathbf{\Omega_c}^{[1]} & \ldots & \mathbf{S}^{*[1,V]} \mathbf{\Omega_c}^{[V]} \\ \vdots & \ddots & \vdots \\ \mathbf{S}^{*[V,1]} \mathbf{\Omega_c}^{[1]} & \ldots & \mathbf{S}^{*[V,V]} \mathbf{\Omega_c}^{[V]} \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha}^{[1]^{(l)}} \\ \vdots \\ \boldsymbol{\alpha}^{[V]^{(l)}} \end{bmatrix} \tag{12}$$

$$= \frac{1}{\rho^{(l)}} \begin{bmatrix} \mathbf{B}^{[1]} & \ldots & \mathbf{0}_{N^{[1]}} \\ \vdots & \ddots & \vdots \\ \mathbf{0}_{N^{[V]}} & \ldots & \mathbf{B}^{[V]} \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha}^{[1]^{(l)}} \\ \vdots \\ \boldsymbol{\alpha}^{[V]^{(l)}} \end{bmatrix}.$$

The matrices $B^{[v]}$ and the kernel matrices are defined in the same manner as for MVKSC.

Instead of explicitly defining the (possibly infinite) feature maps, the kernel matrices are computed through kernel functions as

$$\begin{aligned} \Omega_{ij}^{[v]} &= \boldsymbol{\varphi}^{[v]}(\mathbf{x}_i^{[v]})^T \boldsymbol{\varphi}^{[v]}(\mathbf{x}_j^{[v]}) \\ &= K^{[v]}(\mathbf{x}_i^{[v]}, \mathbf{x}_j^{[v]}). \end{aligned} \tag{13}$$

The kernel functions $K^{[v]} : \mathbb{R}^{d^{[v]}} \times \mathbb{R}^{d^{[v]}} \rightarrow \mathbb{R}$ are similarity functions and have to be positive definite. The degree matrix $\mathbf{D}^{[v]}$ can hence also be computed trough these kernel functions as $D_{ii}^{[v]} = \sum_j K^{[v]}(\mathbf{x}_i^{[v]}, \mathbf{x}_j^{[v]})$. The degree matrix can hence be interpreted as the similarity degree of each point with respect 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 $\frac{1}{\rho^{(l)}}$, and $\gamma^{[v]^{(l)}}$ are the view-specific parameters to be tuned. The constraints are given to the model trough the matrices $\Upsilon^{[v,u]}$ for each $v, u = 1, \ldots, V$.

As for KSC and MVKSC, the final clustering of a datapoint $\mathbf{x}_i^{[v]}$ for $v = 1, \ldots, V$ and $i = 1, \ldots, N^{[v]}$ is determined by the binarized score variables $\text{sign}(e_i^{[v]^{(1)}}), \ldots, \text{sign}(e_i^{[v]^{(k-1)}})$ which are computed as

$$\mathbf{e}^{[v]^{(l)}} = \mathbf{\Omega_c}^{[v]} \boldsymbol{\alpha}^{[v]^{(l)}}. \tag{14}$$

The model also contains an out-of-sample extension, where the score variables of test datapoints $\mathcal{D}_{\text{test}}^{[v]} = \{\mathbf{x}_{\text{test}_i}^{[v]}\}_{i=1}^{N_{\text{test}}^{[v]}} \in \mathbb{R}^{d^{[v]}}$ for $v = 1, \ldots, V$ can be computed as

$$\mathbf{e}_{\text{test}}^{[v]^{(l)}} = \mathbf{\Omega}_{\mathbf{c}_{\text{test}}}^{[v]} \boldsymbol{\alpha}^{[v]^{(l)}} \tag{15}$$

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

$$\mathbf{\Omega}_{\mathbf{c}_{\text{test}}}^{[v]} = \mathbf{M}_{\mathbf{D}}{}_{\text{test}}^{[v]} \mathbf{\Omega}_{\text{test}}^{[v]} \mathbf{L}_{\mathbf{D}}{}^{[v]}. \tag{16}$$

The test kernel matrix can be computed as

$$\mathbf{\Omega}_{\text{test}}^{[v]} = K^{[v]}(\mathbf{x}_{\text{test}_i}^{[v]}, \mathbf{x}_j^{[v]}). \tag{17}$$

The computational complexity of the UPMVKSC model is dominated by solving the eigenvalue problem in Eq.(12), where the left-hand side matrix is of size $(\sum_{v=1}^{V} N^{[v]} \times \sum_{v=1}^{V} N^{[v]})$. For large datasets this could be hard to compute. Because of the out-of-sample extension, however, this can be avoided by only training on a smaller subset of the data and using Eq.(15) to compute the clustering for the remaining datapoints. This approach was followed by Mall et al.[16] for KSC and by Houthuys et al.[13] for MVKSC to cluster large datasets and can be easily applied to the UPMVKSC model as well.

B. Model selection

The results obtained by UPMVKSC depend on the choice of the kernel function and its parameters and on the choice of the regularization parameters.

In correspondence to the experimental setup in the work of Houthuys et al.[13] it was chosen to take the same parameter for each binary cluster problem, hence $\gamma^{[v]} = \gamma^{[v]^{(1)}} = \ldots = \gamma^{[v]^{(k-1)}}$, and to take the same kernel function for each view, in order to decrease tuning complexity.

The tuning of the kernel and regularization parameters is done by simulated annealing. Notice that since the the method is unsupervised, model selection is performed without knowing the true underlying clustering (labels). Hence, the tuning criteria has to be unsupervised as well.

The single-view benchmark model KSC is tuned by means of Balanced Line Fit (BLF) [15] which is an average measure of collinearity and balance. The criteria contains a user-defined parameter η which controls the importance given to the collinearity and the balance. The default chose is $\eta = 0.75$ which entails that more importance is put on collinearity then balance, and will be chosen in the experiments described here. The UPMVKSC model is tuned by means of constrained BLF (cBLF) [10], an modification of BLF where a third term is introduced that measures the accuracy of fulfilling pairwise constraints. This third term is called the constraint fit and is defined as the ratio of fulfilled constraints with respect to the total number of constraints. This criteria contains three user-defined parameters η_l, η_b and η_c, each controlling the importance of the collinearity, balance and constraint fit respectively. The default chose, which will again be used in this paper, is $\eta_l = 0.5, \eta_b = 0.2, \eta_c = 0.3$. cBLF is

evaluated for each view and the mean value is taken as the total performance.

IV. Experiments

This section describes the experimental setup and results performed with the proposed UPMVKSC model.

In order to see the improvement of using constraints we used a similar experimental setup as Qian et al.[8]. We varied the number of constraints ranging from 0 to $10\%(\bar{N})$ with an interval of $1\%(\bar{N})$ and from 0 to $100\%(\bar{N})$ with an interval of $10\%(\bar{N})$, where $\bar{N}$ is the mean dataset size over all views, hence $\bar{N} = 1/V \sum_{v=1}^{V} N^{[v]}$. The constraints were randomly chosen from the entire pool of different inter- and intra-view constraints possible. If a certain constraint between $x_i^{[v]}$ and $x_j^{[u]}$, with $v, w = 1, \ldots, V$, $i = 1, \ldots, N^{[v]}$ and $j = 1, \ldots, N^{[u]}$ is selected, $\Upsilon_{ij}^{[v,u]}$ is adapted as well as $\Upsilon_{ji}^{[u,v]}$ without counting is as a second constraint. The default entries $\Upsilon_{ii}^{[v,v]} = 1$ are not counted in the number of constraints. Notice that the maximum number of different constraints is $\binom{\sum_{v=1}^{V} N^{[v]}}{2}$ so even when using $100\%(\bar{N})$ constraints, we only use a small fraction of all possible constraints.

The experiments are performed for must-link and cannot-link constraints, and for only must-link constraints.

A. Data

A brief description of the four multi-view datasets used is given here. The important statistics of them are summarized in Table I.

- **Synthetic dataset:** This synthetic multi-view dataset was also used by Kumar et al. [17] and consists of three views where each view is generated by a two component Gaussian mixture model from which 2000 points are randomly sampled for each view. 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

$$\mathbf{\Sigma}_1^{[1]} = \begin{pmatrix} 1 & 0.5 \\ 0.5 & 1.5 \end{pmatrix}, \mathbf{\Sigma}_2^{[1]} = \begin{pmatrix} 0.3 & 0.2 \\ 0.2 & 0.6 \end{pmatrix},$$
$$\mathbf{\Sigma}_1^{[2]} = \begin{pmatrix} 1 & -0.2 \\ -0.2 & 1 \end{pmatrix}, \mathbf{\Sigma}_2^{[2]} = \begin{pmatrix} 0.6 & 0.1 \\ 0.1 & 0.5 \end{pmatrix},$$
$$\mathbf{\Sigma}_1^{[3]} = \begin{pmatrix} 1.2 & 0.2 \\ 0.2 & 1 \end{pmatrix}, \mathbf{\Sigma}_2^{[3]} = \begin{pmatrix} 1 & 0.4 \\ 0.4 & 0.7 \end{pmatrix}.$$

- **Reuters Multilingual dataset:** This real-world dataset is described by Amini et al. [18] and consist of documents written in five different languages (English, French, German, Spanish and Italian) and their translations in each of the other four languages. The documents report over 6 common categories and they are represented in bag-of-words style. In this paper the subset of English documents and their French translations is used, and from each cluster 300 documents were randomly sampled.
- **3-Sources Text dataset:** This third dataset is collected by Greene and Cunningham [19] and consist of news stories from three online news sources: BBC, Reuters and The Guardian. The dataset contains 169 news articles reported by all three sources from the periods February to April 2009. Each story was manually annotated with one of the six topical labels: business, entertainment, health, politics, sport and technology.
- **YouTube Video Games dataset:** The dataset proposed by Madani et al. [2] contains three high-level feature families: textual, visual and auditory features describing YouTube videos of video games. In this paper one textual feature (Latent Dirichlet Allocation ran on all of description, title, and tags) and one visual feature (motion feature trough cuboid interest point detection, for more details see the work of Yang and Toderici [20]) is chosen. From each of the seven most occurring labels (excluding the last label, since these datapoints represent videos not belonging to any of the other 30 clusters) 300 videos were randomly sampled.

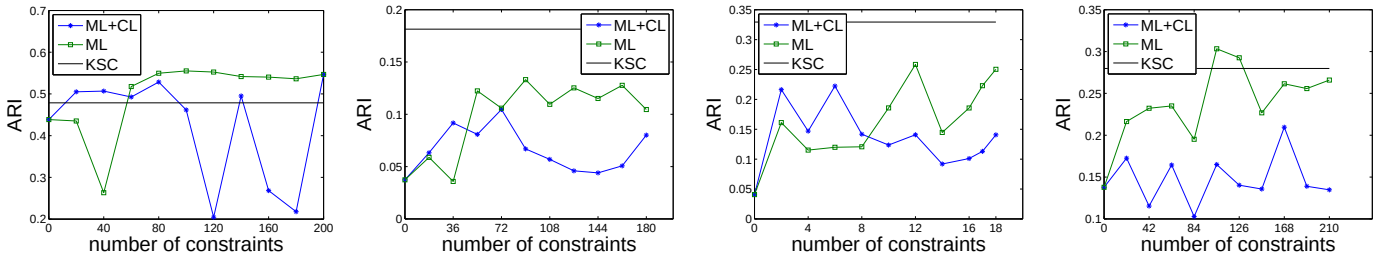
TABLE I: Details of the datasets used in the experiments.

Dataset	$\bar{N}$	V	Dimensions
Synthetic data	2000	3	$d^{[1]} = 2$ $d^{[2]} = 2$ $d^{[3]} = 2$
Reuters	1800	2	$d^{[1]} = 21531$ $d^{[2]} = 24892$
3-Sources	169	3	$d^{[1]} = 3560$ $d^{[2]} = 3631$ $d^{[3]} = 3068$
YouTube Video Games	2100	2	$d^{[1]} = 1000$ $d^{[2]} = 512$

For the synthetic dataset the radial basis function (RBF) kernel is chosen, so the corresponding kernel function is $K(\mathbf{x}_i^{[v]}, \mathbf{x}_j^{[v]}) = \exp\left(-\frac{||\mathbf{x}_i^{[v]} - \mathbf{x}_j^{[v]}||^2}{2\sigma^2}\right)$ for $v = 1, \ldots, V$ and where σ is a kernel parameter to be tuned. Since the real world datasets are very high dimensional using an RBF kernel, bringing the data to a even higher feature space, is not recommended [21]. Therefore a normalized linear kernel is considered for these datasets. So the proposed kernel function for these datasets is $K(\mathbf{x}_i^{[v]}, \mathbf{x}_j^{[v]}) = \frac{\mathbf{x}_i^{[v]^T} \mathbf{x}_j^{[v]}}{\sqrt{(\mathbf{x}_i^{[v]^T} \mathbf{x}_i^{[v]})\ (\mathbf{x}_j^{[v]^T} \mathbf{x}_j^{[v]})}}$ for $v = 1, \ldots, V$. Other appropriate kernel functions for high dimensional data such as Chi-square kernels [22] and String kernels [23] were not considered.

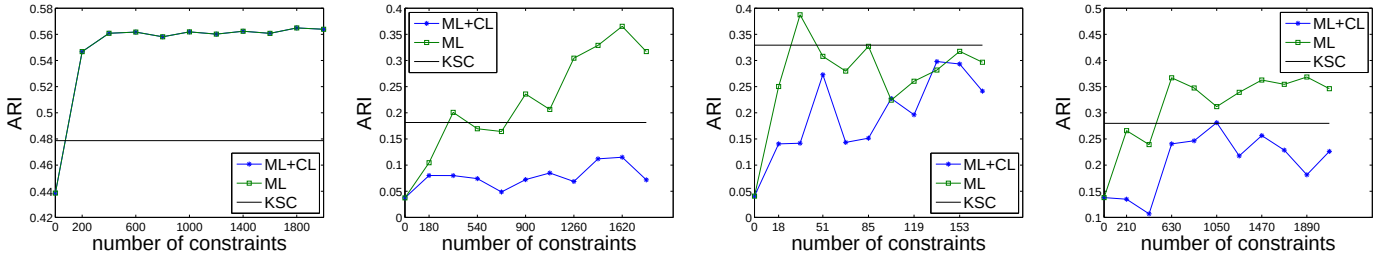
B. Results

Figure 1 and Figure 2 show the experimental results on all four datasets. The performance is measured by the adjusted rand index (ARI) [24], where the mean ARI value over all views is reported. Notice that ARI is a supervised criteria and is hence only used to test the model on test data, and not during training or tuning of the parameters. The performance of UPMVKSC is evaluated when must-link and cannot-link constraints (*ML+CL* in Figure 1 and 2) and when only must-link constraints (*ML* in Figure 1 and 2) are considered, for



(a) Synthetic data (b) Reuters (c) 3-Sources (d) YouTube Video Games

Fig. 1: Performance of UPMVKSC for different number of constraints ranging from 0 to $10\%(\bar{N})$ with an interval of $1\%(\bar{N})$, measured by the ARI value on test data. The performance of KSC is included as a baseline.



(a) Synthetic data (b) Reuters (c) 3-Sources (d) YouTube Video Games

Fig. 2: Performance of UPMVKSC for different number of constraints ranging from 0 to $100\%(\bar{N})$ with an interval of $10\%(\bar{N})$, measured by the ARI value on test data. The performance of KSC is included as a baseline.

different amount of constraints. The UPMVKSC performance is compared to the mean performance of KSC over all views.

Figure 1 shows that only using $10\%(\bar{N})$ or less constraints is usually not enough to achieve a similar performance than KSC. Only for the synthetic datasets and the YouTube Videa Games dataset the UPMVKSC is able to obtain a better results than KSC with very few constraints. Although the figure shows that using more constraints usually leads to a better performance, especially when only using must-link constraints. Sometimes more constraints lead to worse results then using no constraints at all. This could be because of overfitting on the constraints and not being able to generalise well.

These conclusions are more clear when using more constraints as is visible in Figure 2. The UPMVKSC model shows a clear improvement when using more constraints (only for $10\%(\bar{N})$ and $20\%(\bar{N})$ constraints for the YouTube Video Games dataset the performance drops a little with respect to using no constraints). This is again more visible when using only must-link constraints. When using more constraints the UPMVKSC model is able to outperform KSC on all datasets.

While there is usually a big difference in performance with using must-link and cannot-link constraints or only using must-link constraints, this is not the case for the synthetic dataset, as can be seen in Figure 2a. We can see here a big jump in performance between using 0 and $10\%(\bar{N})$ constraints, and almost identical results for using both types of constraints or only must-link. When we look at the results for less constraints, as visualised in Figure 1a, we do see a difference in which types of constraints are used. We can see that when using both types, UPMVKSC is allready able to outperform KSC when using $1\%(\bar{N}) = 20$ constraints, however this is not really stable until using $10\%(\bar{N})$ constraints. When using only ML constraints, however, a clear improvement is seen when using more constraints and from $3\%(\bar{N}) = 60$ on, the results are better than with KSC.

Figure 1b shows that when using a very small set of constraints, UPMVKSC is not able to outperform KSC on the Reuters dataset. It does however shows an improvement when using more constraints, especially visible when using only must-link constraints. When using a little more constraints, Figure 2b does show that UPMVKSC is able to outperform KSC with a rather large difference of almost 0.2 in ARI. This is however only the case when using only must-link constraints.

For the 3-Sources dataset (Figure 1c and 2c) the improvements of using more constraints is less clear than for the other datasets. UPMVKSC is able to outperform KSC when using $20\%(\bar{N}) = 34$ constraints, but the performance drops again when using more constraints.

Finally, for the YouTube Video Games dataset, the performance of using only must-link constraints or using both types is very different as can be seen in Figure 1d and 2d. While the performance of UPMVKSC with both types of constraints is lower than using no constraints for a couple of number of constraints, and it only slightly improves on using KSC for $50\%(\bar{N}) = 1050$ constraints, the results when using only

must-link constraints are much better. When using only must-link constraints the performance grows steadily when using more and more views and even outperforms KSC when using only $5\%(\bar{N}) = 105$ and $6\%(\bar{N}) = 126$ constraints and it outperforms KSC consistently from using $630\%(\bar{N}) = 1050$ constraints on.

V. Conclusion

We have studied the problem of unpaired multi-view clustering with pairwise inter- and intra-view constraints. To do so, we proposed the Unpaired Multi-View Kernel Spectral Clustering (UPMVKSC) model which includes a coupling term that maximizes the correlation of the score variables for two points with a must-link constraint and that minimizes the correlation of the score variables for two points with a cannot-link constraint. To the authors knowledge there has not been any other study to this particular problem.

Experimental results on four multi-view datasets indicate that the proposed model is suitable for unpaired multi-view clustering and that UPMVKSC can improve the results of the single-view method KSC. The results further show that even a relatively small set of constraints can increase the performance of UPMVKSC and using only must-link constraints seem to be preferable to using both types of constraints.

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] T. Kolenda, L. K. Hansen, J. Larsen, and O. Winther, "Independent component analysis for understanding multimedia content," *Proceedings of IEEE Workshop on Neural Networks for Signal Processing*, vol. 12, pp. 757–766, 2002.

[2] O. Madani, M. Georg, and D. A. Ross, "On using nearly-independent feature families for high precision and confidence," *Machine Learning*, vol. 92, pp. 457–477, 2013.

[3] J. Gu, S. Chen, and T. Sun, "Localization with incompletely paired data in complex wireless sensor network," *IEEE Transactions on Wireless Communications*, vol. 10, pp. 2841–2849, 2011.

[4] M. B. Blaschko, C. H. Lampert, and A. Gretton, "Semi-supervised Laplacian regularization of Kernel canonical correlation analysis," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 5211 LNAI, pp. 133–145, 2008.

[5] A. Kimura, H. Kameoka, M. Sugiyama, T. Nakano, E. Maeda, H. Sakano, and K. Ishiguro, "Semicca: Efficient semi-supervised learning of canonical correlations," in *2010 20th International Conference on Pattern Recognition*, Aug 2010, pp. 2933–2936.

[6] B. Zhang, J. Hao, G. Ma, J. Yue, and Z. Shi, *Semi-paired Probabilistic Canonical Correlation Analysis*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 1–10.

[7] X. Zhou, X. Chen, and S. Chen, "Neighborhood correlation analysis for semi-paired two-view data," *Neural Processing Letters*, vol. 37, pp. 335–354, 2013.

[8] Q. Qian, S. Chen, and X. Zhou, "Multi-view classification with cross-view must-link and cannot-link side information," *Knowledge-Based Systems*, vol. 54, pp. 137–146, 2013.

[9] R. Yan, J. Zhang, J. Yang, and A. G. Hauptmann, "A discriminative learning framework with pairwise constraints for video object classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 4, pp. 578–593, Apr. 2006.

[10] C. Alzate and J. A. K. Suykens, "A regularized formulation for spectral clustering with pairwise constraints," *Proceedings of the International Joint Conference on Neural Networks*, pp. 141–148, 2009.

[11] E. Eaton, M. DesJardins, and S. Jacob, "Multi-view constrained clustering with an incomplete mapping between views," *Knowledge and Information Systems*, vol. 38, pp. 231–257, 2014.

[12] Z. Lu and Y. Peng, "Unified Constraint Propagation on Multi-View Data," *Twenty-Seventh AAAI Conference on Artificial Intelligence*, pp. 640–646, 2013.

[13] L. Houthuys, R. Langone, and J. A. K. Suykens, "Multi-View Kernel Spectral Clustering," *Internal Report 17-71, ESAT-SISTA, KU Leuven (Leuven, Belgium)*, 2017.

[14] J. A. K. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor, and J. Vandewalle, *Least Squares Support Vector Machines*. World Scientific, 2002.

[15] C. Alzate and J. A. K. Suykens, "Multiway spectral clustering with out-of-sample extensions through weighted kernel PCA," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 2, pp. 335–347, 2010.

[16] R. Mall, R. Langone, and J. A. K. Suykens, "Kernel spectral clustering for big data networks," *Entropy*, vol. 15, pp. 1567–1586, 2013.

[17] A. Kumar, P. Rai, and H. Daume, "Co-regularized multi-view spectral clustering," *Neural Information Processing Systems 2011*, pp. 1413–1421, 2011.

[18] M.-R. Amini, N. Usunier, and C. Goutte, "Learning from multiple partially observed views - an application to multilingual text categorization," *Advances in Neural Information Processing Systems*, pp. 28–36, 2009.

[19] D. Greene and 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*, pp. 423–438, 2009.

[20] W. Yang and G. Toderici, "Discriminative tag learning on youtube videos with latent sub-tags," in *Computer Vision and Pattern Recognition (CVPR)*, 2011, pp. 3217–3224.

[21] C.-W. Hsu, C.-C. Chang, and C.-J. Lin, "A Practical Guide to Support Vector Classification," *Technical report, University of National Taiwan, Department of Computer Science and Information Engineering*, pp. 1–12, 2003.

[22] P. Li, G. Samorodnitsky, and J. Hopcroft, "Sign Cauchy projections and Chi-square kernel," *Advances in Neural Information Processing Systems*, vol. 26, pp. 2571–2579, 2013.

[23] H. Lodhi, C. Saunders, J. Shawe-Taylor, N. Cristianini, and C. Watkins, "Text Classification using String Kernels," *Journal of Machine Learning Research*, vol. 2, pp. 419–444, 2002.

[24] L. Hubert and P. Arabie, "Comparing partitions," *Journal of Classification*, pp. 193–218, 1985.