BIOINFORMATICS

Vol. 27 ISMB 2011, pages i342–i348
doi:10.1093/bioinformatics/btr204

ccSVM: correcting Support Vector Machines for confounding factors in biological data classification

Limin Li[1,2,*], Barbara Rakitsch[1] and Karsten Borgwardt[1,*]

[1]Machine Learning and Computational Biology Research Group, Max Planck Institutes Tübingen, Tübingen, Germany and [2]Department of Mathematics, Xi'an Jiaotong University, Xi'an 710049, China

ABSTRACT

Motivation: Classifying biological data into different groups is a central task of bioinformatics: for instance, to predict the function of a gene or protein, the disease state of a patient or the phenotype of an individual based on its genotype. Support Vector Machines are a wide spread approach for classifying biological data, due to their high accuracy, their ability to deal with structured data such as strings, and the ease to integrate various types of data. However, it is unclear how to correct for confounding factors such as population structure, age or gender or experimental conditions in Support Vector Machine classification.

Results: In this article, we present a Support Vector Machine classifier that can correct the prediction for observed confounding factors. This is achieved by minimizing the statistical dependence between the classifier and the confounding factors. We prove that this formulation can be transformed into a standard Support Vector Machine with rescaled input data. In our experiments, our confounder correcting SVM (ccSVM) improves tumor diagnosis based on samples from different labs, tuberculosis diagnosis in patients of varying age, ethnicity and gender, and phenotype prediction in the presence of population structure and outperforms state-of-the-art methods in terms of prediction accuracy.

Availability: A ccSVM-implementation in MATLAB is available from http://webdav.tuebingen.mpg.de/u/karsten/Forschung/ISMB11_ccSVM/.

Contact: limin.li@tuebingen.mpg.de; karsten.borgwardt@tuebingen.mpg.de

1 INTRODUCTION

Several of the most intensively studied problems in computational biology are classification tasks: for instance, predicting the function of a gene, the disease state of a patient, the reaction of a patient to a therapy and the phenotype of an individual based on its genotype. The abstract task is to predict the class y of an biological subject based on its features x. Emerging and existing high-throughput technologies allow us to measure the features of genes, proteins and individuals at an unprecedented resolution and scale, and the hope is that this rich knowledge will lead to ever more accurate data classification.

One of the most prominent and most successful classification algorithms are Support Vector Machines (SVMs) (Cortes and Vapnik, 1995; Schölkopf and Smola, 2002). They are based on the idea to separate objects from two classes by means of a hyperplane; new test objects are then predicted to belong to one of these two classes depending on which half-space they are located in. Their popularity is due to several reasons: first, SVMs have shown excellent prediction accuracy in many studies (Noble, 2006). Second, SVMs can be directly applied to structured data, such as strings (Leslie *et al.*, 2002) or graphs (Borgwardt *et al.*, 2005), which are abundant in bioinformatics. Third, SVMs allow for straightforward data integration of several data types (Lanckriet *et al.*, 2004).

However, SVMs suffer from one limitation: it is unclear how to correct for confounding variables in SVM predictions. According to Meinert (Meinert and Tonascia, 1986), a confounder is defined as a variable which is related to two factors of interest, and which falsely obscures or accentuates the relationship between them. In this article, we present an SVM which can correct for observed confounding variables.

The detrimental effects of confounders are observable in many classification tasks in molecular biology, as illustrated by the following two examples: one may want to predict the phenotypes of plants based on their genotype, typically represented by single nucleotide polymorphisms that represent sequence variation in an individual. In this task, population structure, that is systematic ancestry differences between plants with different phenotypes, may have a confounding effect on the prediction (Price *et al.*, 2010). For instance, if there is a correlation between population structure and phenotype, the classifier may rely on SNPs that correlate with population structure, and subsequently, its predictions may be wrong on datasets from different geographic origins where the phenotype–population correlation is less pronounced or not present.

Another example is drug treatment response in patients from gene expression profiles. Confounding factors may be the age, the gender or the ethnicity of the patients, each of which may correlate with the treatment response and the expression levels of certain genes (Holsboer, 2008). When predicting on patients with different age, sex or ethnic background, the learnt classifier may poorly generalize.

Our goal in this article is to define a confounder-correcting Support Vector Machine (ccSVM) that removes the confounding side information to the largest extent possible. To achieve this, we strive to make the classifier base its prediction on features that do not correlate with the confounding variable.

The remainder of this article is structured as follows. In Section 2, we present the ccSVM (Section 2.3), and the classifier (Section 2.1) and the statistical dependence measure (Section 2.2) it is based upon. We prove that the ccSVM can be computed highly efficiently with existing software packages in Section 2.4. In Section 3, we show that our method improves upon several state-of-the-art classifiers in tumor diagnosis (Section 3.3), tuberculosis diagnosis (Section 3.4)

*To whom correspondence should be addressed.

and plant phenotype prediction (Section 3.5). In Section 4, we summarize our findings and give an outlook to future work.

2 ccSVM APPROACH

We first introduce the SVM (Section 2.1) and the Hilbert-Schmidt Independence Criterion (HSIC) (Section 2.2), that is the measure of statistical dependence that we use to then define our confounder-correcting SVM (Section 2.3). In Section 2.4, we show how to efficiently solve the ccSVM optimization problem.

2.1 SVMs

SVMs are supervised learning methods (Schölkopf and Smola, 2002; Vapnik and Chervonenkis, 1974) that are widely used in molecular biology (Schölkopf *et al.*, 2004). The SVM takes a set of input data with corresponding class labels, and predicts to which class a new input belongs. Suppose we are given the data $(\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_m, y_m)$, where $\mathbf{x}_i$ is an observation and y_i is its class label ($+1$ or -1). The original SVM assumes the data are separable by a hyperplane and obtains this hyperplane by maximizing the margin, that is the minimum distance between the hyperplane and points from each class. Once the hyperplane is learnt from the training data, it can be used to predict the class label of new test points. Suppose the hyperplane is in the form of $f(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b$, then the model is as follows:

$$\min_{\mathbf{w}\in\mathbb{R}^n, b\in\mathbb{R}} ||\mathbf{w}||^2 \tag{1}$$

subject to

$$y_i(\langle \mathbf{w}, \mathbf{x}_i\rangle + b) \geq 1 \tag{2}$$

By considering the case when data are non-separable, a soft margin SVM was proposed to punish the training errors as follows (Cortes and Vapnik, 1995):

$$\min_{\mathbf{w}\in\mathbb{R}^n, b\in\mathbb{R}, \xi\in\mathbb{R}^m} ||\mathbf{w}||^2 + C\sum_{i=1}^{m}\xi_i \tag{3}$$

subject to

$$\begin{aligned} &y_i(\langle \mathbf{w}, \mathbf{x}_i\rangle + b) \geq 1 - \xi_i \\ &\xi_i \geq 0, \end{aligned} \tag{4}$$

where C determines the trade-off between margin maximization and training errors minimization, and ξ_i is the term by which the object x_i violates the inequality (2). Once $\mathbf{w}$ and b are obtained, one can predict the class label for a new observation $\mathbf{x}$ by the decision function: $\text{sgn}(\mathbf{w}^T\mathbf{x} + b)$.

The dual problem of (3) is

$$\max_{\alpha}\left\{-\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} y_i y_j \alpha_i \alpha_j \mathbf{x}_i^T\mathbf{x}_j + \sum_{i=1}^{m}\alpha_i\right\} \tag{5}$$

under the constraints of

$$\begin{aligned} &\textstyle\sum_{i=0}^{m} y_i\alpha_i = 0, \\ &0 \leq \alpha_i \leq C, \text{ for } i = 1, \ldots, m \end{aligned} \tag{6}$$

The Karush–Kuhn–Tucker conditions (Kuhn and Tucker, 1951) imply that $\mathbf{w} = \sum_{i=1}^{n} y_i\alpha_i\mathbf{x}_i$. Thus, after we obtain α_i by solving (5), the decision function will be

$$\text{sgn}\left(\sum_{i=1}^{m} y_i\alpha_i\mathbf{x}_i^T\mathbf{x} + b\right).$$

The kernel trick is to replace $\mathbf{x}_i^T\mathbf{x}_j$ by $k(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^T\phi(\mathbf{x}_j)$ in (5), where $k(\mathbf{x}, \mathbf{x}')$ is a kernel function such that its discretization $\mathbf{K}_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$ is a positive definite matrix. The decision function can then be represented as

$$\text{sgn}\left(\sum_{i=1}^{m} y_i\alpha_i k(\mathbf{x}_i, \mathbf{x}) + b\right).$$

2.2 HSIC

The HSIC is a measure of statistical independence (Gretton *et al.*, 2005). Intuitively, HSIC can be thought of as a squared correlation coefficient between two random variables x and z computed in feature spaces $\mathcal{F}$ and $\mathcal{G}$.

In more detail, let x be a random variable from the domain $\mathcal{X}$ and z a random variable from the domain $\mathcal{Z}$. Let $\mathcal{F}$ and $\mathcal{G}$ be feature spaces on $\mathcal{X}$ and $\mathcal{Z}$ with associated kernels $k: \mathcal{X}\times\mathcal{X} \to \mathbb{R}$ and l: $\mathcal{Z}\times\mathcal{Z} \to \mathbb{R}$. If we draw pairs of samples (x, z) and (x', z') from x and z according to a joint probability distribution $p_{(x,z)}$, then the HSIC can be computed in terms of kernel functions via:

$$\begin{aligned} \text{HSIC}(p_{(x,z)}, \mathcal{F}, \mathcal{G}) = &\mathbf{E}_{x,x',z,z'}[k(x,x')l(z,z')] \qquad (7)\\ &+\mathbf{E}_{x,x'}[k(x,x')]\mathbf{E}_{z,z'}[l(z,z')] \\ &-2\mathbf{E}_{x,z}[\mathbf{E}_{x'}[k(x,x')]\mathbf{E}_{z'}[l(z,z')]], \qquad (8) \end{aligned}$$

where $\mathbf{E}$ is the expectation operator. The empirical estimator of HSIC for a finite sample of points X and Z from x and z with $p_{(x,z)}$ was shown in Gretton *et al.* (2005) to be

$$\text{HSIC}((X, Z), \mathcal{F}, \mathcal{G}) \propto \mathbf{tr}(\mathbf{KHLH}), \tag{9}$$

where $\mathbf{tr}$ is the trace of the products of the matrices, $\mathbf{H}$ is a centering matrix $\mathbf{H}_{ij} = \delta_{(i,j)} - \frac{1}{m}$ (where $\delta_{(i,j)} = 1$ if $i = j$ and $\delta_{(i,j)} = 0$ otherwise), $\mathbf{K}$ and $\mathbf{L}$ are the kernel matrices on the two random variables of size $m \times m$ and m is the number of observations. The larger HSIC, the more likely it is that X and Z are not independent from each other.

2.3 The ccSVM

Via HSIC we can now define an SVM that can use side information to avoid confounding. Suppose m samples with their feature vectors $(\mathbf{x}_1, \ldots, \mathbf{x}_m)$, class labels $(y_1, \ldots, y_m)$ and side information $(\mathbf{z}_1, \ldots, \mathbf{z}_m)$ are given. $\mathbf{x}_i$ is a n-dimensional column vector representing the features of sample i, $y_i \in \{-1, +1\}$ is the class label for $\mathbf{x}_i$ and $\mathbf{z}_i$ is the some kind of side information on object i, e.g. region, country, age, gender, lab membership or population structure.

$\mathbf{L} \in \mathbb{R}^{m\times m}$ is a predefined kernel matrix which is generated based on a kernel l on the side information, that is $\mathbf{L}_{ij} = l(\mathbf{z}_i, \mathbf{z}_j)$. We call $\mathbf{L}$ the side information kernel matrix.

We propose to obtain a classifier by minimizing the following objective function:

$$\min_{\mathbf{w}\in\mathbb{R}^n, b\in\mathbb{R}} ||\mathbf{w}||^2 + \lambda\,\text{tr}(\mathbf{KHLH}) \tag{10}$$

subject to

$$\begin{aligned} &y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1 \\ &\mathbf{K}_{ij} = \langle \mathbf{w} \odot \mathbf{x}_i, \mathbf{w} \odot \mathbf{x}_j \rangle, \end{aligned} \tag{11}$$

where $\odot$ represents the element-wise product of two vectors.

The objective function includes two terms. To minimize the first term is to maximize the classifier margin, as in a standard SVM. The second term tr(**KHLH**) is the HSIC, which measures the independence between two kernels, the reweighted kernel matrix **K** and side information kernel matrix **L**. Here the reweighted kernel **K** is the kernel after reweighting each feature by its weight in **w**.

To minimize HSIC is to make the dependence between the reweighted kernel matrix and the side information kernel matrix as small as possible. In other words, besides maximizing the margin, the ccSVM also tries to weaken the effect of the side information on the weight vector **w** of the classifier. It rewards solutions in which the input data—after being reweighted by weight vector **w**—are as independent as possible from the side information, thereby favoring a solution that does not rely on the side information. A constant $\lambda > 0$ determines the trade-off between margin maximization and dependence minimization.

Note that in practice, a separating hyperplane may not exist. A possible soft margin classifier can be obtained by minimizing the following objective function:

$$\min_{\mathbf{w} \in \mathbb{R}^n, b \in \mathbb{R}, \xi \in \mathbb{R}^m} ||\mathbf{w}||^2 + \lambda \mathrm{tr}(\mathbf{KHLH}) + C \sum_{i=1}^{m} \xi_i \tag{12}$$

subject to

$$\begin{aligned} &y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1 - \xi_i \\ &\mathbf{K}_{ij} = \langle \mathbf{w} \odot \mathbf{x}_i, \mathbf{w} \odot \mathbf{x}_j \rangle \\ &\xi_i \geq 0. \end{aligned} \tag{13}$$

Two constants C and λ determine the trade-off among margin maximization, dependence minimization and training error minimization.

2.4 Transformation into SVM problem with rescaled input

Next, we show how to solve the ccSVM optimization problem (12) by rescaling the input of a standard SVM. For this purpose, we denote **HLH** by $\tilde{\mathbf{L}}$, and we define $\mathbf{w} = (w_1, \ldots, w_n)^T$ and $\mathbf{x}_i = (x_{1i}, \ldots, x_{ni})^T$. Then HSIC in (12) can be written as

$$\begin{aligned} &\mathrm{tr}(\mathbf{KHLH}) = \mathrm{tr}(\mathbf{K}\tilde{\mathbf{L}}) \\ &= \sum_{i,j=1}^{m} \tilde{\mathbf{L}}_{ij} \langle \mathbf{w} \odot \mathbf{x}_i, \mathbf{w} \odot \mathbf{x}_j \rangle \\ &= \sum_{i,j=1}^{m} \tilde{\mathbf{L}}_{ij} \sum_{k=1}^{n} w_k^2 x_{ki} x_{kj} \\ &= \sum_{k=1}^{n} w_k^2 \sum_{i,j=1}^{m} \tilde{\mathbf{L}}_{ij} x_{ki} x_{kj} \end{aligned} \tag{14}$$

Let $l_k = \sum_{i,j} \tilde{\mathbf{L}}_{ij} x_{ki} x_{kj}$, then (14) is equal to:

$$\sum_{k=1}^{n} w_k^2 l_k$$

Thus, the objective function in (12) becomes

$$\begin{aligned} &\sum_{k=1}^{n} w_k^2 + \lambda \sum_{k=1}^{n} w_k^2 l_k + C \sum_{i=1}^{m} \xi_i \\ &= \sum_{k=1}^{n} w_k^2 (1 + \lambda l_k) + C \sum_{i=1}^{m} \xi_i \end{aligned}$$

Let

$$\tilde{w}_k = w_k \sqrt{1 + \lambda l_k} \tag{15}$$

and

$$\tilde{x}_{ki} = \frac{x_{ki}}{\sqrt{1 + \lambda l_k}} \tag{16}$$

for $k = 1, \ldots, n$. Denote $\tilde{\mathbf{w}} = (\tilde{w}_1, \ldots, \tilde{w}_n)^T$ and $\tilde{\mathbf{x}}_i = (\tilde{x}_{1i}, \ldots, \tilde{x}_{ni})^T$. Then the optimization problem (12) becomes:

$$\min_{\tilde{\mathbf{w}} \in \mathbb{R}^n, b \in \mathbb{R}, \xi \in \mathbb{R}^m} ||\tilde{\mathbf{w}}||^2 + C \sum_{i=1}^{m} \xi_i \tag{17}$$

subject to

$$\begin{aligned} &y_i(\langle \tilde{\mathbf{w}}, \tilde{\mathbf{x}}_i \rangle + b) \geq 1 - \xi_i \\ &\xi_i \geq 0. \end{aligned} \tag{18}$$

Interestingly, the optimization problem (17) with the constraints in (18) is the standard SVM, which can be solved using libsvm (Chang and Lin, 2001) or other SVM software. Thus, in order to solve the ccSVM problem (12), one only needs to first rescale each feature according to the formula (16) and then solve a standard SVM problem (17). Note that Equation (17) uses a linear kernel $\tilde{\mathbf{X}}^T \tilde{\mathbf{X}}$, where $\tilde{\mathbf{X}} = (\tilde{\mathbf{x}}_1, \ldots, \tilde{\mathbf{x}}_m) \in \mathbb{R}^{n \times m}$. While the rescaling step (16) does not lend itself to kernelization, one can kernelize (17) and (18) by replacing $\tilde{\mathbf{x}}_i^T \tilde{\mathbf{x}}_j$ by $\phi(\tilde{\mathbf{x}}_i)^T \phi(\tilde{\mathbf{x}}_j)$ in its dual problem.

3 EXPERIMENTS

In our experiments, we examine three different applications of the ccSVM in bioinformatics: microarray cross-platform comparability on a simulated dataset, disease outcome prediction with correction for various kinds of side information and phenotype prediction with population structure correction.

3.1 Parameter selection

There are two parameters in the ccSVM model (12): λ and C. We choose the parameters based on cross-validation on the training dataset only. We split all the training data into several (for example, 5) folds, and each time we take 1-fold as test set and the others as training set. We first set $\lambda = 0$ and select the C by which we can get the best average area under curve (AUC) using a standard SVM. C can take one of the values in $\{2^{-8}, 2^{-4}, 2^{-2}, 1, 2^2, 2^4, 2^8\}$. Then we fix C in the ccSVM, and select the λ such that it gives the best average AUC in the ccSVM. λ is chosen from the values $\{10^{-8}, 10^{-4}, 10^{-2}, 1, 10^2, 10^4, 10^8\}$. This parameter selection is performed on the training dataset only.

3.2 Comparison partners

We compare the ccSVM to the following comparison partners:

- Standard SVM: we use linear kernel $\mathbf{K}_{\mathrm{SVM}} = \mathbf{X}^T \mathbf{X}$ in the standard SVM, where $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_m) \in \mathbb{R}^{n \times m}$.
- (K+L)SVM: we integrate the side information with the original features by simply concatenating **X** and **L**. Thus, the number of features are $n+m$, where n is the number of original features,

and m is the number of side features. The linear kernel will be $\mathbf{K}_{(\mathrm{K+L})\mathrm{SVM}} = \mathbf{K}_{\mathrm{SVM}} + \mathbf{L}^T\mathbf{L}$. (K+L)SVM means that we use a standard SVM with kernel matrix $\mathbf{K}_{(\mathrm{K+L})\mathrm{SVM}}$.

- pcaSVM: we consider the first component from principle component analysis (PCA) to be most related to the side information, and then weaken the side-effect by removing it from the kernel matrix. Price *et al.* (2006) used a similar approach to correct for stratification in genome-wide association studies. Suppose the largest eigenvalue of $\mathbf{K}_{\mathrm{SVM}} = \mathbf{X}^T\mathbf{X}$ is σ and its corresponding eigenvector is v, then define the PCA correction kernel $\mathbf{K}_{\mathrm{PCA}} = \mathbf{K}_{\mathrm{SVM}} - \sigma vv^T$. pcaSVM means that we use a standard SVM with kernel matrix $\mathbf{K}_{\mathrm{PCA}}$.
- Confounder correcting logistic regression (ccLR): we consider the following logistic model

$$\ln\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 x_1 + \cdots + \beta_n x_n + u_1 l_1 + \cdots + u_m l_m,$$

where p is the probability of a sample being in one class (e.g. the positive class), β_i and u_i are parameters, x_i are the original features and l_i are the side features included in $\mathbf{L}$. Kang *et al.* (2010) applied a related mixed-model approach to correct for population structure in genome-wide association studies. In contrast to our approach, they are interested in quantitative phenotypes. In our experiments, besides standard logistic regression with maximum likelihood, a sparse Bayesian logistic regression model BLogReg (Cawley and Talbot, 2006) is also used to estimate the parameters β_i and u_i. ccLR with these two parameter estimation methods are denoted as ccLR(ML) and ccLR(BR), respectively.

3.3 Microarray cross-platform comparability

In this experiment, we compared the sensitivity of the ccSVM to a standard SVM on a microarray dataset which consists of samples from two different labs. A synthetic dataset was also generated to compare the ccSVM and standard SVM.

Data: P.Warnat *et al.* (2005) compared two studies on acute myeloid leukemia (AML): Bullinger *et al.* (2004) and Valk *et al.* (2004). The dataset Bullinger consists of 52 patients, and the dataset Valk of 97 patients. Both datasets share gene expression levels for $n = 7102$ genes. The prediction task is to differentiate between cancerous and normal tissue. The experiments of Bullinger *et al.* were carried out on a cDNA platform while Valk *et al.* used oligonucleotide microarrays.

Besides the real data, we also generated a synthetic dataset based on Bullinger and Valk: we picked randomly half of the genes and centered them to zero mean for each gene and each dataset separately, and kept the other half genes uncentered. The centered genes have no correlation with the lab membership while many of the uncentered genes have a strong correlation. Hence, difference in mean expression level seems to distinguish the expression values from these two labs.

We defined the side information matrix $\mathbf{L} \in \mathbb{R}^{m \times m}$ by the lab membership. $\mathbf{L}_{ij} = 1$ if patient i and patient j belong to the same lab, and $\mathbf{L}_{ij} = 0$ if the two patients belong to different labs.

Experimental setting: we first did 50 times 5-fold random cross-validation on the real data using the ccSVM and SVM, and report their average AUCs, standard errors and t-test P-values. For the ccSVM, we split the data randomly into 5-folds. We used 4-folds for training and 1-fold for testing. Then we fixed the parameters λ and C as explained in Section 3.1 with 4-fold cross-validation. With the obtained parameters, we trained the ccSVM on the training set and predicted on the test objects. The experiment was repeated five times until each fold served as test dataset once. For standard SVM and pcaSVM, we used the same experimental protocol, but we only needed to train C from the training data.

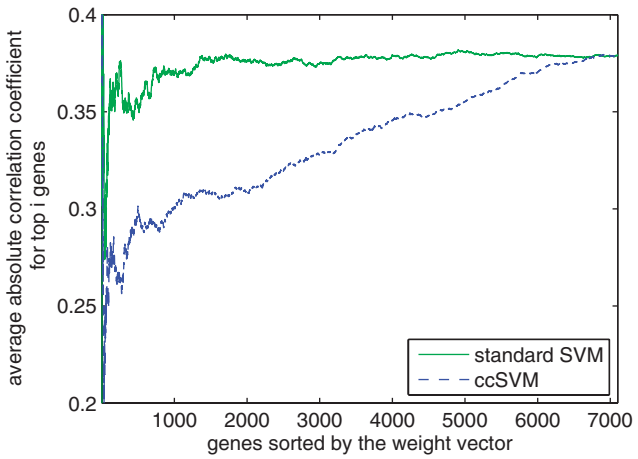

Fig. 1. Genes are sorted according to the weight vector of the ccSVM (blue dashed line) and according to the weight vector of the standard SVM (green line). The correlation coefficient between each gene expression level and lab membership is calculated. The averaged absolute correlation coefficient of the top i genes is plotted for gene i.

We then explored how the ccSVM corrects the normalized weight vector based on the synthetic data. We trained on a subset of the pooled Bullinger and Valk dataset. We determined the parameter C according to the experimental protocol outlined in Section 3.1 and fixed $\lambda = 1$. Therefore, we split the training set into 3-folds. With these optimized parameters, we trained our ccSVM jointly over all training objects and predicted on the test dataset. For training the standard SVM, we used the same experimental protocol.

Results: for the real data, we obtain an average AUC value of 0.911 ± 0.002 for the ccSVM and an AUC value of 0.822 ± 0.003 for the standard SVM. The P-value of the t-test is 4.8e-40. This result shows that our method is superior to the standard SVM.

For the synthetic data, we can see from Figure 1 that the ccSVM assigns large weights to genes that weakly correlate with the lab membership while the standard SVM assigns the weights without paying attention to the correlation to the lab membership.

3.4 Disease outcome prediction with various confounding factors

In this experiment, we analyzed the ability of the ccSVM to predict active tuberculosis based on blood transcriptional profiles. We used ethnicity, age and gender as confounding information.

Data: we obtained the dataset from Berry *et al.* (2010). It includes 103 blood samples from patients with active tuberculosis and 40 blood samples from healthy controls. The transcriptional signature of the blood samples were measured in a subsequent microarray experiment with $n = 48\,803$ gene expression levels.

We used three different confounding factors: ethnicity, gender and age. For ethnicity, we defined the information matrix as

Table 1. AUC and *P*-values for ccSVM, standard SVM, pcaSVM, (K+L)SVM and ccLR for the three different confounding variables on the Tuberculosis dataset

Side information	AUC_{ccSVM}	AUC_{SVM}	p_{SVM}	AUC_{pcaSVM}	p_{pcaSVM}	$AUC_{(K+L)SVM}$	$p_{(K+L)SVM}$	$AUC_{ccLR(ML)}$
Ethnicity	0.955 ± 0.002		6.3e-05		3.6e-09	0.942 ± 0.003	1.2e-04	0.499
Age	0.967 ± 0.002	0.939 ± 0.003	3.8e-12	0.933 ± 0.003	1.5e-18	0.943 ± 0.002	4.0e-16	0.499
Gender	0.938 ± 0.003		2.8e-01		6.2e-01	0.941 ± 0.003	1.7e-01	0.499

follows: $\mathbf{L}_{ij} = 1$ if the patient i and j belong to the same ethnic group, $\mathbf{L}_{ij} = 0$ if they do not. For gender, we defined $\mathbf{L}$ similarly: $\mathbf{L}_{ij} = 1$ if the patient i and j have the same gender, $\mathbf{L}_{ij} = 0$ if the patients have different gender. We used a Gaussian kernel for age as side information.

Experimental setting: for the ccSVM, standard SVM, pcaSVM and (K+L)SVM, we used the same experimental setting as described in Section 3.3. We again utilized the same experimental design for ccLR, but instead of setting the parameters (λ, C), we determined the parameters $\beta_0, \ldots, \beta_n$ and $u_1, \ldots, u_m$.

We ran 50 times random 5-fold cross-validation for standard SVM, pcaSVM,(K+L)SVM and ccSVM, and reported their corresponding average AUCs and standard errors. We also performed a *t*-test between the 50 AUCs of competing partners and 50 AUCs of ccSVM, and recorded the *P*-values. As ccLR and BLogReg did not work well, we performed logistic regression with maximum likelihood estimation in 10 times 5-fold cross-validation and reported the averaged AUC.

Results: Table 1 shows the prediction results for random cross-validation. Regarding the AUC values, ccSVMs with side information of ethnicity and age are slightly better than the other SVM approaches, while ccSVM with gender as side information works similar with the other SVMs. The logistic regression approach is not able to classify the data correctly regardless of which side information is used.

Weight vector analysis: we examined the weight vector of the ccSVM to get a further understanding for its improved performance. Specifically, we trained on four ethnic groups and then used it to predict on a fifth. In Figure 2, we plot the averaged absolute correlation coefficients between membership in one ethnicity (African) and the expression levels of the 10 000 top ranked genes. We can observe that the ccSVM assigns the largest weights to genes that do not correlate with the confounder, while the standard SVM is unaware of the confounder and puts large weight on the features that correlate with the confounding variable.

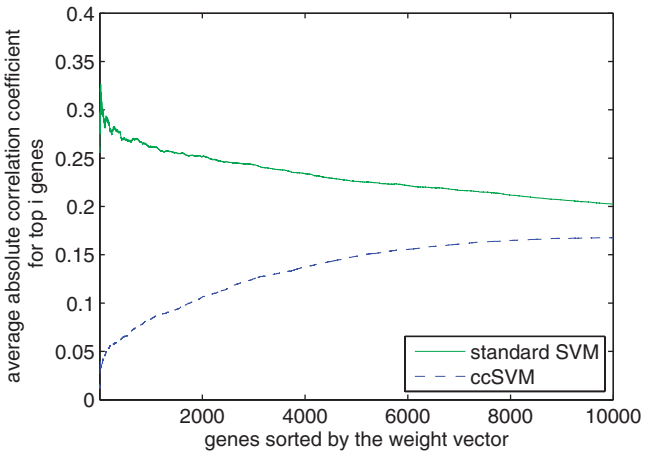

Fig. 2. Gene expression levels are sorted according to the weight vector of ccSVM (blue dashed line) and according to the weight vector of standard SVM (green line). The correlation coefficient between each gene expression level and ethnic origin (African) is calculated. The averaged absolute correlation coefficient of the top i genes is plotted for gene i.

3.5 Phenotype prediction with population structure correction

In this experiment, we assessed the performance of the ccSVM in comparison to the standard SVM, (K+L)SVM, pcaSVM and ccLR on phenotype prediction from SNP data in *Arabidopsis thaliana*.

Data : we used data from the genome-wide association study in *A.thaliana* conducted by Atwell *et al.* (2010). The dataset consists of $m = 177$ samples and $n = 216\,130$ single nucleotide polymorphisms (SNPs). An SNP is a fixed position in the genome which exists in two different variations between individuals. We examined five binary phenotypes, namely the presence and absence of chlorosis at 22°C (PID:169), of anthocyanin at 16°C (PID:171) and at 22°C (PID:172) and of leaf roll at 10°C (PID:176) and at 22°C (PID:178).

We used population structure as side information and computed a side information kernel matrix $\mathbf{L} \in \mathbb{R}^{m \times m}$. Population structure is defined by the different allele frequencies between subpopulations. If the phenotype prevalence also differs between these subpopulations, it can lead to spurious associations between the phenotype and SNPs that are associated with a subpopulation in which one phenotype is prevalent (Marchini *et al.*, 2004). Each entry $\mathbf{L}_{ij}$ is here defined as the number of common SNPs between sample i and sample j.

Experimental setting: for this experiment, we used the same experimental setting as described in Subsection 3.4.

Results: prediction results are reported in Table 2. For all the phenotypes except leaf roll at 22°C (PID:178), ccSVM yields better AUC values than the state-of-the-art competitors. Regarding the *P*-values, we see that the improvement of our method against standard SVM, pcaSVM and (K+L)SVM is significant for the phenotypes chlorosis at 22°C (PID:169), anthocyanin at 16°C (PID:171), anthocyanin at 22°C (PID:172) and leaf roll at 10°C (PID:176).

Weight vector analysis: in Figure 3, we compare the normalized weight vectors obtained by ccSVM and standard SVM for two phenotypes by looking at one representative each. We first pick up the top 100 features selected by standard SVM, and then see how the ccSVM corrects the weights of these features. When the ccSVM curve is lower than the standard SVM curve (negative peak),

Table 2. AUC and *P*-values for ccSVM, standard SVM, pcaSVM, (K+L)SVM and ccLR for the five different *Arabidopsis* phenotypes

PID	Phenotype	AUC_{ccSVM}	AUC_{SVM}	p_{SVM}	AUC_{pcaSVM}	p_{pcaSVM}	$AUC_{(K+L)SVM}$	$p_{(K+L)SVM}$	$AUC_{ccLR(ML)}$	$AUC_{ccLR(BR)}$
169	Chlorosis at 22°C	0.658 ± 0.004	0.623± 0.004	8.3e-10	0.625± 0.004	6.4e-09	0.574± 0.004	2.2e-28	0.632± 0.006	0.523± 0.004
171	Anthocyanin at 16°C	0.590 ± 0.005	0.568± 0.005	1.2e-03	0.570± 0.004	2.1e-03	0.560± 0.004	2.1e-06	0.571± 0.012	0.571± 0.003
172	Anthocyanin at 22°C	0.628 ± 0.003	0.610± 0.003	2.7e-05	0.610± 0.004	1.2e-04	0.576± 0.003	1.8e-21	0.613± 0.004	0.552± 0.004
176	Leaf Roll at 10°C	0.720 ± 0.002	0.695 ± 0.003	2.6e-09	0.697 ± 0.003	3.8e-08	0.653 ± 0.003	3.3e-31	0.691 ± 0.010	0.550± 0.003
178	Leaf Roll at 22°C	0.587 ± 0.007	0.575± 0.006	1.8e-01	0.591 ± 0.005	6.0e-01	0.580 ± 0.006	4.1e-01	0.573± 0.006	0.476± 0.008

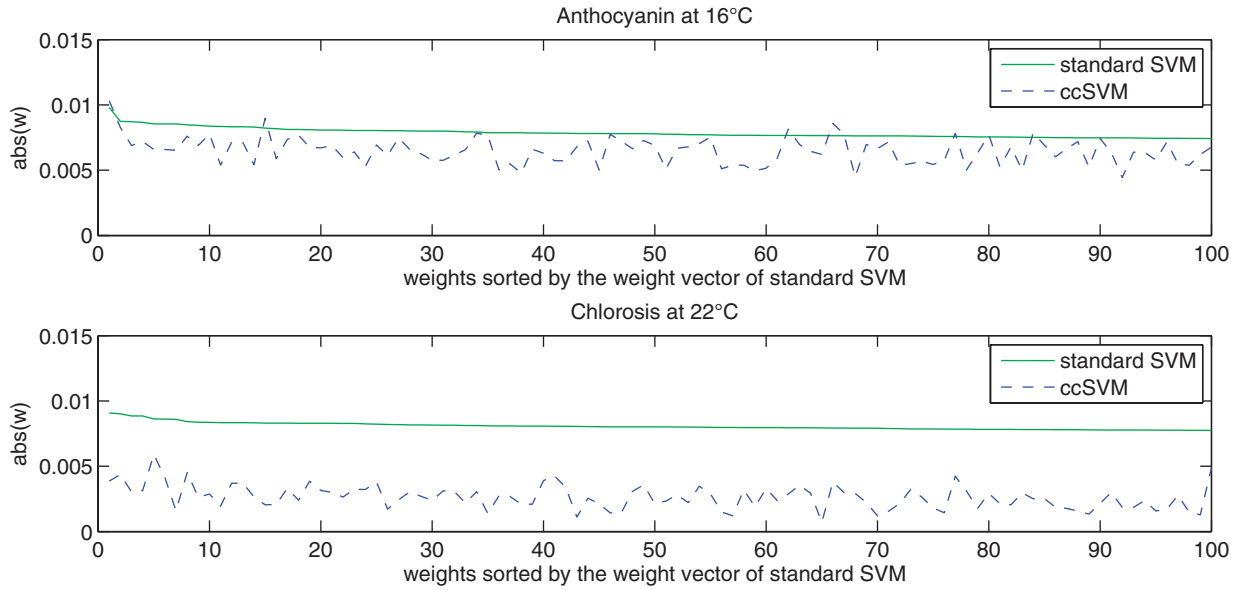

Fig. 3. SNPs are sorted by their absolute weight of the standard SVM. The green line shows the weights of the standard SVM, the blue dashed line shows the weights of ccSVM. Both weight vectors are normalized. The *Arabidopsis* phenotypes are shown in the following order (from top to bottom): anthocyanin at 16°C (PID:171, $\lambda = 10^{-2}$), chlorosis at 22°C (PID:169, $\lambda = 10^{8}$).

it means that the corresponding SNPs are likely to be correlated with the confounder and ccSVM weights them down for classification. The SNPs whose weights are scaled up (positive peaks) are less correlated with the confounding side information.

We can see from the figure that both parameter λ and the number of negative peaks increases from the top to the bottom. This implies the confounding information increases from top to bottom. For the phenotype anthocyanin at 16°C (PID:171), the top figure shows that there are almost no large negative peaks in the ccSVM curve. This implies there are few spurious associations for the ccSVM to correct. For the phenotype chlorosis at 22°C (PID:169), we can see that ccSVM scales all SNPs down which the standard SVM assigns large weights to. It is likely that they are all correlated with the confounding variable.

Functional investigation: we did further analysis for the phenotype chlorosis at 22°C (PID:169). In order to do this, we used the complete dataset as training set and determined λ and *C* via cross-validation as described in Section 3.1.

First, we selected the top 500 SNPs from the weight vector of ccSVM; these are the SNPs that correspond to the 500 largest absolute entries in the weight vector. After normalizing these entries in both weight vectors, we selected all SNPs which were upscaled by the ccSVM by at least a factor of two. For these 217 SNPs, we searched for nearby genes (±15 kb) which are known to be associated with chlorosis by using a candidate gene list from Atwell *et al.* (2010).

Table 3. Summary of ccSVM results for the presence or absence of chlorosis at 22°C (PID:169)

Rank	Chrom	Pos	Gene	Gene ID	dist(Gene)
109	1	22050068			6365
110	1	22056970	PDR8/PEN3	AT1G59870	13267
111	1	22057369			13666
208	4	949836	MOS6	AT4G02150	775
224	1	20910400	AHG2	AT1G55870	8313
267	1	20737467	CPN60B	AT1G55490	14605
363	5	25795239	AT5G64510	AT5G64510	6391
464	5	25795805			5825
489	5	12625100	CDR1	AT5G33340	11918

In the table, Chrom,Pos and dist(Gene) represent chromosome, position and the distance from the SNP to the specified gene, respectively.

The results are shown in Table 3. *pen-3-1* mutants show a chlorosis response after being attacked by *Erysiphe cichoracearum*. It is assumed that the gene *PEN3* contributes to defense at the cell wall and intracellularly (Stein *et al.*, 2006). The *mos6* mutants suppress *snc1* resistance and hence exhibit enhanced disease susceptibility to virulent pathogens (Palma *et al.*, 2005). The gene *CDR1* is known to be involved in disease resistance signaling (Xia *et al.*, 2004), and *ahg2-1* mutants have an elevated resistance to bacterial pathogens (Nishimura *et al.*, 2009).

In total, 9 of the 217 upscaled SNPs are close to candidate genes. Out of 216 130 genome-wide SNPs, 3959 are in close proximity to candidate genes. Hence, SNPs near candidate genes are significantly enriched among the SNPs upscaled by the ccSVM ($P=0.020$, $\alpha=0.05$, Binomial $n=217, p=\frac{3959}{216130}$).

4 DISCUSSION

In this article, we have defined the ccSVM, an SVM with correction for confounding side information. In our experiments, it outperforms several state-of-the-art classifiers with confounder correcting schemes for disease diagnosis in humans and for phenotype prediction in *A.thaliana*.

Our work extends the advantages of SVMs in data integration: while there is lot of work on SVMs for optimally combining several informative sources of data for a joint prediction (Lanckriet *et al.*, 2004), there was no approach for correcting SVMs for observed confounding factors so far. The ccSVM closes this gap. This is of particular importance for bioinformatics, as side information on confounders is abundant in most classification tasks on biological data.

It remains to be discovered if SVMs can be corrected for hidden, unobserved confounders as well, as these tend to frequently occur in gene expression phenotypes. Correcting for these hidden confounders may be one way to further improve the accuracy of our predictions.

On the biological level, our work will focus on applications of the ccSVM to binary phenotype prediction in plant genetics and in personalized medicine. The latter includes improved disease diagnosis, prognosis and therapy outcome prediction for human patients. One challenge we will tackle here is how to optimally account for several confounding factors, that is learning their weights relative to each other to further improve phenotype prediction.

ACKNOWLEDGEMENTS

The authors would like to thank Richard Neher for fruitful discussions.

Conflict of Interest: none declared.

REFERENCES

Atwell,S. *et al*. (2010) Genome-wide association study of 107 phenotypes in Arabidopsis thaliana inbred lines. *Nature*, **465**, 627–631.

Berry,M. *et al*. (2010) An interferon-inducible neutrophil-driven blood transcriptional signature in human tuberculosis. *Nature*, **466**, 973–977.

Borgwardt,K.M. *et al*. (2005) Protein function prediction via graph kernels. *Bioinformatics*, **21** (Suppl. 1), i47–i56.

Bullinger,L. *et al*. (2004) Use of gene-expression profiling to identify prognostic subclasses in adult acute myeloid leukemia. *N. Engl. J. Med.*, **350**, 1605–1616.

Cawley,G. and Talbot,N. (2006) Gene selection in cancer classification using sparse logistic regression with bayesian regularization. *Bioinformatics*, **22**, 2348–2355.

Chang,C.-C. and Lin,C.-J. (2001) *LIBSVM: a library for support vector machines*. Available at http://www.csie.ntu.edu.tw/∼cjlin/libsvm (last accessed May 3, 2011).

Cortes,C. and Vapnik,V. (1995) Support vector networks. *Machine Learn.*, **20**, 273–297.

Gretton,A. *et al*. (2005) Measuring statistical dependence with Hilbert-Schmidt norms. In *Proceedings Algorithmic Learning Theory*, Springer, pp. 63–77.

Holsboer,F. (2008) How can we realize the promise of personalized antidepressant medicines? *Nat. Rev. Neurosci.*, **9**, 638–646.

Kang,H.M. *et al*. (2010) Variance component model to account for sample structure in genome-wide association studies. *Nat. Genet.*, **42**, 348–354.

Kuhn,H.W. and Tucker,A.W. (1951) Nonlinear programming. In *Proceedings of the 2nd Berkeley Symposium on Mathematical Statistics and Probabilistics*, University of California Press, Berkeley, pp. 481–492.

Lanckriet,G. *et al*. (2004) A statistical framework for genomic data fusion. *Bioinformatics*, **20**, 2626–2635.

Leslie,C. *et al*. (2002) The spectrum kernel: A string kernel for SVM protein classification. *Pac. Symp. Biocomput.*, 564–575.

Marchini,J. *et al* (2004) The effects of human population structure on large genetic association studies. *Nat. Genet.*, **36**, 512–517.

Meinert,C. and Tonascia,S. (1986) Clinical trials: design, conduct, and analysis. *Monographs in Epidemiology and Biostatistics*. Oxford University Press, USA.

Nishimura,N. *et al*. (2009) ABA hypersensitive germination2-1 causes the activation of both abscisic acid and salicylic acid responses in Arabidopsis. *Plant Cell Physiol.*, **50**, 2112–2122.

Noble,W.S. (2006) What is a support vector machine? *Nat. Biotech.*, **24**, 1565–1567.

Palma,K. *et al*. (2005) An importin alpha homolog, MOS6, plays an important role in plant innate immunity. *Curr. Biol.*, **15**, 1129–1135.

Price,A.L. *et al*. (2006) Principal components analysis corrects for stratification in genome-wide association studies. *Nat. Genet.*, **38**, 904–909.

Price,A.L. *et al*. (2010) New approaches to population stratification in genome-wide association studies. *Nat. Rev. Genet.*, **11**, 459–463.

Schölkopf,B. and Smola,A.J. (2002) *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Cambridge, MA, London, England.

Schölkopf,B. *et al*. (2004) *Kernel Methods in Computational Biology*. MIT Press, Cambridge, MA.

Stein,M. *et al*. (2006) Arabidopsis PEN3/PDR8, an ATP binding cassette transporter, contributes to nonhost resistance to inappropriate pathogens that enter by direct penetration. *Plant Cell*, **18**, 731–746.

Valk,P.J. *et al*. (2004) Prognostically useful gene-expression profiles in acute myeloid leukemia. *N. Engl. J. Med.*, **350**, 1617–1628.

Vapnik,V. and Chervonenkis,A. (1974) *Theory of Pattern Recognition [in Russian]*. Nauka, Moscow. [German Translation: W. Wapnik & A. Tscherwonenkis (1979) *Theorie der Zeichenerkennung*. Akademie, Berlin, 1979].

Warnat,P. *et al*. (2005) Cross-platform analysis of cancer microarray data improves gene expression based classification of phenotypes. *BMC Bioinformatics*, **6**, 265.

Xia,Y. (2004) An extracellular aspartic protease functions in Arabidopsis disease resistance signaling. *EMBO J.*, **23**, 980–988.