Contents lists available at ScienceDirect

Analytica Chimica Acta

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

A tutorial on support vector machine-based methods for classification problems in chemometrics

Jan Luts[a,*], Fabian Ojeda[a], Raf Van de Plas[a,b], Bart De Moor[a], Sabine Van Huffel[a], Johan A.K. Suykens[a]

[a] *Department of Electrical Engineering (ESAT), Research Division SCD, Katholieke Universiteit Leuven, Kasteelpark Arenberg 10, B-3001 Leuven, Belgium*
[b] *ProMeTa, Interfaculty Centre for Proteomics and Metabolomics, Katholieke Universiteit Leuven, O & N 2, Herestraat 49, B-3000 Leuven, Belgium*

ARTICLE INFO

Article history:
Received 6 January 2010
Received in revised form 11 March 2010
Accepted 15 March 2010
Available online 24 March 2010

Keywords:
Support vector machine
Least squares support vector machine
Kernel logistic regression
Kernel-based learning
Feature selection
Multi-class probabilities

ABSTRACT

This tutorial provides a concise overview of support vector machines and different closely related techniques for pattern classification. The tutorial starts with the formulation of support vector machines for classification. The method of least squares support vector machines is explained. Approaches to retrieve a probabilistic interpretation are covered and it is explained how the binary classification techniques can be extended to multi-class methods. Kernel logistic regression, which is closely related to iteratively weighted least squares support vector machines, is discussed. Different practical aspects of these methods are addressed: the issue of feature selection, parameter tuning, unbalanced data sets, model evaluation and statistical comparison. The different concepts are illustrated on three real-life applications in the field of metabolomics, genetics and proteomics.

1. Introduction

In the machine learning and pattern recognition field, researchers develop methods and computer programs to perform a certain task. In this context, certain learning methods try to optimize a performance criterion using example data or past experience [1,2]. In case of supervised learning the learning methods are adopted based on these examples (i.e. the so-called learning or training phase), which include output values. When the output values indicate the different categories to which the examples belong, this learning task is called classification. Since the parameters of a classifier are obtained from these example data, they are usually called training data. Afterwards, the classifier or classification algorithm can be tested on new examples. If they fit the training data too much, the classification ability on unseen data may degrade. This behavior is called over-fitting. Therefore, the generalization ability of the classifier or classification algorithm is determined during this testing phase. Good generalization is an important property, and often the ultimate goal, of a classifier or classification algorithm since it provides information about their performance on unseen data.

Although linear support vector machines (SVMs) originate from almost 50 years ago, a major breakthrough was realized during the early nineties [3]. Originally, the method became popular in the neural networks and machine learning community, and afterwards it was accepted in many other fields. The machine learning technique provides a methodology for (non)linear function estimation and classification problems [4]. In contrast to the many local minima for multilayer perceptrons (MLPs), the training problem for SVMs is convex. In addition, the number of hidden units is automatically obtained for SVMs, in contrast to MLPs. The technique has already been successfully used for a wide area of applications: image classification, speech recognition, cancer diagnosis, survival analysis, forecasting, bio-informatics [5–8].

This tutorial provides a short introduction to SVMs for pattern classification, but also focuses on related classification techniques. In addition, different important practical aspects, when using these classification methods, are addressed. Multi-class classification, feature selection, determination of tuning parameters, model eval-

Abbreviations: ARD, automatic relevance determination; AUC, area under the receiver operating characteristic curve; BER, balanced error rate; KLR, kernel logistic regression; LOO, leave-one-out; LS-SVM, least squares support vector machine; MALDI-TOF, matrix-assisted laser desorption ionization time-of-flight; MLP, multilayer perceptron; MR, magnetic resonance; MRI, magnetic resonance imaging; MRS, magnetic resonance spectroscopy; MRSI, magnetic resonance spectroscopic imaging; MSI, mass spectral imaging; RBF, radial basis function; RFE, recursive feature elimination; ROC, receiver operating characteristic; SVM, support vector machine; TF, transcription factor; TOF, time of flight; VUS, volume under the surface.

* Corresponding author. Tel.: +32 16 321065; fax: +32 16 321970.
E-mail addresses: jan.luts@esat.kuleuven.be (J. Luts), fabian.ojeda@esat.kuleuven.be (F. Ojeda), raf.vandeplas@esat.kuleuven.be (R. Van de Plas), bart.demoor@esat.kuleuven.be (B. De Moor), sabine.vanhuffel@esat.kuleuven.be (S. Van Huffel), johan.suykens@esat.kuleuven.be (J.A.K. Suykens).

0003-2670/$ – see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.aca.2010.03.030

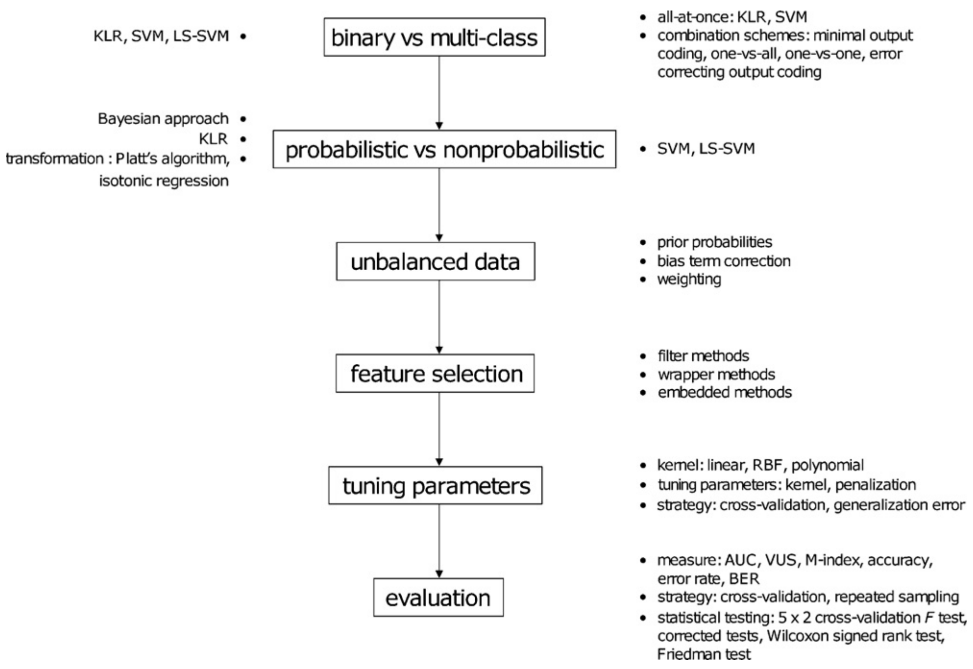
KLR, SVM, LS-SVM
binary vs multi-class
all-at-once: KLR, SVM
combination schemes: minimal output coding, one-vs-all, one-vs-one, error correcting output coding
Bayesian approach
KLR
transformation : Platt's algorithm, isotonic regression
probabilistic vs nonprobabilistic
SVM, LS-SVM
unbalanced data
prior probabilities
bias term correction
weighting
feature selection
filter methods
wrapper methods
embedded methods
tuning parameters
kernel: linear, RBF, polynomial
tuning parameters: kernel, penalization
strategy: cross-validation, generalization error
evaluation
measure: AUC, VUS, M-index, accuracy, error rate, BER
strategy: cross-validation, repeated sampling
statistical testing: 5 x 2 cross-validation F test, corrected tests, Wilcoxon signed rank test, Friedman test

Fig. 1. Flowchart of the different issues that are discussed in this paper. Practitioners should carefully address each of the steps in order to come to the final classification result.

uation, statistical comparison and dealing with unbalanced data sets are included. Fig. 1 indicates the main issues that are discussed in this paper and presents them in a flowchart. Some practical guidelines for novice users are also provided (Table 1). Toy examples are presented for illustration purposes and the ability to deal with real-life problems from chemometrics is demonstrated. These case studies include classification problems within the field of metabolomics, genetics and proteomics [9,10]. This tutorial provides the reader with the global ideas and potentials of the presented techniques. While the tutorial is rather practically oriented, a more extensive tutorial, including theoretical foundations, can be found in Ref. [11]. In addition, various books, which provide an in-depth overview of the topic, have been published [4,12–16].

2. Methods

2.1. Support vector machine classifiers

The key concept of SVMs, which were originally first developed for binary classification problems, is the use of hyperplanes to define decision boundaries separating between data points of different classes. SVMs are able to handle both simple, linear, classification tasks, as well as more complex, i.e. nonlinear, classification problems. Both separable and nonseparable problems are handled by SVMs in the linear and nonlinear case. The idea behind SVMs is to map the original data points from the input space to a high-dimensional, or even infinite-dimensional, feature space such that the classification problem becomes simpler in the feature space. The mapping is done by a suitable choice of a kernel function (Fig. 2).

Consider a training data set $\{x_i, y_i\}_{i=1}^N$, with $x_i \in \mathbb{R}^d$ being the input vectors and $y_i \in \{-1, +1\}$ the class labels. SVMs map the d-dimensional input vector x from the input space to the d_h-dimensional feature space using a (non)linear function $\varphi(\cdot) : \mathbb{R}^d \to \mathbb{R}^{d_h}$. The separating hyperplane in the feature space is then defined as $w^T\varphi(x) + b = 0$, with $b \in \mathbb{R}$ and w an unknown vector with the same dimension as $\varphi(x)$. A data point x is assigned to the first class if $f(x) = \text{sign}(w^T\varphi(x) + b)$ equals $+1$ or to the second class if $f(x)$ equals -1.

In case the data are linearly separable, the separating hyperplane can be defined in many ways. However, SVMs are based on the maximum margin principle, and aim at constructing a hyperplane with maximal distance between the two classes (Fig. 3). The SVM classifier starts from the following formulations

$$w^T\varphi(x_i) + b \geq +1 \quad \text{for } y_i = +1, \tag{1}$$

$$w^T\varphi(x_i) + b \leq -1 \quad \text{for } y_i = -1, \tag{2}$$

equivalent to

$$y_i(w^T\varphi(x_i) + b) \geq 1, \quad i = 1, \ldots, N. \tag{3}$$

The classifier is written as

$$f(x) = \text{sign}(w^T\varphi(x) + b). \tag{4}$$

However, in most real-life applications data of both classes are overlapping, which makes a perfect linear separation impossible. Therefore, a restricted number of misclassifications should be tolerated around the margin. The resulting optimization problem for SVMs, where the violation of the constraints is penalized, is written as

$$\min_{w,\xi,b} \mathcal{J}_1(w, \xi) = \frac{1}{2}w^Tw + C\sum_{i=1}^{N}\xi_i, \tag{5}$$

such that

$$y_i(w^T\varphi(x_i) + b) \geq 1 - \xi_i, \quad i = 1, \ldots, N, \tag{6}$$

$$\xi_i \geq 0, \quad i = 1, \ldots, N, \tag{7}$$

where C is a positive regularization constant. The regularization constant in the cost function defines the trade-off between a large margin and misclassification error (i.e. empirical risk minimization). SVMs are based on the principle of structural risk minimization which balances model complexity (i.e. first term in (5)) and empirical error (i.e. second term in (5)) through regularization. The first set of constraints corresponds to (3) while the second set imposes positive slack variables ξ, tolerating misclassifications. The value of ξ_i indicates the distance of x_i with respect to the decision boundary

Table 1
Guidelines for classification with support vector machine-based methods.

Data preparation	
1	Define the training, validation and test sets in a clear way.
2	Perform multiple random data splits into training, validation and test sets and repeat the analysis for each partition.
3	Consider data standardization (i.e. mean subtraction and variance scaling) or scaling the features within the same range.
4	Do not perform any type of data analysis or preprocessing such as data standardization or feature selection using the full data set (i.e. prior to data splitting).
Binary versus multi-class	
5	Consider creating binary classifiers for each pair of classes for exploratory purposes.
6	Consider multiple combination schemes for multi-class classification.
Probabilistic outputs	
7	Do not interpret uncalibrated output from (LS-)SVMs as probabilities.
8	Transform the latent values from (LS-)SVMs to probabilities using e.g. Platt's method.
Unbalanced data	
9	Choose a performance measure like BER that accounts for an unbalance in the data.
10	Use different weights for data points of different classes to include information about prior class probabilities.
11	Employ data splitting in a stratified way.
Feature selection for predictive modeling	
12	Do not use information from the test set for feature selection.
13	Consider including a simple, fast, filter method as a baseline for comparison with more sophisticated feature selection procedures.
14	Realize that wrapper and embedded methods select features that are relative to the model and for predictive purposes.
Tuning parameters	
15	Do not over-fit on training data.
16	Do not simply fix values for the regularization or kernel parameters (e.g. C, γ, σ) without any motivation (e.g. no specific default values from software packages).
17	Use a global optimization method or define a coarse grid of values for the parameters on a logarithmic scale (e.g. $10^{[-3\,-2\,-1\,0\,1\,2\,3]}$) and evaluate their performance in a cross-validation setting. Choose the parameters C^*, σ^*, γ^* with the best performance.
18	Define a finer grid of values in the neighborhood of C^*, σ^*, γ^* and evaluate the cross-validation performance for fine-tuning.
19	Take into account that σ for an RBF kernel should scale with the input dimension d.
20	Use a linear kernel with properly tuned γ as a baseline.
21	Consider the use of a linear kernel when $d \gg N$.
Evaluation	
22	Specify on which data the mentioned performance was obtained (e.g. AUC on test data).
23	Do not assume that test set performances obtained with data splitting techniques (e.g. cross-validation, repeated sampling) are independent since training and/or test sets overlap.
24	Use an appropriate statistical technique to compare the performance of different algorithms.
25	Consider collecting additional test data after the classifier has been developed.

- $\xi_i \geq 1 : y_i(w^T\varphi(x_i) + b) < 0$ implies that the decision function and the target have a different sign, indicating that x_i is misclassified,
- $0 < \xi_i < 1 : x_i$ is correctly classified, but lies inside the margin,
- $\xi_i = 0 : x_i$ is correctly classified and lies outside the margin or on the margin boundary.

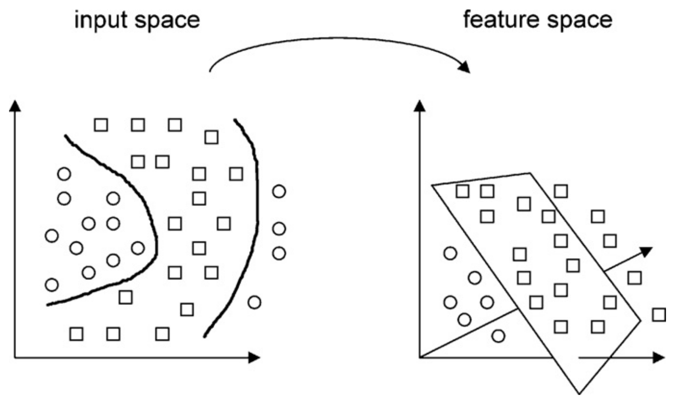

Fig. 2. SVMs allow mapping of the data from the input space to a high-dimensional feature space where a linear separation is obtained.

Typically, the constrained optimization problem in (5)–(7) is referred to as the primal optimization problem. Equivalently, the optimization problem for SVMs can be written in the dual space using the Lagrangian with Lagrange multipliers $\alpha_i \geq 0$ for the first set of constraints (6). The solution for the Lagrange multipliers is obtained by solving a quadratic programming problem. Finally, the SVM classifier takes the form

$$f(x) = \operatorname{sign}\left(\sum_{i=1}^{\#SV} \alpha_i y_i K(x, x_i) + b\right), \qquad (8)$$

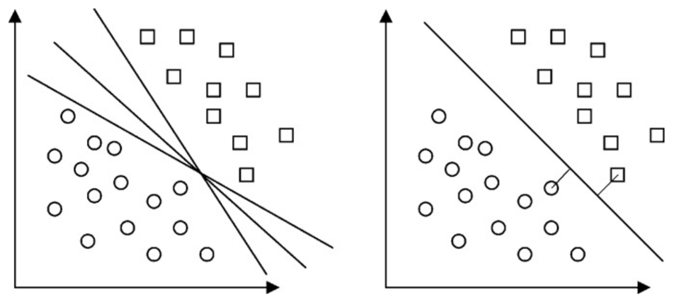
Fig. 3. Left: Various hyperplanes allow to separate the data. Right: SVMs construct a hyperplane that maximizes the margin.

where #SV represents the number of support vectors and the kernel function $K(\cdot,\cdot)$ is positive definite. It satisfies Mercer's condition then, i.e. $K(x, x_i) = \varphi(x)^T \varphi(x_i)$. This relation is also often called the kernel trick since no explicit construction of the mapping $\varphi(x)$ is needed. In the optimization problem only $K(\cdot,\cdot)$ is used which is related to $\varphi(\cdot)$. This enables SVMs to work in a high-dimensional (or infinite-dimensional) feature space, without actually performing calculations in this space. Various types of kernel functions can be chosen

- linear SVM: $K(x, z) = x^T z$,
- polynomial SVM of degree d: $K(x, z) = (\tau + x^T z)^d$, $\tau \geq 0$,
- radial basis function (RBF): $K(x, z) = \exp(-\|x - z\|_2^2/\sigma^2)$,
- MLP: $K(x, z) = \tanh(\kappa_1 x^T z + \kappa_2)$,

where $K(\cdot,\cdot)$ is positive definite for all σ values in the RBF kernel case and $\tau \geq 0$ values in the polynomial case, but not for all possible choices of κ_1, κ_2 in the MLP case.

The solution to the convex quadratic programming problem in (5)–(7) is global and unique when a positive definite kernel is used. For a positive semi-definite kernel the solution to the quadratic programming problem is global but not necessarily unique. An important property of the SVM classifier, called sparseness, is that a number of the resulting Lagrange multipliers, α_i, equal zero. As such, the sum in the resulting classifier in (8) should only be taken over all nonzero α_i values, i.e. support values, instead of all data points. The corresponding vectors x_i are referred to as the support vectors. These data points are located close to the decision boundary and contribute to the construction of the separating hyperplane (Fig. 5).

2.2. Least squares support vector machine classifiers

It has been proposed to modify the SVM methodology by introducing a least squares loss function and equality instead of inequality constraints [17]. Instead of solving a quadratic programming problem, the solution is obtained from a set of linear equations, significantly reducing the complexity and computational effort. The least squares support vector machine (LS-SVM) classifier optimizes the following problem

$$\min_{w,e,b} \mathcal{J}_2(w, e) = \frac{1}{2} w^T w + \gamma \frac{1}{2} \sum_{i=1}^{N} e_i^2, \tag{9}$$

subject to

$$y_i(w^T \varphi(x_i) + b) = 1 - e_i, \quad i = 1, \ldots, N, \tag{10}$$

with $e = [e_1 \ldots e_N]^T$ a vector of error variables to tolerate misclassifications, $\varphi(\cdot) : \mathbb{R}^d \to \mathbb{R}^{d_h}$ a mapping from the input space into a high-dimensional feature space of dimension d_h, w a vector of the same dimension as φ, γ a positive regularization constant and b a bias term. Since the value 1 in the equality constraints is a target value instead of a threshold value, the method is related to kernel Fisher discriminant analysis [15]. The primal problem is expressed in terms of the feature map, the dual problem in terms of the kernel function. The solution for the dual problem is unique because the matrix, which corresponds to a linear system, has full rank. The resulting classifier in the dual space is similar to the standard SVM classifier and is given by

$$f(x) = \text{sign}\left(\sum_{i=1}^{N} \alpha_i y_i K(x, x_i) + b \right), \tag{11}$$

with K the kernel matrix with $K(x, x_i) = \varphi(x)^T \varphi(x_i)$ and α_i the Lagrange multipliers. The support values α_i are proportional to the error of the corresponding training data points. This implies that usually every training data point is a support vector and no sparseness property remains in the LS-SVM formulation. However, high support values indicate a high contribution of the training data point on the decision boundary.

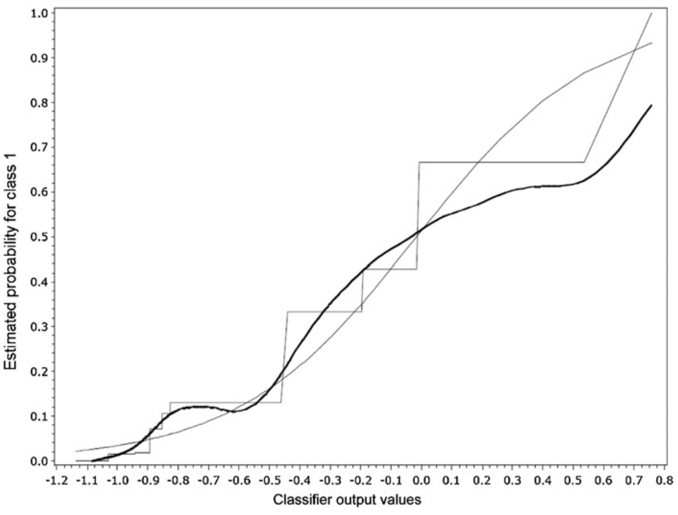

Fig. 4. Visualization of Platt's method (smooth curve) and isotonic regression (step function) for a binary classification problem on an artificial data set. The horizontal axis indicates the original output values from the LS-SVM classifier. These output values are transformed to obtain the class probabilities for class +1, which are visualized on the vertical axis. The bold line indicates the true relationship, which is obtained with nonparametric logistic regression.

2.3. Probabilistic interpretations

Eqs. (8) and (11) illustrate that SVM and LS-SVM classifiers generate 'black or white' output values. In many situations however, one prefers class probabilities which provide information about the uncertainty of belonging to one class or the other. Also, class probabilities are useful when a classifier constitutes only a small part within the overall decision process (Section 2.4). There are various strategies to obtain class probabilities from an (LS-)SVM classifier. This section focuses on the method by Platt [18], an improved version by Lin et al. [19], isotonic regression [20] and Bayesian methods [21,22].

The method by Platt maps the SVM output into probabilities using a sigmoid function. The following parametric form is used for the sigmoid

$$P(y = 1|x) = \frac{1}{1 + \exp\left(A\left(\sum_{i=1}^{N} \alpha_i y_i K(x, x_i) + b\right) + B\right)}. \tag{12}$$

The corresponding values for A and B are estimated based on maximum likelihood estimation from a separate training set, which can be obtained using a cross-validation procedure (Section 2.9), to obtain optimal probability outputs. The algorithm by Platt has been improved in terms of robustness by Lin et al. [19].

Isotonic regression has been proposed to transform the (LS-)SVM output values into class probabilities [20]. This is a nonparametric regression method that maps the values into probabilities using a monotonically increasing function. The pooled adjacent violators method is the isotonic regression method that is used. All training data are ranked with respect to the output values, while class membership is taken as the initial estimated probability of belonging to the second class. In case the initial probabilities are strictly increasing (i.e. isotonic), these values are considered to be the final estimates. In case the probabilities are not isotonic for two cases, both probabilities are averaged to serve as new estimates. This procedure is repeated until the estimated probabilities are isotonic. Training data output values for regression can be obtained based on cross-validation. Fig. 4 presents an example of isotonic regression (step function) and Platt's method (smooth line), compared to the true relationship (bold line).

It has been proposed to apply a Bayesian framework to (LS-)SVM classifiers [21,22]. This methodology is able to directly produce predictive class probabilities. An additional advantage of the Bayesian approach is that no cross-validation is required to obtain values for the regularization and kernel tuning parameters. The Bayesian approach for LS-SVMs uses MacKay's evidence procedure [23]. The model formulation for the Bayesian LS-SVM classifier is slightly modified compared to (9).

$$\min_{w,e,b} \mathcal{J}_3(w,e) = \frac{\mu}{2} w^T w + \frac{\zeta}{2} \sum_{i=1}^{N} e_i^2, \tag{13}$$

such that $\gamma = \zeta/\mu$. The Bayesian framework for LS-SVMs is a hierarchical method, consisting of three levels. At each level Bayes' rule is applied and the posterior is maximized. On the first level of inference, the weights w (or α_i in the dual representation) and the bias term b of the LS-SVM are determined. The hyper-parameters for regularization are calculated on the second level, and the third level performs model comparison to infer the kernel parameters. The Bayesian LS-SVM classifier computes final class probabilities by integrating over the posterior distribution for w and b, using the prior probabilities. Prior probabilities can be obtained by considering the proportion of cases within each class of the training data set.

2.4. Multi-class classification

Up to now, the focus was on methods for binary classification tasks. However, in practice many classification tasks require a multi-class solution. The upgrade of binary (LS-)SVMs to multi-class (LS-)SVMs is not straightforward since SVM-based methods employ direct decision functions [16]. Although there exist direct multi-class extensions of SVMs which determine all decision functions simultaneously (i.e. all-at-once approach) [12,24], the typical procedure is to split up the multi-class problem into a number of binary classification tasks. The combination of binary classifier results into a multi-class decision can be performed in many ways and the optimal choice may depend on the particular classification problem under study. In the next paragraphs we summarize some of the typical methods to tackle multi-class classification using SVM-based methods.

2.4.1. Combination schemes

Minimal output coding uses a unique binary codeword based on k bits or k classifiers for each class to encode $n_C = 2^k$ classes [25]. Error correcting output codes use more than the minimal number of bits for encoding to enhance the generalization of the multi-class classifier system [26]. The one-versus-all procedure constructs n_C binary classifiers for the n_C class problem by separating each class from the combination of all others [27]. A disadvantage of the one-versus-all scheme is that the data set is often heavily unbalanced after combining $n_C - 1$ classes. In case of one-versus-one coding the unbalance in the data set is often smaller [28]. For the n_C class problem $n_C(n_C - 1)/2$ binary classifiers need to be built. If the number of classes increases, this method seems to become cumbersome. However, when the number of classes is not too abundant, each binary classifier needs to be trained on a smaller number of data such that the training and tuning of the classifier can actually become faster [16]. A simple voting scheme or max-wins criterion can be used to decide the final class for the one-versus-one approach.

In practice, the choice between the various combination schemes should also depend on the particular classification task under study. For instance, assume a medical doctor wants to distinguish between different types of brain tumors using magnetic resonance spectroscopic imaging (MRSI) signals. In many practical situations, the clinicians already have an idea about the potential tumor type for a specific patient and they only doubt between a few, e.g. two, types of tissue, such that a simple binary classification approach is sufficient for diagnosing the patient. In this case a one-versus-one scheme might be appropriate since one can interpret the binary classifiers as powerful stand-alone entities that can also be combined when multi-class classification is needed for other cases.

2.4.2. Pairwise combination of probabilities

An interesting research topic is the development of methods to generate multi-class probabilities based on one-versus-one classifiers. These methods are able to provide multi-class probabilities based on pairwise coupling of binary class probabilities [29]. The pairwise probabilities r_{kj}, which are the probabilities to predict class k using a binary (i.e. pairwise) classifier that is only trained on data coming from group k and group j, are denoted as estimates of $\mu_{kj} = P(y_i = k | y_i = k \text{ or } j, x_i)$. The goal of coupling probabilities is to obtain the probability $p_k = P(y_i = k|x_i)$ based on the r_{kj} values. Refregier and Vallet [30], Price et al. [31], Friedman [32], Hastie and Tibshirani [29] and Wu et al. [33] proposed various strategies for retrieving a multi-class classification. Wu et al. argue that in the algorithm of Refregier and Vallet some arbitrary choices about the selection of pairwise probabilities have to be made. It has been pointed out by Price et al. and Wu et al. that the results are very sensitive to this choice and that finding the optimal selection is often very expensive. Wu et al. also found that voting [32] produced higher errors compared to the other methods. In summary, one can apply these pairwise combination methods using binary class probabilities, which can be obtained based on the techniques in Section 2.3, in order to retrieve multi-class probabilities for SVM-based methods.

2.5. Kernel logistic regression

Logistic regression and kernel logistic regression (KLR) have already proven their value in statistics and the machine learning community. In contrast to the risk minimization approach as employed by SVMs, logistic regression and KLR directly yield probabilistic outcomes based on a maximum likelihood argument and the method can be extended to multi-class classification problems in a straightforward way.

In classical binary logistic regression the probability $P(y = 1|x)$ is based on a linear combination of input variables. The output of the model can be interpreted as a probability by using the logit transformation

$$\log\left(\frac{P(y=1|x)}{1-P(y=1|x)}\right) = \beta_0 + \beta^T x, \tag{14}$$

where β_0 denotes the intercept and β represents a vector of weights, where each weight is associated to an input variable. The intercept and weights are often obtained by maximizing the likelihood of the data based on iteratively re-weighted least squares. Finally, the predicted probability for a case x is obtained as

$$P(y=1|x) = \frac{\exp(\beta_0 + \beta^T x)}{1 + \exp(\beta_0 + \beta^T x)}. \tag{15}$$

The binary model can be extended to multi-class logistic regression (multinomial logit) by using baseline category logit equations using a reference class, e.g. class n_C,

$$\begin{cases} P(y=1|x) = \dfrac{\exp(\beta_{01}+\beta_1^T x)}{1+\sum_{i=1}^{n_C-1}\exp(\beta_{0i}+\beta_i^T x)}, \\ P(y=2|x) = \dfrac{\exp(\beta_{02}+\beta_2^T x)}{1+\sum_{i=1}^{n_C-1}\exp(\beta_{0i}+\beta_i^T x)}, \\ \ldots \\ P(y=n_C|x) = \dfrac{1}{1+\sum_{i=1}^{n_C-1}\exp(\beta_{0i}+\beta_i^T x)}, \end{cases} \tag{16}$$

where the intercepts and weights are obtained by maximizing the multinomial log-likelihood. The logistic regression model can be extended to KLR [34] by introducing a feature map [35]. In the kernel version of logistic regression the different models are defined by [35]

$$\begin{cases} P(y=1|x) = \dfrac{\exp(\beta_{01}+\beta_1^T \varphi(x))}{1+\sum_{i=1}^{n_C-1}\exp(\beta_{0i}+\beta_i^T \varphi(x))}, \\ P(y=2|x) = \dfrac{\exp(\beta_{02}+\beta_2^T \varphi(x))}{1+\sum_{i=1}^{n_C-1}\exp(\beta_{0i}+\beta_i^T \varphi(x))}, \\ \ldots \\ P(y=n_C|x) = \dfrac{1}{1+\sum_{i=1}^{n_C-1}\exp(\beta_{0i}+\beta_i^T \varphi(x))}. \end{cases} \tag{17}$$

The resulting model is closely related to SVMs, but they differ with respect to the loss function [34]. While in SVMs a Hinge loss function is used (i.e. the loss is zero for cases on the correct side of the margin and increases linearly for cases on the wrong side of the margin), KLR uses a negative log-likelihood type of loss function. However, unlike SVMs, KLR by its nature is not sparse and needs all training data points in its final model. Different modifications to the original algorithm were proposed to obtain sparseness [34,36]. In Ref. [35], it was proposed to solve the KLR problem by the LS-SVM framework using a iteratively weighted version of LS-SVMs with primal and dual model representations in terms of the feature map and kernel function, respectively.

2.6. Feature selection

Although SVMs directly determine the decision boundaries in the training step and the method can also provide good generalization in high-dimensional input spaces, various researchers reported that feature selection is important for SVMs [16]. In general, the major aims of feature selection for classification are finding a subset of variables that result in more accurate classifiers and constructing more compact models. An amount of variables can be redundant to the classification problem and feature selection can filter out the variables that are irrelevant for the model. Three major types of feature selection methods are distinguished, namely filter, wrapper and embedded methods [37–40].

2.6.1. Filter methods

Filter methods employ an initial analysis as a preprocessing step, where a set of features is selected as input for the classifier using a training data set [39]. The initial analysis constructs subsets of features by ranking the features independently of the classification algorithm. The different features can be ranked according to a specific measure: chi-square test, information gain criterion, mutual information, cross-entropy measure, Fisher discriminant criterion, Pearson correlation coefficient or the Kruskal–Wallis test. The filter model has low computational cost since the learning algorithm is not used. On the other hand, by completely ignoring the learning algorithm, the impact of the feature (sub)set on the classifier is not taken into account. In Ref. [41] it is proposed that the selection procedure should also take the learning algorithm into account.

2.6.2. Wrapper methods

In wrapper models [39], a subset of variables is selected based on a search by actually using the classifier. The best subset of features is obtained by estimating and comparing the performances of the various subsets of features, which are fed to the classifier. An exhaustive search through the input space is often not feasible, therefore heuristic search methods using backward, forward or stepwise variable selection are employed [42]. Backward variable selection starts with all features and consecutively removes the feature that is least significant, until all variables in the model satisfy a level of significance. Forward selection enters variables until a level of significance is reached, while stepwise selection sequentially adds or removes variables from the model. In addition, more sophisticated methods like best-first search are also able to traverse the space of subsets [43,39]. For the evaluation of each subset, n-fold cross-validation or leave-one-out (LOO) cross-validation can be used (Section 2.9). In general the wrapper models result in an increased accuracy because potential effects of the classification algorithm are taken into account [39]. On the other hand, in principle, wrapper methods involve the higher computational cost of a search. A popular algorithm that combines SVMs with a backward search is that of recursive feature elimination (RFE) [44]. In this approach the contribution of a feature is estimated by the change in the cost function when such a feature is removed. For this purpose, the weights in the SVM model are used while support vectors are fixed for fast feature elimination. An efficient algorithm for variable selection in high-dimensional scenarios was proposed in Ref. [45]. Based on an LS-SVM model with linear kernel, rank-one updates allow the LS-SVM model parameters α_i and bias term b to be updated when features are either added or removed. Due to the closed-form solution of the LS-SVM, the LOO evaluation is obtained as a by-product.

2.6.3. Embedded methods

In contrast to filter and wrapper methods, embedded methods aim to immediately integrate the variable selection or weighting procedure in the learning algorithm. Gradient-based algorithms are used to perform feature selection by minimization of generalization error bounds. In Ref. [46] the radius-margin bound for SVM is optimized with respect to feature scaling factors, which reflect the contribution of features to the prediction rule. The work by Chapelle et al. provides derivations for several SVM generalization bounds in terms of the feature scaling factors [47]. Such formulations can serve as a methodology for multiple hyper-parameter tuning. Within this category automatic relevance determination (ARD), aims to identify informative input features as a natural result of optimizing the model selection criterion [47,48]. This task can be most easily achieved using an elliptical RBF kernel. Assuming that the kernel $K(\cdot,\cdot)$ depends on d tuning parameters, which are encoded as $\theta = [\theta_1 \ldots \theta_d]^T$, the following class of decision functions parameterized by α_i, b and θ is considered

$$f(x) = \text{sign}\left(\sum_{i=1}^{N}\alpha_i y_i K_\theta(x, x_i) + b\right), \tag{18}$$

where the jth component of x is denoted by $x_{(j)} \in \mathbb{R}$ with

$$K_\theta(x, z) = \exp\left(-\sum_{j=1}^{d}\theta_j(x_{(j)} - z_{(j)})^2\right). \tag{19}$$

In this way, individual scaling factors can be incorporated for each input dimension. Partial derivatives with respect to the kernel parameters provide the necessary information to perform gradient-based optimization. In this setting upper bounds on the LOO error (e.g. radius-margin bound) can be minimized to optimize the kernel parameter assigned to each feature. Larger θ_j's indicate more important features.

Another set of approaches penalizes the weight norm $||w||_q$, where typically $q = 0$ or $q = 1$. In contrast to the L_2-norm used in standard SVMs (i.e. $w^T w$), both the L_0-norm and L_1-norm force components in w to be exactly zero, thus automatically performing variable selection. However, the use of such norms requires specialized optimization techniques and in some cases the optimization problem is no longer convex and usually more difficult to solve than L_2-norm problems. Weston et al. solve the L_0-norm optimization problem by considering a modified standard SVM via iterative multiplicative rescaling of the training data [49]. Other approaches within this setting include algorithms based on the least absolute shrinkage and selection operator (LASSO) [50], which employ L_1-norm minimization. This method is however intended for parametric models. The LASSO algorithm computes weights for each variable according to their importance using a regularization procedure.

2.7. Tuning parameters

SVMs and other kernel-based models usually depend on several parameters controlling the model complexity. In SVM-based algorithms, model parameters, i.e. α_i and the bias term b, are determined by solving the convex optimization problem, while the positive penalization constant (e.g. C) and the kernel parameters remain user defined. In the simplest case of an SVM with a linear kernel, one needs to define a procedure to properly set the penalization parameter C. In the limit of $C \to 0$, the SVM solution converges to the case where the margin maximization term dominates. Therefore, less emphasis is put on minimizing the misclassification error, which results in a margin with large width. Consequently, as C decreases, the width of the margin increases. In the setting of SVMs with specialized kernel functions an extra parameter usually enters in the formulation. The kernel parameters (e.g. σ in the RBF kernel or τ (≥ 0) in polynomial kernel) determine the shape of the decision boundary as well as the dimension and complexity of the corresponding induced feature space. In practice, defining the best kernel for a particular problem often reduces to comparing several kernels within the same framework and experimental settings.

Existing approaches for hyper-parameter tuning include heuristics-based approaches, bounds on generalization error and cross-validation techniques. A typical a priori setting for the RBF kernel parameter σ, is that of using the distance between the closest points from different classes. However, such type of heuristic is often unreliable and noise sensitive. In the absence of reliable criteria, applications rely on the use of a validation set or cross-validation to choose hyper-parameter values. In practice, cross-validation-based techniques are preferred over generalization error bounds. Even though some of these bounds hold strong theoretical properties closely linked to the SVM foundations, experimental results on a large number of problems still favor cross-validation strategies. Criticism for cross-validation approaches is related to the high computational load involved. An interesting benchmark study developed by Cawley [51] presents an efficient methodology for hyper-parameter tuning and model building using LS-SVMs. The approach is based on the closed form leave-one-out (LOO) computation for LS-SVMs, only requiring the same computational cost of one single LS-SVM training. Additionally in Ref. [52], a smoothed LOO error estimate is optimized with respect to scaling factors $\theta \in \mathbb{R}^d$ as explained in the previous subsection. Performance compares favorably to bound-based techniques with the advantage of exact and efficient LOO computation.

2.8. Unbalanced data sets

In many binary classification problems, data sets are often skewed in favor of one class such that the contributions of false negatives and false positives to the performance assessment criterion are not balanced. In such cases the proportions of positive and negative points in the training data are not truly representative for the operational class frequencies. As mentioned in Section 2.3, appropriate prior class probabilities need to be specified for Bayesian methods and can therefore take an unbalance in the data into account. In case of standard SVMs or LS-SVMs a constant bias term correction can be performed [15]. Furthermore, different weights can be introduced for positive and negative data points, thereby balancing their contributions to the regularized loss function. Taking into account the unbalance in the data set, the objective functions for SVM and LS-SVM become

$$\min_{w,\xi,b}\mathcal{J}_4(w, \xi) = \frac{1}{2}w^T w + C\sum_{i=1}^{N}\nu_i\xi_i, \quad \text{(SVM)} \tag{20}$$

$$\min_{w,e,b}\mathcal{J}_5(w, e) = \frac{1}{2}w^T w + \frac{\gamma}{2}\sum_{i=1}^{N}\nu_i e_i^2, \quad \text{(LS-SVM)} \tag{21}$$

with

$$\nu_i = \begin{cases} \dfrac{N}{2N_P}, & \text{if } y_i = +1, \\ \dfrac{N}{2N_N}, & \text{if } y_i = -1, \end{cases} \tag{22}$$

where N_P and N_N, represent the number of positive and negative data points, respectively. Asymptotically this weighting is equivalent to resampling of the data such that there is an equal number of positive and negative data points.

2.9. Evaluation: experimental setup

2.9.1. Performance measures

An important aspect of building classifiers or the development of classification algorithms is the evaluation of their performance. The accuracy (misclassification error) represents the proportion of cases that is (not) correctly classified. These measures are not optimal since they can be misleading when data are unbalanced (Section 2.8). In case one of the two classes accounts for 90% of the data, a method that always predicts this class (i.e. the majority rule) will already obtain an accuracy of 90%. Based on the balanced error rate (BER), which is the average of the error rate for each separate class, an unbalance in the data can be taken into account. In addition, in some applications misclassification costs should be incorporated by a weighting scheme. Predicting class 1 for a class 2 case can be worse than classifying a class 1 case as class 2. Other well-known evaluation measures are the sensitivity and specificity. The sensitivity is the proportion of cases of the class of interest that are correctly predicted as members of this class. It is the number of true positives divided by the sum of the amount of true positives and false negatives. The specificity is the proportion of cases from the other class that are correctly predicted. It is the number of true negatives divided by the sum of the number of true negatives and false positives. Another interesting measure is the positive predictive value, being the probability of belonging to the first class when this class is predicted. The negative predictive value is the analogue for the other class. Finally, some measures try to quantify the level of discrimination between two classes. A very popular measure for binary problems is the area under the receiver operating characteristic (ROC) curve (AUC) [53]. An ROC curve can be constructed by plotting 1 – specificity against the sensitivity by varying the decision threshold over its entire range. A method that gives random predictions results in an AUC of 0.5, while a perfect method attains an AUC of 1 on the test data. The AUC value is often interpreted as the probability that a random pair (one from each class) of cases will be correctly discriminated. Various measures have been proposed to extend the AUC towards multi-class problems. For three-class problems the volume under the surface (VUS) has been defined. The M-index is a measure that can be applied to any multi-class problem [54].

2.9.2. Evaluation strategies

Apart from the chosen measure, a specific strategy is required to evaluate the performance on the data. However, in Ref. [37], an important distinction is made between the actual classifier (i.e. (8) and (11)) and the classification algorithm. The classification algorithm is the technique that is used to produce the classifier. In this tutorial, all techniques are supervised methods, causing that a training data set with labeled data is needed for the construction of the model and test data are required for the evaluation of the generalization ability of the model to new data. A good classifier should generalize well to new, unseen data and should not over-fit the training data. As such, in case one is only interested in a classifier and depending on the number of points in the available data, it may be required to split the original data in a training set and a test set. The former is used to develop the classifier, while the latter is only used for evaluation purposes. SVM-based classifiers require tuning of (hyper-)parameters (i.e. C, γ and σ for the case of an RBF kernel), and therefore, an additional validation set is required. In case no separate validation set is available, cross-validation techniques can be used for (hyper-)parameter tuning. If one is not interested in a classifier, but in the classification algorithm that produces classifiers, typically, many data sets are generated such that multiple classifiers can be generated. Data splitting techniques are used to repeatedly obtain a training (validation) set and a test set. Afterwards, the results on the test set can be combined. There exist various techniques to perform data splitting.

In case of n-fold cross-validation the data are divided into n subsets of approximately equal size. For each of the subsets, the remaining $n-1$ subsets are combined to form the training data such that the performance can be measured on the remaining subset. This process can be repeated such that different subsets are generated in a random way, after which the results on the test sets can be combined. If one chooses n equal to the number of points in the total data set, N, this strategy is called LOO cross-validation. On the other hand, repeated sampling iteratively divides the data in a set for training and a set for testing in a random way. Often 2/3 of the data are used for training and the remaining data are used for testing. In addition, cross-validation and repeated sampling can be used in a stratified manner. This means that the training sets and test sets are stratified such that they approximately contain the same proportions of labels as the original data. It guarantees a prescribed number of data points from each subpopulation. Finally, a good estimator of the performance can be obtained by testing on data that are collected after the classifier or classification algorithm were developed.

Although the LOO error is an important statistical estimator of the performance of a learning algorithm, its exact computation requires N runs of the learning algorithm. Such an approach becomes infeasible for large numbers of data points. In the context of standard SVMs, many studies have focused on the derivation of approximations or upper bounds for the LOO error [55]. The main reason behind this is the high computational cost involved when solving N quadratic programming problems. On the other hand, the LOO error for LS-SVM classifiers can be obtained exactly in closed form with the computational complexity of a single LS-SVM algorithm [56,51]. In addition, there exist methods (e.g. based on the regularization path) for more efficient parameter tuning in SVMs as well [57].

2.9.3. Statistical analysis

Researchers are often interested in comparing different classification algorithms. Traditionally, these comparisons are performed based on statistical significance testing. However, when data splitting has been used as the strategy to evaluate the performance of a classification algorithm, the assumption of independence, which is usually required for statistical tests, is not valid. For cross-validation the different training sets overlap and for random sampling both training and test sets overlap such that the obtained data points are not independent from each other [37]. In this way, the computation of the variance of the average test set performance becomes more difficult. In Ref. [58], a 5×2 cross-validation F test was constructed to compare two classification algorithms. Different variance estimates were obtained in Ref. [59] and the corrected resampled t-test was established. In addition, it was reported that the effective degree of freedom is lower than the theoretically expected value when the same data are reused [60]. Therefore, it was proposed to use empirically calibrated values. A corrected repeated n-fold cross-validation test was obtained in Ref. [61], while Ref. [62] derived a t-statistic for standard n-fold cross-validation. An approach to find the best of more than two classification algorithms is explained in Ref. [63].

The issue of comparing classification algorithms on multiple data sets is addressed in Ref. [64]. The Wilcoxon signed rank test (for two classification algorithms) and the Friedman test (for more than two classification algorithms), including post-hoc tests for multiple comparison, are two nonparametric statistical tests that have been proposed. Multiple comparison procedures are covered in more detail in Ref. [65], where Shaffer's static and Bergmann–Hommel's procedures are proposed. In Ref. [66], a framework for multiple testing in the presence of dependence has been established, since ignoring the dependence among hypothesis tests can result in highly variable significance measures and bias. Statistical hypothesis testing is a difficult issue and the practice of null hypothesis significance testing has been criticized by statisticians, since blind reliance on a p-value for the interpretation of the results can be dangerous [67,68]. Alternatively, the use of confidence intervals and effect sizes (e.g. a confidence interval for the difference in AUC between two methods) has been proposed to quantify the effect under study. In general, one should verify whether a statistically significant difference is in fact a significant difference in practice.

2.10. Software packages

A large number of software packages are freely available on the web and most of them originate from technical institutions and universities. The kernel machines web portal at www.kernel-machines.org provides an extensive repository with links to manuals, software packages, as well as forum discussions with experts open to all users. Some common cited packages are the

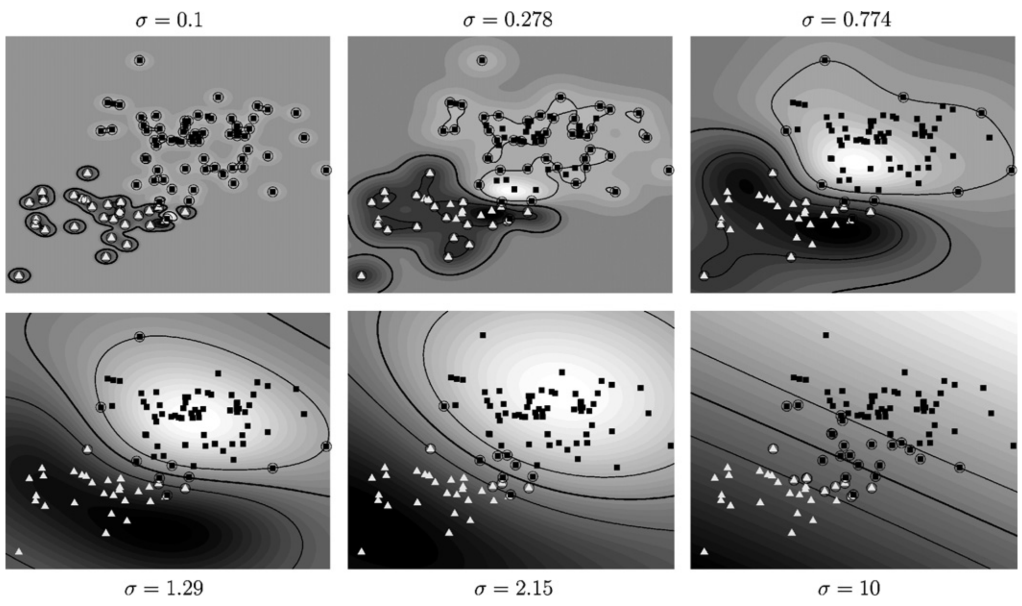

Fig. 5. The influence of varying the RBF kernel width parameter σ for a fixed value of $C = 10$. From left to right and top to bottom, the parameter σ is changed from 0.1 (top left) to 10 (bottom right). Encircled points represent the support vectors. Increasing σ results in smoother decision boundaries, changing from nonlinear to almost linear.

award winning SVM library LIBSVM [69], and kernlab [70], which is a kernel-based machine learning library for the R language. BSVMHT [71] provides probabilistic methods for SVMs. LS-SVMLab [72] is a Matlab/C toolbox for LS-SVMs, while SVM^{light}[73] is a library for large scale problems.

3. Case studies

The following subsections provide examples of the application of kernel-based classifiers to toy problems and real-life problems are provided. The use of SVMs, LS-SVMs and KLR is demonstrated to differentiate brain tumors based on MR data, predict gene regulatory networks or to detect anatomical regions using mass spectral imaging (MSI) data.

3.1. Toy example

In order to highlight the importance of tuning hyper-parameter setting in SVMs, a two-dimensional synthetic data set is randomly drawn, in an unbalanced way, from two normal distributions with different means and equal covariance matrices. The training data are shown in Fig. 5. Since the negative class (■) is more abundant than the positive class (△), the modification presented in (20)–(22) is employed to adjust class frequencies. Due to class overlap, some misclassifications are tolerated by properly setting the penalization parameter C. Starting with a linear kernel, a value of $C = 10$ is obtained via 10-fold cross-validation. From this point, with $C = 10$ fixed, the linear kernel is replaced by the RBF kernel and the σ parameter is varied in the range [0.1, 10]. SVM classifiers are trained with these values and tested on points uniformly sampled from a two-dimensional grid. The decision regions are generated using the predictions on the test data. A model, where every training point is a support vector, is displayed in the top left box. In this case a narrow RBF kernel is centered on each training data point, thus creating isolated class boundaries which over-fit the training data. Smoother surfaces are obtained by increasing σ, yielding simpler models with fewer support vectors and compact class boundaries. The under-fitting phenomenon is clearly observed for large values of σ (e.g. 10) at the bottom right box.

3.2. Brain tumor classification based on magnetic resonance data

Contrast-enhanced magnetic resonance imaging (MRI) is an important tool for the anatomical assessment of brain tumors. For example, Fig. 6 provides a series of high-resolution MR images (i.e. T1-weighted, T2-weighted, proton density-weighted and gadolinium-enhanced T1-weighted MR image) for a patient with a glioma. On the other hand, several diagnostic questions, such as the type and grade of the tumor, are difficult to address using these conventional MR images. Therefore, despite the asso-

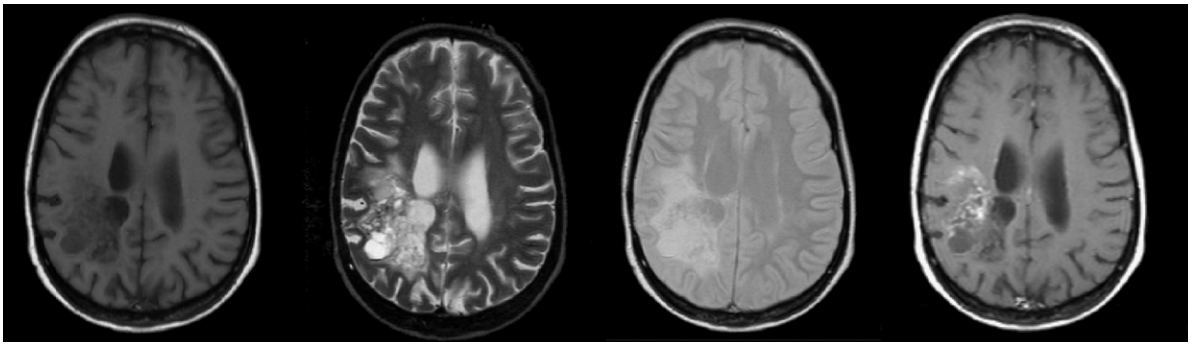

Fig. 6. From left to right: T1-weighted MR image, T2-weighted MR image, proton density-weighted MR image and gadolinium-enhanced T1-weighted MR image for a patient with a glioma.

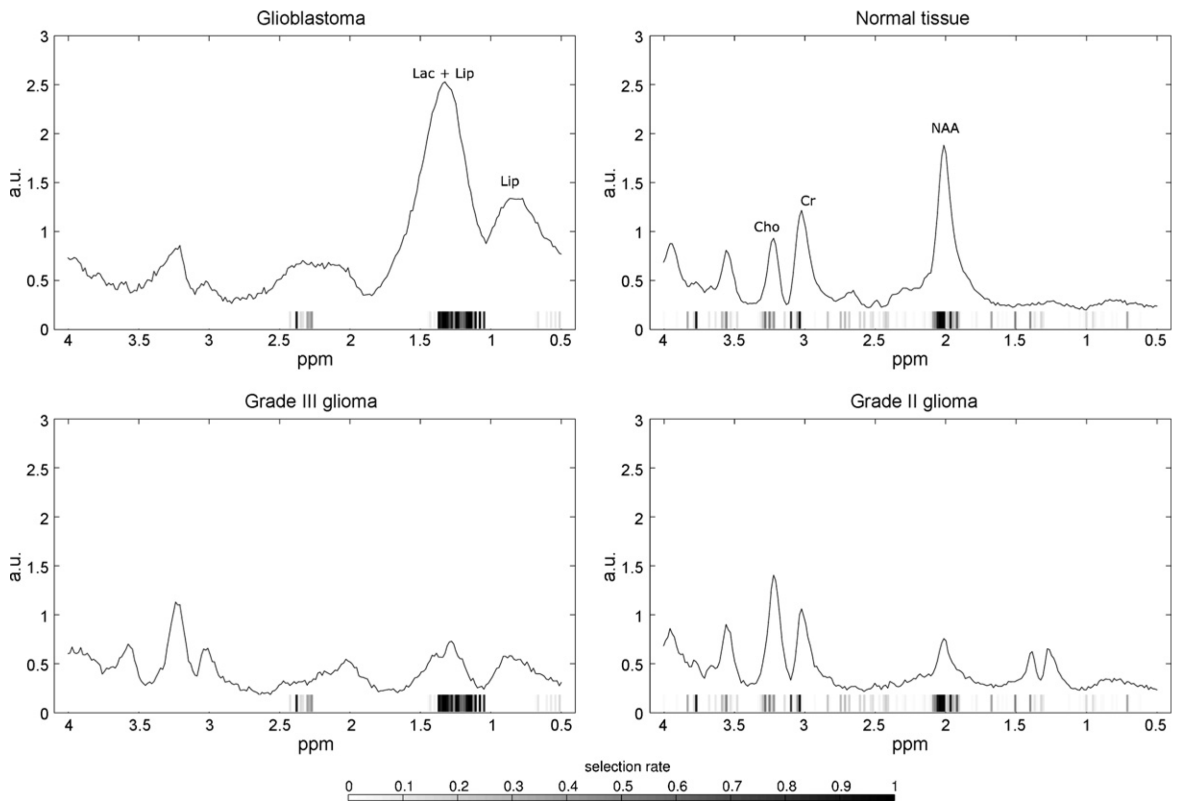

Fig. 7. Mean water-normalized magnitude MR spectra for glioblastoma, grade III glioma, normal tissue and grade II glioma. The bars indicate the selection rates of each feature for the binary classification problems over 100 times of random sampling. For classification of glioblastoma and grade III glioma the filter approach Fisher discriminant criterion mainly selected features from the region with lipids (Lip) and lactate (Lac). For differentiation between normal tissue and grade II glioma, the wrapper method based on the forward search strategy selected features belonging to the metabolites N-acetylaspartate (NAA), choline (Cho) and creatine (Cr).

ciated risks of surgery to obtain a biopsy, the histopathological characterization of a tissue specimen remains the gold standard. More recently, magnetic resonance spectroscopy (MRS) is increasingly being used for more detailed noninvasive evaluation of brain tumors. MR spectra provide metabolic information since the different peaks in the MR spectrum correspond to different chemical metabolites and the area under the peak is representative for their concentration. Fig. 7 illustrates mean MR spectra for different types of tissue: glioblastoma, grade III glioma, normal tissue and grade II glioma. In addition, MRSI measures a whole grid of MR spectra (each

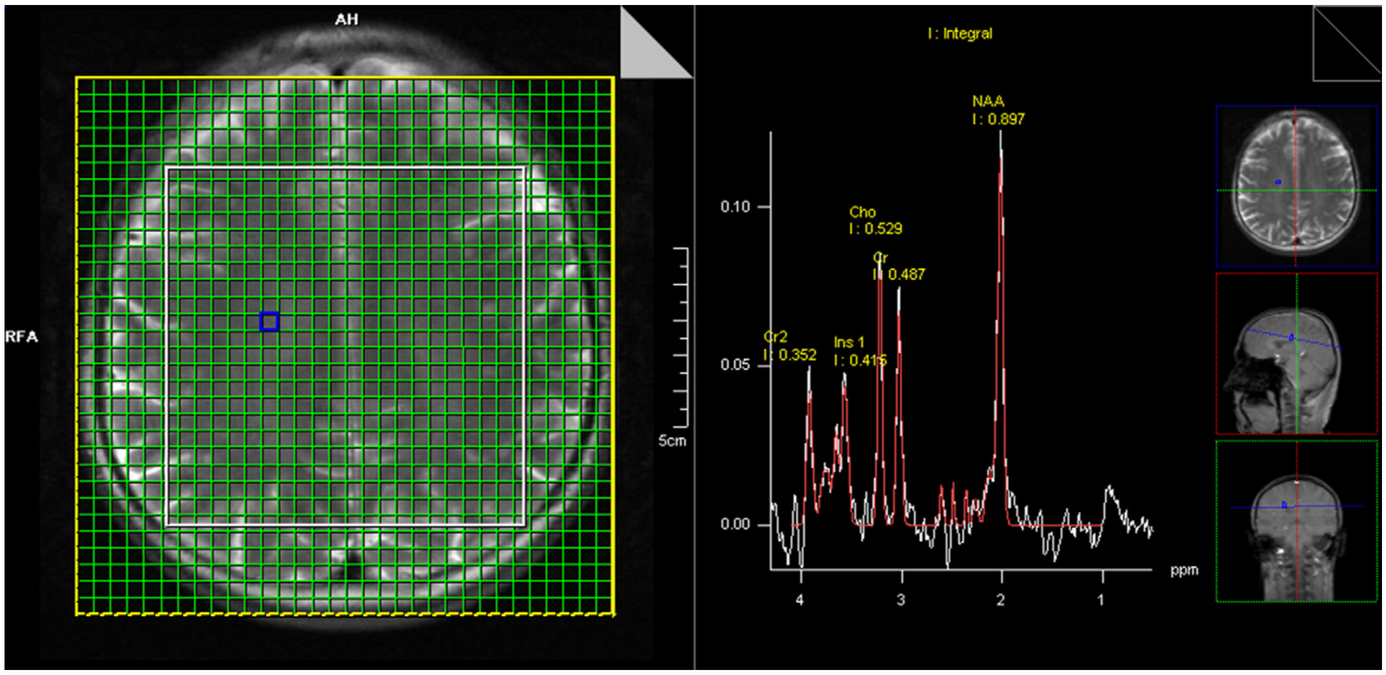

Fig. 8. MRSI grid and a corresponding spectrum for a healthy volunteer.

from a specific voxel) and provides quantitative metabolite maps of the brain. Fig. 8 provides a grid of MRSI spectra for a healthy volunteer. The individual inspection and analysis of the many spectral patterns, obtained by MRSI, remains extremely time-consuming and requires specific spectroscopic expertise. As such, the use of pattern recognition methods to automatically classify MR spectra is an important research area.

MRI and MRSI data were acquired using a 1.5 T Siemens Vision Scanner in University Medical Center Nijmegen and a set of 681 MR spectra was extracted from 24 patients with a brain tumor and 4 healthy volunteers [9]. Six different tissue types (i.e. classes) were included in the training set: normal tissue (218 spectra), cerebrospinal fluid (CSF) (100 spectra), grade II glioma (176 spectra), grade III glioma (69 spectra), glioblastoma (70 spectra) and meningioma (48 spectra). First, a feature selection analysis is demonstrated for two binary classification problems: glioblastoma versus grade III glioma and normal tissue versus grade II glioma. For each classification problem the MR spectra, of which each signal consists of 230 values, are split (in a stratified way) 100 times in a set for training (i.e. 80 %) and a set for testing (i.e. 20 %). For glioblastoma versus grade III glioma the filter method Fisher discriminant criterion is used to select the 20 features with the highest scores using the training data in each iteration. Next, the regularization parameter γ of an LS-SVMs with a linear kernel is tuned based on the 10-fold cross-validation error. Fig. 7 illustrates that the filter method mainly selects features from the region of the lipids and lactate. Lipids are known to be very important markers for high-grade glioma tumors as glioblastomas. The averaged AUCs on the test set for the full feature set and the reduced feature set are 0.9901 and 0.9819, respectively. According to the corrected resampled t-test the performances of the full and reduced feature set are not statistically significantly different (p-value = 0.35) [59]. For the separation of normal tissue from grade II glioma a wrapper feature selection approach is used. The forward search strategy from [45] is used in combination with an LS-SVM with linear kernel. The training data are first used to obtain the regularization parameter γ, thereby minimizing the 10-fold cross-validation error. In a second step, forward feature selection is performed by creating candidate feature subsets by sequentially adding each of the features not yet selected. Feature subsets are ranked according to a 10-fold cross-validation error. In each of the 100 runs the feature selection procedure is stopped once 20 features are selected. Fig. 7 shows that most selected features correspond to the metabolites N-acetylaspartate, choline and creatine. These metabolites are known to be important features to differentiate between normal tissue and grade II gliomas. For the classification of normal tissue and grade II gliomas the averaged AUCs for the full feature set and the reduced feature set are 0.9982 and 0.9989, respectively. The corrected resampled t-test indicates no statistically significant difference (p-value = 0.57) between the performances of the full feature set and the reduced feature set.

Next, a multi-class classifier system is created and for this purpose an estimate of the concentration of important metabolites is derived from the MR spectra by peak integration. Ten different peaks (e.g. corresponding to lipids, N-acetylaspartate, choline, creatine, etc.) were integrated in each MR spectrum and combined with the four types of MRI intensities that corresponded with the voxel. As such, an input pattern for the pattern recognition methods consists of 14 features. To deal with the multi-class problem it is chosen to split it up into binary problems following the one-versus-one scheme. Multi-class probabilities are obtained with the pairwise combination algorithm of Wu et al. Each binary classification problem is solved by a Bayesian LS-SVM classifier with RBF kernel. The unbalance in the data set is handled by adjusting the prior class probabilities for the pairwise Bayesian LS-SVM classifiers. In addition, a direct multi-class approach is also applied by using KLR with an RBF kernel. No additional feature selection is used for the Bayesian LS-SVM classifiers or KLR. Five-fold cross-validation is used to tune the hyper-parameters for the KLR classifier. Evaluation of the Bayesian LS-SVM classifiers and the KLR classifier is done using a leave-one-patient-out strategy. This means that all data from the patient (i.e. the patient in Fig. 6) under study were removed from the training sets.

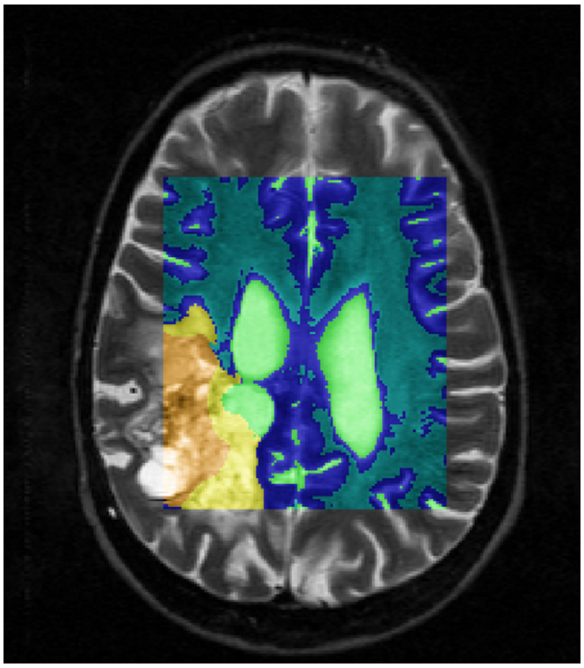
Fig. 9. Nosologic image: light blue reflects white matter, dark blue gray matter, green CSF, yellow grade II glioma, and orange grade III glioma. The multi-class Bayesian LS-SVM classifier differentiates grade II glioma from grade III glioma in the abnormal tissue region.

Fig. 9 shows the final classification result for the patient in Fig. 6. All pixels in the abnormal region are classified by the multi-class Bayesian LS-SVM classifier. Every pixel is assigned to the class, having the highest posterior probability. Each assigned color represents a specific tissue class (light blue = white matter, dark blue = gray matter, green = CSF, yellow = grade II glioma and orange = grade III), resulting in a so-called nosologic image [9]. For this patient the pathologists disagreed between a grade II and a grade III glial tumor. The result from automated pattern recognition shows a heterogeneous tumor with mainly grade III glioma, especially in the part with enhancement after gadolinium. Fig. 10 provides a probability plot of the class probabilities, obtained with the KLR classifier. The more red the probability map is, the higher the probability for grade III glioma. The grade III glioma probabilities are higher at the frontal part of the abnormal tissue region with the enhancement after gadolinium and smaller at the occipital part of the tumor.

3.3. Prediction of gene regulatory networks

Microarrays are designed to understand how the whole set of genes (genome) of a particular organism is functioning. On a single glass slide, thousands of spots, each related to a single gene, are used to simultaneously detect gene expression levels while varying several experimental conditions [8]. Genes working together in certain biological processes are assumed to have similar expression patterns. Therefore, finding similar patterns of genes, having related functions, can deliver evidence to infer the structure of gene

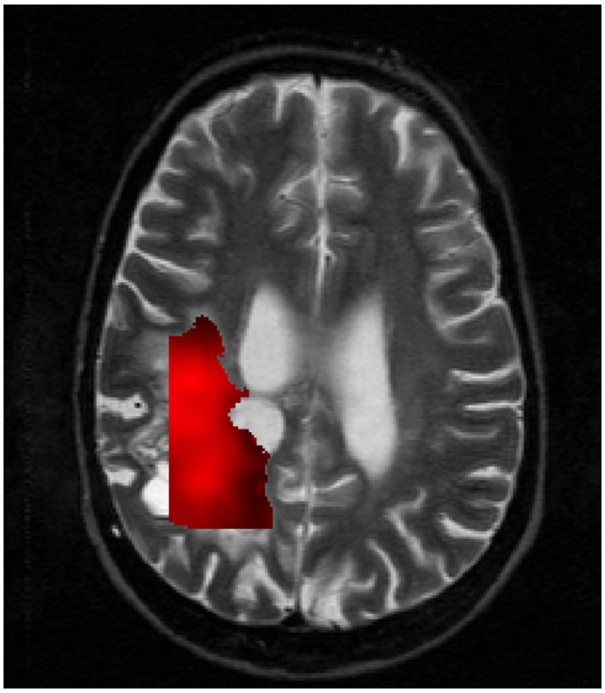

Fig. 10. Probability map for grade III glioma: a higher intensity of red indicates a higher probability for grade III glioma. Probabilities are higher at the frontal part of the abnormal tissue region with the enhancement after gadolinium and smaller at the occipital part of the tumor.

networks [74]. Identification of transcription factors (TFs) binding sites in the genome remains incomplete and often difficult solely through chemical techniques. Therefore machine learning approaches constitute a potential alternative with respect to complex experimental methods. Assuming that the full genome of a certain species is given, a particular gene g starts transcription for protein production when a TF t (a protein) chemically binds to it. The machine learning approach to this problem can be summarized as: given a gene g, determine whether TF t binds to it. SVMs can then be used to categorize new genes assuming one provides a set of genes known to be regulated by a certain TF and a set known to be not co-regulated.

In this case study microarray data sets, describing the gene expression profiles of yeast under several experimental conditions, are considered [8,74]. The Spellman data set contains 77 experiments describing the dynamic changes of yeast genes during the cell cycle, while the Gasch data set consists of 177 experiments, examining gene expression behavior during various stress conditions [8,74]. In order to construct positive and negative data points, a list of known regulation relationships between known TFs and a set of genes is required. Such regulatory information is derived from ChIP-chip interactions data collecting binding information of 204 regulators to their respective target genes [75]. This information has previously been used to correlate regulatory programs with regulators and corresponding motifs to a set of co-expressed genes [76]. Values in this regulatory information data set are given in terms of p-values, indicating whether a TF binds to the set of considered genes in yeast. Positive data points are then defined from interactions with a high confidence value, i.e. a p-value ≤ 0.001 [75,10]. However, the choice of negative data points, i.e. pairs of TF and genes without reported regulatory relationships, is not straightforward since there are few data published establishing that a given TF is found not to regulate a given target gene. Instead of taking a random sample of pairs with unknown interaction, negative data points are selected as those genes with a p-value ≥ 0.8 as indicated by the ChIP-chip interaction data. Such genes are less likely to be bound by the TF. After some preprocessing steps (finding common genes and TFs across data sets, removing rows or columns with too many missing values, considering only those TFs with at least 5 known positive interactions), the final expression data consist of a matrix with 5295 genes $[g_1 g_2 \cdots g_{5295}]$ measured over 250 experimental conditions $[e_1 e_2 \cdots e_{250}]$ and the target matrix with 118 TFs $[t_1 t_2 \cdots t_{118}]$ accounting for about 4000 positive interactions. Fig. 11 illustrates a small gene regulatory network. Known interactions (solid lines) between TFs (red circles) and target genes (green circles) are defined from ChIP-chip experiments and represent the target information in the learning task. Dashed lines represent noninteracting TF-gene pairs. Given a certain gene, the aim is to determine for instance whether TF t_6 binds to it. Learning new TF-gene interactions and building gene regulatory networks (center) is then based on input gene expression data (left) and known TF-gene interactions targets (right).

Similar to the applications reported in Refs. [77,10,78], this case study designs SVM-based classifiers (i.e. LS-SVM with RBF kernel) for each TF. The learning problem is highly unbalanced in all cases with the number of positive data points being outnumbered. Therefore, the weighted cost function in (21) and (22) is used for the LS-SVM classifier. Training multiple classifiers and tuning their parameters (i.e. γ, σ) pose a computational burden, hence, the fast cross-validation implementation for LS-SVM is considered [51,79]. Fig. 12 shows the gene regulatory network inferred by the LS-SVM algorithm. Visualization is done using VisAnt software [80]. TF-gene interactions are obtained based on the output of the method by Platt, which is used to assign a probability value based on the LS-SVM output [18]. TFs with at least 5 new interactions with cut-off probability value greater than 0.95 are shown in the regulatory network.

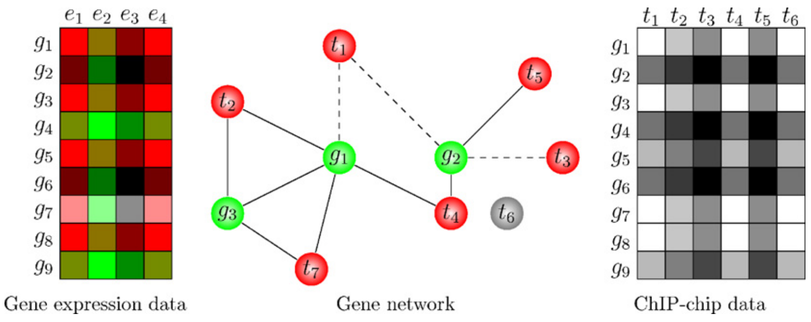
e_1 e_2 e_3 e_4
g_1 g_2 g_3 g_4 g_5 g_6 g_7 g_8 g_9
Gene expression data
t_1 t_2 t_3 t_4 t_5 t_6 t_7
g_1 g_2 g_3
Gene network
t_1 t_2 t_3 t_4 t_5 t_6
g_1 g_2 g_3 g_4 g_5 g_6 g_7 g_8 g_9
ChIP-chip data

Fig. 11. Gene regulatory network scheme representing known interactions (solid lines) between TF (red circles) and target genes (green circles). Dashed lines represent noninteracting TF-gene pairs. For instance, given the expression levels $[e_1 e_2 \cdots e_4]$ of a particular gene g, the learning task aims to determine whether the TF t_6 binds to it. Learning new TF-gene interactions and building gene regulatory networks (center) is based on gene expression data (left) and known TF-gene interactions (right).

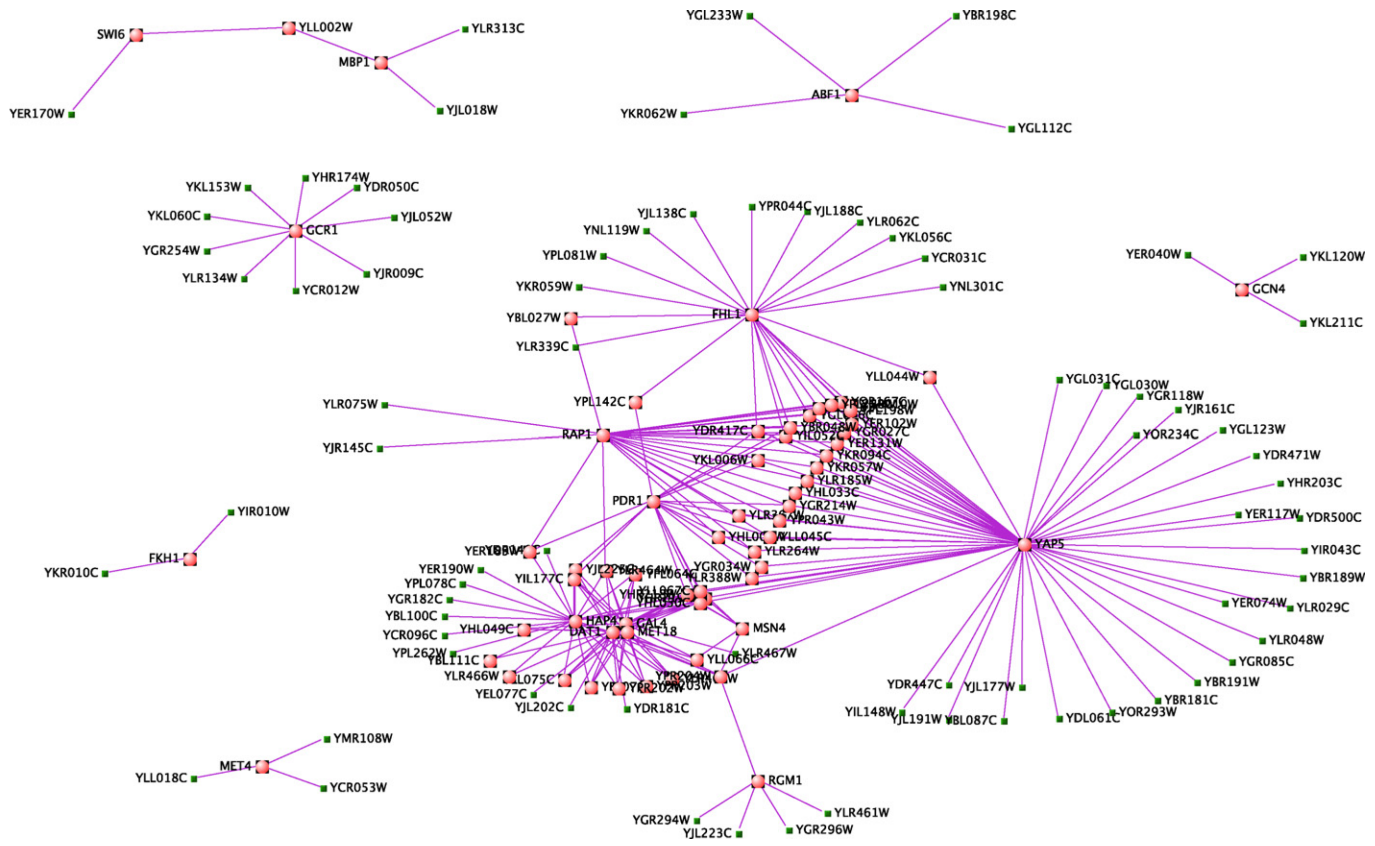

Fig. 12. Gene network constructed using TFs with at least five new predicted interactions by the LS-SVM classifier. The cutoff probability threshold is set to 0.95, such that false positives are less likely to be ranked as new bindings.

3.4. *Detection of anatomical regions in tissue using mass spectral imaging*

Mass spectral imaging (MSI) is a developing technology that facilitates the detection of biomolecules such as proteins, peptides, and metabolites directly from organic tissue while keeping the spatial information intact [81]. MSI enables to study the spatial tissue distribution of any detectable molecule that falls within a user-specified molecular mass range [82]. The technology differentiates itself from traditional molecular imaging modalities such as fluorescence microscopy and immunostaining in that it does not require chemical labeling of the molecules of interest. As a result MSI does not need a prior hypothesis, making the technique particularly suited for exploratory roles in addition to confirmatory imaging. Additionally, MSI exhibits a high molecular specificity and can potentially track hundreds of molecules in a single experiment. These characteristics make that MSI is gaining considerable momentum in the tissue biomarker community. In this case study MSI is used to elucidate the spatial biochemical topology of a sagittal section of mouse brain as shown in Fig. 13.

A typical MSI experiment consists of a grid of measurement locations or pixels covering the tissue section, with an individual mass spectrum attached to each pixel. The resulting data structure can be considered as a three-dimensional array or tensor with two spatial dimensions (x and y) and one mass-over-charge (m/z) dimension. The MSI data tensor can be reshaped into a matrix format by reducing the two spatial dimensions to a single pixel dimension, in order to accommodate matrix decomposition methods such as principal component analysis [83,84]. Other approaches to mine the MSI data tensor include spatial querying [85], and supervised classification of the spectra also known as digital staining [86]. Along the lines of [86], this case study demonstrates the use of SVM as a means of digital tissue staining.

The data set was acquired at University Hospitals Leuven and consists of a MSI data tensor acquired from a sagittal section of mouse brain [85]. The spatial grid covering the tissue has a size of 51×34 measurement locations (i.e. 1734 pixels). At each measurement location a mass spectrum is acquired using a matrix-assisted laser desorption ionization time-of-flight (MALDI-TOF) mass spectrometer (ABI 4800 MALDI-TOF/TOF Analyzer) spanning a mass range from 2800 to 25000 Dalton in 6490 m/z bins. Thus, the data structure consists of 1734 mass spectra measuring 6490 m/z variables per spectrum. Partial labeling information is provided by a pathologist for four distinct anatomical regions within the tissue. The labeled pixels represent high-confidence membership to a particular anatomical structure in the brain. The four provided labels are the cerebellar cortex (cc), Ammon's horn in the hippocampus (ca), the cauda-putamen (cp), and the lateral ventricle (vl) area. Fig. 13 depicts the four partially labeled regions overlayed on a gray level microscopic image of the mouse brain section and the corresponding averaged spectra. A total of 279 pixels (and corresponding spectra) comprise the training set for the 4 classes. Spectra are normalized with respect to the total ion current and are baseline corrected. The standard SVM with linear kernel is used for this high-dimensional classification problem. The regularization constant C is tuned using 3-fold cross-validation, with the misclassification error as the performance measure, together with a one-against-all multi-class scheme. Predictions for the whole set of spectra (i.e. 1734) are reconstructed into the image in Fig. 14.

By translating the predicted labels back to their position in the spatial domain, one can directly assess the performance of the classification algorithm via visual inspection. The classification model

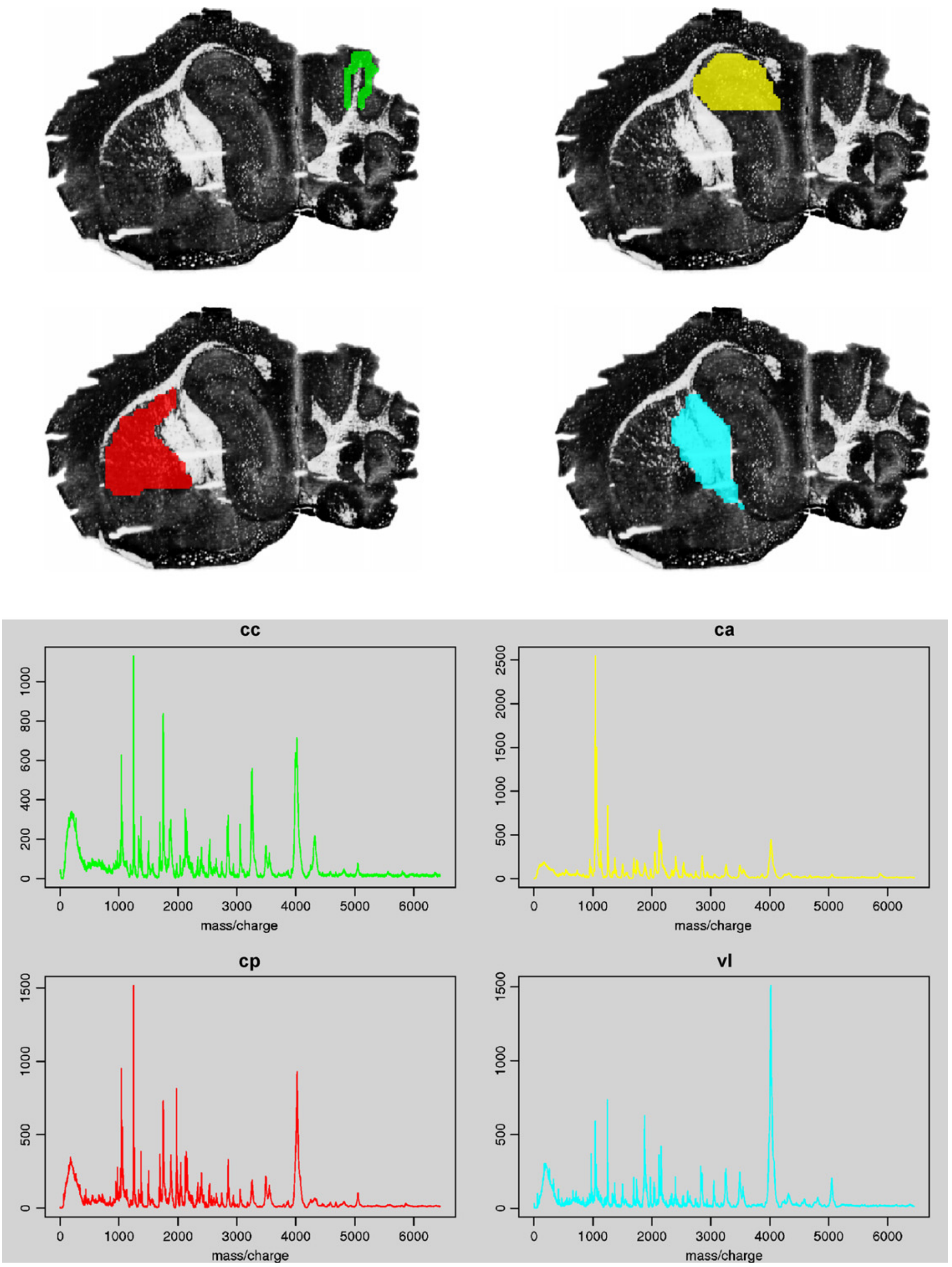

Fig. 13. Partially labeled tissue regions that constitute the training data set. Green reflects the cerebellar cortex (cc), yellow the Ammon's horn section of the hippocampus (ca), red the cauda-putamen (cp) region and blue the lateral ventricle (vl) area. Averaged spectra are presented for each tissue region.

effectively separates the lateral ventricle (vl) and cauda-putamen (cp) from the surrounding tissue. The classification for the ventricle area additionally draws in the elongated corpus callosum and cerebellar nucleus regions as well, as these regions share a panel of common molecules within the measured mass range. The cerebellar cortex (cc) label exceeds its intended boundaries encapsulating only the lobes of the nucleus, and instead delineates all non-nucleus tissue of the cerebellum (right part of the tissue containing the tree-like cerebellar nucleus). The remaining hippocampus label (ca) extends to capture the complete hippocampus and most of the remaining unlabeled areas of the tissue. This case study shows that classification using a simple linear kernel can already provide significant informative insight for the clinician and histopathologist to assess tissue type, structure, and content. Such considerations become increasingly important as MSI moves from the fundamental toward the clinical path in settings such as tumor detection and assessment. The digital staining that was demonstrated also holds value for more fundamental exploratory studies of tissue as it can highlight similarity in content between different cell types and tissue areas.

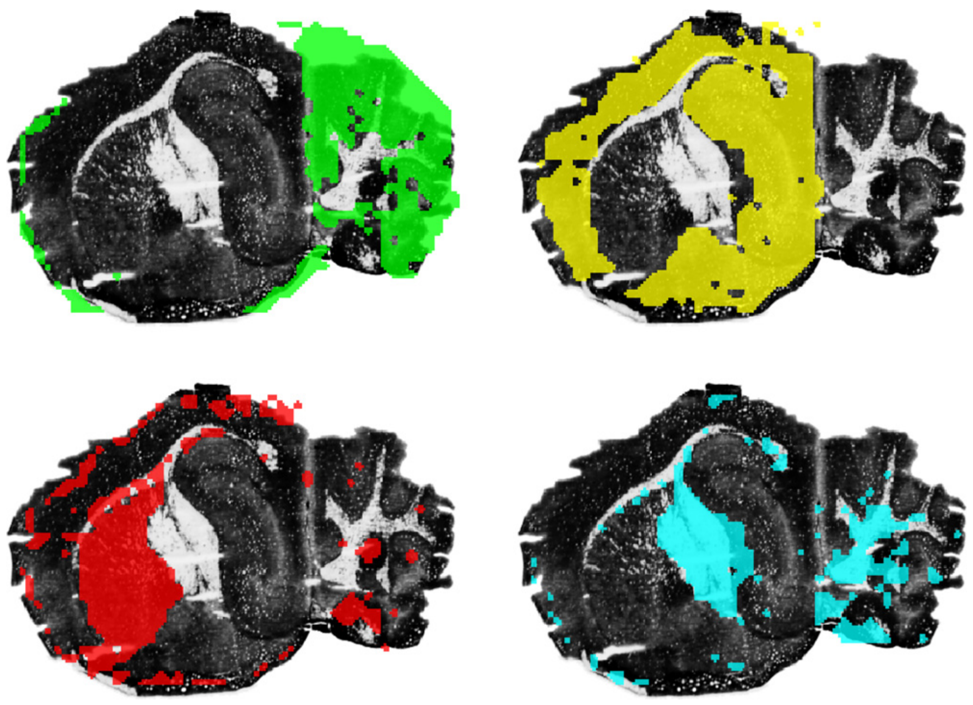

Fig. 14. SVM predictions for the whole set of spectra. Green reflects the cerebellar cortex (cc), yellow the Ammon's horn section of the hippocampus (ca), red the cauda-putamen (cp) region and blue the lateral ventricle (vl) area.

4. Conclusions

This tutorial presented an overview on SVMs and closely related techniques. These methods can be used for pattern classification to learn from data and predict future data. SVMs employ a trade-off between model complexity and empirical risk minimization. This involves the maximum margin principle, being the construction of a hyperplane that maximally separates the classes. Maximizing this margin corresponds to minimizing a weight vector. SVMs incorporate a mapping of the input data to a high-(or even infinite-)dimensional feature space. The mapping is defined through a kernel, i.e. a similarity measure, to take care of nonlinearity. Apart from the primal formulation, the problem can be formulated in the dual space by means of Lagrange multipliers. The unique solution can be found by solving a convex quadratic programming problem. SVMs possess the sparseness property since the solution is defined based on a set of support vectors. The tuning of hyper-parameters for regularization or for the kernel can be performed using a cross-validation strategy. LS-SVMs are based on the SVM formulation as the inequality constraints are replaced with equality constraints and a least squares loss function is introduced. This results in solving a set of linear equations, which significantly reduces the complexity and computational effort, but loosing the sparseness property. SVMs and LS-SVMs are in the first place binary classification methods. However, the methodology can be extended for multi-class classification or the multi-class problem can be converted into a number of binary classification tasks. SVMs and LS-SVMs output values can be modified into probabilistic output by various procedures. A popular strategy is to use a sigmoid function. On the other hand, KLR directly yields probabilistic outcomes and can be extended to multi-class classification in a straightforward way. Various feature selection approaches were summarized since SVM-based methods can benefit from them. An important aspect is the evaluation of classifiers or classification algorithms, which require the use of appropriate statistical tests for comparison. An interesting performance measure is the AUC, while a popular strategy for evaluation is cross-validation. Finally, the use of the classification techniques has been illustrated on toy examples and different real-life classification problems.

Acknowledgements

This research is funded by a PhD grant of the Institute for the Promotion of Innovation through Science and Technology in Flanders (IWT-Vlaanderen); Research supported by Research Council KUL: GOA-AMBioRICS, several PhD/postdoc and fellow grants, Centers-of-Excellence Optimisation, IDO 05/010 EEG-fMRI, GOA/2004/05 (Mixing and Analyzing Real and Virtual Environments and Lighting); University Medical Center Nijmegen for acquiring the MRI and MRSI data; Institute for Molecules and Materials, Analytical Chemistry, Chemometrics Research Department of the Radboud University Nijmegen for preprocessing the MRSI data; Arend Heerschap; Marinette van der Graaf; Ben Van Calster; Etienne Waelkens; Flemish Government: FWO: PhD/postdoc grants, projects, G.0360.05 (Advanced EEG analysis techniques for epilepsy monitoring), G.0519.06 (Noninvasive brain oxygenation), G.0341.07 (data fusion), G.0321. 06 (Numerical tensor techniques for spectral analysis), G.0302.07 (Support vector machines and kernel methods), G.0566.06 (Computational strategies for shape modeling and matching and their application in medical image analysis), research communities (ICCoS, ANMMM); IWT: PhD Grants; Belgian Federal Government: DWTC (IUAP IV-02 (1996–2001), IUAP V-22 (2002–2006): Dynamical Systems and Control: Computation, Identification & Modelling) and Belgian Federal Science Policy Office IUAP P6/04 (dynamical systems, control and optimization, 2007–2011); EU: BIOPATTERN (contract no. FP6-2002-IST 508803), eTUMOUR (contract no. FP6-2002-LIFESCIHEALTH 503094), HealthAgents (contract no. FP6-2005-IST 027213), FAST (contract no. FP6-019279-2), ESA: Cardiovascular Control (Prodex-8 C90242).

References

[1] E. Alpaydin, Introduction to Machine Learning, MIT Press, Cambridge, 2004.

[2] C.M. Bishop, Pattern Recognition and Machine Learning, Springer, New York, 2006.
[3] B.E. Boser, I.M. Guyon, V.N. Vapnik, A training algorithm for optimal margin classifiers, in: Proceedings of the Annual Workshop on Computational Learning Theory, Pittsburgh, 1992, pp. 144–152.
[4] V.N. Vapnik, The Nature of Statistical Learning Theory, Springer, New York, 1995.
[5] C. Cortes, V.N. Vapnik, Support-vector networks, Mach. Learn. 20 (1995) 273–297.
[6] L. Lukas, A. Devos, J.A.K. Suykens, L. Vanhamme, F.A. Howe, C. Majos, A. Moreno-Torres, M. Van der Graaf, A.R. Tate, C. Arus, S. Van Huffel, Brain tumor classification based on long echo proton MRS signals, Artif. Intell. Med. 31 (2004) 73–89.
[7] Y.-J. Lee, O.L. Mangasarian, W.H. Wolberg, Breast cancer survival and chemotherapy: a support vector machine analysis, DIMACS Ser. Discrete Math. Theor. Comput. Sci. 55 (2000) 1–10.
[8] M.P.S. Brown, W.N. Grundy, D. Lin, N. Cristianini, C.W. Sugnet, T.S. Furey, M. Ares, D. Haussler Jr., Knowledge-based analysis of microarray gene expression data by using support vector machines, Proc. Natl. Acad. Sci. U. S. A. 97 (1) (2000) 262–267.
[9] J. Luts, T. Laudadio, A.J. Idema, A.W. Simonetti, A. Heerschap, D. Vandermeulen, J.A.K. Suykens, S. Van Huffel, Nosologic imaging of the brain: segmentation and classification using MRI and MRSI, NMR Biomed. 22 (2009) 374–390.
[10] D.T. Holloway, M. Kon, C. DeLisi, Machine learning for regulatory analysis and transcription factor target prediction in yeast, Syst. Synth. Biol. 1 (1) (2007) 25–46.
[11] C.J.C. Burges, A tutorial on support vector machines for pattern recognition, Data Min. Knowl. Discov. 2 (1998) 121–167.
[12] V.N. Vapnik, Statistical Learning Theory, Wiley, New York, 1998.
[13] N. Cristianini, J. Shawe-Taylor, An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods, Cambridge University Press, Cambridge, 2000.
[14] B. Schölkopf, A.J. Smola, Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond, MIT Press, Cambridge, 2001.
[15] J.A.K. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor, J. Vandewalle, Least Squares Support Vector Machines, World Scientific, Singapore, 2002.
[16] S. Abe, Support Vector Machines for Pattern Classification, Springer, New York, 2005.
[17] J.A.K. Suykens, J. Vandewalle, Least squares support vector machine classifiers, Neural Process. Lett. 9 (3) (1999) 293–300.
[18] J.C. Platt, Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods, in: P.J. Bartlett, B. Schölkopf, D. Schuurmans, A.J. Smola (Eds.), Advances in Large Margin Classifiers, MIT Press, Cambridge, 2000, pp. 61–74.
[19] H.-T. Lin, C.-J. Lin, R.C. Weng, A note on Platt's probabilistic outputs for support vector machines, Mach. Learn. 68 (2007) 267–276.
[20] B. Zadrozny, C. Elkan, Transforming classifier scores into accurate multiclass probability estimates, in: Proceedings of the International Conference on Knowledge Discovery and Data Mining, Edmonton, 2002, pp. 694–699.
[21] P. Sollich, Bayesian methods for support vector machines: evidence and predictive class probabilities, Mach. Learn. 46 (2002) 21–52.
[22] T. Van Gestel, J.A.K. Suykens, G. Lanckriet, A. Lambrechts, B. De Moor, J. Vandewalle, Bayesian framework for least squares support vector machine classifiers, Gaussian processes and kernel Fisher discriminant analysis, Neural Comput. 14 (2002) 1115–1147.
[23] D.J.C. MacKay, Probable networks and plausible predictions – a review of practical Bayesian methods for supervised neural networks, Netw. Comput. Neural Syst. 6 (1995) 469–505.
[24] K. Crammer, Y. Singer, On the learnability and design of output codes for multiclass problems, Mach. Learn. 47 (2002) 201–233.
[25] J.A.K. Suykens, J. Vandewalle, Multiclass least squares support vector machines, in: Proceedings of the International Joint Conference on Neural Networks, Washington, DC, 1999, pp. 900–903.
[26] T.G. Dietterich, G. Bakiri, Solving multiclass learning problems via error-correcting output codes, J. Artif. Intell. Res. 2 (1994) 263–286.
[27] E.A. Allwein, R.E. Schapire, Y. Singer, Reducing multiclass to binary: a unifying approach for margin classifiers, J. Mach. Learn. Res. 1 (2001) 113–141.
[28] U.H.-G. Kreßel, Pairwise classification and support vector machines, in: B. Schölkopf, C.J.C. Burges, A.J. Smola (Eds.), Advances in Kernel Methods: Support Vector Learning, MIT Press, Cambridge, 1999, pp. 255–268.
[29] T. Hastie, R. Tibshirani, Classification by pairwise coupling, Ann. Stat. 26 (1998) 451–471.
[30] P. Refregier, F. Vallet, Probabilistic approach for multiclass classification with neural networks, in: Proceedings of the International Conference on Artificial Networks, Amsterdam, 1991, pp. 1003–1007.
[31] D. Price, S. Knerr, L. Personnaz, G. Dreyfus, Pairwise neural network classifiers with probabilistic outputs, in: Proceedings of the Conference on Neural Information Processing Systems, Denver, 1994, pp. 1109–1116.
[32] J. Friedman, Another Approach to Polychotomous classification, Technical Report, Stanford University (1996).
[33] T.-F. Wu, C.-J. Lin, R.C. Weng, Probability estimates for multi-class classification by pairwise coupling, J. Mach. Learn. Res. 5 (2004) 975–1005.
[34] J. Zhu, T. Hastie, Kernel logistic regression and the import vector machine, J. Comput. Graph. Stat. 14 (1) (2005) 185–205.
[35] P. Karsmakers, K. Pelckmans, J.A.K. Suykens, Multi-class kernel logistic regression: a fixed-size implementation, in: Proceedings of the International Joint Conference on Neural Networks, Orlando, 2007, pp. 1756–1761.
[36] S.S. Keerthi, K.B. Duan, S.K. Shevade, A.N. Poo, A fast dual algorithm for kernel logistic regression, Mach. Learn. 61 (2005) 151–165.
[37] T.G. Dietterich, Approximate statistical tests for comparing supervised classification learning algorithms, Neural Comput. 10 (-1923) 1895.
[38] D. Wettschereck, D.W. Aha, T. Mohri, A review and empirical evaluation of feature weighting methods for a class of lazy learning algorithms, Artif. Intell. Rev. 11 (1997) 273–314.
[39] R. Kohavi, G.H. John, Wrappers for feature subset selection, Artif. Intell. 97 (1997) 273–324.
[40] A.L. Blum, P. Langley, Selection of relevant features and examples in machine learning, Artif. Intell. 97 (1997) 245–271.
[41] G.H. John, R. Kohavi, K. Pfleger, Irrelevant features and the subset selection problem, in: Proceedings of the International Conference on Machine Learning, New Brunswick, 1994, pp. 121–129.
[42] J. Neter, M.H. Kutner, W. Wasserman, C.J. Nachtsheim, Applied Linear Statistical Models, McGraw-Hill, Boston, 1996.
[43] M.L. Ginsberg, Essentials of Artificial Intelligence, Morgan Kaufmann, Palo Alto, 1993.
[44] I.M. Guyon, J. Weston, S. Barnhill, V.N. Vapnik, Gene selection for cancer classification using support vector machines, Mach. Learn. 46 (2002) 389–422.
[45] F. Ojeda, J.A.K. Suykens, B. De Moor, Low rank updated LS-SVM classifiers for fast variable selection, Neural Netw. 21 (2–3) (2008) 437–449.
[46] J. Weston, S. Mukherjee, O. Chapelle, M. Pontil, T. Poggio, V.N. Vapnik, Feature selection for SVMs, in: Proceedings of the Conference on Neural Information Processing Systems, Denver, 2000, pp. 668–674.
[47] O. Chapelle, V.N. Vapnik, O. Bousquet, S. Mukherjee, Choosing multiple parameters for support vector machines, Mach. Learn. 46 (1–3) (2002) 131–159.
[48] C.E. Rasmussen, C.K.I. Williams, Gaussian Processes for Machine Learning, MIT Press, Cambridge, 2006.
[49] J. Weston, A. Elisseeff, B. Schölkopf, M. Tipping, Use of the zero-norm with linear models and kernel methods, J. Mach. Learn. Res. 3 (2003) 1439–1461.
[50] R. Tibshirani, Regression shrinkage and selection via the lasso, J. R. Stat. Soc. Ser. B: Stat. Methodol. 58 (1996) 267–288.
[51] G.C. Cawley, Leave-one-out cross-validation based model selection criteria for weighted LS-SVMs, in: Proceedings of the International Joint Conference on Neural Networks, Vancouver, 2006, pp. 1661–1668.
[52] G.C. Cawley, N.L.C. Talbot, Preventing over-fitting during model selection via Bayesian regularisation of the hyper-parameters, J. Mach. Learn. Res. 8 (2007) 841–861.
[53] J.A. Hanley, B.J. McNeil, The meaning and use of the area under a receiver operating characteristic (ROC) curve, Radiology 143 (1982) 29–36.
[54] D.J. Hand, R.J. Till, A simple generalisation of the area under the ROC curve for multiple class classification problems, Mach. Learn. 45 (2001) 171–186.
[55] V.N. Vapnik, O. Chapelle, Bounds on error expectation for support vector machines, Neural Comput. 12 (9) (2000) 2013–2036.
[56] G.C. Cawley, N.L.C. Talbot, Fast exact leave-one-out cross-validation of sparse least-squares support vector machines, Neural Netw. 17 (2004) 1467–1475.
[57] T. Hastie, S. Rosset, R. Tibshirani, J. Zhu, The entire regularization path for the support vector machine, J. Mach. Learn. Res. 5 (2004) 1391–1415.
[58] E. Alpaydin, Combined 5×2cv F test for comparing supervised classification learning algorithms, Neural Comput. 11 (1999) 1885–1892.
[59] C. Nadeau, Y. Bengio, Inference for the generalization error, Mach. Learn. 52 (2003) 239–281.
[60] R.R. Bouckaert, Choosing between two learning algorithms based on calibrated tests, in: Proceedings of the International Conference on Machine Learning, Washington, DC, 2003, pp. 51–58.
[61] R.R. Bouckaert, E. Frank, Evaluating the replicability of significance tests for comparing learning algorithms, Lect. Notes Comput. Sci. 3056 (2004) 3–12.
[62] Y. Grandvalet, Y. Bengio, Hypothesis Testing for Cross-Validation, Technical Report 1285, Departement d'Informatique et Recherche Operationnelle, Universite de Montreal (2006).
[63] O.T. Yildiz, E. Alpaydin, Ordering and finding the best of $K > 2$ supervised learning algorithms, IEEE Trans. Pattern Anal. Mach. Intell. 28 (2006) 392–402.
[64] J. Demsar, Statistical comparisons of classifiers over multiple data sets, J. Mach. Learn. Res. 7 (2006) 1–30.
[65] S. Garcia, F. Herrera, An extension on "Statistical comparisons of classifiers over multiple data sets" for all pairwise comparisons, J. Mach. Learn. Res. 9 (2008) 2677–2694.
[66] J.T. Leek, J.D. Storey, A general framework for multiple testing dependence, Proc. Natl. Acad. Sci. U. S. A. 105 (2008) 18718–18723.
[67] J. Cohen, The earth is round ($p < .05$), Am. Psychol. 49 (1994) 997–1003.
[68] J.A.C. Sterne, G.D. Smith, Sifting the evidence – what's wrong with significance tests? Br. Med. J. 322 (2001) 226–231.
[69] C.-C. Chang, C.-J. Lin, LIBSVM: A Library for Support Vector Machines, National Taiwan University, http://www.csie.ntu.edu.tw/~cjlin/libsvm/ (2001).
[70] A. Karatzoglou, A. Smola, K. Hornik, A. Zeileis, kernlab – an S4 package for kernel methods in R, J. Stat. Softw. 11 (9) (2004) 1–20, http://www.jstatsoft.org/v11/i09/.
[71] P. Sollich, C. Gold, Bayesian Support Vector Machine Hyperparameter Tuning, King's College London, http://www.mth.kcl.ac.uk/~psollich/BayesSVM/ (2005).
[72] K. Pelckmans, J.A.K. Suykens, T. Van Gestel, J. De Brabanter, L. Lukas, B. Hamers, B. De Moor, J. Vandewalle, LS-SVMlab: A Matlab/C Toolbox for Least Squares

Support Vector Machines, ESAT-SISTA, Katholieke Univiversiteit Leuven, Belgium, http://www.esat.kuleuven.be/sista/lssvmlab/ (2002).
[73] T. Joachims, Learning to Classify Text Using Support Vector Machines, Kluwer, 2002.
[74] A.P. Gasch, P.T. Spellman, C.M. Kao, O. Carmel-Harel, M.B. Eisen, G. Storz, D. Botstein, P.O. Brown, Genomic expression programs in the response of yeast cells to environmental changes, Mol. Biol. Cell 11 (12) (2000) 4241–4257.
[75] C.T. Harbison, D.B. Gordon, T.I. Lee, N.J. Rinaldi, K.D. Macisaac, T.W. Danford, N.M. Hannett, J.-B. Tagne, D.B. Reynolds, J. Yoo, E.G. Jennings, J. Zeitlinger, D.K. Pokholok, M. Kellis, P.A. Rolfe, K.T. Takusagawa, E.S. Lander, D.K. Gifford, E. Fraenkel, R.A. Young, Transcriptional regulatory code of a eukaryotic genome, Nature 431 (7004) (2004) 99–104.
[76] K. Lemmens, T. Dhollander, T. De Bie, P. Monsieurs, K. Engelen, B. Smets, J. Winderickx, B. De Moor, K. Marchal, Inferring transcriptional modules from ChIP-chip, motif and microarray data, Genome Biol. 7 (5) (2006) R37.
[77] J. Qian, J. Lin, N.M. Luscombe, H. Yu, M. Gerstein, Prediction of regulatory networks: genome-wide identification of transcription factor targets from gene expression data, Bioinformatics 19 (15) (2003) 1917–1926.
[78] F. Mordelet, J.-P. Vert, SIRENE: supervised inference of regulatory networks, Bioinformatics 24 (16) (2008) 76–i82.
[79] S. An, W. Liu, S. Venkatesh, Fast cross-validation algorithms for least squares support vector machine and kernel ridge regression, Pattern Recogn. 40 (8) (2007) 2154–2162.
[80] Z. Hu, J. Mellor, J. Wu, C. DeLisi, VisANT: an online visualization and analysis tool for biological interaction data, BMC Bioinformatics 5 (1) (2004) 17.
[81] M. Stoeckli, P. Chaurand, D.E. Hallahan, R.M. Caprioli, Imaging mass spectrometry: a new technology for the analysis of protein expression in mammalian tissues, Nat. Med. 7 (4) (2001) 493–496.
[82] L.A. McDonnell, R.M.A. Heeren, Imaging mass spectrometry, Mass Spectrom. Rev. 26 (4) (2007) 606–643.
[83] R. Van de Plas, F. Ojeda, M. Dewil, L. Van Den Bosch, B. De Moor, E. Waelkens, Prospective exploration of biochemical tissue composition via imaging mass spectrometry guided by principal component analysis, in: Proceedings of the Pacific Symposium on Biocomputing 12, Maui, 2007, pp. 458–469.
[84] R. Van de Plas, B. De Moor, E. Waelkens, Imaging mass spectrometry based exploration of biochemical tissue composition using peak intensity weighted PCA, in: IEEE/NIH Life Science Systems and Applications Workshop, Bethesda, 2007.
[85] R. Van de Plas, K. Pelckmans, B. De Moor, E. Waelkens, Spatial querying of imaging mass spectrometry data: a nonnegative least squares approach, in: Neural Information Processing Systems Workshop on Machine Learning in Computational Biology, 2007.
[86] M. Hanselmann, U. Kothe, M. Kirchner, B.Y. Renard, E.R. Amstalden, K. Glunde, R.M.A. Heeren, F.A. Hamprecht, Toward digital staining using imaging mass spectrometry and random forests, J. Proteome Res. 8 (7) (2009) 3558–3567.