Neurocomputing 282 (2018) 78–88

Contents lists available at ScienceDirect

Neurocomputing

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

Multi-View Least Squares Support Vector Machines Classification

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

Article history:
Received 21 April 2017
Revised 28 August 2017
Accepted 11 December 2017
Available online 14 December 2017

Communicated by Dr. Chenping Hou

Keywords:
Multi-view learning
Classification
LS-SVM

ABSTRACT

In multi-view learning, data is described using different representations, or views. Multi-view classification methods try to exploit information from all views to improve the classification performance. Here a new model is proposed that performs classification when two or more views are available. The model is called Multi-View Least Squares Support Vector Machines (MV-LSSVM) Classification and is based on solving a constrained optimization problem. The primal objective includes a coupling term, which minimizes a combination of the errors from all views. The model combines the benefits from both early and late fusion, it is able to incorporate information from all views in the training phase while still allowing for some degree of freedom to model the views differently. Experimental comparisons with similar methods show that using multiple views improves the results with regard to the single view classifiers and that it outperforms other state-of-the-art algorithms in terms of classification accuracy.

© 2017 Elsevier B.V. All rights reserved.

1. Introduction

In many application domains data is described by several means of representation or *views* [1,2]. For example, web pages can be classified based on both the page content (text) and hyperlink information [3,4], for social networks one could use the user profile but also the friend links [5], images consist of the pixel arrays but can also have captions associated with them [6], and so on. Although each of the views by itself might already perform sufficiently for a certain learning task, improvements can be obtained by combining the information provided by several representations of the data.

The information from the views can be combined in different ways as well as in different stages of the training process. In early fusion techniques the information is combined before any training process is performed. This can be achieved by means of a simple concatenation of the data from all views, e.g. Zilca and Bistritz [7], or a more complex method like for example the work done by Yu et al. [8]. In late fusion techniques, a different classifier for each view is separately trained and later a weighted combination is taken as the final model. These models are also called committee networks [9]. Here, one has the freedom to model the views differently, which is a strong advantage when the data is inherently different over the views (e.g. in the case of text data and pixel arrays). Existing multi-view classification methods are usually a form of late fusion. For example, Bekker et al. [10] uses a stochastic combination of two classifiers to determine whether a breast microcalcification is benign or malignant, Mayo and Frank [11] perform multi-view multi-instance learning by a weighted combination of separate classifiers on each view and uses it to do image classification and Wozniak and Jackowski [12] give a comparative overview of methods which perform classification based on a weighted voting of the results of separate classifiers on the views individually.

A third option is to combine the benefits of both fusion types. To perform the multi-view learning so that it has some degree of freedom to model the views differently but to also ensure that information from other views is already exploited during the training phase. This idea was already partially examined by Koco and Capponi [13] who update the initial separate classifiers for each view based on the information of the other classifiers by means of a boosting scheme. At the end of the training phase a weighted combination of the classifiers is taken as the final classifier. Minh et al. [14] use this technique to develop a multi-view semi-supervised classification model based on Support Vector Machines (SVM) with within-view as well as between-view regularization. This model, just like SVM, results in having to solve a quadratic programming problem.

In this paper a multi-view classification model called *Multi-View Least Squares Support Vector Machines (MV-LSSVM) Classification* is introduced which is cast in the primal-dual optimization setting typical to Least Squares Support Vector Machines (LS-SVM) [15], where multiple classification formulations in the primal model are combined in such a way that a combination of the error variables

* 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).

https://doi.org/10.1016/j.neucom.2017.12.029
0925-2312/© 2017 Elsevier B.V. All rights reserved.

from all views is minimized. This model combines the benefits of late and early fusion by allowing for a different regularization parameter and a different kernel function for the different views while the coupling term enforces the product of all error variables to be small.

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 classification problem in case there are more than two classes.

The rest of this paper is organized as follows: Section 2 gives a summary of the LS-SVM classification and discusses the multiclass extension. Section 3 discusses the proposed model MV-LSSVM. It shows the mathematical formulation and the multiclass extension for the training data and shows the resulting classifier for the out-of-sample test data. Section 4 discusses the experiments done with MV-LSSVM and compares it to using only one view and to other multi-view methods. Section 4 further discusses the obtained results and shows a parameter and a complexity study. Finally, in Section 5 some conclusions are drawn.

2. LS-SVM classification

This section summarizes the *Least Squares Support Vector Machine* model as described by Suykens et al. [15]. LS-SVM is a modification to the Support Vector Machine (SVM) model as introduced by [16] with a least squares loss function and equality constraints, where the dual solution can be found by solving a linear system instead of quadratic programming problem. As for SVM, LS-SVM maps the data into a high dimensional feature space in which one constructs a linear separating hyperplane.

Given a training set of N data points $\{y_k, \mathbf{x}_k\}_{k=1}^N$ where $\mathbf{x}_k \in \mathbb{R}^d$ denotes the kth input pattern and $y_k \in \{-1, 1\}$ the kth label, the primal formulation of the LS-SVM classification model is:

$$\min_{\mathbf{w},\mathbf{e},b} \quad \frac{1}{2}\mathbf{w}^T\mathbf{w} + \frac{1}{2}\gamma\mathbf{e}^T\mathbf{e} \tag{1}$$

$$\text{s.t. } \mathbf{Z}^T\mathbf{w} + \mathbf{y}\, b = \mathbf{1}_N - \mathbf{e}$$

where $\mathbf{e} \in \mathbb{R}^N$ are error variables such that misclassifications are tolerated in case of overlapping distributions, $\mathbf{y} = [y_1; \ldots; y_N]$ denotes the target vector, b is a bias term and γ a positive real constant. $\mathbf{Z}^T \in \mathbb{R}^{N \times d_h}$ is defined as $\mathbf{Z}^T = [\varphi(\mathbf{x}_1)^T y_1; \ldots; \varphi(\mathbf{x}_N)^T y_N]$ where $\varphi : \mathbb{R}^d \to \mathbb{R}^{d_h}$ is the feature map which maps the input to a high dimensional feature space. The function φ is usually not defined explicitly, but rather implicitly through a positive definite kernel function $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$. Based on Mercer's condition [17] we can formulate the kernel function as $K(\mathbf{x}_i, \mathbf{x}_j) = \varphi(\mathbf{x}_i)^T\varphi(\mathbf{x}_j)$ and we can thus work in the high dimensional feature space without having to explicitly define it.

In the case of a high dimensional or implicitly defined feature space it is not practical to work with this formulation. By taking the Lagrangian of the primal problem, deriving the KKT optimality conditions and eliminating the primal variables $\mathbf{w}$ and $\mathbf{e}$ we obtain the following dual problem:

$$\left[\begin{array}{c|c} 0 & \mathbf{y}^T \\ \hline \mathbf{y} & \boldsymbol{\Omega} + \mathbb{I}_N/\gamma \end{array}\right]\left[\begin{array}{c} b \\ \hline \boldsymbol{\alpha} \end{array}\right] = \left[\begin{array}{c} 0 \\ \hline \mathbf{1}_N \end{array}\right] \tag{2}$$

where $\mathbf{1}_N$ is a one column vector of dimension N and $\mathbb{I}_N$ is the identity matrix of dimension $N \times N$. $\boldsymbol{\Omega} = \mathbf{Z}^T\mathbf{Z}$ is the labeled kernel matrix and the kernel trick can be applied within the kernel matrix as follows:

$$\begin{aligned} \Omega_{ij} &= y_i y_j \varphi(\mathbf{x}_i)^T\varphi(\mathbf{x}_j) \\ &= y_i y_j K(\mathbf{x}_i, \mathbf{x}_j), \quad k, j = 1, \ldots, N. \end{aligned} \tag{3}$$

The resulting classifier in the dual space takes the form

$$y(\mathbf{x}) = \operatorname{sign}\left(\sum_{k=1}^{N} \alpha_k y_k K(\mathbf{x}, \mathbf{x}_k) + b\right). \tag{4}$$

While in SVM many support vector values are typically equal to zero, for LS-SVM a support value α_k is proportional to the error at the data point $\mathbf{x}_k$.

This binary classification problem can easily be extended to a multiclass problem by taking additional output variables, similarly to the neural networks approach. Instead of one output value $\mathbf{y}$, we take m outputs $\mathbf{y}^{(1)}, \ldots, \mathbf{y}^{(m)}$. The number of output values m depend on the coding used to encode n_c classes. A popular choice for the encoding is the *one-versus-all (1vsA)* encoding where $m = n_c$, which makes binary decisions between each class and all other classes. When the number of classes is very high *minimum output encoding (MOC)* can be considered which uses m outputs to encode up to $n_c = 2^m$ classes.

The primal formulation of the LS-SVM multiclass classification model [15] is:

$$\min_{\mathbf{w}^{(l)},\mathbf{e}^{(l)},b^{(l)}} \quad \frac{1}{2}\sum_{l=1}^{m} \mathbf{w}^{(l)T}\mathbf{w}^{(l)} + \frac{1}{2}\sum_{l=1}^{m} \gamma^{(l)}\mathbf{e}^{(l)T}\mathbf{e}^{(l)} \tag{5}$$

$$\begin{aligned} &\text{s.t. } \mathbf{Z}^{(1)T}\mathbf{w}^{(1)} + \mathbf{y}^{(1)}\, b^{(1)} = \mathbf{1}_N - \mathbf{e}^{(1)} \\ &\quad\vdots \\ &\mathbf{Z}^{(m)T}\mathbf{w}^{(m)} + \mathbf{y}^{(m)}\, b^{(m)} = \mathbf{1}_N - \mathbf{e}^{(m)}. \end{aligned}$$

By taking the Lagrangian of the primal problem, deriving the KKT optimality conditions and eliminating the primal variables $\mathbf{w}^{(l)}$ and $\mathbf{e}^{(l)}$ we get the following dual problem:

$$\left[\begin{array}{c|c} \mathbf{0}_{m\times m} & \mathbf{Y_M}^T \\ \hline \mathbf{Y_M} & \boldsymbol{\Omega_M} + \mathbf{D_M} \end{array}\right]\left[\begin{array}{c} \mathbf{b_M} \\ \hline \boldsymbol{\alpha_M} \end{array}\right] = \left[\begin{array}{c} \mathbf{0}_m \\ \hline \mathbf{1}_{Nm} \end{array}\right] \tag{6}$$

where $\mathbf{0}_m$ and $\mathbf{1}_{Nm}$ are zero and one column vectors of dimension m and Nm, respectively, $\mathbf{0}_{m\times m}$ is a zero matrix of dimension $m \times m$ and with given matrices

$$\begin{aligned} \mathbf{Y_M} &= \operatorname{blockdiag}\left\{\mathbf{y}^{(1)}, \ldots, \mathbf{y}^{(m)}\right\} \\ \boldsymbol{\Omega_M} &= \operatorname{blockdiag}\left\{\boldsymbol{\Omega}^{(1)}, \ldots, \boldsymbol{\Omega}^{(m)}\right\} \\ \mathbf{D_M} &= \operatorname{blockdiag}\left\{\mathbf{D}^{(1)}, \ldots, \mathbf{D}^{(m)}\right\} \\ \mathbf{b_M} &= \begin{bmatrix} b^{(1)} \\ \vdots \\ b^{(m)} \end{bmatrix}, \boldsymbol{\alpha_M} = \begin{bmatrix} \boldsymbol{\alpha}^{(1)} \\ \vdots \\ \boldsymbol{\alpha}^{(m)} \end{bmatrix} \end{aligned} \tag{7}$$

and where $D_{ij}^{(l)} = \delta_{ij}/\gamma^{(l)}$ and where δ_{ij} denotes the Kronecker delta ($\delta_{ij} = 1$ if $i = j$ and 0 otherwise).

The linear system in Eq. (6) however does not need to be solved as a whole. Because of the block-diagonal structure the problem can be decomposed into m smaller subproblems like Eq. (2).

3. Multi-View LS-SVM Classification

In this section the model *Multi-View Least Squares Support Vector Machines (MV-LSSVM) Classification* is introduced. This is an extension to multiclass LS-SVM classification where 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 and n_c classes, a training set of N data points $\{y_k^{(l)}, \mathbf{x}_k^{[v]}\}_{k=1,l=1}^{k=N,l=m}$ for each view $v = 1, \ldots, V$ where $\mathbf{x}_k^{[v]} \in \mathbb{R}^{d^{[v]}}$ denotes the kth input pattern and $y_k^{(l)} \in \{-1, 1\}$ the lth output unit for the kth label, the primal formulation of the proposed model is:

$$\min_{\substack{\mathbf{w}^{[v]^{(l)}}, \mathbf{e}^{[v]^{(l)}}, \\ b^{[v]^{(l)}}}} \frac{1}{2}\sum_{l=1}^{m}\sum_{v=1}^{V}\mathbf{w}^{[v]^{(l)T}}\mathbf{w}^{[v]^{(l)}} + \frac{1}{2}\sum_{l=1}^{m}\sum_{v=1}^{V}\gamma^{[v]^{(l)}}\mathbf{e}^{[v]^{(l)T}}\mathbf{e}^{[v]^{(l)}} + \rho\sum_{l=1}^{m}\sum_{v,u=1;v\neq u}^{V}\mathbf{e}^{[v]^{(l)T}}\mathbf{e}^{[u]^{(l)}} \tag{8}$$

$$\text{s.t. } \mathbf{Z}^{[v]^{(1)T}}\mathbf{w}^{[v]^{(1)}} + \mathbf{y}^{(1)}\, b^{[v]^{(1)}} = \mathbf{1}_N - \mathbf{e}^{[v]^{(1)}}$$
$$\vdots$$
$$\mathbf{Z}^{[v]^{(m)T}}\mathbf{w}^{[v]^{(m)}} + \mathbf{y}^{(m)}\, b^{[v]^{(m)}} = \mathbf{1}_N - \mathbf{e}^{[v]^{(m)}} \qquad \text{for } v = 1, \ldots, V$$

where $l = 1, \ldots, m$ denote the binary subproblems needed to classify n_c classes and m depends on the coding used. $\mathbf{e}^{[v]^{(l)}} \in \mathbb{R}^N$ are error variables related to the vth view such that misclassifications are tolerated in case of overlapping distributions, $\mathbf{y}^{(l)} = [y_1^{(l)}; \ldots; y_N^{(l)}]$ denotes the target vector, $b^{[v]^{(l)}}$ are bias terms and $\gamma^{[v]^{(l)}}$ are positive real constants. $\mathbf{Z}^{[v]^{(l)T}} \in \mathbb{R}^{N\times d_h^{[v]^{(l)}}}$ are defined as $\mathbf{Z}^{[v]^{(l)T}} = [y_1^{(l)}\varphi^{[v]^{(l)}}(\mathbf{x}_1^{[v]})^T; \ldots; y_N^{(l)}\varphi^{[v]^{(l)}}(\mathbf{x}_N^{[v]})^T]$ where $\varphi^{[v]^{(l)}} : \mathbb{R}^{d^{[v]}} \to \mathbb{R}^{d_h^{[v]^{(l)}}}$ are the mappings to high dimensional feature spaces. This primal optimization function is a sum of V different classification objectives (one for each view) coupled by means of the *coupling term* $\sum_{l=1}^{m}\sum_{v,u=1;v\neq u}^{V}\mathbf{e}^{[v]^{(l)T}}\mathbf{e}^{[u]^{(l)}}$, where ρ is an additional regularization constant and will be called the *coupling parameter*. This term minimizes the product of the error variables of both views. In this way, information from both views is incorporated in the model and high error variables for a certain point in one view can be compensated by a corresponding low error variable in the other view.

The Lagrangian of the primal problem is

$$\mathcal{L}\left(\mathbf{w}^{[v]^{(l)}}, \mathbf{e}^{[v]^{(l)}}, b^{[v]^{(l)}}; \boldsymbol{\alpha}^{[v]^{(l)}}\right) = \frac{1}{2}\sum_{l=1}^{m}\sum_{v=1}^{V}\mathbf{w}^{[v]^{(l)T}}\mathbf{w}^{[v]^{(l)}} + \frac{1}{2}\sum_{l=1}^{m}\sum_{v=1}^{V}\gamma^{[v]^{(l)}}\mathbf{e}^{[v]^{(l)T}}\mathbf{e}^{[v]^{(l)}} + \rho\sum_{l=1}^{m}\sum_{\substack{v,u=1\\v\neq u}}^{V}\mathbf{e}^{[v]^{(l)T}}\mathbf{e}^{[u]^{(l)}} - \boldsymbol{\alpha}^{[v]^{(l)T}}\left(\mathbf{Z}^{[v]^{(l)T}}\mathbf{w}^{[v]^{(l)}} + \mathbf{y}^{(l)}\, b^{[v]^{(l)}} - \mathbf{1}_N + \mathbf{e}^{[v]^{(l)}}\right) \tag{9}$$

with conditions of optimality

$$\begin{cases} \dfrac{\partial\mathcal{L}}{\partial\mathbf{w}^{[v]^{(l)}}} = 0 \to \mathbf{w}^{[v]^{(l)}} = \mathbf{Z}^{[v]^{(l)}}\boldsymbol{\alpha}^{[v]^{(l)}}, \\ \dfrac{\partial\mathcal{L}}{\partial\mathbf{e}^{[v]^{(l)}}} = 0 \to \boldsymbol{\alpha}^{[v]^{(l)}} = \gamma^{[v]^{(l)}}\mathbf{e}^{[v]^{(l)}} + \rho\sum_{\substack{v,u=1\\v\neq u}}^{V}\mathbf{e}^{[u]^{(l)}}, \\ \dfrac{\partial\mathcal{L}}{\partial b^{[v]^{(l)}}} = 0 \to \mathbf{y}^{(l)^T}\boldsymbol{\alpha}^{[v]^{(l)}} = 0, \\ \dfrac{\partial\mathcal{L}}{\partial\boldsymbol{\alpha}^{[v]^{(l)}}} = 0 \to \mathbf{Z}^{[v]^{(l)T}}\mathbf{w}^{[v]^{(l)}} + \mathbf{y}^{(l)}\, b^{[v]^{(l)}} = \mathbf{1}_N - \mathbf{e}^{[v]^{(l)}}, \end{cases} \tag{10}$$

where $v = 1, \ldots, V$ and $l = 1, \ldots, m$. Eliminating the primal variables $\mathbf{w}^{[v]^{(l)}}$, $\mathbf{e}^{[v]^{(l)}}$ leads to the following dual problem:

$$\left[\begin{array}{c|c} \mathbf{0}_{V\times V} & \mathbf{Y}_{\mathbf{M}}^{(l)^T} \\ \hline \boldsymbol{\Gamma}_{\mathbf{M}}^{(l)}\mathbf{Y}_{\mathbf{M}}^{(l)} + \rho\,\mathcal{I}_{\mathbf{M}}\mathbf{Y}_{\mathbf{M}}^{(l)} & \boldsymbol{\Gamma}_{\mathbf{M}}^{(l)}\boldsymbol{\Omega}_{\mathbf{M}}^{(l)} + \mathbb{I}_{NV} + \rho\,\mathcal{I}_{\boldsymbol{M}}\boldsymbol{\Omega}_{\mathbf{M}}^{(l)} \end{array}\right]\left[\begin{array}{c} \mathbf{b}_{\mathbf{M}}^{(l)} \\ \hline \boldsymbol{\alpha}_{\mathbf{M}}^{(l)} \end{array}\right] = \left[\begin{array}{c} \mathbf{0}_V \\ \hline \boldsymbol{\Gamma}_{\mathbf{M}}^{(l)}\mathbf{1}_{NV} + (V-1)\rho\,\mathbf{1}_{NV} \end{array}\right] \tag{11}$$

for $l = 1, \ldots, m$ where $\mathbf{0}_V$ and $\mathbf{1}_{NV}$ are zero and one column vectors of dimension V and NV, respectively, $\mathbf{0}_{V\times V}$ is a zero matrix of dimension $V \times V$ and $\mathbb{I}_{NV}$ is the identity matrix of dimension $NV \times NV$. The other matrices are defined as follows:

$$\mathbf{Y}_{\mathbf{M}}^{(l)} = \text{blockdiag}\left\{\underbrace{\mathbf{y}^{(l)}, \ldots, \mathbf{y}^{(l)}}_{V \text{ times}}\right\} \in \mathbb{R}^{N\cdot V\times V}$$
$$\boldsymbol{\Gamma}_{\mathbf{M}}^{(l)} = \text{blockdiag}\left\{\boldsymbol{\gamma}_{\mathbf{M}}^{[1]^{(l)}}, \ldots, \boldsymbol{\gamma}_{\mathbf{M}}^{[V]^{(l)}}\right\} \in \mathbb{R}^{N\cdot V\times N\cdot V}$$
$$\boldsymbol{\gamma}_{\mathbf{M}}^{[v]^{(l)}} = \text{diag}\left\{\underbrace{\gamma^{[v]^{(l)}}, \ldots, \gamma^{[v]^{(l)}}}_{N \text{ times}}\right\} \in \mathbb{R}^{N\times N}$$
$$\mathcal{I}_{\boldsymbol{M}} = \begin{bmatrix} 0 & \mathbb{I}_N & \cdots & \mathbb{I}_N \\ \mathbb{I}_N & 0 & \cdots & \mathbb{I}_N \\ \vdots & \vdots & \ddots & \vdots \\ \mathbb{I}_N & \mathbb{I}_N & \cdots & 0 \end{bmatrix} \in \mathbb{R}^{N\cdot V\times N\cdot V}$$
$$\boldsymbol{\Omega}_{\mathbf{M}}^{(l)} = \text{blockdiag}\{\boldsymbol{\Omega}^{[1]^{(l)}}, \ldots, \boldsymbol{\Omega}^{[V]^{(l)}}\} \in \mathbb{R}^{N\cdot V\times N\cdot V}$$
$$\mathbf{b}_{\mathbf{M}}^{(l)} = \begin{bmatrix} b^{[1]^{(l)}} \\ \vdots \\ b^{[V]^{(l)}} \end{bmatrix} \in \mathbb{R}^V,\ \boldsymbol{\alpha}_{\mathbf{M}}^{(l)} = \begin{bmatrix} \boldsymbol{\alpha}^{[1]^{(l)}} \\ \vdots \\ \boldsymbol{\alpha}^{[V]^{(l)}} \end{bmatrix} \in \mathbb{R}^{N\cdot V}, \tag{12}$$

where $\boldsymbol{\alpha}^{[v]^{(l)}}$ are the dual variables. $\boldsymbol{\Omega}^{[v]^{(l)}}$ are the labeled kernel matrices where $\boldsymbol{\Omega}^{[v]^{(l)}} = Z^{[v]^{(l)T}}Z^{[v]^{(l)}}$ and

$$\begin{aligned}\Omega_{ij}^{[v]^{(l)}} &= y_i^{(l)}y_j^{(l)}\varphi^{[v]^{(l)}}\left(\mathbf{x}_i^{[v]}\right)^T\varphi^{[v]^{(l)}}\left(\mathbf{x}_j^{[v]}\right) \\ &= y_i^{(l)}y_j^{(l)}K^{[v]^{(l)}}\left(\mathbf{x}_i^{[v]}, \mathbf{x}_j^{[v]}\right)\end{aligned} \tag{13}$$

with the kernel functions $K^{[v]^{(l)}} : \mathbb{R}^{d^{[v]}} \times \mathbb{R}^{d^{[v]}} \to \mathbb{R}$ being positive definite.

It is clear that the multiclass problem can be decomposed into m binaryclass MV-LSSVM subproblems.

For ease of notation we will omit the $^{(l)}$ superscript when the statement is true for all subproblems.

3.2. Decision rule

The linear model stated in Eq. (11) will be solved based on available training data. The extracted dual variables $\boldsymbol{\alpha}^{[v]}$ and bias terms $b^{[v]}$ are used to construct the classifier $\hat{y}^{[v]}(\mathbf{x_t}^{[v]})$ that is able to classify a new unseen test data point $\mathbf{x_t}^{[v]}$. Let

$$g^{[v]}(\mathbf{x_t}^{[v]}) = \sum_{k=1}^{N}\alpha_k^{[v]}y_kK^{[v]}(\mathbf{x_t}^{[v]}, \mathbf{x}_k^{[v]}) + b^{[v]}$$

for each view v, the classifier can than be defined as:

$$\hat{y}(\mathbf{x_t}^{[v]}) = \text{sign}\left(\sum_{u=1}^{V}\beta_u\, g^{[u]}(\mathbf{x_t}^{[v]})\right), \tag{14}$$

which entails that $\hat{y}(\mathbf{x_t}^{[1]}) = \cdots = \hat{y}(\mathbf{x_t}^{[V]})$, so the classification is equal over all views. The value of β_u for each $u = 1, \ldots, V$ can be $1/V$, or can be calculated based on the error covariance matrix. In this last case the value of β_u can be chosen so that it

minimizes the error, similarly to how it is done for committee networks [18]. Alternatively, also the median could be considered. Since in our experiments, we generally noticed that taking the mean (hence $\beta_1 = \cdots = \beta_V = 1/V$) produces good results we will use this throughout the rest of the paper.

3.3. Model selection

To decrease tuning computational complexity we considered the same regularization parameter and kernel function (including parameters) for each binary subproblem, thus $\gamma^{[v]} = \gamma^{[v](1)} = \cdots = \gamma^{[v](m)}$ and $K^{[v]} = K^{[v](1)} = \cdots = K^{[v](m)}$. The resulting algorithm is described in Algorithm 1, where the superscript $^{[1:V]}$ is shorthand

Algorithm 1 MV-LSSVM.

Input: $\mathcal{X}^{[1:V]} = \{y_k^{(l)}, \mathbf{x}_k^{[1:V]}\}_{k=1,l=1}^{k=N,l=m}, K^{[1:V]}, \theta^{[1:V]}, \gamma^{[1:V]}, \quad \rho, \mathcal{X}_t^{[1:V]} = \{\mathbf{x_t}_k^{[1:V]}\}_{k=1}^{k=N_t}$

1: **for** $l = 1$ **to** m **do**
2: **for** $v = 1$ **to** V **do**
3: $\mathbf{\Omega}^{[v](l)} \leftarrow$ Eq. (13)$(\mathcal{X}^{[v]}, K^{[v]}, \theta^{[v]})$
4: **end for**
5: $\mathbf{b}_\mathbf{M}^{(l)}, \boldsymbol{\alpha}_\mathbf{M}^{(l)} \leftarrow$ Eq. (11)$(\mathbf{\Omega}^{[1:V](l)}, \gamma^{[1:V]}, \rho, \mathbf{y}^{(l)})$
6: $\hat{y}(\mathbf{x_t}^{[1:V]}) \leftarrow$ Eq. (14)$(\boldsymbol{\alpha}_\mathbf{M}^{(l)}, \mathbf{y}^{(l)}, \mathbf{b}_\mathbf{M}^{(l)}, K^{[1:V]}, \theta^{[1:V]}, \mathcal{X}_t^{[1:V]})$
7: **end for**

Output: $\hat{y}(\mathbf{x_t}^{[1:V]})$

for 'for all views $v = 1, \ldots, V$' and $\theta^{[1:V]}$ denote the kernel parameters (if any).

Algorithm 2 describes the model selection process. The parameters are found here by means of Simulated Annealing and 5-fold cross validation using only the training set. The model is evaluated using an independent test set $\mathcal{X}_t^{[1:V]}$ of size N_t.

Algorithm 2 Model selection.

Input: for $\mathcal{X}^{[1:V]} = \{y_k^{(l)}, \mathbf{x}_k^{[1:V]}\}_{k=1,l=1}^{k=N,l=m}$, $K^{[1:V]}, \mathcal{X}_t^{[1:V]} = \{\mathbf{x_t}_k^{[1:V]}\}_{k=1}^{k=N_t}$

1: $\theta^{[1:V]}, \gamma^{[1:V]}, \rho \leftarrow$ Simulated Annealing & 5-fold crossvalidation (Algorithm 1, $\mathcal{X}^{[1:V]}, K^{[1:V]}$) with criteria: classification accuracy
2: $\hat{y}(\mathbf{x_t}^{[1:V]}) \leftarrow$ Algorithm 1 $(\mathcal{X}^{[1:V]}, K^{[1:V]}, \theta^{[1:V]}, \gamma^{[1:V]}, \rho, \mathcal{X}_t^{[1:V]})$

Output: $\hat{y}(\mathbf{x_t}^{[1:V]})$

4. Experiments

In this section the results of MV-LSSVM are shown and compared to other multi-view classification methods. The results will be discussed on several synthetic and 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 datasets:* A number of synthetic datasets are generated, similar to the datasets described by [13]. All datasets consist of two views with data points belonging to one of three classes. The data in each view is generated by a three component Gaussian mixture model where the distributions slightly overlap. The distribution means for both views are $\boldsymbol{\mu_1} = (1\ 1)$, $\boldsymbol{\mu_2} = (2\ 6)$ and $\boldsymbol{\mu_3} = (-1.5\ 2)$. The covariances for the both views are

$$\boldsymbol{\Sigma_1} = \begin{pmatrix} 0.5 & 0.5 \\ 0.5 & 1.5 \end{pmatrix}, \boldsymbol{\Sigma_2} = \begin{pmatrix} 0.3 & 0 \\ 0 & 0.6 \end{pmatrix}.$$

For each view 999 points are sampled for training as well as for testing, 333 for each class. The encoding scheme used is one-versus-all. In some views noise is added. This is achieved by generating a certain rate of data points using a uniform distribution. Two types of datasets are considered.
  - *Noise:* The noise in the two views has the same noise rate η. By varying η from 0 to 0.5 with steps of 0.05, eleven datasets are generated. An example of such a dataset is given in Fig. 1 where $\eta = 0.50$.
  - *DiffNoise:* The noise rate of the second view $\eta^{[2]}$ equals $\eta^{[2]} = (3 - 2\eta^{[1]})/4$ where $\eta^{[1]}$ is the noise rate of the first view. Again, eleven datasets are generated where $\eta^{[1]}$ varies from 0 to 0.5 with steps of 0.05. An example of such a dataset is given in Fig. 2 where $\eta^{[1]} = 0$ and $\eta^{[2]} = 0.75$.
- *Flower species dataset:* This dataset, originally proposed by Nilsback and Zisserman [19,20], consist of 1360 images of 17 flower species segmented out from the background.[1] Like Minh et al. [14] we use the following seven features as views: HOG, HSV histogram, boundary SIFT, foreground SIFT, and three features derived from color, shape and texture vocabularies.
- *Image-caption web dataset:* This dataset consist of images retrieved from the Internet with their associated caption. We thank the authors of [6] for providing the dataset. The data is divided into three classes namely Sport, Aviation and Paintball images. For each class 400 records are provided. The data is represented by three views where the first two views represent two extracted feautures of the images (HSV colour and image Gabor texture)[2] and the third view consist of the term frequencies of the associated caption text.
- *YouTube Video dataset:* This dataset, originally proposed by Madani et al. [21], describes YouTube videos of video games by means of three high-level feature families: textual, visual and auditory features.[3] For this paper we selected the textual feature Latent Dirichlet Allocation (ran on all of description, title, and tags of the videos) and the visual feature Motion feature through Cuboid Interest Point Detection (for more details see the work of Yang and Toderici [22]) as two views. 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 classes) 300 videos were randomly sampled.

Table 1
Details of the datasets used in the experiments. N and Nt denote, respectively, the number of data points in the training and test set. V denotes the number of views and n_c the number of classes of the dataset.

Dataset	N	Nt	V	n_c	Encoding
Noise	999	999	2	3	1VsA
DiffNoise	999	999	2	3	1VsA
Flower species	1088	272	7	17	MOC
Image-caption web	960	240	3	3	1VsA
YouTube Video Games	1680	420	2	7	MOC
UCI Digits	1600	400	2	10	MOC
Reuters	29953	–	5	6	MOC

[1] The complete data is available at http://www.robots.ox.ac.uk/~vgg/data/flowers/17/index.html.

[2] Detailed description of these features can be found in Kolenda et al. [6].

[3] The data is available at http://archive.ics.uci.edu/ml/datasets/youtube+multiview+video+games+dataset.

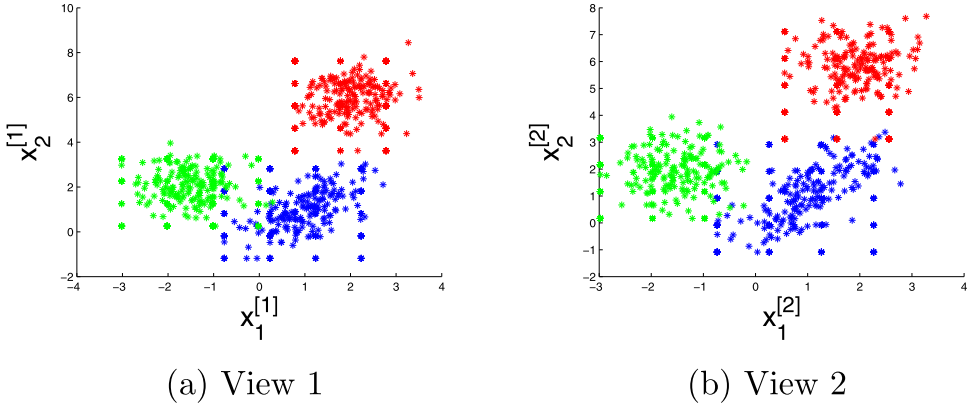


Fig. 1. A synthetic dataset of type Noise where the noise rate η equals 0.5 for both views.

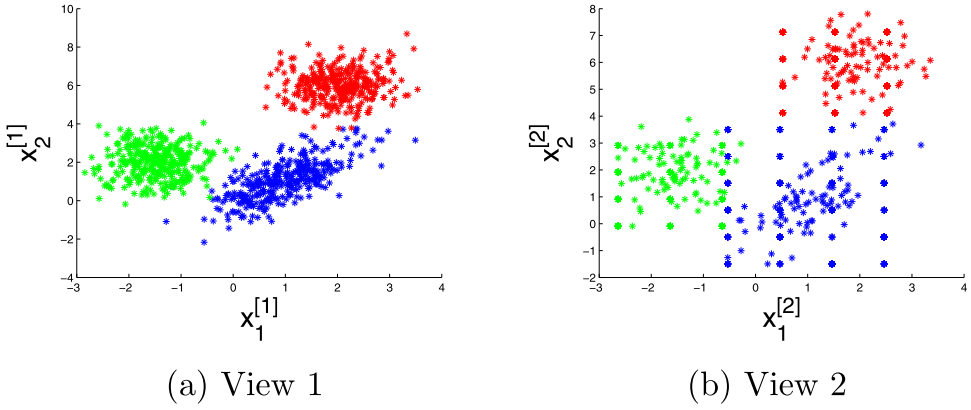


Fig. 2. A synthetic dataset of type DiffNoise where the noise rate differs for each view. This dataset was generated with $\eta^{[1]} = 0$ and $\eta^{[2]} = 0.75$.

- *UCI Digits dataset:* This dataset represent handwritten digits (0–9) and is taken from the UCI repository [23].[4] The dataset consist of 2000 digits which are represented through the profile correlations as view one and by the Fourier coefficients as view two.
- *Reuters dataset:* This multilingual text dataset is described by Amini et al. [24] and available through the UCI repository [23].[5] The dataset consist of documents originally written in five different languages and their translations in each of the other four languages, over a common set of six categories, represented by a bag-of-words style feature. We took the largest possible Reuters set, which consists of documents written in German for one view and translations of them in English, French, Spanish and Italian for the other four views. This set contains 29,953 documents and the dimension of the data over the views range from 11,547 to 34,279.

For the Flower species dataset data is already provided as kernels. For the synthetic datasets, the UCI Digits dataset and the first two views of the Image-caption web dataset, the radial basis function (RBF) kernel is chosen. For the YouTube Video dataset, the Reuters dataset and the third view of the Image-caption web dataset, the features are sparse and very high dimensional, using an RBF kernel, and hence bringing the data to a even higher feature space, is not recommended [25]. Therefore a linear kernel is chosen for these views. Since this simple kernel function resulted in a good performance other appropriate kernel functions for text-data such as polynomial kernels of a order two, Chi-square kernels [26] or String kernels [27] were not considered.

For the real world datasets (except for the large Reuters dataset, see Section 4.6), the data is randomly divided into a test and training set three times where 80% of the data belongs to the training set. The results shown are averaged over the three splits.

4.2. Baseline algorithms

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

- *Best Single View (BSV):* The results of applying classification on the most informative view, i.e., the one on which LS-SVM achieves the best performance.
- *Feature Concatenation (FC):* Early fusion where the features of all views are concatenated and LS-SVM is used to do classification on this concatenated view representation.
- *Kernel Addition (KA):* Early fusion where for each view an appropriate kernel matrix is constructed in the same way as for MV-LSSVM but the kernels are simply combined by adding them. LS-SVM is then used to do the classification on the combined kernels.
- *Committee LS-SVM (Comm):* A typical example of late fusion where a separate model is trained for each view and a weighted average is taken as the final classifier [9]. For this baseline method, LS-SVM is applied on each view separately and the final classifier is defined in the same way as for MV-LSSVM (Eq. (14)) (although notice that for Committee LS-SVM the views are completely independently trained). The weights are calculated based on the error covariance matrix in the

[4] The data is available at https://archive.ics.uci.edu/ml/datasets/Multiple+Features.
[5] The data is available at https://archive.ics.uci.edu/ml/datasets/Reuters+RCV1+RCV2+Multilingual,+Multiview+Text+Categorization+Test+collection.

Table 2
Figures from the test set, misclassified only by LS-SVM using the first or second view (the image views), and misclassified only by using the third view (the caption view). The table further shows the correct class the figure belongs to and the incorrect prediction made by LS-SVM. Notice that MV-LSSVM is able to correctly classify all four figures.

	Miss by image views only		Miss by caption view only	
Class Prediction	Aviation Sport	Paintball Aviation	Sport Paintball	Sport Aviation
Image				
Caption	Check out the various kinds of seals that were seen on the antarctic voyage of the mv hanseatic.	New paintball strategy boards on the market by bambi bullard	feb 9, 19:43 pt	Kwan celebrates at the end of her preformance. Kwan takes first step towards gold michelle kwan did not make many friends among the judges, but she's closer to making gold. The american star won the short program in the marquee ladies' figure skating competition tuesday night, moving a step closer to that elusive gold medal. Four years after winning silver in nagano, kwan turned in one of five exceptionally clean performances in a talented field and won five of nine judges for a slight edge over irina slutskaya of russia and fellow american sasha cohen. Meanwhile, olympic pairs co-champions jamie sale and david pelletier announced that they will give their first post-salt lake city skating exhibition in edmonton on march 12.

same way as for Committee LS-SVM regression as described by Suykens et al. [15].

- *Multi-View Learning with Least Square loss function (MVL- LS):* This method, proposed by Minh et al. [14], is a multi-view classification model based on SVM that can handle labeled as well as unlabeled data (semi-supervised setting). To fairly compare with our proposed MS LS-SVM method we use the same (labeled) data and do not add unlabeled data. The method has three regularization parameters as well as kernel parameters to be tuned.
- *SimpleMKL:* This method is a multiple kernel learning method based on SVM, proposed by Rakotomamonjy et al. [28], where the kernel is defined as a linear combination of multiple kernels. The SimpleMKL problem is defined through a weighted 2-norm regularization formulation with a constraint on the weights that encourages sparse kernel combinations. The method has one regularization parameter (for the resulting SVM problem) as well as kernel parameters to be tuned.

The parameters of the baseline algorithms are selected in the same way as MV-LSSVM (see Algorithm 2).

4.3. Influence coupling term

To show the importance of the coupling term, we look at some results on the Image-caption web dataset. To do this we ran LS-SVM on all views separate and MV-LSSVM on all views together. Table 2 shows two figures that were incorrectly classified by LS-SVM on the first two views only, which describe the image, and two figures that were incorrectly classified by LS-SVM on the third view only, which describes the associated caption. Notice that MV-LSSVM was able to correctly classify all the figures in Table 2.

Looking at the first two figures, it is clear to see why LS-SVM is not able to classify them well, based only on the image. The image of the seal is very different from the other images belonging to the Aviation class, which are mostly planes. The strategy board is also a rather unusual image in the class Paintball, which mostly contains images of people playing paintball. The associated captions, however, contain some key words of both classes (like 'voyage' for the first figure and 'paintball' for the second), so it is not surprising that LS-SVM classifies them well using the caption view. The last two figures however have images very typical to the class sport, and hence LS-SVM is able to classify them well using the image views. Using the caption view, however, LS-SVM classifies the figures incorrectly. The third figure's caption is just a timestamp, so it is impossible to classify using only this. The fourth caption is very long, hence it contains a lot of terms so it is harder to know which are important, and it contains some words that are also used to describe images of planes (e.g. 'american', 'star', 'field' etc.).

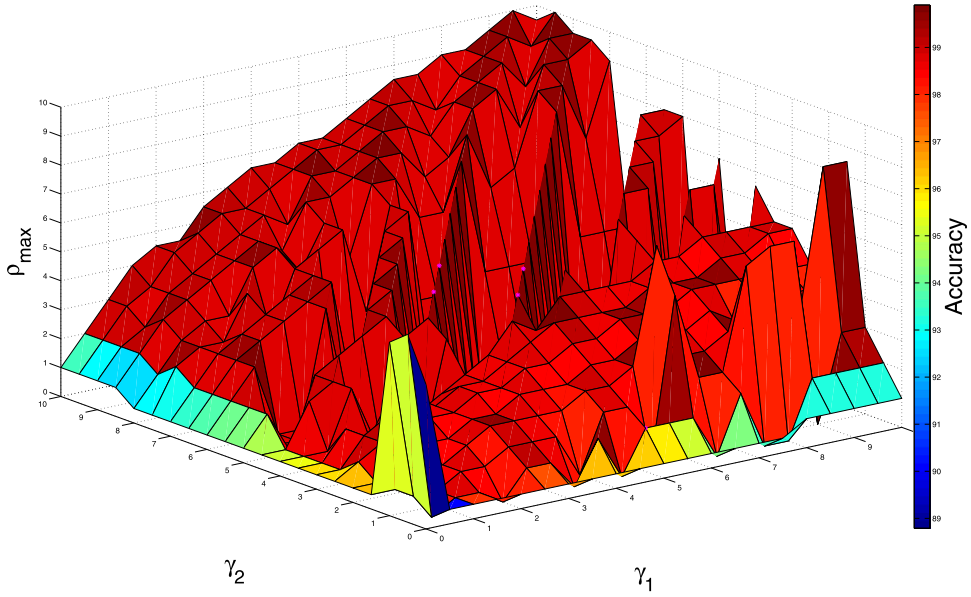
By introducing the coupling term in Eq. (8), MV-LSSVM is able to incorporate the information from both the image views and the caption view. By minimizing the product of the error variables it can allow for a larger error variable in one view, if it is compensated by the other view. In this example this means that e.g. the error variable for the first figure might be high for the first two views, but it will be low for the third view. The influence of the coupling term is controlled by the coupling parameter ρ, and its influence is discussed in the next session.

4.4. Parameter study

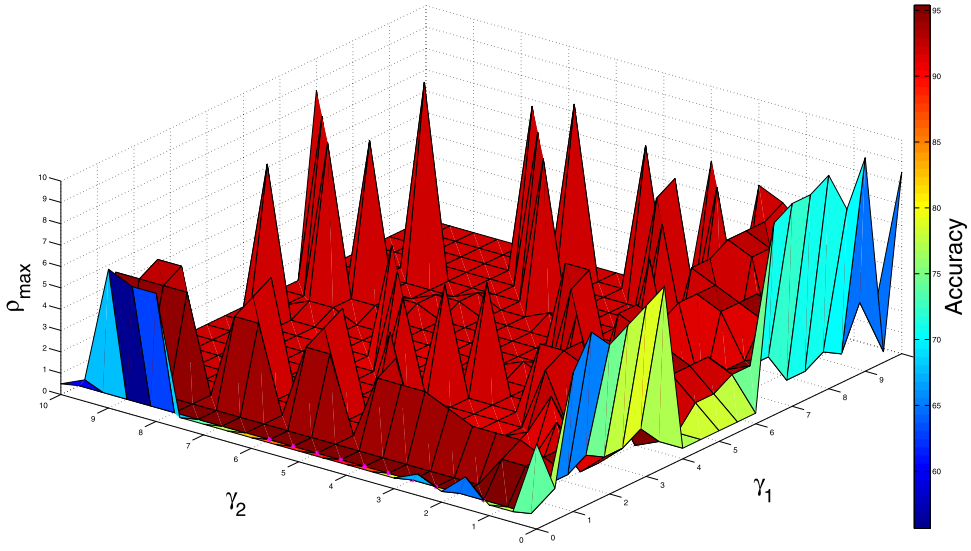
In order to study the influence of the coupling parameter ρ we looked at two synthetic datasets from the type DiffNoise, namely the dataset with the biggest difference in noise rate (Fig. 2) and the dataset with the same noise rate (Fig. 1) in both views. By taking these datasets we are able to compare the influence of the coupling parameter when the information in both views is very different to when the views are alike.

For both datasets we varied ρ, $\gamma^{[1]}$ and $\gamma^{[2]}$ from 0 to 10 with steps of 0.5 and computed the accuracy on the test data. Some results are visualized in Fig. 3.

Fig. 3 shows the value of ρ corresponding to the highest obtained accuracy for each combination of $\gamma^{[1]}$ and $\gamma^{[2]}$. The color indicates this accuracy. A first observation is that ρ generally has a higher value for the dataset where the views differ much than for the dataset where the views are alike. Fig. 3a further shows that a high value of ρ usually corresponds to a high accuracy and that a high accuracy is mostly related to a high value of ρ. We can also



(a) $\eta^{[1]} = 0,\ \eta^{[2]} = 0.75$



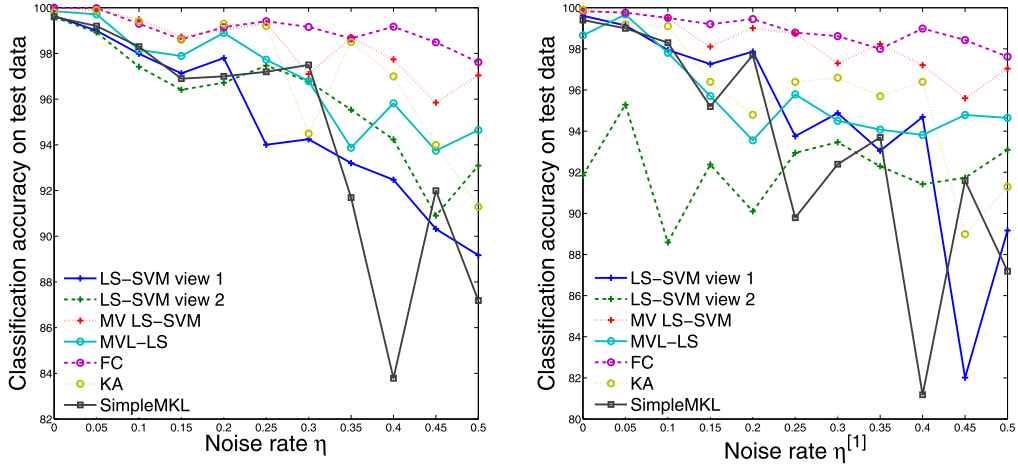
(b) $\eta^{[1]} = \eta^{[2]} = 0.5$

Fig. 3. The value of ρ corresponding to the highest obtained accuracy on test data for each combination of $\gamma^{[1]}$ and $\gamma^{[2]}$. The color indicates the accuracy. The purple asterisks indicate the combinations corresponding to the overall highest accuracy.

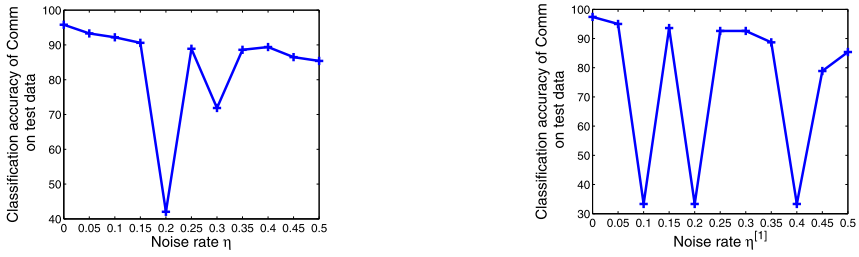
see that when $\gamma^{[1]}$ and $\gamma^{[2]}$ are increasing, the value of ρ corresponding to the maximum accuracy also increases. This correlation is especially clear for $\gamma^{[2]}$. This indicates that the model puts a high importance in minimizing the combined error variables (large ρ) and the error variables belonging to the view with the most noise (large $\gamma^{[2]}$).

Fig. 3b on the other hand shows a very different influence of ρ. This graph shows that ρ is usually high when $\gamma^{[1]}$ or $\gamma^{[2]}$ is rather low (except for a few peaks) and that a high accuracy is mostly related to a low value of ρ. It also shows that ρ does not increase when $\gamma^{[1]}$ and $\gamma^{[2]}$ do, but instead has rather high values when $\gamma^{[1]}$ or $\gamma^{[2]}$ have a very low value. So the model only puts a rather high importance on minimizing the combined error variables (rather large ρ) when a very low importance is put on minimizing the error variables of one of the two views ($\gamma^{[1]}$ or $\gamma^{[2]}$ very small). In fact the best results are obtained when $\rho = 0$, which results in no coupling.

These results indicate that the coupling term in the primal formulation (Eq. (8)) is of most importance when the views provide enough diverse information. It also indicates that when the information from the different views is too similar, the multi-view method is less suited. This is line with the findings about another type of fusion, namely committee networks where the independent models can not be too similar [9].

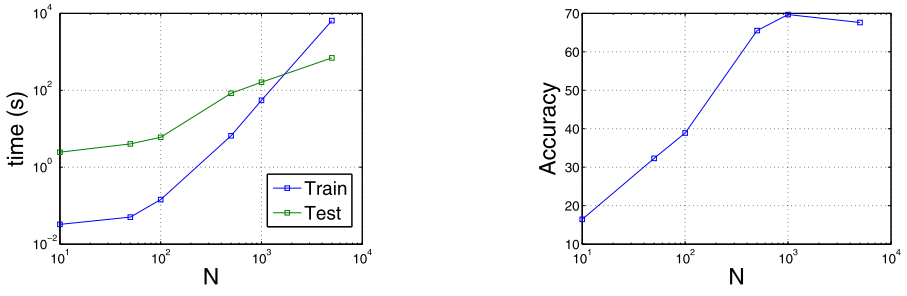


(a) Noise dataset (b) DiffNoise dataset



(c) Noise dataset - Comm (d) DiffNoise dataset - Comm

Fig. 4. Classification accuracy on test data with respect to the noise rate for the two types of synthetic datasets.



(a) Runtime (s) for the training and test phase of MV-LSSVM. (b) Classification accuracy on the test data.

Fig. 5. Timing and classification accuracy results for the large-scale Reuters dataset when the number of training instances is increased from $N = 10^1$ to $N = 0.5 \cdot 10^4$.

4.5. Experimental results

Fig. 4 show the classification accuracy as a function of the noise rates for both types of synthetic datasets. The figures show the performance of an LS-SVM classifier applied on each view separately, the performance of the proposed MV-LSSVM method and the performances of the baselines algorithms MVL-LS, FC, KA, SimpleMKL and Comm.

Fig. 4a and c shows a decrease in accuracy as the noise rate increases. This is expected since the classes are harder to classify when more noise is present. It further shows that MV-LSSVM is able to outperform LS-SVM on the separate views for all noise rates.

In Fig. 4b it is also visible that MV-LSSVM performs better than the BSV model. Even in the extreme case where $\eta^{[1]} = 0$ and $\eta^{[2]} = 0.75$, MV-LSSVM using the two views obtains better results than applying LS-SVM on only the view with no noise.

Both figures show that for this synthetic dataset MV-LSSVM is competitive with the multiview method MVL-LS and SimpleMKL and obtains a higher accuracy for most noise rates.

The results further show that for these synthetic datasets the early fusion techniques with simple coupling schemes FC and KA performing well. FC even outperforms the proposed multi-view method for most noise rates. Intuitively the reason for this good performance is that this synthetic data from both views is rather similar, in the sense that it is simple 2-dimensional data drawn from the same Gaussian mixture models (albeit with different noisy samples). The degree of freedom to model the views differently which the multi-view methods offers is hence not that important and the early fusion techniques will work very well. In

Table 3
Classification accuracy on the test set for the real-world datasets. The standard deviation is shown between brackets. The highest accuracies are indicated in bold.

Method	Flower species	Image-caption	YouTube video	UCI digits
BSV	44.31 (±1.96)	96.81 (±0.24)	90.95 (±1.04)	74.25 (±10.03)
FC	5.49 (±1.39)	79.17 (±1.50)	93.25 (±2.54)	10.42 (±0.88)
KA	22.16 (±1.28)	32.5 (±3.31)	92.30 (±3.91)	75.83 (±0.38)
MVL-LS	8.43 (±4.42)	97.5 (±1.82)	91.13 (±5.23)	**83.87** (±1.12)
Comm	30.98 (±14.3)	32.5 (±3.31)	72.38 (±10.01)	67.42 (±8.52)
SimpleMKL	10.88 (±5.88)	97.64 (±1.27)	94.16 (±1.12)	73.50 (±6.36)
MV-LSSVM	**49.91** (±3.40)	**98.06** (±1.97)	**95.40** (±0.96)	75.92 (±1.01)

real-world datasets however, the data from different views is usually not much alike (e.g. image and text data) and, as we will show further on, these simple coupling schemes will not be sufficient anymore.

This of course entails that the late fusion technique Comm will also not work well. Fig. 4c and d show the results for Comm. The results are shown in a different graph because of the poor results which would hinder the visibility of the figures. As expected, Comm does not perform well on the synthetic datasets, the accuracy is even lower than the single view LS-SVM for most noise rates.

Table 3 shows the accuracy of all baseline algorithms and of the proposed MV-LSSVM model on the real-world datasets.

The BSV results was obtained with the foreground SIFT features for the Flowers species dataset, with the term frequencies of the caption for the Image-caption dataset, with the Latent Dirichlet Allocation text feature for the YouTube Video dataset and with the Fourier coefficients for the UCI Digits dataset. When applying FC on the Image-caption database an RBF as well as a linear kernel were considered. The best result was achieved using a linear kernel thus only this result is reported here.

As for the results on the synthetic datasets, Table 3 shows the improvement of using multiple views. For all four real-world datasets MV-LSSVM obtains a higher classification accuracy than the BSV method and for three of the four datasets it is able to outperform all the considered baseline algorithms. Where the simple coupling schemes KA and FC performed well on the synthetic datasets, it is clear that they are insufficient for the real-world datasets. In fact these simple coupling schemes do not improve on using only the best view. Table 3 also shows that MV-LSSVM is definitely able to compete with SimpleMKL and with the state-of-the-art method MVL-LS, since it is able to obtain a higher accuracy on the Flower species, Image-caption and YouTube Video datasets. Only on the UCI Digits dataset, MVL-LS outperforms our method, although MV-LSSVM is still the second best method for this dataset.

4.6. Complexity and large-scale experiment

To investigate the behavior of MV-LSSVM when dealing with large-scale data, we use the Reuters dataset.

The time complexity of MV-LSSVM can be split into two parts, namely the training part (lines 2–5 in Algorithm 1) and the test part (line 6 in Algorithm 1). The training part consist of two time-consuming steps, the calculation of the kernel matrices (Eq. (13)) and the solving of the linear system in Eq. (11). This first step has a time complexity of $O(VN^2\bar{d})$, where $\bar{d}$ is the mean of the data dimensions over all views. In real-life datasets V is rarely larger than 10 and hence usually $V \ll N$. The dimensions of the datasets are usually either small or the features are very sparse (as is the case for the Reuters dataset). Since most numeric programming languages have fast routines to multiply sparse matrices (like e.g. Matlab), one can usually assume a complexity of $O(N^2)$. The second step of the training part is of time complexity $O(N^3V^3)$. The complete time complexity of the training part can hence be considered $O(N^3)$. The test part consist of calculation of the test kernel matrices (Eq. (13)) and the calculation of the classifier (Eq. (14)). These steps have a time complexity of $O(VNN_t\bar{d})$ and $O(VN_t)$, respectively. The complete time complexity of the training part can hence be considered $O(NN_t)$.

This complexity study shows that when N is very large, the training part will take up a lot of time. To deal with this, a common approach is to train on only a small part of the data and assume the model will generalize well to the unseen test data. For the Reuters dataset we looked at the runtime of the training and test phase of MV-LSSVM and the accuracy on the test set for $N \in \{10^1,\ 0.5\cdot 10^2,\ 10^2,\ 0.5\cdot 10^3,\ 10^3,\ 0.5\cdot 10^4\}$. We randomly chose N datapoints from the total set (of size 29,953), and the remaining datapoints are considered as the test set. The size of the test data will hence be $N_t \in \{29,943; 29,903; 29,853; 29,453; 28,953; 24,953\}$.

Fig. 5 displays the results of the experiments on the Reuters dataset. The runtime results in Fig. 5a show that the training time increases rapidly with the number of training points, which is in line with the found complexity of $O(N^3)$. The test time also increases with the number of training points, but a lot less fast. This is again in line with the complexity of the test part $O(NN_t)$. We can see that when $N \geq 10^3$ the training phase takes a lot of time, even more than the testing time. However in Fig. 5b it is clear that the test accuracy does not improve drastically when $N > 0.5\cdot 10^3$ so the model seems to generalize well with a relatively small training size.

These results indicate that MV-LSSVM can be efficiently used on the large Reuters dataset by the simple approach of using a relatively small training set. Of course for some other datasets, there might still be the need to train a model on a large number of instances. For this purpose a variety of single-view models have been developed in the past, like fixed-size LS-SVM [29], the use of ensemble methods [30] or using a weighted linear loss function [31], which deal with a large number of training samples. It might be useful to extend these techniques to the MV-LSSVM method in the future.

5. Conclusion and perspectives

In this paper we proposed a new model called Multi-View Least Squares Support Vector Machines (MV-LSSVM) Classification that exploits information from two or more views when performing classification. The model is based on LS-SVM classification where coupling of the different views is obtained by an additional coupling term in the primal model. The aim of this new model is to improve the classification accuracy by incorporating information from multiple views. The model is tested on synthetic and real-world datasets where the obtained results show the improvement of using multiple views. It also shows that the proposed model MV-LSSVM is able to outperform some early and late fusion methods and some state-of-the-art multi-view methods on real-world datasets. The complexity of the method is discussed and results

on a large-scale dataset are discussed. A parameter study shows that the model is particularly suited when the information from the views is diverse enough.

Acknowledgments

The research leading to these results has received funding from the European Research Council under the European Union's 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); Ph.D./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, in: Proceedings of the International Conference on Machine Learning, 2009, pp. 129–136.
[2] C. Xu, D. Tao, C. Xu, A survey on multi-view learning, eprint arXiv:1304.5634 (2013).
[3] A. Blum, T. Mitchell, Combining labeled and unlabeled data with co-training, in: Proceedings of the Conference on Learning Theory, 1998, pp. 92–100.
[4] X.-Y. Jing, Q. Liu, F. Wu, B. Xu, Y. Zhu, S. Chen, Web page classification based on uncorrelated semi-supervised intra-view and inter-view manifold discriminant feature extraction, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2015, pp. 2255–2261.
[5] Y. Yang, C. Lan, X. Li, J. Huan, B. Luo, Automatic social circle detection using multi-view clustering, in: Proceedings of the ACM Conference on Information and Knowledge Management (CIKM), 2014, pp. 1019–1028.
[6] T. Kolenda, L.K. Hansen, J. Larsen, O. Winther, Independent component analysis for understanding multimedia content, in: Proceedings of IEEE Workshop on Neural Networks for Signal Processing, 12, 2002, pp. 757–766.
[7] R.D. Zilca, Y. Bistritz, Feature concatenation for speaker identification, in: Proceedings of the 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] M.P. Perrone, L.N. Cooper, in: When Networks Disagree: Ensemble Methods for Hybrid Neural Networks, Chapman and Hall, 1993, pp. 126–142.
[10] A. Bekker, M. Shalhon, H. Greenspan, J. Goldberger, Multi-view probabilistic classification of breast microcalcifications, IEEE Trans. Med. Imaging 35 (2) (2016) 645–653.
[11] M. Mayo, E. Frank, Experiments with multi-view multi-instance learning for supervised image classification, in: Proceedings of the Image and Vision Computing New Zealand (IVCNZ), 2011, pp. 363–369.
[12] M. Wozniak, K. Jackowski, Some Remarks on Chosen Methods of Classifier Fusion Based on Weighted Voting, Springer Berlin Heidelberg, Berlin, Heidelberg, 541–548.
[13] S. Koço, C. Capponi, A boosting approach to multiview classification with cooperation, in: Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases, 2, 2011, pp. 209–228.
[14] H.Q. Minh, L. Bazzani, V. Murino, A unifying framework in vector-valued reproducing kernel hilbert spaces for manifold regularization and co-regularized multi-view learning, J. Mach. Learn. Res. 17 (2016) 1–72.
[15] J.A.K. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor, J. Vandewalle, Least Squares Support Vector Machines, World Scientific, 2002.
[16] V. Vapnik, The Nature of Statistical Learning Theory, Springer-Verlag, New-York, 1995.
[17] J. Mercer, Functions of positive and negative type, and their connection with the theory of integral equations, in: Proceedings of the Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character, 209, 1909, pp. 415–446.
[18] C.M. Bishop, Neural Networks for Pattern Recognition, Oxford University Press, 1995.
[19] M.-E. Nilsback, A. Zisserman, A visual vocabulary for flower classification, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2, 2006, pp. 1447–1454.
[20] M.-E. Nilsback, A. Zisserman, Automated flower classification over a large number of classes, in: Proceedings of the Indian Conference on Computer Vision, Graphics and Image Processing (ICVGIP), 2008, pp. 722–729.
[21] O. Madani, M. Georg, D.A. Ross, On using nearly-independent feature families for high precision and confidence, Mach. Learn. 92 (2013) 457–477.
[22] W. Yang, G. Toderici, Discriminative tag learning on youtube videos with latent sub-tags, in: Proceedings of the Computer Vision and Pattern Recognition (CVPR), 2011, pp. 3217–3224.
[23] Lichman, M. (2013). UCI Machine Learning Repository. Irvine, CA: University of California, School of Information and Computer Science. http://archive.ics.uci.edu/ml.
[24] M.-R. Amini, N. Usunier, C. Goutte, Learning from multiple partially observed views – an application to multilingual text categorization, in: Proceedings of the Advances in Neural Information Processing Systems, 2009, pp. 28–36.
[25] C.-W. Hsu, C.-C. Chang, C.-J. Lin, A practical guide to support vector classification, in: Technical report, University of National Taiwan, Department of Computer Science and Information Engineering, 2003, pp. 1–12.
[26] P. Li, G. Samorodnitsky, J. Hopcroft, Sign Cauchy projections and Chi-square kernel, in: Proceedings of the Advances in Neural Information Processing Systems, 26, 2013, pp. 2571–2579.
[27] H. Lodhi, C. Saunders, J. Shawe-Taylor, N. Cristianini, C. Watkins, Text classification using string kernels, J. Mach. Lear. Res. 2 (2002) 419–444.
[28] A. Rakotomamonjy, F.R. Bach, S. Canu, Y. Grandvalet, SimpleMKL, J. Mach. Learn. Res. 9 (2008) 2491–2521.
[29] M. Espinoza, J.A.K. Suykens, B.D. Moor, Fixed-size least squares support vector machines: a large scale application in electrical load forecasting, Comput. Manag. Sci. 3 (2) (2006) 113–129, doi:10.1007/s10287-005-0003-7.
[30] A. Schwaighofer, V. Tresp, The Bayesian Committee Support Vector Machine, Springer Berlin Heidelberg, Berlin, Heidelberg, 411–417.
[31] Y.-H. Shao, Z. Wang, Z.-M. Yang, N.-Y. Deng, Weighted linear loss support vector machine for large scale problems, Procedia Comput. Sci. 31 (2014) 639–647. 2nd International Conference on Information Technology and Quantitative Management. 10.1016/j.procs.2014.05.311.



Lynn Houthuys was born in Leuven Belgium, May 23, 1990. In 2011 she received a Bachelor's degree in Informatics and in 2013 a Master's degree in Engineering Computer Science with the thesis ; "Parallelization of tensor computations through OpenCL", both at the KU Leuven. She is currently a doctoral student in machine learning, at the STADIUS research division of the Department of Electrical Engineering (ESAT) at KU Leuven, under the supervision of prof. Johan A. K. Suykens. Currently Lynn serves as a teaching assistant for several courses involving neural networks and support vector machines, included in the master programs organized by the KU Leuven. Lynn's scientific interests include multi-view learning, kernel methods, neural networks, multi-task learning and coupled data-driven models in general.



Rocco Langone was born in Potenza, Italy, in 1983. He received the bachelors degree in physics and information technology, the masters degree in physics with the thesis titled A Neural Network Model for Studying the Attribution of Global Circulation Atmospheric Patterns on the Climate at a Local Scale, and the second masters degree in scientific computing with the thesis titled Stochastic Volatility Models for European Calls Option Pricing from the Sapienza University of Rome, Rome, Italy, in 2002, 2008, and 2010, respectively. He was a Researcher with the National Research Council, Rome, until 2008, where he developed neural networks models for climate studies. He was a Ph.D. fellow in machine learning from 2010 to 2014 and after, for two years, a postdoctoral researcher in machine learning with the STADIUS Research Division, Department of Electrical Engineering, KU Leuven. In this period his research focused on kernel methods, optimization, unsupervised learning (clustering and community detection), big data, fault detection. He is currently a data scientist at Deloitte Belgium where he builds machine learning models for several business applications.



Johan A.K. Suykens was born in Willebroek Belgium, May 18 1966. He received the master degree in Electro-Mechanical Engineering and the Ph.D. degree in Applied Sciences from the Katholieke Universiteit Leuven, in 1989 and 1995, respectively. In 1996 he has been a Visiting Postdoctoral Researcher at the University of California, Berkeley. He has been a Postdoctoral Researcher with the Fund for Scientific Research FWO Flanders and is currently a full Professor with KU Leuven. He is author of the books Artificial Neural Networks for Modelling and Control of Non-linear Systems (Kluwer Academic Publishers) and Least Squares Support Vector Machines (World Scientific), co-author of the book Cellular Neural Networks, Multi-Scroll Chaos and Synchronization (World Scientific) and editor of the books Nonlinear Modeling: Advanced Black-Box Techniques (Kluwer Academic Publishers), Advances in Learning Theory: Methods, Models and Applications (IOS Press) and Regularization, Optimization, Kernels, and Support Vector Machines (Chapman & Hall/CRC). In 1998 he organized an International Workshop on Nonlinear Modelling with Time-series Prediction Competition. He has served as associate editor for the IEEE Transactions on Circuits and Systems (19971999 and 20042007), the IEEE Transactions on Neural Networks (19982009) and the IEEE Transactions on Neural

Networks and Learning Systems (from 2017). He received an IEEE Signal Processing Society 1999 Best Paper Award and several Best Paper Awards at International Conferences. He is a recipient of the International Neural Networks Society INNS 2000 Young Investigator Award for significant contributions in the field of neural networks. He has served as a Director and Organizer of the NATO Advanced Study Institute on Learning Theory and Practice (Leuven 2002), as a program co-chair for the International Joint Conference on Neural Networks 2004 and the International Symposium on Nonlinear Theory and its Applications 2005, as an organizer of the International Symposium on Synchronization in Complex Networks 2007, a co-organizer of the NIPS 2010 workshop on Tensors, Kernels and Machine Learning, and chair of ROKS 2013. He has been awarded an ERC Advanced Grant 2011 and has been elevated IEEE Fellow 2015 for developing least squares support vector machines.