Noname manuscript No.
(will be inserted by the editor)

Implicitly Constrained Semi-Supervised Least Squares Classification

Jesse H. Krijthe · Marco Loog

Received: date / Accepted: date

Abstract We introduce a novel semi-supervised version of the least squares classifier. In implicitly constrained least squares (ICLS), we minimize the squared loss on the labeled data among the set of parameters implied by all possible labelings of the unlabeled data. Unlike previous discriminative semi-supervised methods, our approach does not introduce explicit additional assumptions into the objective function, but leverages implicit assumptions already present in the choice of the supervised least squares classifier. We show this classifier can be formulated as a quadratic programming problem and its solution can be found using a simple gradient descent procedure. In a specific 1-dimensional case without intercept, we prove that this method can never lead to worse performance than supervised least squares classification, while in the multidimensional case we prove improvement of the parameter estimates for an adapted version of the proposed procedure. Experimental results corroborate the theoretical results and indicate desirable properties over alternative semi-supervised least squares approaches.

Keywords Semi-supervised learning, Least Squares Classification, Constrained Learning

1 Introduction

We consider the problem of semi-supervised learning of binary classification functions. Like in the supervised paradigm, the goal in semi-supervised learning is to construct a classification rule that maps objects in some input space to a target outcome, such that future objects map to correct target outcomes as closely as possible. In the supervised paradigm this mapping is learned using a set of N_l training objects and their corresponding outputs. In the semi-supervised scenario we are given an additional and often large set of N_u unlabeled objects. The challenge of semi-supervised learning is to incorporate this additional information to improve the classification rule.

Leiden University Medical Center
· Pattern Recognition Laboratory, Delft University of Technology

In this work, we present a novel approach to semi-supervised learning for the least squares classifier. We will refer to this approach as implicitly constrained semi-supervised learning (ICLS). The goal is to leverage the implicit assumptions present in supervised least squares classification to construct a semi-supervised version. That is, we exploit constraints inherent in the choice of supervised classifier, whereas current state-of-the-art typically relies on imposing additional extraneous, and possibly incorrect, assumptions.

In least squares classification, classes are encoded as numerical outputs after which a linear regression model is applied (see also, Section 3). By placing a threshold on the output of this model, we can use the linear function to predict class labels. In a different neural network formulation, this classifier is also known as Adaline [1]. There are several reasons why this is a particularly interesting classifier to study. First of all, this is a discriminative classifier. Some have claimed semi-supervised learning without additional assumptions is impossible for discriminative classifiers [2]. Our results, like [3], show this may not strictly hold. Secondly, as we will show in section 3.2, it allows us to formulate our approach as a quadratic programming problem, that can be solved through a simple gradient descent with boundary constraints. Lastly, least squares classification is a useful and adaptable classification technique allowing for straightforward use of, for instance, regularization, sparsity penalties or kernelization [4–7]. Using these formulations, it has been shown to be competitive with state-of-the-art methods based on loss functions other than the squared loss [5] as well as computationally efficient on large datasets [8].

Ultimately, the goal of our research is to build a semi-supervised least squares classier that has the property that, at least in expectation, its performance is not worse than supervised least squares classification. While it may seem like an obvious requirement for any semi-supervised method, current semi-supervised methods do not have this property. In fact, performance can significantly degrade as more unlabeled data is added, as has been shown in [9]. Also, many semi-supervised learning procedures are formulated as non-convex objective functions which are hard to optimize. A more satisfactory state of affairs would therefore be computationally efficient methods that on average do not lead to worse classification performance than their supervised alternatives.

The proposed semi-supervised least squares classifier works by constraining the solution of the supervised least squares classifier based on the unlabeled data. These constraints follow from the choice of the supervised classifier and require only minimal assumptions. A limited though important result, which follows from this formulation, is that in the 1 dimensional case without intercept, this procedure *never* leads to a decrease in generalization performance. In our experiments, we indeed find that this new approach sports many of the properties we deem desirable in a semi-supervised learner, namely that on average performance increases as we increase the amount of unlabeled data and that we efficiently converge to a global optimum.

The main contributions of this paper are

– A novel convex formulation for robust semi-supervised learning under squared loss (Equation 5)

– An intuition through a proof of non-degradation for the 1-dimensional case (Theorem 1) and improvement in parameter estimation for an adapted procedure in the multivariate case (Theorem 2).
– An empirical evaluation of the properties proposed by these theories (Section 4)

The rest of this paper is organized as follows. Section 2 discusses related work on semi-supervised learning. Section 3 introduces our semi-supervised version of the least squares classifier. We then derive a quadratic programming formulation and present a simple way to solve this problem through bounded gradient descent. Section 4 contains a proof that this classifier works under some assumptions about the data. Section 5 presents an empirical evaluation of the proposed approach on benchmark datasets. The final sections discuss the results and conclude.

2 Related Work

Many diverse semi-supervised learning techniques have been proposed [10, 11]. While these have proven successful in particular applications, such as document classification [12], it has also been observed that these techniques may give worse performance than their supervised counterparts [13, 9]. In these cases, disregarding the unlabeled data would lead to better performance. Some [14, 15] have argued that *agnostic* semi-supervised learning, which [14] define as semi-supervised learning that is at least no worse than supervised learning, can be achieved by cross-validation on the limited labeled data. Agnostic semi-supervised learning follows if we only use semi-supervised methods when their estimated cross-validation error is significantly lower than those of the supervised alternatives. As the results of [14] indicate, this criterion may be too conservative: given the small amount of labeled data, a semi-supervised method will only be preferred if the difference in performance is very large. If the difference is less distinct, the supervised learner will always be preferred and we potentially ignore useful information from the unlabeled objects. Moreover, this cross-validation approach can be computationally demanding.

A simple approach to semi-supervised learning is offered by the self-learning procedure [16], also known as Yarowsky's algorithm [17, 18]. Taking any classifier, we first estimate its parameters on only the labeled data. Using this trained classifier we label the unlabeled points and add the most confident label predictions to the training set. The classifier parameters are re-estimated using these labeled objects to get a new classifier. One iteratively applies this procedure until the predicted labels of the unlabeled data no longer change.

One of the advantages of this procedure is that it can be applied to any supervised classifier. It has also shown practical success in some application domains, particularly document classification [12, 17]. Unfortunately, the process of self-training can also lead to severely decreased performance, compared to the supervised solution [13, 9]. One can imagine that once an object is incorrectly labeled and added to the training set, its incorrect label may be reinforced, leading the solution away from the optimum. Self-learning is closely related to the expectation maximization (EM) based approaches [18]. Indeed, expectation maximization suffers from the same issues as self-learning [11].

More recent work has focused on introducing useful assumptions about the unlabeled data that can help link information about the distribution of the features $P(X)$ to the posterior of the classes $P(Y|X)$. Commonly used assumptions are the smoothness assumption, objects that are close in the feature space likely share the same label; the cluster assumption, objects in the same cluster share a label; and the low density assumption enforcing that the decision boundary should be in a region of low density.

The low-density assumption is used in entropy regularization [19] as well as for support vector classification in the transductive support vector machine (TSVM) [20]. When used in the semi-supervised setting this is closely related to the formulation of S^3VM [21,22]. Like in entropy regularization, for the semi-supervised versions of SVM, an additional term is added to the objective function to push the decision boundary away from dense regions. The resulting objective function is non-convex, owing to the possible labelings that can be assigned to the unlabeled objects. Several approaches have been put forth to solve this difficult optimization problem, such as a convex concave procedure [23] and difference convex programming [22,24].

In the approaches presented above, a parameter controls the importance of the unlabeled points. When the parameter is correctly set, it is clear, as [15] claims, that TSVM is always better than supervised SVM. It is, however, non-trivial to choose this parameter, given that semi-supervised learning is most interesting in cases where we have limited labeled objects, making a choice using cross-validation very unstable. In practice, therefore, TSVM may also lead to worse performance than the supervised support vector machine. [25] tries to guard against this deterioration by proposing safe semi-supervised SVM (S4VM). In some way, this method tries to find the safest decision boundary among all low-density decision boundaries identified by S3VM. While the approach is successful in protecting against deterioration when compared to supervised SVM, the price to pay is smaller performance increases on many datasets as well as significantly higher computational cost.

[26,27] attempt to guard against the possibility of deterioration in performance by not introducing additional assumptions, but instead leveraging implicit assumptions already present in supervised classifiers. These assumptions link parameters estimates that depend on labeled data to parameter estimates that rely on all data. By exploiting these links, semi-supervised versions of the nearest mean classifier and the linear discriminant are derived. Because these links are unique to each classifier, the approach does not generalize directly to other classifiers. The method presented here is similar in spirit, but unlike [26,27], no explicit equations have to be formulated to link parameter estimates using only labeled data to parameter estimates based on all data. This makes the current approach more flexible.

Little work has been done on applying semi-supervised learning to the least squares classifier specifically. For least squares regression [28] studied the value of knowing $\mathbb{E}[\mathbf{X}'\mathbf{X}]$, where $\mathbf{X}$ is the $N_l \times m$ design matrix containing the feature values for each observation. If we assume the number of unlabeled data points is large, this is similar to the semi-supervised situation. It is shown that if the size of the parameters is small compared to the noise, the variance of a procedure that plugs in $\mathbb{E}[\mathbf{X}'\mathbf{X}]$ as the estimate of $\mathbf{X}'\mathbf{X}$ has a lower variance than supervised least squares regression. As the size of the parameters increases, this effect reverses. In

fact, the paper demonstrates that in this semi-supervised setting no best linear unbiased estimator for the regression coefficients exists. In Section 5, we compare our approach to using this plug-in estimate by substituting the matrix $\mathbf{X}'\mathbf{X}$ by a version based on both labeled and unlabeled data. A similar plug-in procedure has been used by [29] for the dimensionality reduction technique that often is referred to as linear discriminant analysis. Here the (normalized) total scatter matrix, which plays a similar role to the $\mathbf{X}'\mathbf{X}$ matrix in least squares regression is exchanged with the more accurate estimate of the total scatter based on both labeled and unlabeled data.

3 Implicitly Constrained Least Squares Classification

Given a limited set of N_l labeled objects and a potentially large set N_u of unlabeled data, the goal of implicitly constrained least squares classification is to use the latter to improve the solution of the least squares classifier trained on just the labeled data. We start with a sketch of this approach, before discussing the details.

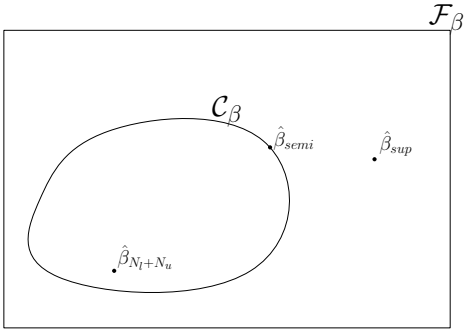

Fig. 1 A visual representation of implicitly constrained semi-supervised learning. $\mathcal{F}_\beta$ is the space of all linear models. $\hat{\beta}_{sup}$ denotes the solution given only a small amount of labeled data. $\mathcal{C}_\beta$ is the subset of the space which contains all the solutions we get when applying all possible labelings to the unlabeled data. $\hat{\beta}_{semi}$ is a projection of $\hat{\beta}_{sup}$ onto $\mathcal{C}_\beta$. $\hat{\beta}_{N_l+N_u}$ is the supervised solution if we would have the labels for all the objects.

Given the supervised least squares classifier, consider the hypothesis space of all possible parameter vectors, which we will denote as $\mathcal{F}_\beta$, see Figure 1. Given a set of labeled objects, we can determine the supervised parameter vector $\hat{\beta}_{sup}$. Suppose we also have a potentially large number N_u of unlabeled objects. Assume that these object have a label, it is merely unknown to us. If these labels were to be revealed, it is clear how the additional objects can improve classification performance: we estimate the least squares classifier using all the data to obtain the parameter vector $\hat{\beta}_{N_l+N_u}$. Since this estimate is based on more objects, we expect the parameter estimate to be better. These real labels are unknown, but we still can consider all possible labelings of unlabeled objects, and estimate corresponding parameters based on these imputed labelings. In this way, we get a set of possible parameters for our classifier, which form the set denoted by $\mathcal{C}_\beta$, which is a subset of all possible solutions $\mathcal{F}_\beta$. Clearly one of these labelings corresponds to the real,

but unknown, labeling, so one of the parameter estimates in this set corresponds to the solution we would obtain using all the correct labels of both the labeled and unlabeled objects. Because these are the only possible classifiers when the true labels would be revealed, we propose to look within this set $\mathcal{C}_\beta$ for an improved semi-supervised solution.

Two issues then remain: how do we choose the best parameters from this set and how do we find these without having to enumerate all possible labelings?

Looking at the first problem, we reiterate that the goal of semi-supervised learning is to find a good classification rule and, therefore, still the obvious way to evaluate this rule is by the loss on the labeled training points. In other words, we choose the classifier from the parameter set that minimizes the loss on the labeled points. Note this approach is rather different from other approaches to semi-supervised learning where the loss is adapted by including a term that depends on the unlabeled data points. In our formulation, the loss function is still the regular, supervised loss of our classification procedure. We can the interpret the minimization of this loss under the constraint that its solution needs to be in $\mathcal{C}_\beta$ as a projection of $\hat{\beta}_{sup}$ onto the subset $\mathcal{C}_\beta$. We will denote this solution by $\hat{\beta}_{semi}$.

As for the second issue, after relaxing the constraint that we need hard labels for the data points, we will derive the gradient of the loss on the labeled training points with respect to the imputed labels of the unlabeled objects. This will allow us to find the optimal labeling through a simple gradient descent procedure without having to go through all possible labelings of the unlabeled data.

3.1 Multivariate Least Squares Classification

Least squares classification [4, 5] is the direct application of well-known ordinary least squares regression to a classification problem. In other words, a linear model is assumed and the parameters are minimized under squared loss. Let $\mathbf{X}$ be an $N_l \times (m+1)$ design matrix with N_l rows containing vectors of length equal to the number of features m plus one for the intercept. Vector $\mathbf{y}$ denotes an $N_l \times 1$ vector of class labels. Without loss of generality one class is encoded as 0 and the other by 1. The multivariate version of the empirical risk function for least squares regression is given by

$$\hat{R}(\beta) = \frac{1}{n} \left\| \mathbf{X}\boldsymbol{\beta} - \mathbf{y} \right\|_2^2 \tag{1}$$

The well known closed-form solution for this problem is found by setting derivative with respect to $\boldsymbol{\beta}$ equal to $\mathbf{0}$ and solving for $\boldsymbol{\beta}$, giving:

$$\hat{\boldsymbol{\beta}} = \left(\mathbf{X}^T \mathbf{X} \right)^{-1} \mathbf{X}^T \mathbf{y} \tag{2}$$

In case $\mathbf{X}^T\mathbf{X}$ is not invertible (for instance when $n < (m+1)$), a pseudo-inverse is applied. As we will see, the convexity and subsequent closed form solution to this problem will enable us to formulate our semi-supervised learning approach in terms of a standard quadratic programming problem..

3.2 Implicitly Constrained Least Squares Classification

In the semi-supervised setting, apart from a design matrix $\mathbf{X}$ and target vector $\mathbf{y}$, an additional set of measurements $\mathbf{X}_u$ of size $N_u \times (m+1)$ *without* a corresponding target vector $\mathbf{y}_u$ is given. In what follows, $\mathbf{X}_e = \begin{bmatrix}\mathbf{X}^T & \mathbf{X}_u^T\end{bmatrix}^T$ denotes the extended design matrix which is simply the concatenation of the design matrices of the labeled and unlabeled objects.

In the implicitly constrained approach, we propose that a sensible solution to incorporate this additional information is to search within the set of classifiers that can be obtained by all possible labelings $\mathbf{y}_u$, for the one classifier that minimizes the *supervised* empirical risk function (1). This set, $\mathcal{C}_{\boldsymbol{\beta}}$, is formed by the $\boldsymbol{\beta}$'s that would follow from training supervised classifiers on all (labeled and unlabeled) objects going through all possible soft labelings for the unlabeled samples, i.e., using all $\mathbf{y}_u \in [0,1]^{N_u}$. Since these supervised solutions have a closed form, this can be written as:

$$\mathcal{C}_{\boldsymbol{\beta}} := \left\{ \boldsymbol{\beta} = \left(\mathbf{X}_e^T \mathbf{X}_e\right)^{-1} \mathbf{X}_e^T \begin{bmatrix}\mathbf{y} \\ \mathbf{y}_u\end{bmatrix} : \mathbf{y}_u \in [0,1]^{N_u} \right\} \tag{3}$$

Note that soft labelings are allowed. This is both a relaxation for computational reasons as well as a strategy to deal with label uncertainty. We can interpret these fractions as "responsibilities", a type of class posterior for the unlabeled objects.

This constrained region $\mathcal{C}_{\boldsymbol{\beta}}$, combined with the supervised loss that we want to optimize (1), gives the following definition for implicitly constrained semi-supervised least squares classification:

$$\begin{aligned} \underset{\boldsymbol{\beta}\in\mathbb{R}^{m+1}}{\operatorname{argmin}} \quad & \frac{1}{n}||\mathbf{X}\boldsymbol{\beta} - \mathbf{y}||^2 \\ \text{subject to} \quad & \boldsymbol{\beta} \in \mathcal{C}_{\boldsymbol{\beta}} \end{aligned} \tag{4}$$

Since $\boldsymbol{\beta}$ is fixed for a particular choice of $\mathbf{y}_u$ and has a closed form solution (2), we can rewrite the minimization problem in terms of $\mathbf{y}_u$ instead of $\boldsymbol{\beta}$:

$$\begin{aligned} \underset{\mathbf{y}_u}{\operatorname{argmin}} \quad & \frac{1}{n}\left\| X\left(\mathbf{X}_e^T\mathbf{X}_e\right)^{-1}\mathbf{X}_e^T\begin{bmatrix}\mathbf{y}\\ \mathbf{y}_u\end{bmatrix} - \mathbf{y}\right\|_2^2 \\ \text{subject to} \quad & \mathbf{y}_u \in [0,1]^{N_u} \end{aligned} \tag{5}$$

Solving this optimization problem provides an optimal $\mathbf{y}_u$. The corresponding solution for $\boldsymbol{\beta}$ then follows from equation (2) by using this imputed labeling as the labels for the unlabeled data. The problem defined in (5), is a standard quadratic programming problem of the form:

$$\begin{aligned} &\min_{\mathbf{y}_u} \frac{1}{2}\mathbf{y}_u^T\mathbf{Q}\mathbf{y}_u + \mathbf{c}^T\mathbf{y}_u \\ &\text{subject to:} \quad \begin{bmatrix}\mathbf{I}_{N_u} \\ -\mathbf{I}_{N_u}\end{bmatrix}\mathbf{y}_u \leq \begin{bmatrix}\mathbf{1}_{N_u} \\ \mathbf{0}_{N_u}\end{bmatrix} \end{aligned} \tag{6}$$

where

$$\mathbf{Q} = \frac{1}{n}\mathbf{X}_u\left(\mathbf{X}_e^T\mathbf{X}_e\right)^{-1}\mathbf{X}^T\mathbf{X}\left(\mathbf{X}_e^T\mathbf{X}_e\right)^{-1}\mathbf{X}_u^T$$

and

$$\mathbf{c} = \frac{1}{n}\mathbf{X}_u\left(\mathbf{X}_e^T\mathbf{X}_e\right)^{-1}\mathbf{X}^T\mathbf{y} + \frac{1}{2n}\mathbf{X}_u\left(\mathbf{X}_e^T\mathbf{X}_e\right)^{-1}\mathbf{X}^T\mathbf{X}\left(\mathbf{X}_e^T\mathbf{X}_e\right)^{-1}\mathbf{X}^T\mathbf{y}.$$

Where $\mathbf{I}_{N_l}$ denotes the $N_u \times N_u$ identity matrix and $\mathbf{1}_{N_u}$ and $\mathbf{0}_{N_u}$ denote column vectors of respectively ones and zeros.

Since the matrix $\mathbf{Q}$ is a product of a matrix and its transpose, it is guaranteed to be positive semi-definite. The problem is typically not positive definite because there are different labelings that will lead to one and the same minimum objective.

The quadratic problem defined above can be solved using, for instance, an interior point method. We have found a gradient descent approach to be easier to implement. Ignoring the constraint $\mathbf{y}_u \in [0,1]^{N_u}$ in (5), taking the derivative to $\mathbf{y}_u$ and rearranging the terms we find:

$$\begin{aligned}\frac{\partial \mathbf{L}}{\partial \mathbf{y}_u} = \quad & \frac{2}{n}\mathbf{X}_u\left(\mathbf{X}_e^T\mathbf{X}_e\right)^{-1}\mathbf{X}^T\mathbf{X}\left(\mathbf{X}_e^T\mathbf{X}_e\right)^{-1}\mathbf{X}^T\mathbf{y} \\ & + \frac{2}{n}\mathbf{X}_u\left(\mathbf{X}_e^T\mathbf{X}_e\right)^{-1}\mathbf{X}^T\mathbf{X}\left(\mathbf{X}_e^T\mathbf{X}_e\right)^{-1}\mathbf{X}_u^T\mathbf{y}_u \\ & + \frac{2}{n}\mathbf{X}_u\left(\mathbf{X}_e^T\mathbf{X}_e\right)^{-1}\mathbf{X}^T\mathbf{y}\end{aligned} \tag{7}$$

Because of its convexity, this problem can be solved efficiently using a quasi-Newton approach that allows for the simple [0,1] box bounds, such as L-BFGS-B [30]. Finally, the optimal labeling $\mathbf{y}_u$ (as defined by the supervised loss function) gives us the semi-supervised estimate of $\boldsymbol{\beta}$.

4 Theoretical Results

4.1 Strong performance result in 1D

Consider the case where we have just one feature x, a limited set of labeled instances and assume we know the probability density function of this feature $f_X(x)$ exactly. This last assumption is similar to having unlimited unlabeled data and is also considered, for instance, in [3]. We consider a linear model with no intercept: $y = x\beta$ where y, without loss of generality, is set as 0 for one class and 1 for the other. For new data points, estimates $\hat{y}$ can be used to determine the predicted label of an object by using a threshold set at, for instance, 0.5.

The expected squared loss, or risk, for this model is given by:

$$R^*(\beta) = \sum_{y=\{0,1\}} \int_{-\infty}^{\infty} (x\beta - y)^2 f_{X,Y}(x,y)\mathrm{d}x \tag{8}$$

Where $f_{X,Y}$ is the joint density of X and Y. The Bayes optimal solution β^* is given by the β that minimizes this loss:

$$\beta^* = \operatorname*{argmin}_{\beta \in \mathbb{R}} R^*(\beta) \tag{9}$$

Setting the derivative with respect to β to 0 and rearranging we get:

$$\beta = \left(\int_{-\infty}^{\infty} x^2 f_X(x)\mathrm{d}x\right)^{-1} \sum_{y=\{0,1\}} \int_{-\infty}^{\infty} xy f_{X,Y}(x,y)\mathrm{d}x \tag{10}$$

$$= \left(\int_{-\infty}^{\infty} x^2 f_X(x)\mathrm{d}x\right)^{-1} \int_{-\infty}^{\infty} x f_X(x) \sum_{y=\{0,1\}} yP(y|x)\mathrm{d}x \tag{11}$$

$$= \left(\int_{-\infty}^{\infty} x^2 f_X(x)\mathrm{d}x\right)^{-1} \int_{-\infty}^{\infty} x f_X(x)\mathbb{E}[y|x]\mathrm{d}x \tag{12}$$

In this last equation, since the assume $f_X(x)$ as given, the only unknown is the function $\mathbb{E}[y|x]$, the expectation of the label y, given x. Now suppose we consider every possible labeling of the unlimited number of unlabeled objects including fractional labels, that is, every possible function where $\mathbb{E}[y|x] \in [0,1]$. Given this restriction on $\mathbb{E}[y|x]$, the latter integral becomes a re-weighted version of the expectation operation $\mathbb{E}[x]$. By changing the choice of $\mathbb{E}[y|x]$ one can vary the value of this integral, but it will always be bounded on an interval on $\mathbb{R}$. It follows that all possible β's also form an interval on $\mathbb{R}$, which is the constrained set $\mathcal{C}_{\boldsymbol{\beta}}$. The Bayes optimal solution has to be in this interval, since it corresponds to a particular but unknown labeling $\mathbb{E}[y|x]$.

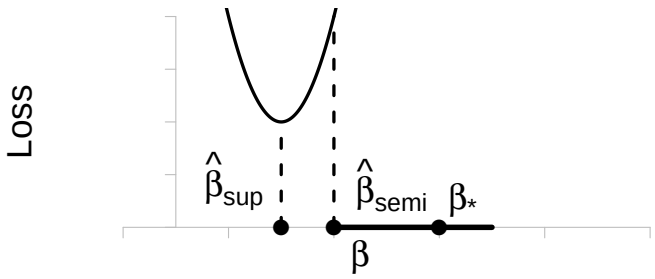

Fig. 2 An example where implicitly constrained optimization always improves performance. The supervised solution $\hat{\beta}_{sup}$ which minimizes the supervised loss (shown), is not part of the interval of allowed solutions. The solution that minimizes this supervised loss within the allowed interval is $\hat{\beta}_{semi}$. This solution is closer to the Bayes optimal solution β^{bayes} than the supervised solution $\hat{\beta}_{sup}$.

Using the set of labeled data, we can construct a supervised solution $\hat{\beta}_{sup}$ that minimizes the loss on the training set of N_l labeled objects, see Figure 2:

$$\hat{\beta}_{sup} = \underset{\beta\in\mathbb{R}}{\operatorname{argmin}} \sum_{i=1}^{N_l} (x_i\beta - y_i)^2 \tag{13}$$

Now, either this solution falls within the constrained region, $\hat{\beta}_{sup} \in \mathcal{C}_{\boldsymbol{\beta}}$ or not, $\hat{\beta}_{sup} \notin \mathcal{C}_{\boldsymbol{\beta}}$, with different consequences:

1. If $\hat{\beta}_{sup} \in \mathcal{C}_{\boldsymbol{\beta}}$ there is a labeling of the unlabeled points that gives us the same value for β. Therefore, the solution falls within the allowed region and there is no reason to update our estimate. Therefore $\hat{\beta}_{semi} = \beta_{sup}$.
2. Alternatively, if $\hat{\beta}_{sup} \notin \mathcal{C}_{\boldsymbol{\beta}}$, the solution is outside of the constrained region (as shown in Figure 2): there is no possible labeling of the unlabeled data that will give the same solution for β. We then update the β to be the β within the constrained region that minimizes the loss on the supervised training set. As can be seen from Figure 2, this will always be a point on the boundary of the interval. Note that $\hat{\beta}_{semi}$ is now closer to β^* than $\hat{\beta}_{sup}$. Since the true loss function $R^*(\beta)$ is convex and achieves its minimum in the Bayes optimal solution, the true loss of our semi-supervised solution will always be equal to or lower than the loss of the supervised solution.

Thus, the proposed update either improves the estimate of the parameter β or it does not change the supervised estimate. In no case will the semi-supervised solution be worse than the supervised solution, in terms of the expected squared loss. We summarize this result in the following theorem:

Theorem 1 *Given a linear model without intercept, $y = x\beta$, and $f_X(x)$ known, the estimate obtained through implicitly constrained least squares performs at least as good as the regular least squares solution: $R^*(\hat{\beta}_{semi}) \leq R^*(\hat{\beta}_{sup})$.*

In particular, if $f_{X,Y}$ is continuous and $f_{X,Y}(0,1) > 0$, then $\mathbb{E}[R^*(\hat{\beta}_{semi})] < \mathbb{E}[R^*(\hat{\beta}_{sup})]$.

The last part of this theorem gives a general condition when, in expectation, our semi-supervised approach will outperform the supervised learner. Because $\hat{\beta}_{semi}$ will never be worse than $\hat{\beta}_{sup}$, to prove this we only need to show that for some observation of a labeled point with positive $f_{X,Y}(x,y) > 0$, the estimated $\hat{\beta}^{SL}$ is outside of the interval $\mathcal{C}_{\boldsymbol{\beta}}$, in which case $R^*(\hat{\beta}_{semi}) < R^*(\hat{\beta}_{sup})$.

If we observe an object labeled 1 with feature value x, the corresponding estimate $\hat{\beta}_{sup} = \frac{1}{x}$. Since the improvement in loss will only result if this estimate is not in the constrained region, we need to show that:

$$P(\tfrac{1}{x} \notin \mathcal{C}_{\boldsymbol{\beta}}, y = 1) > 0 \tag{14}$$

To do this, consider the bounds of the interval $\mathcal{C}_{\boldsymbol{\beta}}$. These most extreme values are obtained whenever all negative values of x are assigned label 0 while the positive x get labels 1, or the other way around. From (12) and writing $\mathbb{E}(X^2) = \int_{-\infty}^{\infty} x^2 f_X(x)\mathrm{d}x$ we find the interval is given by:

$$\mathcal{C}_{\boldsymbol{\beta}} = \left[\frac{\int_{-\infty}^{0} x f_X(x)\mathrm{d}x}{\mathbb{E}(X^2)}, \frac{\int_{0}^{\infty} x f_X(x)\mathrm{d}x}{\mathbb{E}(X^2)} \right] \tag{15}$$

Using this and rearranging (15), we get the following condition:

$$P\left(\frac{\mathbb{E}(X^2)}{\int_{-\infty}^{0} x f_X(x)\mathrm{d}x} < x < 0 \vee 0 < x < \frac{\mathbb{E}(X^2)}{\int_{0}^{\infty} x f_X(x)\mathrm{d}x}, y = 1 \right) > 0 \tag{16}$$

Since $f_{X,Y}$ is assumed to be continuous, $\mathbb{E}[X^2] > 0$ and the lower bound in this equation is always smaller than 0, while the upper bound is always larger than 0. Therefore, (15) holds whenever $f_{X,Y}(0,1) > 0$. The assumption of the continuity of $f_{X,Y}$ ensures that (16) holds. The property $f_{X,Y}(0,1) > 0$ is satisfied by many distributions of the data. The result, therefore, indicates, that improvement is not only possible, but will occur in many cases.

4.2 Euclidean Projection Estimator for Multivariate ICLS

In the multivariate case (with intercept), we will prove that the solution vector obtained through an adapted version of the implicitly constrained least squares classifier is always as close or closer to the real coefficients β than the supervised solution, when using Euclidean distance as a measure of closeness. While not directly equivalent to the goal of classification error, this different notion of statistical risk is commonly used in many areas of statistics [31]. We can use this risk as an alternative way to study in what sense the constrained parameter space may offer an improved estimator over the supervised estimator.

In the adapted ICLS procedure considered, we minimize a slightly different risk function than the squared loss on the labeled objects used in ICLS:

$$\hat{\boldsymbol{\beta}}_{adapted} = \underset{\boldsymbol{\beta} \in \mathcal{C}_{\boldsymbol{\beta}}}{\operatorname{argmin}} ||\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}_{sup}||^2 \tag{17}$$

This can be interpreted as finding the projection of the supervised solution $\hat{\boldsymbol{\beta}}_{sup}$ onto the constrained space $\mathcal{C}_{\boldsymbol{\beta}}$ defined in 3, where the projection is the minimum distance to this region, measured by the Euclidean distance. Note that the difference between this and the regular ICLS procedure is the measure used to calculate this distance. $\hat{\boldsymbol{\beta}}_{adapted}$ uses Euclidean distance, while $\hat{\boldsymbol{\beta}}_{semi}$ uses the loss on the labeled objects to determine the best element in $\mathcal{C}_{\boldsymbol{\beta}}$. The subset of the parameter space onto which we project is still the same. For this adapted ICLS we can prove the following:

Theorem 2 *Given a multivariate linear model, $y = \boldsymbol{X}\boldsymbol{\beta}$, and $f_X(\boldsymbol{x})$ known, we have for $\hat{\boldsymbol{\beta}}_{adapted}$ as defined in Equation* (17)*, that*

$$||\hat{\boldsymbol{\beta}}_{adapted} - \boldsymbol{\beta}|| \leq ||\hat{\boldsymbol{\beta}}_{sup} - \boldsymbol{\beta}||. \tag{18}$$

To prove this result, we first show that the constrained region $\mathcal{C}_{\boldsymbol{\beta}}$ is convex. Note that the constrained space in terms of $\mathbf{y}_u$, $\mathcal{C}_{\mathbf{y}_u} = [0,1]^{N_u}$ is convex. Now, for every pair $b_1, b_2 \in \mathcal{C}_{\boldsymbol{\beta}}$, their corresponding labelings $\mathbf{y}_{u_1}, \mathbf{y}_{u_2}$ and all $c \in [0,1]$, we have that:

$$\begin{aligned} & cb_1 + (1-c)b_2 \\ =& c\left(\mathbf{X}_e^T\mathbf{X}_e\right)^{-1}\mathbf{X}_e^T\begin{bmatrix}\mathbf{y}\\ \mathbf{y}_{u_1}\end{bmatrix} + (1-c)\left(\mathbf{X}_e^T\mathbf{X}_e\right)^{-1}\mathbf{X}_e^T\begin{bmatrix}\mathbf{y}\\ \mathbf{y}_{u_2}\end{bmatrix} \\ =& \left(\mathbf{X}_e^T\mathbf{X}_e\right)^{-1}\mathbf{X}_e^T\begin{bmatrix}\mathbf{y}\\ c\mathbf{y}_{u_1} + (1-c)\mathbf{y}_{u_2}\end{bmatrix} \in \mathcal{C}_\beta \end{aligned} \tag{19}$$

Where the conclusion that this estimate is an element of $\mathcal{C}_\beta$ follows from the fact that the space of labelings $\mathcal{C}_{y_u} = [0,1]^{N_u}$ is convex.

We now note the following result which can be found in, for instance, Proposition 1.4.1ii in [32, p.17]: For any two elements a, b in a Hilbert space H we have that $||P_C(a) - P_C(b)|| \leq ||a - b||$, where P_C is the best approximation projector onto a closed convex subset $C \subset H$.

Now, take a to be $\hat{\boldsymbol{\beta}}_{\boldsymbol{sup}}$, and b to be $\boldsymbol{\beta}$, the optimal parameter vector. Since $\boldsymbol{\beta} \in \mathcal{C}_\beta$, its projection is $P_{\mathcal{C}_\beta}(\boldsymbol{\beta}) = \boldsymbol{\beta}$. The projection of $\hat{\boldsymbol{\beta}}_{sup}$ is $\hat{\boldsymbol{\beta}}_{adapted}$. By the theorem above, we have $||\hat{\boldsymbol{\beta}}_{adapted} - \boldsymbol{\beta}|| \leq ||\hat{\boldsymbol{\beta}}_{sup} - \boldsymbol{\beta}||$ which proves the result.

Note that in 1D the Euclidean projection of Equation (17) and the regular ICLS projection are the same, which can also be seen in Figure 2. In the multivariate case, the two projections are not the same, meaning a better estimator in terms of Euclidean distance does not necessarily imply a better estimator in terms of squared loss. While the projection of regular ICLS onto the subset is slightly different from the Euclidean projection, the result in Theorem 2 indicates some form of improvement is possible by projecting onto the subset $\mathcal{C}_{\boldsymbol{\beta}}$.

5 Empirical Results

The empirical properties of our proposed approach are evaluated in two ways. Firstly, we study the behavior of the error rates of our semi-supervised approach for increasing amounts of unlabeled data. The goal is to study whether the ICLS procedure exhibits the improvement in expectation that motivated this method. Secondly, we compare the classification performance of these approaches in a cross-validation setting. This setting is in line with how these procedures are used in practice and allows us to study whether the procedure offers expected improvements there as well.

Since we extended the least squares classifier to the semi-supervised setting, we compare how, for different sizes of the unlabeled sample, our semi-supervised least squares approach fares against supervised least squares classification (LS) without the constraints. For comparison we included two alternative semi-supervised strategies, namely self-learning applied to the least squares classifier (SLLS) and a procedure were the matrix $\mathbf{X}^T\mathbf{X}$ is replaced by an appropriately scaled matrix $\mathbf{X}_e^T\mathbf{X}_e$ similar to the estimator in [28]. We will refer to the latter as updated covariance least squares (UCLS) classification. We also included the performance of the least squares classifier if all unlabeled object were to be labeled (LSoracle). This serves as the unattainable upper bound on the performance of any semi-supervised learner.

A description of the datasets used for our experiments is given in Table 1. We use datasets from both the UCI repository [33] and six of the benchmark datasets proposed by [10]. While the benchmark datasets proposed in [10] are useful, in our experience, the results on these datasets are very homogeneous because of the similarity in the dimensionality and their low Bayes errors. The UCI datasets are more diverse both in terms of the number of objects and features as well as the nature of the underlying problems. Taken together, this collection allows us to investigate the properties of our approach for a wide range of problems.

All the code used to run the experiments is available from the authors' website.

Table 1 Description of the datasets used in the experiments

Name	# Objects	#Features	Source
Haberman	305	4	[33]
Ionosphere	351	33	[33]
Parkinsons	195	20	[33]
Pima	768	9	[33]
Sonar	208	61	[33]
SPECT	265	23	[33]
SPECTF	265	45	[33]
Transfusion	748	4	[33]
WDBC	568	30	[33]
Mammographic	960	4	[33]
Digit1	1500	242	[10]
USPS	1500	242	[10]
COIL2	1500	242	[10]
BCI	400	118	[10]
g241c	1500	242	[10]
g241n	1500	242	[10]

5.1 Comparison of Learning Curves

We study the behavior of the expected classification error of the ICLS procedure for different sizes for the unlabeled set. As we noted in the introduction, this statistic has two desired properties. First of all it should never be higher than the expected classification error of the supervised solution. Secondly, the expected classification error should not increase as we add more unlabeled data.

Experiments were conducted as follows. For each dataset, N_l labeled points were randomly chosen, where we make sure it contains at least 1 object from each of the two classes. With fewer than m samples, the least squares classifier is known to deteriorate in performance as more data is added, a behavior known as peaking [34,35]. Since this is not the topic of this work, we will only consider the situation in which the labeled design matrix is of full rank, which is ensured by setting $N_l = m+5$, the dimensionality and intercept of the dataset plus five observations. For all datasets we ensure a minimum of $N_l = 20$ labeled objects.

Next, we create unlabeled subsets of increasing size $N_u = [2, 4, 8, ..., 1024]$ by randomly selecting points from the original dataset without replacement. The classifiers are trained using these subsets and the classification performance is evaluated on the remaining objects. Since the test set decreases in size as the number of unlabeled objects increases, the standard error is slightly increases with the number of unlabeled objects.

This procedure of sampling labeled and unlabeled points is repeated 100 times. We report the mean classification error as well as the standard error of this mean. As can be seen from the tight confidence bands, this offers an accurate estimate of the expected classification error.

The results of these experiments are shown in Figures 3 and 4. Note that the error-axis in the figures does not extend to 0 in order to show the differences between the various methods more clearly. UCLS is left out of this analysis, since its performance was often on a very different scale than that of the other procedures.

We find that the ICLS procedure has monotonically decreasing error curves as the number of unlabeled samples increases. Unlike self-learning, there is no

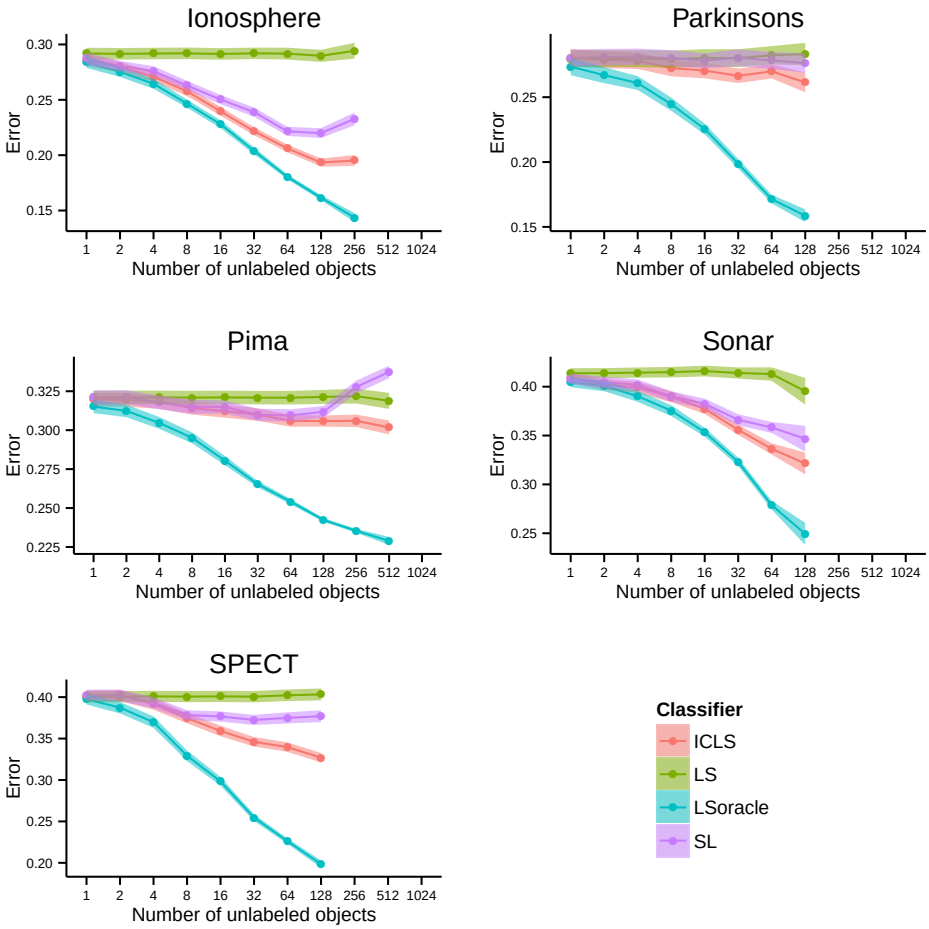

Fig. 3 Mean classification error for $N_l = \max(m + 5, 20)$ and 100 repeats. The error bounds are $+/-$ the standard error of the mean.

deterioration in performance. This is especially apparent, for instance, on the Pima dataset. For the datasets presented in Figure 4, it is interesting to see how the ICLS curves resemble a scaled version of the LS classifier were we assume all data points are labeled. Self-learning does not share this property.

5.2 Benchmark performance

We now consider the cross-validation setting for these datasets. This experiment is set up as follows. For each dataset, the objects are randomly divided into 10 folds. We iteratively go through the folds using 1 fold as validation set, and the other 9 as the training set. From this training set, we then randomly select $N_l = max(m + 5, 20)$ labeled objects, as in the previous experiment, and use the rest as unlabeled data. After predicting labels for the validation set for each fold, the classification error is then determined by comparing the predicted labels to the

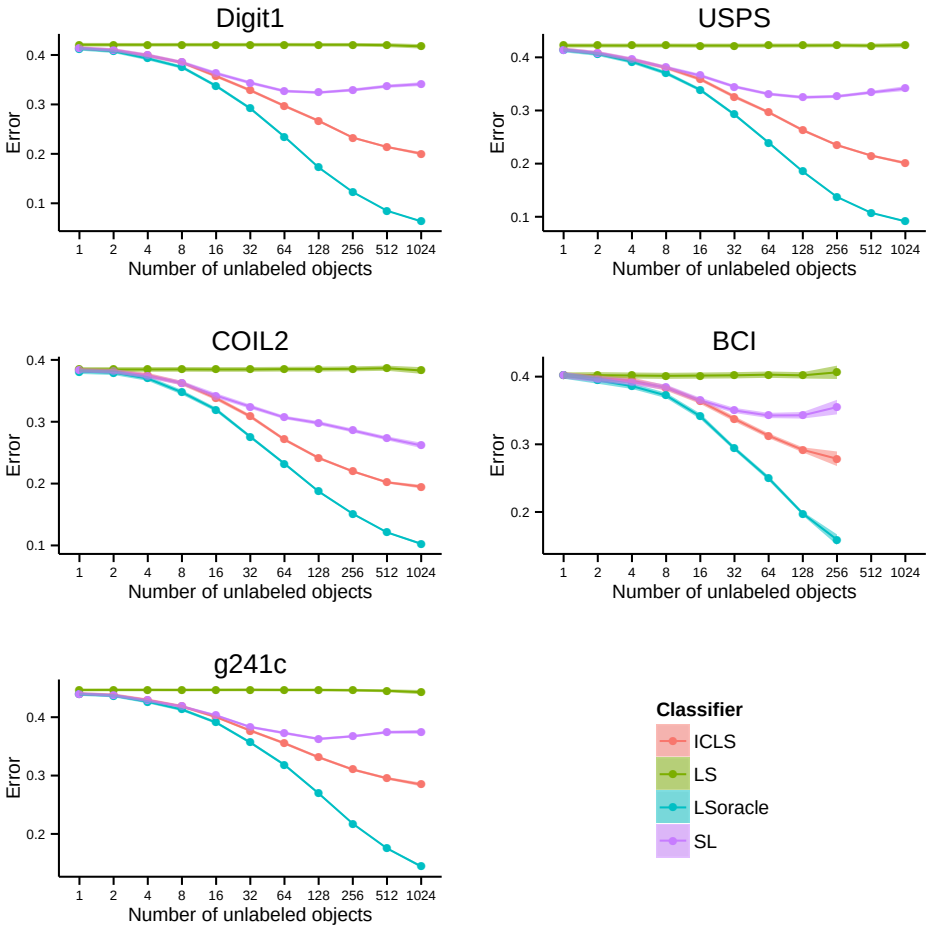

Fig. 4 Mean classification error for $N_l = \max(m + 5, 20)$ and 100 repeats. The error bounds are $+/-$ the standard error of the mean.

real labels. This is repeated 10 times, while randomly assigning objects to folds in each iteration.

The cross-validation procedure used here is slightly different from that prescribed in [10], to make it more closely relate to the cross-validation procedure that is usually employed. More specifically, this procedures ensures the validation sets are independent (non-overlapping), such that, after going over all the folds, each object is in the validation set only once. This is different from the procedure in [10], were the authors ensure the *labeled* sets or non-overlapping. We have not found a qualitative difference in the error rates, however, when using the procedure proposed in [10]. The advantage of the procedure employed here is that every object gets a single predicted label, allowing for the direct comparison of predictions of different classifiers.

The results shown in Table 2 tell a similar story to those in the previous section. Most importantly for the purposes of this paper, ICLS, in general, offers solutions that give at least no higher expected classification error than the su-

Table 2 Average 10-fold cross-validation error and standard deviation over 10 repeats. The classifiers that have been compared are supervised Least Squares (LS), Implicitly constrained least squares (ICLS), self-learned least squares (SLLS), updated covariance least squares (UCLS, see text) and for comparison a supervised least squares classifier that has access to all the labels (LSoracle). Indicated in **bold** is whether a semi-supervised classifier significantly outperform the supervised LS classifier, as measured using a t-test with a 0.05 significance level. Underlined indicates whether a semi-supervised classifier is (significantly) best among the three semi-supervised classifiers considered.

Dataset	LS	ICLS	SLLS	UCLS	LSoracle
Haberman	0.28 ± 0.02	0.28 ± 0.02	0.28 ± 0.01	0.39 ± 0.02	0.26 ± 0.01
Ionosphere	0.28 ± 0.02	$\mathbf{0.19 \pm 0.02}$	$\mathbf{0.24 \pm 0.01}$	0.35 ± 0.05	0.14 ± 0.01
Parkinsons	0.27 ± 0.02	$\mathbf{0.24 \pm 0.02}$	0.25 ± 0.04	0.40 ± 0.03	0.16 ± 0.01
Pima	0.32 ± 0.02	0.30 ± 0.01	0.35 ± 0.01	0.40 ± 0.01	0.23 ± 0.00
Sonar	0.44 ± 0.02	$\mathbf{0.35 \pm 0.03}$	$\mathbf{0.39 \pm 0.02}$	0.43 ± 0.03	0.25 ± 0.01
SPECT	0.42 ± 0.05	$\mathbf{0.33 \pm 0.03}$	0.42 ± 0.03	0.45 ± 0.04	0.18 ± 0.01
SPECTF	0.44 ± 0.03	$\mathbf{0.37 \pm 0.03}$	$\mathbf{0.39 \pm 0.02}$	0.46 ± 0.02	0.23 ± 0.01
Transfusion	0.26 ± 0.01	0.25 ± 0.01	0.28 ± 0.03	0.34 ± 0.02	0.23 ± 0.00
WDBC	0.10 ± 0.02	$\mathbf{0.08 \pm 0.01}$	0.12 ± 0.02	0.29 ± 0.02	0.05 ± 0.00
Mammographic	0.28 ± 0.02	0.28 ± 0.02	0.28 ± 0.02	0.41 ± 0.04	0.20 ± 0.00
Digit1	0.42 ± 0.02	$\mathbf{0.20 \pm 0.01}$	$\mathbf{0.34 \pm 0.02}$	$\mathbf{0.39 \pm 0.02}$	0.06 ± 0.00
USPS	0.42 ± 0.01	$\mathbf{0.20 \pm 0.01}$	$\mathbf{0.34 \pm 0.02}$	$\mathbf{0.38 \pm 0.02}$	0.09 ± 0.00
COIL2	0.38 ± 0.01	$\mathbf{0.20 \pm 0.01}$	$\mathbf{0.26 \pm 0.01}$	0.40 ± 0.01	0.10 ± 0.00
BCI	0.41 ± 0.03	$\mathbf{0.28 \pm 0.03}$	$\mathbf{0.36 \pm 0.04}$	0.41 ± 0.02	0.16 ± 0.02
g241c	0.45 ± 0.01	$\mathbf{0.28 \pm 0.01}$	$\mathbf{0.39 \pm 0.01}$	$\mathbf{0.42 \pm 0.01}$	0.14 ± 0.00
g241n	0.45 ± 0.02	$\mathbf{0.28 \pm 0.01}$	$\mathbf{0.39 \pm 0.01}$	$\mathbf{0.43 \pm 0.02}$	0.13 ± 0.00

pervised procedure. Exceptions are the Transfusion dataset were both ICLS and the self-learning approach show (non statistically significant) deterioration in performance, and Pima where ICLS shows non-statistically significant deterioration while self-learning does show statistically significant deterioration. Particularly on the last six datasets, ICLS offers large improvement in classification accuracy over the supervised solution. The differences in performance between ICLS and self-learning can also be quite substantial, where ICLS outperforms self-learning on most of the datasets.

Self-learning also performs quite well in the experiments, especially considering its low computational cost. Unlike ICLS, it does lead to significantly higher classification error than supervised learning on the 'Pima' and 'WDBC' datasets.

UCLS often shows major deterioration in performance. It does show some minor improvements on some of the SSL benchmark datasets, in particular USPS, g241c and g241n. In the latter cases, improvements are small. It is unclear whether these effects are real, given the small deterioration in performance on related datasets.

6 Discussion

The theoretical results in Section 4 show that projecting onto a constrained subset $\mathcal{C}_\beta$ leads to improvement in terms of squared loss (Theorem 1) and in terms of Euclidean distance of the parameter estimate (Theorem 2). These results are encouraging in the light of negative theoretical performance results in the semi-supervised literature [9]. The empirical results in the previous section indicate that in terms of the expected classification error, ICLS never significantly deteriorates

with increasing amounts of unlabeled data on our collection of datasets. These empirical observations are all the more interesting considering that the loss evaluated in Section 5 is misclassification error and not the squared loss that was considered in Theorem 1 or the Euclidean parameter distance of Theorem 2. Furthermore the experiments were carried out on limited unlabeled data, not the unlimited setting considered in the theorems. This indicates that projecting onto the subset $\mathcal{C}_\beta$, leads to a semi-supervised learner with desirable behavior, both theoretically in terms of various measures of risk and empirically in terms of classification error.

Some have argued that, for discriminative classifiers, semi-supervised learning is impossible without additional assumptions about the link between labeled and unlabeled objects [2]. ICLS, however, is both a discriminative classifier and no explicit additional assumptions about this link are made. Any assumptions that are present follow, implicitly, from the choice of squared loss as the loss function. Furthermore, no additional parameters need to be correctly set for the results in Sections 4 and 5. There is, for instance, no weight to be chosen for the importance of the unlabeled data. Therefore, implicitly constrained semi-supervised learning isa very different approach to semi-supervised learning than the methods discussed in Section 2.

The quadratic programming formulation of ICLS presented in Section 3 allows one to use the standard and constantly improving tools from convex optimization to find the ICLS estimator. Unfortunately one has to go from a convex problem with m variables in the supervised case to a constrained convex problem with N_u variables for ICLS. For very large N_u, this may not currently be computationally feasible. Improvements in quadratic programming solvers may change this. Additionally, instead of finding the exact optimal labeling $\mathbf{y}_u$, as was employed in our experiments, approximations to these optima may take less time to compute without having a large effect on the final estimate of $\hat{\boldsymbol{\beta}}_{semi}$.

Compared to ICLS, self-learning is much more favorable in terms of computational cost. Self-learning usually converges in a few iterations, where each iteration has the cost of one supervised least squares estimation. As we noted in Section 5 the self-learning approach can increase performance, but large amounts of unlabeled data can also have a detrimental effect. Also, the performance of ICLS is significantly better on many of the datasets considered in our experiments. Hence the price one pays for the low computational cost is in terms of classification error. Note that the solution provided by self-learning is, by construction, also in the constrained subset $\mathcal{C}_\beta$. The difference with ICLS is that in ICLS the choice of estimate from $\mathcal{C}_\beta$ is based on information of the labeled objects only, while SLLS also uses the imputed labels on the unlabeled objects. This may lead to self-deception: if the imputed labels are wrong, a good fit for these wrongly imputed labels does not necessarily lead to a good estimate of $\boldsymbol{\beta}$.

The plug-in version of the LS, UCLS, while fast and intuitive, does not perform well. We found that it only offers some improvement on datasets with low Bayes error. This does not correspond to the observations of [28] that the covariance update only decreases the parameter value in high noise settings. While we do not currently fully understand this behavior, it may be related to the finite sample estimate of $\mathbb{E}[\mathbf{X}'\mathbf{X}]$ that we consider or the differences in modeling assumptions when going from the regression setting considered in [28] to the classification setting considered here.

In Figure 1, we illustrate that projecting onto the subset $\mathcal{C}_\beta$ causes improvement as long as a better solution $\hat{\beta}_{N_l+N_u}$ than the supervised solution is within $\mathcal{C}_\beta$. A smaller $\mathcal{C}_\beta$ will give a larger improvement, since the projection is going to be closer to $\hat{\beta}_{N_l+N_u}$. In the extreme case where only $\hat{\beta}_{N_l+N_u}$ forms the subset, this clearly gives a great improvement over supervised learning. It therefore makes sense to think about reducing the size of $\mathcal{C}_\beta$. In the approach presented in this work, however, to ensure a better solution $\hat{\beta}_{N_l+N_u}$ than the supervised solution is always within the constrained set with probability $P(\hat{\beta}_{N_l+N_u} \in \mathcal{C}_\beta) = 1$, our choice of $\mathcal{C}_\beta$ is conservatively large. It contains elements corresponding to all labelings of the unlabeled points, even extremely unlikely ones. By excluding unlikely labelings from the subset, the size of the $\mathcal{C}_\beta$ may shrink, while the probability that it includes $\hat{\beta}_{N_l+N_u}$ remains high. For instance, one might exclude labelings with class priors that are very unlikely to occur, given the class priors that are observed in the labeled data. Changes to $\mathcal{C}_\beta$ may, therefore, allow for larger improvements in terms of the risk or classification error, while introducing a small chance of deterioration in performance.

While the results presented in this work are promising for squared loss, a worthwhile extension would be to other loss functions. In this work, we were able to derive a quadratic programming formulation for ICLS because there is a closed-form solution of the supervised least squares problem. For many loss functions, closed-form solutions do not exist, which complicates the derivation of their semi-supervised counterparts following the same lines expounded in this work. One of the main difficulties is that, even if the loss considered is differentiable, one cannot straightaway apply techniques like gradient descent to the parameters as this typically leads to solutions that are outside of the set $\mathcal{C}_\beta$. Besides these issues, there is the other open question of what loss functions could benefit from constraining the solution to a subset like $\mathcal{C}_\beta$ in the first place.

Finally, a rather interesting application of the idea presented here is to semi-supervised regression. [36] argues that in least squares regression, unlabeled data may not help. However, one of the things we exploit in the classification setting is that the labels have bounded outputs. Using bounds on the outputs may therefore also lead to improvement in the semi-supervised regression setting. In some regression tasks, bounded outputs could be a natural assumption.

7 Conclusion

This contribution introduced a new semi-supervised approach to least squares classification. By considering all possible labelings of the unlabeled objects and choosing the one that best matches the labeled observations, we derived a robust classifier through a simple quadratic programming formulation. For this procedure, in the univariate setting with a linear model without intercept, we can prove it never degrades performance in terms of squared loss (Theorem 1). An additional theoretical result shows that in the multivariate case, an adapted procedure never degrades in terms of the Euclidean distance of the parameter estimates (Theorem 2). Experimental results indicate that in expectation this robustness also holds in terms of classification error on real datasets. Hence, semi-supervised learning for

least squares classification without additional assumptions can lead to improvements over supervised least squares classification both in theory and in practice.

Acknowledgements Part of this work was funded by project P24 of the Dutch public-private research community COMMIT.

References

1. Widrow, B., Hoff, M.E.: Adaptive switching circuits. In: IRE WESCON Convention Record 4. (1960) 96–104
2. Seeger, M.: Learning with labeled and unlabeled data. Technical report (2001)
3. Sokolovska, N., Cappé, O., Yvon, F.: The asymptotics of semi-supervised learning in discriminative probabilistic models. In Cohen, W.W., McCallum, A., Roweis, S.T., eds.: Proceedings of the 25th International Conference on Machine Learning, Helsinki, Finland, ACM Press (2008) 984–991
4. Hastie, T., Tibshirani, R., Friedman, J.H.: The elements of statistical learning. Spinger (2001)
5. Rifkin, R., Yeo, G., Poggio, T.: Regularized least-squares classification. Nato Science Series Sub Series III Computer and Systems Sciences 190 (2003)
6. Tibshirani, R.: Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society. Series B **58**(1) (1996) 267–288
7. Poggio, T., Smale, S.: The Mathematics of Learning: Dealing with Data. Notices of the AMS (2003) 537–544
8. Bottou, L.: Large-scale machine learning with stochastic gradient descent. In: Proceedings of COMPSTAT'2010, Springer (2010) 177–186
9. Cozman, F., Cohen, I.: Risks of Semi-Supervised Learning. In Chapelle, O., Schölkopf, B., Zien, A., eds.: Semi-Supervised Learning. MIT press (2006) 56–72
10. Chapelle, O., Schölkopf, B., Zien, A.: Semi-supervised learning. MIT press (2006)
11. Zhu, X., Goldberg, A.B.: Introduction to Semi-Supervised Learning. Volume 3. Morgan & Claypool (January 2009)
12. Nigam, K., McCallum, A.K., Thrun, S., Mitchell, T.: Text classification from labeled and unlabeled documents using EM. Machine learning **34** (2000) 1–34
13. Cozman, F.G., Cohen, I., Cirelo, M.C.: Semi-Supervised Learning of Mixture Models. Proceedings of the Twentieth International Conference on Machine Learning (2003)
14. Goldberg, A.B., Zhu, X.: Keepin'it real: semi-supervised learning with realistic tuning. NAACL HLT 2009 Workshop on Semi-supervised Learning for Natural Language Processing (2009)
15. Wang, J., Shen, X., Pan, W.: On transductive support vector machines. Contemporary Mathematics **443** (2007) 7–19
16. McLachlan, G.J.: Iterative Reclassification Procedure for Constructing an Asymptotically Optimal Rule of Allocation in Discriminant Analysis. **70**(350) (1975) 365–369
17. Yarowsky, D.: Unsupervised word sense disambiguation rivaling supervised methods. Proceedings of the 33rd annual meeting on Association for Computational Linguistics - (1995) 189–196
18. Abney, S.: Understanding the yarowsky algorithm. Computational Linguistics **30**(3) (2004) 365–395
19. Grandvalet, Y., Bengio, Y.: Semi-supervised learning by entropy minimization. In Saul, L.K., Weiss, Y., Bottou, L., eds.: Advances in Neural Information Processing Systems 17, Cambridge, MA, MIT Press (2005)
20. Joachims, T.: Transductive inference for text classification using support vector machines. In: Proceedings of the 16th International Conference on Machine Learning, Morgan Kaufmann Publishers (1999) 200–209
21. Bennett, K.P., Demiriz, A.: Semi-supervised support vector machines. In: Advances in Neural Information Processing Systems 11. (1998)
22. Sindhwani, V., Keerthi, S.S.: Large scale semi-supervised linear SVMs. In: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval, New York, New York, USA, ACM Press (2006) 477

23. Collobert, R., Sinz, F., Weston, J., Bottou, L.: Large scale transductive SVMs. Journal of Machine Learning Research **7** (2006) 1687–1712
24. Wang, J., Shen, X.: Large margin semi-supervised learning. Journal of Machine Learning Research **8** (2007) 1867–1891
25. Li, Y.F., Zhou, Z.h.: Towards making unlabeled data never hurt. In: Proceedings of the 28th International Conference on Machine Learning. (2011)
26. Loog, M.: Constrained Parameter Estimation for Semi-Supervised Learning: The Case of the Nearest Mean Classifier. In: Proceedings of the 2010 European Conference on Machine learning and Knowledge Discovery in Databases. (2010) 291–304
27. Loog, M.: Semi-supervised linear discriminant analysis through moment-constraint parameter estimation. Pattern Recognition Letters **In press** (March 2013)
28. Shaffer, J.P.: The Gauss-Markov Theorem and Random Regressors. The American Statistician **45**(4) (1991) 269–273
29. Fan, B., Lei, Z., Li, S.Z.: Normalized LDA for semi-supervised learning. In: 8th IEEE International Conference on Automatic Face & Gesture Recognition, Ieee (September 2008) 1–6
30. Byrd, R.H., Lu, P., Nocedal, J., Zhu, C.: A limited memory algorithm for bound constrained optimization. SIAM Journal on Scientific Computing (1995)
31. Berger, J.O.: Statistical decision theory and Bayesian analysis. Springer (1985)
32. Aubin, J.P.: Applied functional analysis. Volume 47. John Wiley & Sons (2000)
33. Bache, K., Lichman, M.: {UCI} Machine Learning Repository (2013)
34. Raudys, S., Duin, R.P.: Expected classification error of the Fisher linear classifier with pseudo-inverse covariance matrix. Pattern Recognition Letters **19**(5-6) (April 1998) 385–392
35. Opper, M., Kinzel, W.: Statistical Mechanics of Generalization. In Domany, E., Hemmen, J.L., Schulten, K., eds.: Models of Neural Networks III. Springer, New York (1996) 151–209
36. Culp, M., Michailidis, G.: An iterative algorithm for extending learners to a semi-supervised setting. Journal of Computational and Graphical Statistics **17**(3) (2008) 545–571