Multi-view Multi-class Classification for Identification of Pathogenic Bacterial Strains

Evgeni Tsivtsivadze[1], Tom Heskes[2], and Armand Paauw[1]

[1] MSB Group, The Netherlands Organization for Applied Scientific Research, Zeist, The Netherlands
firstname.lastname@tno.nl
[2] Institute for Computing and Information Sciences, Radboud University, The Netherlands
firstname.lastname@science.ru.nl

Abstract. In various learning problems data can be available in different representations, often referred to as views. We propose multi-class classification method that is particularly suitable for multi-view learning setting. The algorithm uses co-regularization and error-correcting techniques to leverage information from multiple views and in our empirical evaluation notably outperforms several state-of-the-art classification methods on publicly available datasets. Furthermore, we apply the proposed algorithm for identification of the pathogenic bacterial strains from the recently collected biomedical dataset. Our algorithm gives a low classification error rate of 5%, allows rapid identification of the pathogenic microorganisms, and can aid effective response to an infectious disease outbreak.

1 Introduction

Frequently the problem at hand requires considering a classification task involving more than two classes, namely when the label y is chosen from the set $\mathcal{Y} = \{1, \ldots, \kappa\}$, where $\kappa > 2$. A number of methods have been proposed to deal with this problem and they can be divided into two main categories: *i*) methods that reduce the multi-class problem into simpler binary classification tasks and combine the obtained results afterwards (e.g. [1–3]) and *ii*) genuine multi-class classification algorithms (e.g. [4, 5, 3]) that learn a single function for discriminating between the multiple classes. In [3], authors provide a detailed overview of the methods that are frequently used for multi-class classification. They refer to the algorithms that learn a single function for discriminating between different classes as a "single machine" approach. For instance, in the one-versus-all method (e.g. [3]) the aim is to create a binary classification problem such that examples $y = l_1$ belong to the positive class and all other examples having class labels $l_{2,\ldots,\kappa}$ belong to the negative class. Another way to deal with the multi-class learning problem is described in [2], where all possible pairs of classes $l_1, l_2 \in \mathcal{Y}$ are considered. This means that $\binom{\kappa}{2}$ hypotheses have to be generated and combined. The method is referred to as the all pairs approach.

A more general suggestion on how to treat multi-class classification problem was proposed in [1] and later extended in [6]. It is known as the error-correcting output codes (ECOC) approach. The key idea is to construct the coding matrix $C \in \{-1, +1\}^{\kappa \times p}$, where p is some positive integer, such that the rows of the matrix have good error correcting properties (e.g. a large Hamming distance). The binary learning algorithm is then run once for each column of the output matrix, whose rows correspond to the encodings of the appropriate y labels induced by the coding matrix C. For example, this setting is a generalization of one-versus-all scheme and it can be represented with $\kappa \times \kappa$ coding matrix having all diagonal elements equal to 1 and all other elements equal to -1. Given a new example $\mathbf{x}'$ from input space $\mathcal{X}$, we can predict the corresponding label y' by finding the row of the coding matrix that is "closest" to $\mathbf{f} = (f_1(\mathbf{x}') \dots f_p(\mathbf{x}'))$, where $f_s(\mathbf{x}'), s = 1, \dots, p$ are the prediction functions constructed for each column of the output matrix.

Another group of algorithms that can be considered as genuine multi-class classifiers is described in e.g. [7, 4, 5]. These algorithms learn a single prediction function that can properly discriminate between different classes. Some of the algorithms (e.g. [4, 5]) use so-called joint feature maps/views $\Phi(\mathbf{x}, y)$ on the data from $\mathcal{X}$ and labels $\mathcal{Y}$ to learn the prediction function. In this case the predicted class is the one that maximizes the output of the learnt function for the new data point. We note that the initial problem considered in [4] is structured output prediction and the presented formulation allows considering multi-class classification as a special case. In fact, using multiple views for learning has been shown to be beneficial for the predictive performance of the algorithm in many tasks beyond multi-class classification (e.g. [8]). Multi-class classification algorithms that construct a single prediction function appear to have good generalization performance and are usually computationally more efficient compared to approaches that need to train several binary classifiers to solve the problem. On the other hand, methods such as ECOC have the attractive property of error correction, they are extensively used in practice and continuously improved (see, for example, a recent work on decreasing number of binary classifiers needed for class prediction [9]).

This work aims to combine the benefits of the methods described above and presents a novel multi-class classification algorithm that is particularly suitable for multi-view learning setting. We extends co-regularization [10] framework to be applicable to multi-class problems as well as describe a loss-based decoding approach for estimating class labels when using multiple views/feature representations. In our empirical evaluation on publicly available datasets from the UCI, Statlog, and other repositories proposed algorithm outperforms several state-of-the-art multi-class classification methods. Furthermore, we apply our algorithm to the task of identification of pathogenic bacterial species from the dataset containing MS spectra of highly infectious Brucella microorganism [11]. The genus Brucella contains infectious species that have been found to cause infections in a wide variety of mammals. Most Brucella species have a narrow host range. Infection in humans arises from direct or indirect contact with infected animals

or through consumption of contaminated meat or dairy products. We demonstrate that our algorithm gives a low classification error rate of 5% and allows rapid identification of the pathogenic strains. Proposed algorithm can be helpful for timely and effective response to an infectious disease outbreak, regardless of whether the outbreak is natural or deliberate.

2 Preliminaries

In the following subsections we briefly describe our framework for constructing our multi-view multi-class classification algorithm and outline related research directions. We are interested in the selection of suitable prediction function $f \in \mathcal{H}$. Following [12], we consider the reproducing kernel Hilbert space (RKHS) determined by the input space $\mathcal{X}$ and the positive definite kernel function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. Using the RKHS $\mathcal{H}$ as our hypothesis space, we consider the optimization problem

$$\min_{f \in \mathcal{H}} J(f) = c(f, \mathcal{D}) + \lambda \|f\|_{\mathcal{H}}^2 \tag{1}$$

where $c(\cdot, \cdot)$ is the loss measuring the error of the prediction function f on the training set $\mathcal{D} = (X, Y)$, $\|\cdot\|_{\mathcal{H}}$ denotes the norm in $\mathcal{H}$, and $\lambda \in \mathbb{R}^+$ is a regularization parameter controlling the tradeoff between the error on the training set and the complexity of the hypothesis. We note that by specializing the loss in the above formulation we can obtain support vector machines or regularized least-squares (RLS) [13] as well as other closely related algorithms such as proximal vector machines and ridge regression.

Next, we describe a straightforward extension of the RLS algorithm to handle multiple outputs. Suppose instead of having a single column matrix for the outputs, we now have an $n \times p$-matrix, where p is the number of outputs. Slightly overloading our notation, let the output matrix be denoted as $Y \in \mathbb{R}^{n \times p}$. In the context of multi-class classification using ECOC the rows of Y would be the same as those of the coding matrix C. We use the dataset $\mathcal{D} = (X, Y)$ originating from a set $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^n$ of data points, where $X = (\mathbf{x}_1, \ldots, \mathbf{x}_n)^t \in \mathcal{X}^n$, $Y = (\mathbf{y}_1, \ldots, , \mathbf{y}_n)^t \in \mathbb{R}^{n \times p}$. We write the minimization problem as

$$\min_{\mathbf{f} \in \mathcal{H}} J(\mathbf{f}, \mathcal{D}) = \sum_{i=1}^{p} \sum_{j=1}^{n} (y_{ji} - f_i(\mathbf{x}_j))^2 + \lambda \|f_i\|_{\mathcal{H}}^2, \tag{2}$$

thus, the problem at hand boils down to solving p independent regression tasks. We note that using a square loss function leads to an efficient multi-output regression solution, namely we obtain predictions for each output by inverting the kernel matrix only once, therefore, complexity of the algorithm is hardly increased compared to a standard single output problem. On the other hand when using methods similar to SVMs for prediction of multiple outputs, the complexity of solving a single task is multiplied by the number of outputs.

3 Co-regularized Multi-class Classification

Co-regularization (e.g. [10, 14]) is naturally applicable in situations where more than one feature representation of the same object exists. Formally, consider M RKHSs $\mathcal{H}_1, \ldots, \mathcal{H}_M$ along with their corresponding kernel functions $k^{(v)} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}, 1 \leq v \leq M$. In the classification task, we search for a vector $\mathbf{f} = (\mathbf{f}_1, \ldots, \mathbf{f}_M) \in \mathcal{H}_1 \times \ldots \times \mathcal{H}_M$ of prediction functions which minimizes

$$J(\mathbf{f}, \mathcal{D}) = \sum_{v=1}^{M} c(\mathbf{f}^{(v)}, \mathcal{D}) + \lambda \sum_{v=1}^{M} \|\mathbf{f}^{(v)}\|^2_{\mathcal{H}_v} + \nu \sum_{v,u=1}^{M} \tilde{c}(\mathbf{f}^{(v)}, \mathbf{f}^{(u)}, \mathcal{D}), \qquad (3)$$

where $\lambda, \nu \in \mathbb{R}^+$ are regularization parameters and $\tilde{c}$ is a loss function measuring the disagreement between the prediction functions of the views. A classical example is a web-document classification task where the document can be represented by the word features or link features it contains, thus, creating two distinct views of the same data point [15]. The setting described above is suitable for multi-view approaches, however, in many cases it is not trivial to decide which set of features is most appropriate for co-regularization (e.g. when the views are too similar the approach will not work well in practice, as the co-regularization term would not contribute to the learning).

In case of multi-class classification, the encoding represents a natural choice for constructing a different view of the data. In some sense, the encoding can be considered as a structured output that uniquely describes the label. We suggest that by taking into account correlations among such structured outputs in addition to the inputs can be beneficial for the performance of the learning algorithm. We will construct the views on input-output representations and use them for training of our algorithm. We will show that when dealing with multi-class classification problems co-regularization among the views constructed from the input-output representations leads to improved classification performance. Our key contribution is an efficient algorithm for multi-class classification that retains benefits of an error-correcting approach and can learn from multiple views based on expressive input-output feature representations.

One problem associated with using multi-view representation of the inputs and outputs for multi-class classification arises when estimating the class label of the new data point. Due to the kernel function used to construct the feature space, the encoding of the label of the new data point has to be provided to the algorithm. We solve above problem by making use of the fact that in multi-class classification tasks the set of possible class labels that can be assigned to the new data point is known. The strategy is to compute predictions by considering all possible labels. Once predictions are available loss-based decoding [6] can be used to find the true class label.

Below we propose a simple loss-based decoding approach for estimating the class label when using multiple feature representations. More formally, applying the representer theorem [12] in this context of the co-regularization problem described in (3) shows that the minimizers $\mathbf{f}^{(v)} \in \mathcal{H}^{(v)}$ for $v = 1, \ldots, M$ have the

form $\mathbf{f}^{(v)}(\mathbf{x}', \mathbf{y}') = \sum_{i=1}^{n} \mathbf{a}_i^{(v)} k^{(v)}((\mathbf{x}', \mathbf{y}'), (\mathbf{x}_i, \mathbf{y}_i))$, where $\mathbf{x}'$ is an unseen example, $\mathbf{y}' \in \{C_{1,\cdot}, \dots C_{\kappa,\cdot}\}$ is the encoding of the label, and $\mathbf{a}_1^{(v)}, \dots, \mathbf{a}_n^{(v)} \in \mathbb{R}^p$ are the coefficients. We take the average over all views $\mathbf{f}^*(\mathbf{x}', \mathbf{y}') = \frac{\sum_{v=1}^{M} \mathbf{f}^{(v)}(\mathbf{x}', \mathbf{y}')}{M}$, and define the loss based decoding function that calculates the distance between the prediction and the rows of the coding matrix $d_L(C_{i,\cdot}, \mathbf{f}^*(\mathbf{x}', \mathbf{y}')) = \sum_{j=1}^{p} (C_{i,j} - f_j^*(\mathbf{x}', \mathbf{y}'))^2$. Finally, we select the label to be assigned to the new data point $\mathbf{x}'$ by choosing the class i^* with the smallest distance: $i^* = \operatorname{argmin}_i d_L(C_{i,\cdot}, \mathbf{f}^*(\mathbf{x}', \mathbf{y}'_i))$. In our empirical evaluation we have tried several strategies for selecting the class label for a new data point, however, assigning the label based on smallest distance usually leads to the best results. We note that constructing views on inputs-outputs has been shown to be beneficial (e.g. [8]) in structured output prediction problems. We suggest that our approach can be adapted for such tasks, however, in this work we primarily aim to address multi-class classification problem.

4 Computational Issues

We have mentioned that using square or hinge loss functions leads to two closely related algorithms and argued that our choice of the square loss leads to considerable computational benefits when addressing the problem of multiple output prediction. Thus, using square loss and matrix notations we can reformulate (3) as

$$J(\mathbf{A}) = \sum_{v=1}^{M} \operatorname{tr}\Big(Y - K^{(v)}A^{(v)}\Big)^t \Big(Y - K^{(v)}A^{(v)}\Big) + \lambda \sum_{v=1}^{M} \operatorname{tr} {A^{(v)}}^t K^{(v)} A^{(v)} + \nu \sum_{v,u=1}^{M} \operatorname{tr}\Big(K^{(v)}A^{(v)} - K^{(u)}A^{(u)}\Big)^t \Big(K^{(v)}A^{(v)} - K^{(u)}A^{(u)}\Big),$$

where $A^{(v)} = (\mathbf{a}_1^{(v)}, \dots, \mathbf{a}_n^{(v)})^t \in \mathbb{R}^{n \times p}$ and $\mathbf{A} = (A_1^t, \dots, A_M^t)^t \in \mathbb{R}^{Mn \times p}$. The matrix $K^{(v)} \in \mathbb{R}^{n \times n}$ has entries of the form $\left[K^{(v)}\right]_{i,j} = k^{(v)}((\mathbf{x}_i, \mathbf{y}_i), (\mathbf{x}_j, \mathbf{y}_j))$.

The function $k((\mathbf{x}, \mathbf{y}), (\mathbf{x}', \mathbf{y}'))$ is referred to as a joint kernel. The main idea behind the joint kernel is to describe the similarity between input-output pairs by mapping pairs into a joint space [8]. A joint kernel can encode more than just information about inputs or outputs independent of each other: it can also encode known dependencies between inputs and outputs. For example, several variations of joint kernels have been proposed in [8]. One way to define a joint kernel k is to multiply kernels constructed on input data $k_{\mathcal{X}} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ with kernel constructed on outputs $k_{\mathcal{Y}} : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$:

$$k((\mathbf{x}, \mathbf{y}), (\mathbf{x}', \mathbf{y}')) = k_{\mathcal{X}}(\mathbf{x}, \mathbf{x}') \cdot k_{\mathcal{Y}}(\mathbf{y}, \mathbf{y}'). \tag{4}$$

This feature space corresponds to the tensor product of the features space on $\mathcal{X}$ with that on $\mathcal{Y}$. The kernel we use in the experiments is the standard Gaussian

kernel:

$$k((\mathbf{x},\mathbf{y}),(\mathbf{x}',\mathbf{y}')) = \exp\left[\frac{-\left|\left|(\mathbf{x},\mathbf{y})-(\mathbf{x}',\mathbf{y}')\right|\right|^2}{2\sigma^2}\right].$$

Note that this kernel can be decomposed into a tensor product as in (4) with Gaussian kernels both on inputs and outputs.

4.1 Derivation of the Algorithm

Consider following optimization problem $\min_{\mathbf{A}\in\mathbb{R}^{Mn\times p}} J(\mathbf{A})$. Given this formulation of our optimization problem, we can follow the framework described in [16] to find a closed form for the solution by taking the partial derivative of $J(\mathbf{A})$ with respect to $A^{(v)}$

$$\begin{aligned}\frac{d}{dA^{(v)}}J(\mathbf{A}) = &-2K^{(v)t}(Y-K^{(v)}A^{(v)}) + 2\lambda K^{(v)}A^{(v)} \\ &-4\nu\sum_{u=1,u\neq v}^{M} K^{(v)t}(K^{(u)}A^{(u)}-K^{(v)}A^{(v)}).\end{aligned}$$

By defining $G_\nu^{(v)} = 2\nu(M-1)K^{(v)t}K^{(v)}$, $G_\lambda^{(v)} = \lambda K^{(v)}$ and $G^{(v)} = K^{(v)t}K^{(v)}$, we can rewrite the above term as

$$\frac{d}{dA^{(v)}}J(\mathbf{A}) = 2(G^{(v)}+G_\nu^{(v)}+G_\lambda^{(v)})A^{(v)} - 2K^{(v)t}Y - 4\nu\sum_{u=1,u\neq v}^{M} K^{(v)t}K^{(u)}A^{(u)}.$$

At the optimum we have $\frac{d}{dA^{(v)}}J(\mathbf{A}) = 0$ for all views, thus we get the exact solution by solving

$$\begin{pmatrix} \bar{G}^{(1)} & -2\nu K^{(1)t}K^{(2)} & \dots \\ -2\nu K^{(2)t}K^{(1)} & \bar{G}^{(2)} & \dots \\ \vdots & \vdots & \ddots \end{pmatrix}\begin{pmatrix} A^{(1)} \\ A^{(2)} \\ \vdots \end{pmatrix} = \begin{pmatrix} K^{(1)t}Y \\ K^{(2)t}Y \\ \vdots \end{pmatrix}$$

with respect to $A^{(1)},\dots,A^{(M)}$, where $\bar{G}^{(v)} = G^{(v)}+G_\nu^{(v)}+G_\lambda^{(v)}$. The left-hand side matrix is positive definite and therefore invertible. By defining

$$B = \begin{pmatrix} G^{(1)} & 0 & \dots \\ 0 & G^{(2)} & \dots \\ \vdots & \vdots & \ddots \end{pmatrix} \quad D = \begin{pmatrix} G_\lambda^{(1)} & 0 & \dots \\ 0 & G_\lambda^{(2)} & \dots \\ \vdots & \vdots & \ddots \end{pmatrix} E = \begin{pmatrix} K^{(1)t}Y \\ K^{(2)t}Y \\ \vdots \end{pmatrix}$$

$$C = \begin{pmatrix} G_\nu^{(1)} & -2\nu K^{(1)t}K^{(2)} & \dots \\ -2\nu K^{(2)t}K^{(1)} & G_\nu^{(2)} & \dots \\ \vdots & \vdots & \ddots \end{pmatrix}$$

Table 1. Classification errors in % on resampled datasets of the RANDOM baseline, mSVMLIN, mSVM, mRLS, and mvRLS algorithms. Bold numbers indicate the method with the lowest classification error on the particular dataset.

DATASET	RANDOM	mSVMLIN	mSVM	mRLS	mvRLS
COVERTYPE	85.7	46.2	45.6	46.1	**44.1**
LETTER	96.1	68.2	67.7	67.9	**67.1**
MNIST	90	41.6	41.1	41.0	**39.5**
NEWS20	95	82.3	81.3	81.7	**78.6**
POKER	90	55.5	55.1	55.2	**54.5**
SVMGUIDE4	83.3	56.6	56.2	56.7	**55.7**
USPS	90	32.3	31.8	32.5	**30.4**
VOWEL	90.9	60.2	59.9	59.9	**59.4**

we can formulate the solution of the system as follows:

$$\mathbf{A} = (B + C + D)^{-1} E. \tag{5}$$

Furthermore, we suggest an approach that allows searching for the optimal parameter without increasing the computational cost (see online supplementary material). The computational complexity of constructing the vector E is $\mathcal{O}(Mn^2p)$. Further, the matrices B and D can be constructed in $\mathcal{O}(Mn^3)$, and the matrix C in $\mathcal{O}(M^2n^3)$ time. The resulting matrix $(B + C + D) \in \mathbb{R}^{Mn \times Mn}$ can be inverted in $\mathcal{O}(M^3n^3)$. Thus, for fixed parameters $\lambda, \nu \in \mathbb{R}^+$ the solution of the optimization problem can be found in $\mathcal{O}(M^3n^3 + Mn^2p)$ time (we demonstrate how to further decrease computation complexity of the algorithm in the online supplementary material).

5 Empirical Evaluation

In the following section we refer to the multi-output regularized least-squares algorithm for multi-class classification as mRLS. The shorthand notation for our multi-view multi-class classification algorithm is mvRLS. Further, we denote a multi-class SVM[3] using a linear and Gaussian kernel with shorthand notation mSVMLIN and mSVM, respectively.

Various datasets are used to evaluate performance of the multi-class classifiers. Following previous works, we test the methods on the benchmark datasets from the UCI, Statlog, and other repositories that are made publicly available at the LIBSVM webpage[4] in the format directly usable by mSVM and our algorithms. We note that the attributes in the datasets are linearly scaled to [-1,1].

For the mvRLS algorithm we construct two views: one using a joint kernel over inputs and outputs, another one purely based on inputs. We use training data points and their encodings for constructing the views. For estimating

[3] `http://svmlight.joachims.org/svm_multiclass.html`

[4] `http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/multiclass.html`

the class of the unseen data point using these two views we follow the procedure described in Section 3. Similar to [16] we set the width of $k(\mathbf{x}, \mathbf{x}') = \exp\left[-(\mathbf{x}-\mathbf{x}')^2/2\sigma^2\right]$ to $\sigma^2 = 1/n\sum_{i,j=1}^{n}(\mathbf{x}_i - \mathbf{x}_j)^2$. For the joint kernel the width parameter is chosen identically. We have checked on several datasets whether the computed width is in a good agreement with the one estimated using 10-fold cross-validation. In all cases we have found the computed value to be near optimal. For selecting the regularization parameters for the MVRLS as well as for the other learning algorithms we use a 10-fold cross-validation procedure. Estimating optimal regularization parameters can be computationally prohibitive when conducting cross-validation. For the MVRLS we use the procedure described in Section 4 to efficiently search for the parameters. Namely, for each ν on a grid, we apply the fast parameter selection procedure to obtain results for all λ's with a single run. A similar approach can be applied to find the optimal regularization parameter for the MRLS algorithm.

5.1 Experimental Setup

Following [6] we construct a dense encoding for the multi-class classification problems. The encoding has $10\log_2(\kappa)$ columns where each entry is chosen to be -1 or 1 with equal probability. We generate 10000 random matrices and select the one having the largest Hamming distance among the rows. We also conduct several experiments using sparse random codes [6] (e.g. in addition to -1 and 1, entries containing 0s are allowed), as well as with one-vs-all encoding. Using dense encoding leads to slightly better performance in these trials and we choose it for comparing the algorithms in our final experiments. Somewhat similar observations were reported in [3], where it is also noted that when parameters for underlying binary classifiers are tuned correctly, the one-vs-all scheme becomes quite competitive to error-correcting approaches.

We consider an experimental setup similar to [17], where the classification performance of the binary classifiers is compared on large UCI datasets. From every dataset we randomly sample 100 examples. We ensure that examples belonging to different classes are present in the randomly selected subset. Two thirds of these examples are used for training, while one third is reserved for testing. On the training set we use 10-fold cross-validation to estimate regularization parameters. We select parameters that on average lead to the best performance over 10-folds. With these parameters fixed, we retrain the algorithm on the training set (two thirds) and test it on the test set (one third). Finally, we repeat the complete experiment 50 times and average the results. The obtained results for each dataset are reported in the Table 1.

We note that although the MSVM algorithm is efficient with a linear kernel, its runtime when using a nonlinear kernel is large[5]. Thus, one of the reasons for adapting the experimental setup proposed in [17] is the possibility to compare our algorithm to the non-linear MSVM method. It can clearly be seen that MVRLS consistently has good classification performance. We use the Friedman

[5] See `http://svmlight.joachims.org/svm_multiclass.html` for details.

test [18] with significance level ($p < 0.01$) on the results obtained from 50 independent runs of the algorithm and demonstrate that statistical significance of the obtained results. To further study the performance differences between the algorithms we use post-hoc Nemenyi test. Only in one case (`vowel` dataset) where MVRLS performs better than the other methods the obtained differences are not significant.

6 Identification of Pathogenic Bacterial Strains

We apply proposed multi-view multi-class classification algorithm to recently collected biomedical dataset described in [11]. The dataset obtained via matrix-assisted laser desorption/ionization time-of-flight mass spectrometry consists of spectra of 170 Brucella isolates. The genus Brucella contains highly infectious species that have been found to cause infections in a wide variety of mammals. Infection in humans arises from direct or indirect contact with infected animals or through consumption of contaminated meat or dairy products. Also, diagnostic laboratory workers are also at risk; 2% of all cases of brucellosis are laboratory acquired. Brucella species have a low infectious dose and are capable of transmission via aerosols, and the treatment of infections is lengthy with a risk of complications. The Brucella species primarily considered to be pathogenic for humans are B. melitensis, B. suis, B. abortus, and sporadically B. canis, B. suis biovars. The collected data requires preprocessing such as re-sampling, de-noising, baseline correction, normalization and finally peak detection and qualification before it can be used for training the classifier. We follow [19] to extract relevant features for each data point. We evaluate performance of the proposed algorithm on multi-class classification problem with 5 classes containing pathogenic Brucella species (B. melitensis, B. suis, B. abortus, B. canis, and B. suis biovars) and one additional class containing the rest of non-pathogenic microorganisms. We also test classification performance of our method with lower number of classes representing pathogenic species. A motivation for such experiment is an observation that B. canis and B. suis biovars have not been documented to be human pathogens and can be merged into separate class. In all classification experiments we follow the same parameter estimation and view construction procedure as described in Section 5. We randomly split complete dataset into training set (two thirds) and test it on the test set (one third). We ensure that examples belonging to different classes are represented in both subsets. Once the parameters are estimated via 10-fold cross-validation on the training set, we retrain the algorithm on the training set and test it on the test set. The obtained classification performance on the test set is depicted in the Figure 1. Similarly to the experiments on the benchmark datasets it can observed that MVRLS consistently outperforms other algorithms. The performance differences are statistically significant according to the Friedman test [18] ($p < 0.01$). The main result is shown in the subfigure (a). Experiment 1 corresponds to the multi-class classification problem with 5 classes of pathogenic and 1 class of non-pathogenic species. Experiment 2 corresponds to the classification problem with B. canis and B. suis biovars

species merged into a single class. The rest of the experiments correspond to different reductions of 5 pathogenic classes into lower number of classes. With the exception of experiment 1, the subfigures (a), (b), and (c) depict results on 5-class classification and the subfigures (d), (e), and (f) depict results on to 4-class classification tasks.

7 Conclusions and Discussion

This work concerns a novel multi-class classification algorithm that is particularly suitable for multi-view learning setting. We formulate the algorithm by extending co-regularization [10] framework to be applicable to multi-class problems as well as suggest new loss-based decoding approach for estimating class labels when using multiple views/feature representations. In our empirical evaluation on benchmark datasets the proposed algorithm outperforms several state-of-the-art multi-class classification methods. We apply our algorithm to the task of identification of pathogenic bacterial species of highly infectious Brucella microorganism. The results show low classification error rate of 5% on this task, which suggests that our method can be helpful for timely and effective response to an infectious disease outbreak.

Our approach can be extended in various situations. For instance, the algorithm can be adapted for the task of structured output prediction. It also allows efficient construction of different views for the co-regularization problem (see online supplementary material) and leads to notable speed up when dealing with large-scale learning tasks.

8 Acknowledgements

This work was financially supported by the Dutch Ministry of Defense, grant number V1036. We also acknowledge support from the Netherlands Organization for Scientific Research (NWO), in particular Vici grant (639.023.604).

References

1. Dietterich, T.G., Bakiri, G.: Solving multiclass learning problems via error-correcting output codes. Journal of Artificial Intelligence Res. **2** (1995) 263–286
2. Hastie, T., Tibshirani, R.: Classification by pairwise coupling. In: Proceedings of the Neural Information Processing Systems, Cambridge, MA, USA, MIT Press (1998) 507–513
3. Rifkin, R., Klautau, A.: In defense of one-vs-all classification. Journal of Machine Learning Research **5** (2004) 101–141
4. Tsochantaridis, I., Hofmann, T., Joachims, T., Altun, Y.: Support vector machine learning for interdependent and structured output spaces. In: Proceedings of the International Conference on Machine learning, ACM (2004) 104
5. Zien, A., Ong, C.S.: Multiclass multiple kernel learning. In: Proceedings of the International Conference on Machine learning, New York, NY, USA, ACM (2007) 1191–1198

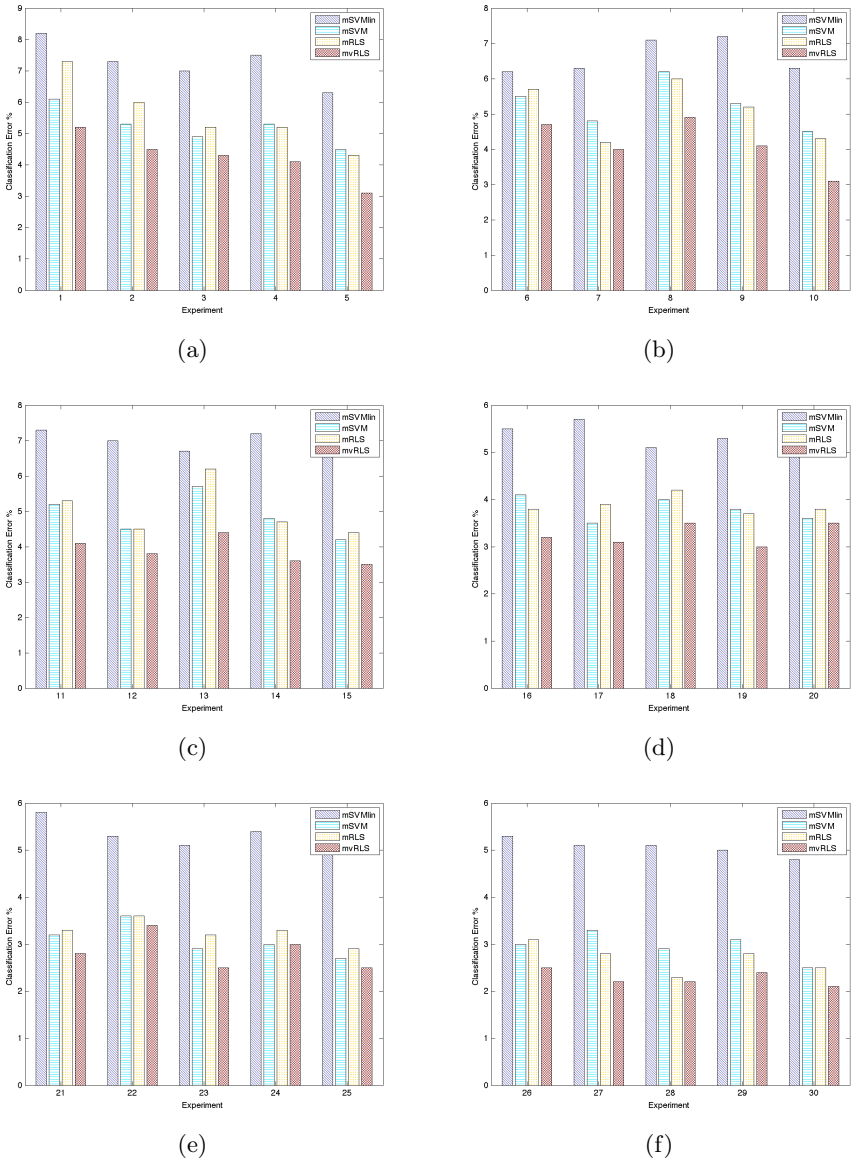

Fig. 1. Experiment 1 (see subfigure (a)) corresponds to the multi-class classification problem with 5 classes of pathogenic and 1 class of non-pathogenic species. The rest of the experiments correspond to different reductions of 5 pathogenic classes into lower number of classes. With the exception of experiment 1, the subfigures (a), (b), and (c) depict results on 5-class classification and the subfigures (d), (e), and (f) depict results on to 4-class classification tasks.

6. Allwein, E.L., Schapire, R.E., Singer, Y.: Reducing multiclass to binary: a unifying approach for margin classifiers. Journal of Machine Learning Research **1** (2001) 113–141
7. Crammer, K., Singer, Y.: On the algorithmic implementation of multiclass kernel-based vector machines. Journal of Machine Learning Research **2** (2002) 265–292
8. Weston, J., Schölkopf, B., Bousquet, O.: Joint kernel maps. In: International Work-Conference on Artificial Neural Networks,. Volume 3512 of Lecture Notes in Computer Science., Springer (2005) 176–191
9. Park, S.H., Fürnkranz, J.: Efficient decoding of ternary error-correcting output codes for multiclass classification. In: Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases, Berlin, Heidelberg, Springer-Verlag (2009) 189–204
10. Sindhwani, V., Niyogi, P., Belkin, M.: A co-regularization approach to semi-supervised learning with multiple views. In: Proceedings of ICML Workshop on Learning with Multiple Views. (2005)
11. Lista, F., Reubsaet, F., De Santis, R., Parchen, R., de Jong, A., Kieboom, J., van der Laaken, A., Voskamp-Visser, I., Fillo, S., Jansen, H.J., Van der Plas, J., Paauw, A.: Reliable identification at the species level of brucella isolates with maldi-tof-ms. BMC Microbiology **11**(1) (2011) 267
12. Schölkopf, B., Herbrich, R., Smola, A.J.: A generalized representer theorem. In: Proceedings of the Conference on Computational Learning Theory, London, Springer (2001) 416–426
13. Rifkin, R., Yeo, G., Poggio, T.: Regularized least-squares classification. In: Advances in Learning Theory: Methods, Model and Applications, Amsterdam, IOS Press (2003) 131–154
14. Tsivtsivadze, E., Pahikkala, T., Boberg, J., Salakoski, T., Heskes, T.: Co-regularized least-squares for label ranking. In Hüllermeier, E., Fürnkranz, J., eds.: Preference Learning. (2010) 107–123
15. Blum, A., Mitchell, T.: Combining labeled and unlabeled data with co-training. In: Proceedings of the eleventh annual conference on Computational learning theory, New York, NY, USA, ACM (1998) 92–100
16. Brefeld, U., Gärtner, T., Scheffer, T., Wrobel, S.: Efficient co-regularised least squares regression. In: Proceedings of the International Conference on Machine learning, New York, NY, USA, ACM (2006) 137–144
17. Brefeld, U., Scheffer, T.: Auc maximizing support vector learning. In: Proceedings of ICML workshop on ROC Analysis in Machine Learning. (2005)
18. Demšar, J.: Statistical comparisons of classifiers over multiple data sets. Journal of Machine Learning Research **7** (2006) 1–30
19. Liu, Q., Sung, A.H., Qiao, M., Chen, Z., Yang, J.Y., Yang, M.Q.Q., Huang, X., Deng, Y.: Comparison of feature selection and classification for MALDI-MS data. BMC genomics **10 Suppl 1** (2009)