Implicitly Constrained Semi-Supervised Least Squares Classification

Jesse H. Krijthe and Marco Loog

[1] Leiden University Medical Center
[2] Pattern Recognition Laboratory, Delft University of Technology

Abstract. We introduce a novel semi-supervised version of the least squares classifier. In implicitly constrained least squares (ICLS) classification, 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 give an intuitive proof that this method can never lead to worse performance than supervised least squares classification. In the more general multidimensional case we prove that a slightly adapted procedure leads to guaranteed improvements both in terms of the parameter estimates and, more importantly, in terms of squared loss. The latter result illustrates that strong theoretical guarantees for semi-supervised procedures are possible, at least for specific classifiers and loss functions. Experimental results corroborate the theoretical results and indicate desirable properties over alternative semi-supervised approaches to least squares classification.

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

1 Introduction

We consider the problem of semi-supervised learning of binary classification functions. As 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 L training objects and their corresponding outputs. In the semi-supervised scenario we are given an additional and often large set of U unlabeled objects. The challenge of semi-supervised learning is to incorporate this additional information to improve the classification rule.

The goal of our research is to build a semi-supervised least squares classifier 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 [**?**]. 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.

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 the supervised classifier, whereas current state-of-the-art semi-supervised learning approaches typically rely 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 3.1). 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 [**?**]. 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 [**?**,**?**]. Our results, like [**?**], show this may not strictly hold. Secondly, as we will show in section 3.2, the closed-form solution for the supervised least squares classifier allows us to both formulate our semi-supervised approach as a quadratic programming problem, that can be solved through a simple gradient descent with boundary constraints, and study its theoretical properties. Lastly, least squares classification is a useful and adaptable classification technique allowing for straightforward use of, for instance, regularization, sparsity penalties or kernelization [**?**,**?**,**?**,**?**]. 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 [**?**] as well as computationally efficient on large datasets [**?**].

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. In the univariate setting without intercept we show this procedure *never* gives worse performance in terms of the squared loss criterion compared to the supervised least squares classifier. We prove a similar result in the more general setting allowing for multi-variate inputs and an intercept term for an extension of our procedure. In our experiments, we indeed find that this new approach displays many 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)
- A proof of improvement in terms of parameter estimation and squared loss for an adapted procedure in the multivariate case (Theorem 2).
- An empirical evaluation of the properties of this classifier (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 several proofs of the improvements of the ICLS classifier and slight adaptations over the supervised alternative. 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 [**?**,**?**]. While these have proven successful in particular applications, such as document classification [**?**], it has also been observed that these techniques may give worse performance than their supervised counterparts [**?**,**?**]. In these cases, disregarding the unlabeled data would lead to better performance. Some [**?**,**?**] have argued that *agnostic* semi-supervised learning, which [**?**] defines 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 [**?**] 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 [**?**], also known as Yarowsky's algorithm [**?**,**?**]. 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 [**?**,**?**]. Unfortunately, the process of self-training can also lead to severely decreased performance, compared to the supervised solution [**?**,**?**]. 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 [**?**]. Indeed, expectation maximization suffers from the same issues as self-learning [**?**].

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 [**?**] as well as for support vector classification in the transductive support vector machine (TSVM) [**?**]. When used in the semi-supervised setting this is closely related to the formulation of S^3VM [**?**,**?**]. 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 the convex concave procedure [**?**] and difference convex programming [**?**,**?**].

Some have argued unlabeled data can *only* help if $P(X)$ and $P(Y|X)$ are somehow linked [**?**]. The goal of our work is to show that in some cases (i.e. the least squares classifier) we do not need explicit assumptions about those links for semi-supervised learning to be possible. Instead, we leverage the implicit assumptions, including possible misspecification, that are already present in the supervised classifier. Similar to [**?**] we also study the finite sample case.

In the approaches presented above, a parameter controls the importance of the unlabeled points. When the parameter is correctly set, it is clear, as [**?**] indeed also 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 can also lead to worse performance than the supervised support vector machine. [**?**] 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 S^3VM. While the approach is often 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.

[?,?] attempt to guard against the possibility of deterioration in performance by not introducing additional assumptions, but instead leveraging implicit assumptions already present in the choice of the supervised classifier. 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 [?,?], 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.

While least squares classification has been widely used and studied [?,?,?], little work has been done on applying semi-supervised learning to the least squares classifier specifically. For least squares regression, [?] studied the value of knowing $\mathbb{E}[\mathbf{X}^\top\mathbf{X}]$, where $\mathbf{X}$ is the $L \times d$ 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 $\mathbb{E}[\mathbf{X}^\top\mathbf{X}]$ as the estimate of $\mathbf{X}^\top\mathbf{X}$ 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 $\mathbf{X}^\top\mathbf{X}$ by a version based on both labeled and unlabeled data. A similar plug-in procedure has been used by [?] 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 $\mathbf{X}^\top\mathbf{X}$ 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 L labeled objects and a potentially large set of U unlabeled objects, 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.

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 U of unlabeled objects. Assume that these objects 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 $\hat{\beta}_{oracle}$. Since this estimate is based on

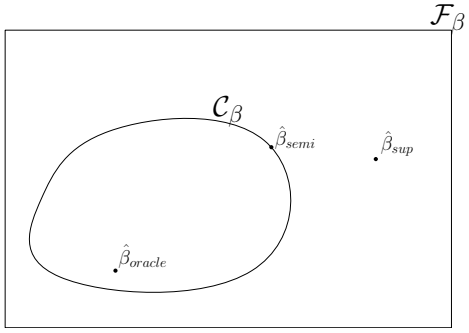

Fig. 1. A visual representation of implicitly constrained semi-supervised learning. $\mathcal{F}_\beta$ is the space of all linear models. $\hat{\beta}_{sup}$ denotes the solution given only a small amount of labeled data. $\mathcal{C}_\beta$ is the subset of the space which contains all the solutions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

more objects, we expect the parameter estimate to be better. These real labels are unknown, but we can still 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 $\mathcal{C}_\beta \subset \mathcal{F}_\beta$. 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 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 $\mathcal{C}_\beta$, an interpretation we will leverage during the proofs in section 4.2. 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 can 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 [**?**,**?**] is the direct application of well-known ordinary least squares regression to a classification problem. A linear model is assumed and the parameters are minimized under squared loss. Let $\mathbf{X}$ be an $L \times (d+1)$ design matrix with L rows containing vectors of length equal to the number of features d plus one for the intercept. Vector $\mathbf{y}$ denotes an $L \times 1$ vector of class labels. Without loss of generality we encode one class 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 $n < (d+1)$), 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 $U \times (d+1)$ *without* a corresponding target vector $\mathbf{y}_u$ is given. In what follows, $\mathbf{X}_e = \left[\mathbf{X}^T\ \mathbf{X}_u^T\right]^T$ 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 $\mathbf{y}_u \in [0,1]^U$. Since these supervised solutions have a closed form, this can be written as:

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

The soft labeling provides 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:

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

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 $\boldsymbol{\beta}$:

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

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:

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

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

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

Where $\mathbf{I}_U$ denotes the $U\times U$ identity matrix and $\mathbf{1}_U$ and $\mathbf{0}_U$ 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 $\mathbf{y}_u\in[0,1]^U$ 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 [**?**]. Finally, the optimal labeling $\mathbf{y}_u$ (as determined 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 [**?**]. 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:

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

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

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

Setting the derivative of (8) with respect to β to 0 and rearranging we get:

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

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

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

In this last equation, since we 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 we will refer to as

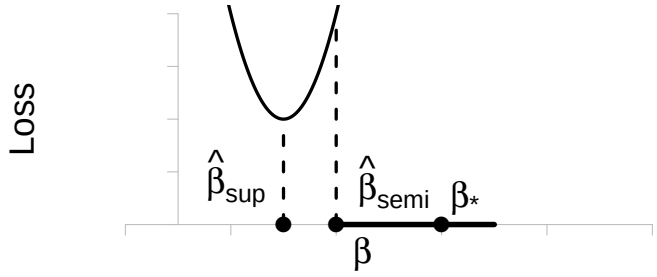

Fig. 2. An example where implicitly constrained optimization improves performance. The supervised solution $\hat{\beta}_{sup}$ which minimizes the supervised loss (shown), is not part of the interval of allowed solutions. The solution that minimizes this supervised loss within the allowed interval is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

the constrained set $\mathcal{C}_{\boldsymbol{\beta}}$. The optimal solution has to be in this interval, since it corresponds to a particular but unknown labeling $\mathbb{E}[y|x]$.

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

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

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 $\hat{\beta}_{semi} = \hat{\beta}_{sup}$.
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 as $\hat{\beta}_{sup}$. 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 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 optimal solution, corresponding to the true labeling, the risk 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 always has an equal or lower risk than the supervised solution:*

$$R^*(\hat{\beta}_{semi}) \leq R^*(\hat{\beta}_{sup})$$

. In particular, for 1 labeled sample, if $f_{X,Y}$ is continuous with bounded second moment 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 $\hat{\beta}_{sup}$ 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:

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

Combining this with (14), 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 in the case of 1 labeled sample improvement is not only possible, but will occur in many cases. When we have multiple labeled examples, this effect will likely become smaller. This makes sense: the more labeled data we have to estimate the parameter, the smaller the impact of the unlabeled objects will be.

4.2 Euclidean Projection Estimator for Multivariate ICLS

In the multivariate case (where we can include an intercept by including a constant feature in the feature vector $\mathbf{x}$), we will first prove that the solution vector obtained through an adapted version of the implicitly constrained least squares classifier is always as to $\boldsymbol{\beta}_{oracle}$ than the supervised solution when using Euclidean distance as a measure of closeness, where $\boldsymbol{\beta}_{oracle}$ is the solution we would obtain if we would have access to all the labels. We will then use a similar method to prove improvement in terms of squared loss in the next section.

While the Euclidean distance between a parameter estimate and the optimal parameter estimate is not directly equivalent to the goal of minimizing classification error, this different notion of statistical risk is commonly used in many areas of statistics [**?**]. We can use this risk as an alternative way to study in what sense the constrained parameter space may offer an improved estimators 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:

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

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 Equation (3), where the projection is the parameter vector in the constrained region with minimum distance to $\hat{\boldsymbol{\beta}}_{sup}$, measured by 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 'closest' 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 will prove the following:

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

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

Note that if we know $f_X(\boldsymbol{x})$, then $\hat{\boldsymbol{\beta}}_{oracle} = \boldsymbol{\beta}^*$, the optimal parameter that we would find using infinite labeled training data. To prove the theorem, we first show that the constrained region $\mathcal{C}_{\boldsymbol{\beta}}$ is convex. Note that the constrained space in terms of $\mathbf{y}_u$, $\mathcal{C}_{\mathbf{y}_u} = [0,1]^U$ is convex. Now, for every pair $\boldsymbol{b}_1, \boldsymbol{b}_2 \in \mathcal{C}_{\boldsymbol{\beta}}$, their corresponding labelings $\mathbf{y}_{u_1}, \mathbf{y}_{u_2}$ and all $c \in [0,1]$, we have that:

$$\begin{aligned} & c\boldsymbol{b}_1 + (1-c)\boldsymbol{b}_2 \\ =& c\left(\mathbf{X}_e^\top \mathbf{X}_e\right)^{-1} \mathbf{X}_e^\top \begin{bmatrix} \mathbf{y} \\ \mathbf{y}_{u_1} \end{bmatrix} + (1-c)\left(\mathbf{X}_e^\top \mathbf{X}_e\right)^{-1} \mathbf{X}_e^\top \begin{bmatrix} \mathbf{y} \\ \mathbf{y}_{u_2} \end{bmatrix} \\ =& \left(\mathbf{X}_e^\top \mathbf{X}_e\right)^{-1} \mathbf{X}_e^\top \begin{bmatrix} \mathbf{y} \\ c\mathbf{y}_{u_1} + (1-c)\mathbf{y}_{u_2} \end{bmatrix} \in \mathcal{C}_\beta \end{aligned} \tag{19}$$

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

We now note the following result which can be found in, for instance, Proposition 1.4.1ii in [**?**, 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 $\hat{\boldsymbol{\beta}}_{sup}$, and b to be $\hat{\boldsymbol{\beta}}_{oracle}$, the parameter vector we would obtain if all the labels were known. Since $\hat{\boldsymbol{\beta}}_{oracle} \in \mathcal{C}_\beta$, its projection is $P_{\mathcal{C}_\beta}(\hat{\boldsymbol{\beta}}_{oracle}) = \hat{\boldsymbol{\beta}}_{oracle}$. The projection of $\hat{\boldsymbol{\beta}}_{sup}$ is $\hat{\boldsymbol{\beta}}_{adapted}$. By the theorem above, we have $||\hat{\boldsymbol{\beta}}_{adapted} - \hat{\boldsymbol{\beta}}_{oracle}|| \leq ||\hat{\boldsymbol{\beta}}_{sup} - \hat{\boldsymbol{\beta}}_{oracle}||$ 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 necessarily the same, meaning a better estimator in terms of Euclidean distance does not necessarily imply a better estimator in terms of squared loss. We can however, get a strong result in terms of the squared loss if we change the distance measure used in the projection.

4.3 Semi-Supervised Projection Estimator for Multivariate ICLS

Using a similar procedure as for the Euclidean distance we can show a stronger results: we can define a semi-supervised least squares classifier than will never increase the squared loss, measured on both labeled and unlabeled objects, as compared to the supervised least squares classifier. Consider the following semi-supervised classifier:

$$\hat{\boldsymbol{\beta}}_{extended} = \underset{\boldsymbol{\beta} \in \mathcal{C}_{\boldsymbol{\beta}}}{\operatorname{argmin}} \sqrt{(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}_{sup})^\top \mathbf{X}_e^\top \mathbf{X}_e (\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}_{sup})} \tag{20}$$

This projects the supervised solution onto the constrained space, but, instead of using the Euclidean distance as in Equation (17), we use a slightly different distance measure based on the labeled and unlabeled training data. For this procedure we can prove the following:

Theorem 3. *Given* $\mathbf{X}$, $\mathbf{X}_\mathrm{u}$ *and* $\mathbf{y}$, $\mathbf{X}_e^\top \mathbf{X}_e$ *positive definite and* $\hat{\boldsymbol{\beta}}_{sup}$ *given by* (2)*. For the projected estimator* $\hat{\boldsymbol{\beta}}_{extended}$ *defined in* (20)*, the following result holds:*

$$L(\hat{\boldsymbol{\beta}}_{extended}, \mathbf{X}_e, \mathbf{y}_\mathrm{e}^*) \leq L(\hat{\boldsymbol{\beta}}_{sup}, \mathbf{X}_e, \mathbf{y}_\mathrm{e}^*)$$

Where $\mathbf{y}_\mathrm{e}^*$ is the true labeling of all, labeled and unlabeled, objects. In other words: $\hat{\boldsymbol{\beta}}_{extended}$ will *always* be at least as good or better than $\hat{\boldsymbol{\beta}}_{sup}$, in terms of the squared surrogate loss, when evaluated on the true labels of both the labeled and unlabeled objects.

The proof of this result follows from a simple geometric interpretation of our procedure. Consider the following inner product, used in equation (20):

$$\langle \mathbf{a}, \mathbf{b} \rangle = \mathbf{a}^\top \mathbf{X}_\mathrm{e}^\top \mathbf{X}_\mathrm{e} \mathbf{b} \tag{21}$$

Let $\mathcal{H}_{\mathbf{X}_e} = (\mathbb{R}^d, \langle ., . \rangle)$ be the inner product space corresponding with this inner product and let $d(a, b) = \sqrt{\langle a - b, a - b \rangle}$. Due to the similarity of the induced metric to a type of weighted Euclidean distance, this is clearly a Hilbert space, as long as $\mathbf{X}_e^\top \mathbf{X}_e$ is positive definite. As we have shown before, $\mathcal{C}_\beta$ is convex. By construction $\hat{\boldsymbol{\beta}}_{extended}$ is the closest projection of $\hat{\boldsymbol{\beta}}_{sup}$ onto this convex constrained set $\mathcal{C}_\beta$ in $\mathcal{H}_{\mathbf{X}_e}$. By the Hilbert space projection theorem, we now have that

$$d(\hat{\boldsymbol{\beta}}_{extended}, \boldsymbol{\beta}) \leq d(\hat{\boldsymbol{\beta}}_{sup}, \boldsymbol{\beta}) \tag{22}$$

for any $\boldsymbol{\beta} \in \mathcal{C}_\beta$. In particular consider $\boldsymbol{\beta} = \hat{\boldsymbol{\beta}}_{oracle}$, which by construction is within $\mathcal{C}_\beta$. That is, all possible labelings correspond to an element in $\mathcal{C}_\beta$, so this also holds for the true labeling $\mathbf{y}_{\mathrm{u}}^*$. Plugging in the closed form solution of $\hat{\boldsymbol{\beta}}_{oracle}$ into (22) and after some manipulations we find:

$$d(\hat{\boldsymbol{\beta}}_{extended}, \hat{\boldsymbol{\beta}}_{oracle})^2 = L(\hat{\boldsymbol{\beta}}_{extended}, \mathbf{X}_e, \mathbf{y}_{\mathrm{e}}^*) + C$$

and

$$d(\hat{\boldsymbol{\beta}}_{sup}, \hat{\boldsymbol{\beta}}_{oracle})^2 = L(\hat{\boldsymbol{\beta}}_{sup}, \mathbf{X}_e, \mathbf{y}_{\mathrm{e}}^*) + C$$

where C is a constant that is equal for both cases. From this the result in Theorem 3 follows directly.

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 using a cross-validation setup on various datasets to study how the properties of our approach compare to other semi-supervised least squares classification approaches.

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 $\mathbf{X}_e^\top \mathbf{X}_e$ similar to the estimator in [**?**]. 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 (LS_{oracle}). 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 [**?**] and six of the benchmark datasets proposed by [**?**]. While the benchmark datasets proposed in [**?**] 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 first author's website.

Table 1. Description of the datasets used in the experiments. Features indicates the dimensionality of the design matrix after categorical features are expanded into dummy variables.

Name	# Objects	#Features	Source
Haberman	306	4	[?]
Ionosphere	351	35	[?]
Parkinsons	195	23	[?]
Pima	768	9	[?]
Sonar	208	61	[?]
SPECT	267	23	[?]
SPECTF	267	45	[?]
Transfusion	748	110	[?]
WDBC	569	31	[?]
Mammography	961	10	[?]
Thoraric	470	25	[?]
Fertility	100	17	[?]
Digit1	1500	242	[?]
USPS	1500	242	[?]
COIL2	1500	242	[?]
BCI	400	118	[?]
g241c	1500	242	[?]
g241n	1500	242	[?]

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, 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 d samples, the least squares classifier is known to deteriorate in performance as more data is added, a behavior known as peaking [**?**,**?**]. 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 we ensure

by setting $L = d + 5$, the dimensionality and intercept of the dataset plus five observations. For all datasets we ensure a minimum of $L = 20$ labeled objects.

Next, we create unlabeled subsets of increasing size $U = [2, 4, 8, ..., 1024]$ 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.

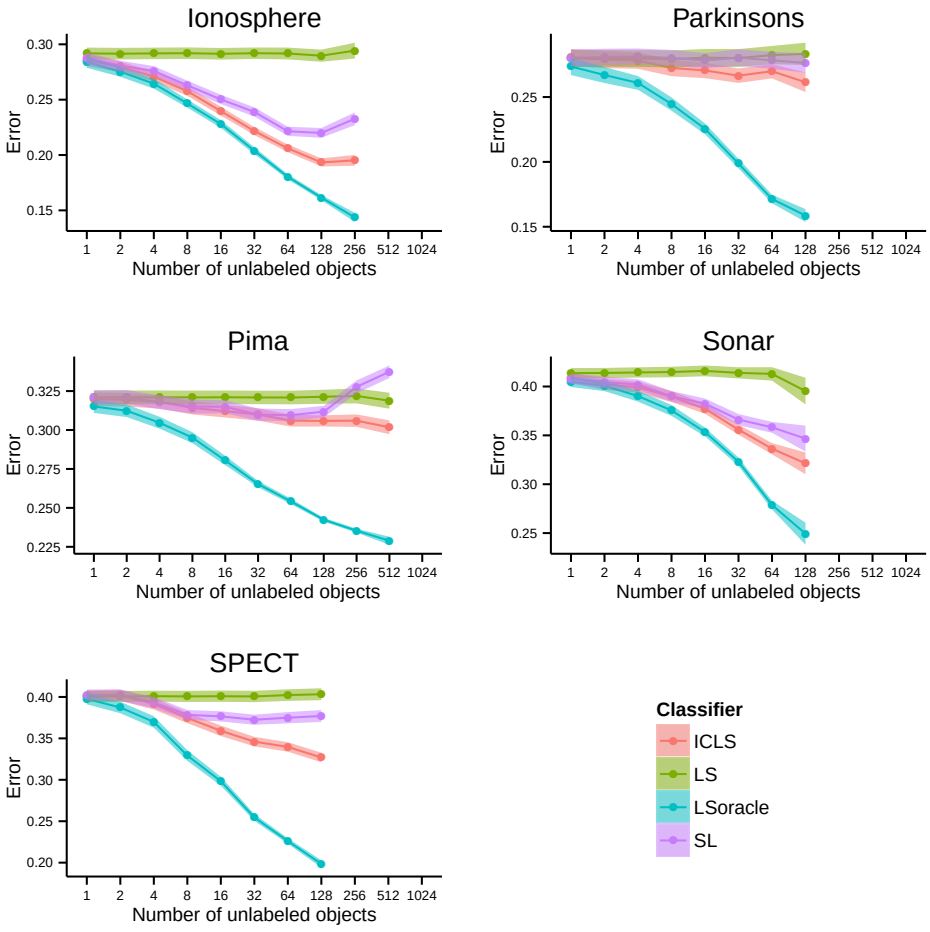

Fig. 3. Mean classification error for $L = \max(d + 5, 20)$ and 100 repeats. The error bounds are $+/-$ the standard error of the mean.

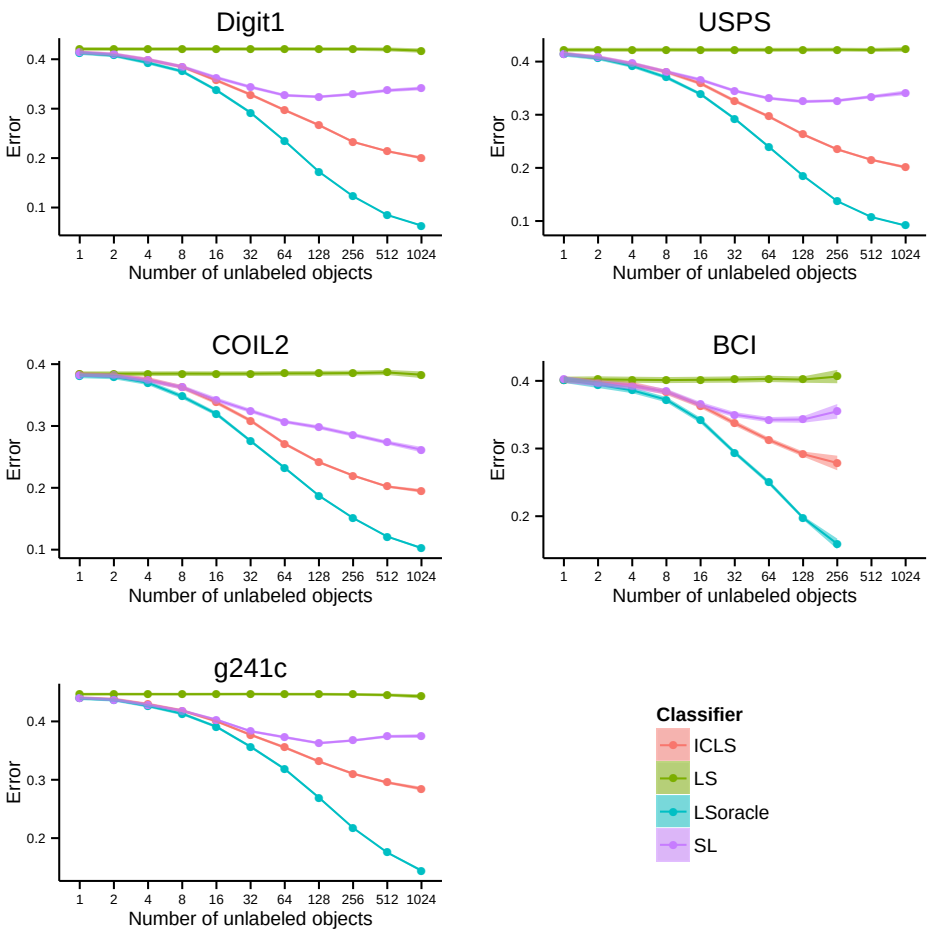

Fig. 4. Mean classification error for $L = \max(d + 5, 20)$ and 100 repeats. The error bounds are $+/-$ the standard error of the mean.

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 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 where 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 $L = d + 5$ 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 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 described in [**?**], 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 [**?**], 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 [**?**]. 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 supervised procedure. On these datasets, the self-learning approach seems to share this property. However, if we look at for how many of the cross-validation repeats the ICLS and self-learning give lower error than the supervised solution, there is a clear difference. The self-learning solution gives a higher error on more of the repeats than ICLS, for most of the datasets. The results also show that unlabeled information can be of use. 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.

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.

Looking at the results of the adapted and extended ICLS procedures, we find that the adapted procedure, which makes use of the Euclidean projection onto the constrained space, performs much worse than the ICLS classifier. The extended procedure, on the other hand does offer improvements over the supervised solution on most of the datasets, which its improvements are almost always smaller than those offered by the original ICLS procedure. While it has stronger theoretical guarantees against deterioration in performance than the

Table 2. Average 10-fold cross-validation error and standard deviation over 20 repeats. The classifiers that have been compared are supervised Least Squares (LS), Implicitly constrained least squares (ICLS), the extended ICLS presented in section 4.3 (ICLS_{ext}), the adapted ICLS procedure from section 4.2 (ICLS_{adp}), self-learned least squares (SLLS), updated covariance least squares (UCLS, see text) and the supervised least squares classifier that has access to all the labels (LS_{oracle}). 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	ICLS_{ext}	ICLS_{adp}	SLLS	UCLS	LS_{oracle}
Haberman	0.36(0)	0.34(3)	0.35(0)	0.35(8)	0.35(7)	0.46(20)	0.26(0)
Ionosphere	0.28(0)	**0.19(0)**	**0.23(0)**	0.39(20)	**0.24(1)**	0.36(18)	0.14(0)
Parkinsons	0.24(0)	0.24(9)	0.24(6)	0.31(20)	**0.22(1)**	0.41(20)	0.14(0)
Diabetes	0.37(0)	**0.33(0)**	0.36(1)	0.42(19)	0.36(9)	0.42(19)	0.23(0)
Sonar	0.42(0)	**0.34(0)**	**0.38(0)**	0.42(11)	**0.37(2)**	0.44(15)	0.25(0)
SPECT	0.41(0)	**0.33(0)**	0.41(4)	0.45(18)	0.40(7)	0.45(17)	0.17(0)
SPECTF	0.45(0)	**0.35(0)**	**0.42(2)**	0.45(11)	**0.40(1)**	0.45(11)	0.23(0)
Transfusion	0.25(0)	0.25(14)	0.24(4)	0.26(17)	0.24(11)	0.34(20)	0.23(0)
WDBC	0.10(0)	**0.10(4)**	0.10(10)	0.30(20)	0.10(10)	0.30(20)	0.05(0)
Mammography	0.31(0)	0.30(3)	0.31(12)	0.37(20)	0.32(13)	0.41(20)	0.20(0)
Thoraric	0.27(0)	0.26(5)	0.26(2)	0.35(20)	**0.24(1)**	0.35(20)	0.17(0)
Fertility	0.28(0)	**0.24(4)**	0.27(3)	0.31(16)	**0.25(3)**	0.37(18)	0.13(0)
Digit1	0.42(0)	**0.20(0)**	**0.38(0)**	0.44(15)	**0.35(0)**	**0.40(6)**	0.06(0)
USPS	0.42(0)	**0.20(0)**	**0.38(0)**	0.44(19)	**0.34(0)**	**0.39(2)**	0.09(0)
COIL2	0.38(0)	**0.20(0)**	**0.35(0)**	0.44(20)	**0.27(0)**	0.40(17)	0.10(0)
BCI	0.41(0)	**0.27(0)**	**0.36(0)**	0.40(9)	**0.35(0)**	0.42(12)	0.16(0)
g241c	0.44(0)	**0.28(0)**	**0.41(0)**	0.43(5)	**0.39(0)**	**0.43(5)**	0.14(0)
g241n	0.45(0)	**0.28(0)**	**0.41(0)**	**0.44(2)**	**0.39(0)**	**0.42(1)**	0.13(0)

original ICLS procedure, the extended procedure may be so conservative that it will not give large improvements when unlabeled data can be of use.

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 3) 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 [**?**]. 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 this 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 and 3 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.

It has been argued that, for discriminative classifiers, semi-supervised learning is impossible without additional assumptions about the link between labeled and unlabeled objects [**?**,**?**]. 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 and from the hypothesis space. One could argue that constraining the solutions to $\mathcal{C}_\beta$ is an assumption as well. While this is true, it corresponds to a very weak assumption about the supervised classifier: that it will improve when we add additional labeled data. This lack of additional assumptions has another advantage: 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 is a 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 estimate. Unfortunately one has to go from a convex problem with d variables in the supervised case to a constrained convex problem with U variables for ICLS. For very large U, this may not currently be computationally feasible. Improvements in quadratic programming solvers may help alleviate this problem. 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 [**?**] 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 $\mathbb{E}[\mathbf{X}^\top\mathbf{X}]$ that we consider or the differences in modeling assumptions when going from the regression setting considered in [**?**] to the classification setting considered here.

The alternative distance measures used for the projection onto $\mathcal{C}_\beta$ and introduced in section 4 did not give rise to the same increase in performance in terms of classification error as the original ICLS procedure. This was to be expected for the Euclidean distance measure, since the corresponding objective that is minimized is potentially far removed from the goal of classification. It is interesting, however, that the distance measure used in $\text{ICLS}_{extended}$, which does offer the strong theoretical guarantees in terms of the loss function, gives rise to behaviour that is perhaps too conservative. In terms of robustness, this classifier corresponds most directly to our goal of constructing a semi-supervised version that is never worse than the supervised alternative. In terms of performance in practice, however, less conservative projections, such as the original ICLS may strike a better balance between robustness and improvement in performance.

In Figure 1, we illustrate that projecting onto the subset $\mathcal{C}_\beta$ causes improvement as long as a better solution $\hat{\beta}_{oracle}$ 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 $\hat{\beta}_{oracle}$. In the extreme case where only $\hat{\beta}_{oracle}$ 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 $\hat{\beta}_{oracle}$ than the supervised solution is always within the constrained set with probability $P(\hat{\beta}_{oracle} \in \mathcal{C}_\beta) = 1$, 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 $\mathcal{C}_\beta$ may shrink, while the probability that it includes $\hat{\beta}_{oracle}$ 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, a strategy which is also employed in Transductive SVMs where it is necessary to converge to meaningful local optima. 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 prohibits a straightforward formulation of their implicitly constrained semi-supervised counterparts. In particular, in the derivation of ICLS, we made use of the closed-form solution given an imputed labeling to derive a quadratic programming problem in terms of the labels. Without such a solution, 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. For logistic loss, for instance, the use of the logistic function ensures that posteriors are always between $[0, 1]$. In that case it seems that the current definition of $\mathcal{C}_\beta$ does not constrain the solutions at all. Treating negative log likelihood as a loss function, on the other hand, does lead to interesting semi-supervised classifiers, for instance in linear discriminant analysis [**?**]. Even when $\mathcal{C}_\beta$ does not constrain the solution, we could still be able to construct other constrained sets with interesting performance guarantees. For instance, many semi-supervised classifiers make the additional assumption that the mean posterior labeling of the unlabeled objects is similar to the observed label prior in the labeled objects [**?**,**?**] or is assumed to be known [**?**]. Rather than ensuring non degradation for every transductive set, such as the result in Theorem 3, such additions to the definition of $\mathcal{C}_\beta$ could lead to improvement over the supervised solution with high probability.

Finally, a rather interesting application of the idea presented here is to semi-supervised regression. [**?**] argues that in least squares regression, unlabeled data may not help. But assuming bounded outputs, which may be justified in certain settings, our approach can again provide improvement guarantees.

7 Conclusion

This contribution introduced a new semi-supervised approach to least squares classification. By implicitly 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) as well as in terms of the squared loss. 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.