Tensor Learning in Multi-view Kernel PCA

Lynn Houthuys(✉) and Johan A. K. Suykens

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

Abstract. In many real-life applications data can be described through multiple representations, or views. Multi-view learning aims at combining the information from all views, in order to obtain a better performance. Most well-known multi-view methods optimize some form of correlation between two views, while in many applications there are three or more views available. This is usually tackled by optimizing the correlations pairwise. However, this ignores the higher-order correlations that could only be discovered when exploring all views simultaneously. This paper proposes novel multi-view Kernel PCA models. By introducing a model tensor, the proposed models aim to include the higher-order correlations between all views. The paper further explores the use of these models as multi-view dimensionality reduction techniques and shows experimental results on several real-life datasets. These experiments demonstrate the merit of the proposed methods.

Keywords: Kernel PCA · Multi-view learning · Tensor learning

1 Introduction

Principal component analysis (PCA) [12] is an unsupervised learning technique that transforms the initial space to a lower dimensional subspace while maintaining as much information as possible. The technique is wildly used in applications like dimensionality reduction, denoising and pattern recognition. PCA consist of taking the eigenvectors corresponding to the n_p largest eigenvalues, also known as the *principal components*, of the covariance matrix of a dataset, which span a subspace that retains the maximum variance of the dataset. For dimensionality reduction these principal components make up the lower dimensional dataset, and thus the new dimension equals n_p.

Several nonlinear extensions to PCA were proposed. One well-known extension is kernel PCA (KPCA) [21]. Instead of working on the data directly, it first applies a, possibly nonlinear, transformation on the data that maps the input data to a high-dimensional feature space.

In multi-view learning the input data is described through multiple representations or *views*. A dataset could for example consist of images and the associated captions [14], video clips could be classified based on image as well as audio

V. Kůrková et al. (Eds.): ICANN 2018, LNCS 11140, pp. 205–215, 2018.
https://doi.org/10.1007/978-3-030-01421-6_21

features [13], news stories could be covered by multiple sources [7], and so on. Multi-view learning has been applied in numerous applications both as supervised [3,28] and unsupervised [2,4] learning schemes. Multi-view dimensionality reduction reduces the multi-view dataset to a lower dimensional subspace to compactly represent the heterogeneous data, where each datapoint in the newly formed subspace is associated with multiple views. Dimensionality reduction is often beneficial for the learning process, especially when the data contains some sort of noise [6,8].

Most multi-view methods optimize a certain correlation between variables of two views. For example, in CCA [10] the correlation between the score variables is maximized, and in Multi-view LS-SVM [11] the product of the error variables is minimized. In real-world applications, however, data is often described through three views or more. This is usually accounted for by optimizing the sum of the pairwise correlations between different views. Due to this approach, higher-order correlations that could only be discovered by simultaneously considering all views, are ignored. This issue was pointed out by Luo et al. [16], where the authors propose an extension to CCA, called Tensor CCA, that analyzes a covariance tensor over the data from all views. The model is formed by performing a tensor decomposition, which has a computational cost that is significantly higher than the cost of regular CCA. This idea of including tensor learning is presented in Fig. 1.

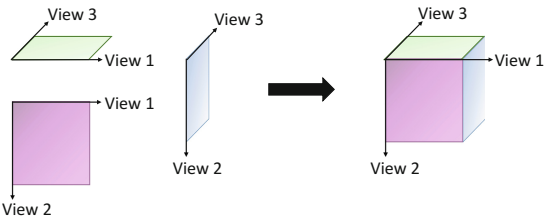

Fig. 1. An example with three views to motivate tensor learning in multi-view learning. (left) The standard coupling: only the pairwise correlations between the views are taken into account. (right) The tensor approach: the higher-order correlations between all views are modeled in a third order tensor.

Tensor learning in machine learning methods has been studied before. For example, Signoretto et al. [22] propose a tensor-based framework to perform learning when the data is multi-linear and Wimalawarne et al. [27] collect the weight vectors corresponding to separate tasks in one weight tensor to achieve multi-task learning.

This paper investigates the use of tensor learning in multi-view KPCA, in order to include the higher-order correlations. The paper proposes three multi-view KPCA methods, where the first two are special cases of the last method. Experiments, where the multi-view KPCA methods are used to reduce the dimensionality for clustering purposes, show the merit of our proposed methods.

We will denote matrices as bold uppercase letters, vectors as bold lowercase letters and higher-order tensors by calligraphic letters. The superscript $^{[v]}$ will denote the vth view for the multi-view method. Whereas the superscript $^{(j)}$ will correspond to the jth principal component.

2 Kernel PCA

Suykens et al. [26] formulated the kernel PCA problem in the primal-dual framework typical of Least Squares Support Vector Machines (LS-SVM) [25], where the dual problem is equivalent to the original kernel PCA formulation of Schölkopf et al. [21]. An advantage of the primal-dual framework is that it allows to perform estimations in the primal space, which can be used for large-scale applications when solving the dual problem becomes infeasible. The formulation further provides an out-of-sample extension to deal with new unseen test data.

Suykens [24] later formulated the kernel PCA in the Restricted Kernel Machines (RKM) framework, which preserves the advantages of the previous formulation. The primal and dual model are formed by means of conjugate feature duality, and give an expression in terms of visible and hidden layers respectively, in analogy with Restricted Boltzmann Machines (RBM) [9]. The dual problem is equivalent to the LS-SVM formulation (and hence the original formulation) up to a parameter. Furthermore it is shown how multiple RKMs can be coupled to form a Deep RKM, which combines deep learning with kernel based methods.

Given data $\{\mathbf{x}_k\}_{k=1}^N \subset \mathbb{R}^d$, the primal formulation of KPCA in the RKM framework is as follows:

$$\min_{\mathbf{w},h_k} \quad \frac{\eta}{2}\mathbf{w}^T\mathbf{w} - \sum_{k=1}^{N} \varphi(\mathbf{x}_k)^T\mathbf{w}\ h_k + \frac{\lambda}{2}\sum_{k=1}^{N} h_k^2 \tag{1}$$

for $k = 1, \ldots, N$. The feature map $\varphi(\cdot) : \mathbb{R}^d \to \mathbb{R}^{d_h}$ maps the input data to a high-dimensional (possible infinite) feature space. λ and η are positive regularization constants and the hidden features h_k correspond to the projected values. The dual problem related to this primal formulation is:

$$\frac{1}{\eta}\boldsymbol{\Omega}\ \mathbf{h} = \lambda\ \mathbf{h} \tag{2}$$

where $\mathbf{h} = [h_1; \ldots; h_N]$ and $\boldsymbol{\Omega} \in \mathbb{R}^{N\times N}$ is a centered kernel matrix defined as

$$\Omega_{kl} = \left(\varphi(\mathbf{x}_k) - \hat{\boldsymbol{\mu}}\right)^T \left(\varphi(\mathbf{x}_l) - \hat{\boldsymbol{\mu}}\right), \quad k, l = 1, \ldots, N \tag{3}$$

with $\hat{\boldsymbol{\mu}} = (1/N)\sum_{k=1}^{N} \varphi(\mathbf{x}_k)$. The feature map $\varphi(\cdot)$ is usually not explicitly defined, but rather through a positive definite kernel function $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$. Based on Mercer's condition [20] we can formulate the kernel function as $K(\mathbf{x}_k, \mathbf{x}_l) = \varphi(\mathbf{x}_k)^T\varphi(\mathbf{x}_l)$.

Every eigenvalue-eigenvector pair $(\lambda - \mathbf{h})$ can be seen as a candidate solution of Eq. (1). The first principal component, i.e. the direction of maximal variance in

the feature space, is determined by the eigenvector corresponding to the highest eigenvalue of $\frac{1}{\eta}\boldsymbol{\Omega}$. The maximum number of components that can be extracted equals the number of datapoints N.

For an unseen test point $\mathbf{x}$, the projection into the subspace spanned by the jth principal component, i.e. the *score variable* $\hat{e}(\mathbf{x})^{(j)}$, can be obtained as

$$\hat{e}(\mathbf{x})^{(j)} = \frac{1}{\eta}\boldsymbol{\Omega}_{\text{test}}\mathbf{h}^{(j)} \tag{4}$$

where $\mathbf{h}^{(j)}$ is the eigenvector corresponding to the jth largest eigenvalue λ and $\boldsymbol{\Omega}_{\text{test}}$ is the centered test kernel matrix calculated through the kernel function $K(\mathbf{x}_k, \mathbf{x}) = \varphi(\mathbf{x}_k)^T\varphi(\mathbf{x})$ for all $k = 1, \ldots, N$.

If KPCA is used to perform dimensionality reduction, the new dimension of the data equals the number of selected components n_p.

3 Multi-view Kernel Principal Component Analysis

In this section we conceive a KPCA model when the data is described through different representations, or *views*. Instead of coupling the different views pairwise, we formulate an overall model so that also higher order correlations between the different views are considered.

3.1 KPCA-ADD: Adding Kernel Matrices

A first model, called KPCA-ADD, is formed by adding up the different KPCA objectives and assuming that all views share the same hidden features $\mathbf{h}$.

Let V be the number of views, given data $\{\mathbf{x}_k^{[v]}\}_{k=1}^N \subset \mathbb{R}^{d^{[v]}}$ the primal formulation is stated as follows:

$$\min_{\mathbf{w}^{[v]}, h_k} \quad \frac{\eta}{2}\sum_{v=1}^{V}\mathbf{w}^{[v]^T}\mathbf{w}^{[v]} - \sum_{v=1}^{V}\sum_{k=1}^{N}\varphi^{[v]}(\mathbf{x}_k^{[v]})^T\mathbf{w}^{[v]}\ h_k + \frac{\lambda}{2}\sum_{k=1}^{N}h_k^2 \tag{5}$$

The stationary points of this objective function, denoted as $\mathcal{J}$, in the primal formulation are characterized by:

$$\begin{cases} \dfrac{\partial \mathcal{J}}{\partial h_k} = 0 \rightarrow \lambda h_k = \displaystyle\sum_{v=1}^{V}\mathbf{w}^{[v]^T}\varphi^{[v]}(\mathbf{x}_k^{[v]}), \\ \dfrac{\partial \mathcal{J}}{\partial \mathbf{w}^{[v]}} = 0 \rightarrow \mathbf{w}^{[v]} = \dfrac{1}{\eta}\displaystyle\sum_{k=1}^{N}\varphi^{[v]}(\mathbf{x}_k^{[v]})h_k, \\ \qquad\qquad \text{where } k = 1, \ldots, N \text{ and } v = 1, \ldots, V. \end{cases} \tag{6}$$

By eliminating the weights $\mathbf{w}^{[v]}$, the dual formulation is obtained:

$$\frac{1}{\eta}\left(\boldsymbol{\Omega}^{[1]} + \ldots + \boldsymbol{\Omega}^{[V]}\right)\ \mathbf{h} = \lambda\ \mathbf{h} \tag{7}$$

where $\boldsymbol{\Omega}^{[v]}$ is the centered kernel matrix corresponding to view v, defined as $\Omega_{kl}^{[v]} = \left(\varphi^{[v]}(\mathbf{x}_k^{[v]}) - \hat{\boldsymbol{\mu}}^{[v]}\right)^T \left(\varphi^{[v]}(\mathbf{x}_l^{[v]}) - \hat{\boldsymbol{\mu}}^{[v]}\right)$ for $k, l = 1, \ldots, N$.

Notice that this coupling results in adding up the kernel matrices belonging to the different views.

The score variables corresponding to a test point $\mathbf{x}$ can be calculated by:

$$\hat{e}(\mathbf{x})^{(j)} = \frac{1}{\eta} \sum_{v=1}^{V} \boldsymbol{\Omega}_{\text{test}}^{[v]} \mathbf{h}^{(j)}. \tag{8}$$

4 Including Tensor Learning in Multi-view KPCA

Even though in the KPCA-ADD formulation the views are coupled by the shared hidden features, there is still a model weight vector $\mathbf{w}^{[v]} \in \mathbb{R}^{d_h^{[v]}}$ for each view v. In order to introduce more coupling, a model tensor $\mathcal{W} \in \mathbb{R}^{d_h^{[1]} \times \ldots \times d_h^{[V]}}$ is presented. By using a tensor comprised of the weights of all views, instead of coupling them pairwise, it becomes possible to model higher order correlations.

4.1 KPCA-PROD: Product of Kernel Matrices

The introduction of a model tensor $\mathcal{W}$ leads to the KPCA-PROD model, where the primal formulation is given by:

$$\min_{\mathcal{W}, h_k} \quad \frac{\eta}{2}\langle \mathcal{W}, \mathcal{W} \rangle - \sum_{k=1}^{N} \langle \Phi_{(k)}, \mathcal{W} \rangle \; h_k + \frac{\lambda}{2} \sum_{k=1}^{N} h_k^2 \tag{9}$$

where $\langle \cdot, \cdot \rangle$ is the tensor inner product defined as

$$\langle \mathcal{A}, \mathcal{B} \rangle := \sum_{i_1=1}^{I_1} \cdots \sum_{i_M=1}^{I_M} \mathcal{A}_{i_1 \cdots i_M} \mathcal{B}_{i_1 \cdots i_M} \tag{10}$$

for two M-th order tensors $\mathcal{A}, \mathcal{B} \in \mathbb{R}^{I_1 \times \ldots \times I_M}$. The rank-1 tensor $\Phi_{(k)} \in \mathbb{R}^{d_h^{[1]} \times \ldots \times d_h^{[V]}}$ is composed by the outer product of the feature maps of all views, i.e. $\Phi_{(k)} = \varphi^{[1]}(x_k^{[1]}) \otimes \ldots \otimes \varphi^{[V]}(x_k^{[V]})$.

The stationary points of the objective function $\mathcal{J}$ in the primal formulation are characterized by:

$$\begin{cases} \dfrac{\partial \mathcal{J}}{\partial h_k} = 0 \rightarrow \lambda h_k = \langle \Phi_{(k)}, \mathcal{W} \rangle = \displaystyle\sum_{i_1=1}^{d_h^{[1]}} \cdots \sum_{i_V=1}^{d_h^{[V]}} \varphi^{[1]}(\mathbf{x}_k^{[1]})_{i_1} \cdots \varphi^{[V]}(\mathbf{x}_k^{[V]})_{i_V} \mathcal{W}_{i_1 \ldots i_V} \\ \dfrac{\partial \mathcal{J}}{\partial \mathcal{W}_{i_1 \ldots i_V}} = 0 \rightarrow \mathcal{W}_{i_1 \ldots i_V} = \dfrac{1}{\eta} \displaystyle\sum_{k=1}^{N} \varphi^{[1]}(\mathbf{x}_k^{[1]})_{i_1} \cdots \varphi^{[V]}(\mathbf{x}_k^{[V]})_{i_V} h_k, \\ \qquad \text{where } k = 1, \ldots, N \;\text{ and } i_v = 1, \ldots, d_h^{[v]} \text{ for } v = 1, \ldots, V. \end{cases} \tag{11}$$

By eliminating the weights, the following dual problem is derived:

$$\frac{1}{\eta}\left(\boldsymbol{\Omega}^{[1]} \odot \ldots \odot \boldsymbol{\Omega}^{[V]}\right)\ \mathbf{h} = \lambda\ \mathbf{h} \tag{12}$$

where $\odot$ denotes the element-wise product. Notice that the dual problem results in element-wise multiplication of the view-specific kernel matrices.

The score variable corresponding to an unseen test point $\mathbf{x}$ can hence be calculated by:

$$\hat{e}(\mathbf{x})^{(j)} = \frac{1}{\eta}\bigodot_{v=1}^{V} \boldsymbol{\Omega}^{[v]}_{\text{test}}\mathbf{h}^{(j)} \tag{13}$$

where $\bigodot$ is the element-wise multiplication operator.

4.2 KPCA-ADDPROD

Taking the element-wise product of kernel matrices can have some unwanted results. Take for example kernel matrices comprised of linear kernel functions. An element of such a linear kernel matrix could be negative, indicating a low similarity between two points. By multiplying the elements of the kernel matrices, highly negative values could result in a high positive value for a certain datapoint pair, which would indicate a very high similarity which is clearly unwanted. Even for kernel matrices comprised of RBF kernel functions, where the values lie between zero and one, a poor view indicating a certain datapoint pair as non-similar and hence assigning a value close to zero, could influence the final result to harshly.

Therefore a last model is proposed, called KPCA-ADDPROD, where the two principles of the previous models are combined. A parameter ρ is added in order to determine the influence of each part. The primal formulation is given by:

$$\begin{aligned}\min_{\mathcal{W},\mathbf{w}^{[v]},h_k}\quad & \frac{\eta}{2}\langle\mathcal{W},\mathcal{W}\rangle - \sqrt{\rho}\sum_{k=1}^{N}\langle\Phi_{(k)},\mathcal{W}\rangle\ h_k + \frac{\lambda}{2}\sum_{k=1}^{N}h_k^2 \\ & + \frac{\eta}{2}\sum_{v=1}^{V}\mathbf{w}^{[v]^T}\mathbf{w}^{[v]} - \sqrt{(1-\rho)}\sum_{v=1}^{V}\sum_{k=1}^{N}\varphi^{[v]}(\mathbf{x}_k^{[v]})^T\mathbf{w}^{[v]}h_k\end{aligned} \tag{14}$$

where $\rho \in [0,1] \subset \mathbb{R}$. By deriving the stationary points of the objective and eliminating the weights, the following dual problem is obtained:

$$\frac{1}{\eta}\left((1-\rho)\sum_{v=1}^{V}\boldsymbol{\Omega}^{[v]} + \rho\bigodot_{v=1}^{V}\boldsymbol{\Omega}^{[v]}\right)\ \mathbf{h} = \lambda\ \mathbf{h}. \tag{15}$$

Note that if $\rho = 0$ the model is equivalent to KPCA-ADD, and if $\rho = 1$ it is equivalent to KPCA-PROD.

5 Experiments

This section describes the experiments performed to evaluate the multi-view KPCA models, as dimensionality reduction techniques. To assess the performance, the KPCA methods are used as a preprocessing step for clustering, and the clustering accuracy is regarded as the evaluation criterion.

Two clustering methods are considered: k-means (KM) [18], a well known linear clustering algorithm and Kernel Spectral Clustering (KSC) [1], a non-linear clustering technique within the LS-SVM framework. To determine the clustering accuracy, the NMI [23] is reported[1]. Due to the local optima solutions found by KM, these results are averaged over 50 runs.

The performances of the proposed multi-view models are compared to the performances on the views separately. Both by clustering the views directly, and by clustering after KPCA was performed.

Model Selection. The parameter η is set to 1 in all experiments, since this parameter is of most importance when multiple RKMs are stacked to form a deep RKM. The RBF kernel function was used for all experiments, both for the KPCA methods as for KSC. The performance of the (multi-view) KPCA models depend on the (view-specific) kernel parameter and the number of principal components n_p. For KPCA-ADDPROD it will also depend on the parameter ρ. Both KSC and KM depend on the number of clusters, and KSC also on the kernel parameter. These parameters are tuned through a grid search with 5-fold crossvalidation. Since the methods are all unsupervised, the model selection criteria has to be unsupervised as well. Here the Davies-Bouldin index (DB) [5] criterion is used.

Datasets. A brief description of each dataset used is given here:

- **Image-caption dataset:** A dataset comprised of images, together with their associated captions. We thank the authors of [14] for providing the dataset. Each image-caption pair represent a figure related to sport, aviation or paintball. For each of these categories, 400 records are available. The first two views consist of different features describing the image (HSV colour and image Gabor texture). The third view describes the associated caption text by its term frequencies. Gaussian white noise is added to the first two views.
- **YouTube Video dataset:** A dataset describing YouTube videos of video gaming, was originally proposed by Madani et al. [19][2]. The videos are described through textual, visual and auditory features. For this paper we selected the textual feature LDA, the visual Motion feature through CIPD [29] and the audio feature MFCC [17] as three views. From each of the seven

[1] To calculate the NMI, and hence asses the performance, the labels of the dataset are used. However, notice that they are never used in the training or validation phase of KM, KSC or the proposed multi-view KPCA models.

[2] http://archive.ics.uci.edu/ml/datasets/youtube+multiview+video+games+dataset.

most occurring labels (excluding the last label, since these datapoints represent videos not belonging to any of the other 30 classes) 300 videos were randomly sampled.

– **UCI Ads dataset:** This dataset, as described by Kushmerick [15][3], was constructed for the task of predicting whether a certain hyperlink corresponds to an advertisement or not. The features are divided over three views in the same way as was done by Luo et al. [16]. The dataset consist of 2821 instances not corresponding to advertisements, and 458 instances that do.

Results. The results of the performed experiments are depicted in Table 1. The table shows the clustering accuracy found by using the clustering techniques on the views directly, and when KPCA was applied as a dimensionality reduction technique first. It further shows the accuracy when the proposed multi-view KPCA techniques are applied. For the KPCA-ADDPROD method, also the found optimal value for ρ is noted.

Table 1. NMI results, where the proposed methods function as dimensionality reduction methods for KM and KSC. The best performing methods, are indicated in bold.

Method	Image-caption			YouTube Video			Ads		
View	1	2	3	1	2	3	1	2	3
KM	0.502	0.301	0.206	**0.434**	0.200	0.052	0.068	0.028	0.071
KPCA+KM	0.516	0.328	0.412	0.375	0.207	0.065	0.016	0.021	0.047
KPCA-ADD+KM	0.596			0.273			0.016		
KPCA-PROD+KM	0.154			0.076			**0.291**		
KPCA-ADDPROD+KM	**0.643** ($\rho = 0.4$)			0.279 ($\rho = 0.2$)			**0.291** ($\rho = 1$)		
KSC	0.061	0.107	0.066	0.028	0.025	0.030	0.017	0.077	0.312
KPCA+KSC	0.474	0.330	0.295	0.243	0.167	0.037	0.013	0.094	0.046
KPCA-ADD+KSC	0.520			0.166			0.085		
KPCA-PROD+KSC	0.031			0.025			**0.147**		
KPCA-ADDPROD+KSC	**0.568** ($\rho = 0.4$)			**0.248** ($\rho = 0.2$)			**0.147** ($\rho = 1$)		

A first observation is that the performance usually improves when using KPCA as a dimensionality reduction method, when clustering the views separately. This encourages the use of dimensionality reduction in these datasets. A notable exception is the accuracy when using KM on the first view of the YouTube Video dataset.

A second observation is that the multi-view KPCA methods are able to improve the clustering accuracy in five out of the six experiments, suggesting the merit of using the multi-view techniques independently of the choice of clustering technique. Only for YouTube Video dataset, the (multi-view) dimensionality reduction is not able to improve the result of applying KM on the first

[3] http://archive.ics.uci.edu/ml/datasets/Internet+Advertisements.

view directly. Another interesting observation is that the found optimal ρ for each dataset is equal for both clustering methods. Since ρ determines the importance of the tensor model vector, this could be an indication of the number of relevant higher order correlations in a dataset. For the first two datasets ρ is relatively small. For these two datasets KPCA-ADD outperforms KPCA-PROD considerably, which is to be expected as it is shown that these two models are actually special cases of KPCA-ADDPROD with $\rho = 0$ and $\rho = 1$ respectively. For the Ads dataset the found optimal ρ equals 1, and hence only the tensor model vector is taken into account, suggesting a high importance of higher order correlations.

6 Conclusion

This paper introduced novel Multi-view Kernel Principal Component Analysis methods to perform KPCA when the data is represented by multiple views. Techniques from tensor learning are applied in order to account for higher order correlations between the views.

The paper starts from the primal RKM formulation of KPCA and shows three approaches for a multi-view extension. It is shown that, when assuming shared hidden features, the dual model results in kernel addition. It further shows that introducing a model tensor, containing the information of all views, results in kernel product in the dual formulation. Finally a third method is suggested combining the two techniques.

The gain of these multi-view techniques is shown by using it as a dimensionality reduction step before clustering. Experiments on multiple real-world datasets with two well known clustering techniques, show the improvement of using multiple views. The parameter controlling the importance of the model tensor seems to indicate the importance of the higher order correlations.

Acknowledgments.. Research supported by Research Council KUL: CoE PFV/10/002 (OPTEC), PhD/Postdoc grants Flemish Government; FWO: projects: G0A4917N (Deep restricted kernel machines), G.088114N (Tensor based data similarity), ERC Advanced Grant E-DUALITY (787960).

References

1. Alzate, C., Suykens, J.A.K.: Multiway spectral clustering with out-of-sample extensions through weighted kernel PCA. IEEE Trans. Pattern Anal. Mach. Intell. **32**(2), 335–347 (2010)
2. Andrew, G., Arora, R., Bilmes, J., Livescu, K.: Deep canonical correlation analysis. In: ICML, pp. 1247–1255 (2013)
3. Bekker, A., Shalhon, M., Greenspan, H., Goldberger, J.: Multi-view probabilistic classification of breast microcalcifications. IEEE Trans. Med. Imaging **35**(2), 645–653 (2016)
4. Blum, A., Mitchell, T.: Combining labeled and unlabeled data with co-training. In: COLT, pp. 92–100 (1998)

5. Davies, D.L., Bouldin, D.W.: A cluster separation measure. IEEE Trans. Pattern Anal. Mach. Intell. **1**(2), 224–227 (1979)
6. Foster, D.P., Kakade, S.M., Zhang, T.: Multi-view dimensionality reduction via canonical correlation analysis. Toyota Technical Institute-Chicago (2008)
7. Greene, D., Cunningham, P.: A matrix factorization approach for integrating multiple data views. In: Buntine, W., Grobelnik, M., Mladenić, D., Shawe-Taylor, J. (eds.) ECML PKDD 2009. LNCS (LNAI), vol. 5781, pp. 423–438. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-04180-8_45
8. Han, Y., Wu, F., Tao, D., Shao, J., Zhuang, Y., Jiang, J.: Sparse unsupervised dimensionality reduction for multiple view data. IEEE Trans. Circ. Syst. Video Technol. **22**(10), 1485–1496 (2012)
9. Hinton, G.E.: What kind of a graphical model is the brain? In: Proceedings of the 19th International Joint Conference on Artificial Intelligence, IJCAI 2005, pp. 1765–1775. Morgan Kaufmann Publishers Inc., San Francisco (2005)
10. Hotelling, H.: Relations between two sets of variates. Biometrica **28**, 321–377 (1936)
11. Houthuys, L., Langone, R., Suykens, J.A.K.: Multi-view least squares support vector machines classification. Neurocomputing **282**, 78–88 (2018)
12. Jolliffe, I.T.: Principal Component Analysis. Springer, New York (1986). https://doi.org/10.1007/978-1-4757-1904-8
13. Kidron, E., Schechner, Y.Y., Elad, M.: Pixels that sound. In: CVPR, vol. 1, pp. 88–95 (2005)
14. Kolenda, T., Hansen, L.K., Larsen, J., Winther, O.: Independent component analysis for understanding multimedia content. In: IEEE Workshop on Neural Networks for Signal Processing, vol. 12, pp. 757–766 (2002)
15. Kushmerick, N.: Learning to remove internet advertisements. In: AGENTS 1999, pp. 175–181 (1999)
16. Luo, Y., Tao, D., Ramamohanarao, K., Xu, C., Wen, Y.: Tensor canonical correlation analysis for multi-view dimension reduction. IEEE Trans. Knowl. Data Eng. **27**(11), 3111–3124 (2015)
17. Lyon, R.F., Rehn, M., Bengio, S., Walters, T.C., Chechik, G.: Sound retrieval and ranking using sparse auditory representations. Neural Comput. **22**(9), 2390–2416 (2010)
18. Macqueen, J.: Some methods for classification and analysis of multivariate observations. In: Berkeley Symposium on Mathematical Statistics and Probability, pp. 281–297 (1967)
19. Madani, O., Georg, M., Ross, D.A.: On using nearly-independent feature families for high precision and confidence. Mach. Learn. **92**, 457–477 (2013)
20. Mercer, J.: Functions of positive and negative type, and their connection with the theory of integral equations. Philos. Trans. R. Soc. London. Ser. A Contain. Pap. Math. Phys. Character **209**, 415–446 (1909)
21. Schölkopf, B., Smola, A., Müller, K.R.: Nonlinear component analysis as a kernel eigenvalue problem. Neural Comput. **10**(5), 1299–1319 (1998)
22. Signoretto, M., Tran Dinh, Q., De Lathauwer, L., Suykens, J.A.K.: Learning with tensors: a framework based on convex optimization and spectral regularization. Mach. Learn. **94**, 303–351 (2014)
23. Strehl, A., Ghosh, J.: Cluster ensembles - a knowledge reuse framework for combining multiple partitions. J. Mach. Learn. Res. **3**, 583–617 (2002)
24. Suykens, J.A.K.: Deep restricted kernel machines using conjugate feature duality. Neural Comput. **29**(8), 2123–2163 (2017)
25. Suykens, J.A.K., Van Gestel, T., De Brabanter, J., De Moor, B., Vandewalle, J.: Least Squares Support Vector Machines. World Scientific, Singapore (2002)

26. Suykens, J.A.K., Van Gestel, T., Vandewalle, J., De Moor, B.: A support vector machine formulation to PCA analysis and its kernel version. IEEE Trans. Neural Netw. **14**(2), 447–450 (2003)
27. Wimalawarne, K., Sugiyama, M., Tomioka, R.: Multitask learning meets tensor factorization: Task imputation via convex optimization. In: NIPS, vol. 4, pp. 2825–2833 (2014)
28. Wozniak, M., Jackowski, K.: Some remarks on chosen methods of classifier fusion based on weighted voting. In: Corchado, E., Wu, X., Oja, E., Herrero, Á., Baruque, B. (eds.) HAIS 2009. LNCS (LNAI), vol. 5572, pp. 541–548. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-02319-4_65
29. Yang, W., Toderici, G.: Discriminative tag learning on Youtube videos with latent sub-tags. In: CVPR, pp. 3217–3224 (2011)