mathematics

MDPI

Article

A Generalized Linear Joint Trained Framework for Semi-Supervised Learning of Sparse Features

Juan Carlos Laria [1,*], Line H. Clemmensen [2], Bjarne K. Ersbøll [2] and David Delgado-Gómez [1]

[1] Department of Statistics, University Carlos III of Madrid, Calle Madrid 126, 28903 Getafe, Spain
[2] Department of Applied Mathematics and Computer Science, Technical University of Denmark, Düsternbrooker Weg 20, 24105 Lyngby, Denmark
* Correspondence: juank.laria@gmail.com

Abstract: The elastic net is among the most widely used types of regularization algorithms, commonly associated with the problem of supervised generalized linear model estimation via penalized maximum likelihood. Its attractive properties, originated from a combination of ℓ_1 and ℓ_2 norms, endow this method with the ability to select variables, taking into account the correlations between them. In the last few years, semi-supervised approaches that use both labeled and unlabeled data have become an important component in statistical research. Despite this interest, few researchers have investigated semi-supervised elastic net extensions. This paper introduces a novel solution for semi-supervised learning of sparse features in the context of generalized linear model estimation: the generalized semi-supervised elastic net (s^2net), which extends the supervised elastic net method, with a general mathematical formulation that covers, but is not limited to, both regression and classification problems. In addition, a flexible and fast implementation for s^2net is provided. Its advantages are illustrated in different experiments using real and synthetic data sets. They show how s^2net improves the performance of other techniques that have been proposed for both supervised and semi-supervised learning.

Keywords: covariate shift; elastic net; semi-supervised classification; semi-supervised regression; unlabeled data

MSC: 62-08

check for updates

Citation: Laria, J.C.; Clemmensen, L.H.; Ersbøll, B.K.; Delgado-Gómez, D. A Generalized Linear Joint Trained Framework for Semi-Supervised Learning of Sparse Features. *Mathematics* **2022**, *10*, 3001. https://doi.org/10.3390/math10163001

Academic Editor: Liangxiao Jiang

Received: 25 July 2022
Accepted: 17 August 2022
Published: 19 August 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

A large proportion of statistical learning has focused on supervised techniques. However, there are many problems where the available labeled observations are scarce or where obtaining the labels is costly, but where substantial amounts of unlabeled data exist. For example, thousands of medical images are collected every day, but labeling them is a time-consuming and expensive process [1]. An expert is required to manually identify and locate regions of interest, such as those containing aneurysms, emphysemas, or polyps. Besides locating the regions of interest, it is usually necessary to have them analyzed by a laboratory to confirm the diagnosis and assess their severity. Another example is found in deoxyribonucleic acid (DNA) analysis, where obtaining the three-dimensional folder structure of a protein can take months of expensive lab work to a crystallographer [2]. A third example can be found in the field of clinical psychology, where a clinician has the responses given by a small group of patients to a given test together with the diagnosis assigned after a laborious clinical interview. In addition, the answers from a separate undiagnosed group that has filled this test online are also collected. Other examples can be found in research areas, such as natural language processing [3], image classification and segmentation [4,5], image quality assessment [6], graph-based classification [7], cancer detection [8], or web content classification [9].

The accessibility of partially labeled databases, such as those obtained in the previous problems, has caused semi-supervised learning to receive enormous interest in recent years. This interest can be seen in the various review articles showing how several techniques developed in the fields of statistics and machine learning have been adapted to the semi-supervised framework [10,11]. These include decision trees [12], support vector machines [13], neural networks [14], discriminant techniques [15] or regressions [16].

Despite this enormous effort in adapting the supervised techniques to the semi-supervised paradigm, one widely used technique that has received limited attention is the elastic net [17–19]. Among the first techniques that were developed to extend the elastic net to the semi-supervised paradigm was the joint trained elastic net (JT) [20]. JT internally assigns labels to the unlabeled data in conjunction with the optimization of the objective function. One aspect to note is that this objective function contains a term that controls the importance that is given to the unlabeled data. A weakness of this work, as indicated by the author himself in a later paper, is that there was no analysis of the conditions for which the model was expected to be useful [21]. In the latter work, this weakness was addressed and they demonstrated that JT was able to handle covariate shifts, that is, scenarios in which the distribution of features in labeled and unlabeled data were different.

Although they provided a mathematical proof of performance bounds, Larsen et al. pointed out the difficulty in tuning and interpreting their model parameters [22]. The latter authors recently extended the work of Culp and Ryan so that the shift in mean value and the covariance structure are modeled explicitly, providing greater interpretability. It is important to note that, to date, the joint trained methodology is only applicable to linear regression problems, and there is no available software implementation of semi-supervised elastic net in the generalized linear framework.

Regarding classification with unlabeled data, early extensions of logistic models to handle unlabeled observations are found in the work by Amini and Gallinari [23]. More recent approaches to deal with classification in the semi-supervised framework are described by Culp and Ryan [24] and Krijthe and Loog [25]. However, none of these research works have considered approaching the problem from the elastic net perspective.

In this article, a methodological and algorithmic approach called s^{2}net is developed to extend the elastic net to semi-supervised generalized models. Therefore, it tackles the problem of feature selection in semi-supervised contexts. Its mathematical formulation is presented from a general perspective, covering a wide range of models. We will focus on linear and logistic responses, but the implementation can be easily extended to other losses in generalized models. In addition, a flexible and fast implementation in R is provided.

This paper is organized as follows. Section 2 provides an overview of previous works that are closely related to the technique s^{2}net. Then, Section 3 provides the mathematical framework of our methodology. Details regarding the algorithm and its implementation are discussed in Section 4. Sections 5 and 6 explore its properties using synthetic and real data sets, respectively. Some conclusions are drawn in the final section.

2. Related Works

In this section, the most related works to the proposed methodology are presented from a mathematical point of view.

The elastic net technique was introduced for generalized linear models in the supervised context by Friedman et al. [26]. It is formulated as

$$\underset{\boldsymbol{\beta}}{\operatorname{argmin}}\{\mathcal{R}(\boldsymbol{y}_{\mathrm{L}}, \mathbf{X}_{\mathrm{L}}\boldsymbol{\beta}) + \lambda_1\|\boldsymbol{\beta}\|_1 + \lambda_2\|\boldsymbol{\beta}\|_2\}, \tag{1}$$

where $\mathbf{X}_{\mathrm{L}}$ is the (standardized) matrix that contains the labeled observations, $\boldsymbol{y}_{\mathrm{L}}$ is the vector with the corresponding labels, $\boldsymbol{\beta}$ is the vector containing the weights of the regression, and $\lambda_1\|\boldsymbol{\beta}\|_1 + \lambda_2\|\boldsymbol{\beta}\|_2$ is the penalization term. The notations $\|\cdot\|_1$ and $\|\cdot\|_2$ refer to the ℓ_1 and ℓ_2 norms, and $\mathcal{R}$ represents the risk function (for example, the squared error or the logistic error).

Later, Culp [20], motivated by the automatic text analysis, introduced the elastic net regularization in the semi-supervised framework. Nowadays, it is possible to access numerous and diverse online documents, such as free books or comments on social networks. There is interest in associating labels to these documents, such as the genre of the book or whether the comments are favorable. Obtaining the predictors such as the bag of words is straightforward. However, obtaining the dependent variable is complicated and costly. It would involve reading the books, and each of the millions of comments. To tackle this problem, Culp proposed the following semi-supervised elastic net formulation:

$$\underset{\boldsymbol{\beta},\boldsymbol{\alpha}}{\operatorname{argmin}}\left\{\|\boldsymbol{y}_{\alpha}-\mathbf{X}\boldsymbol{\beta}\|_2^2+\lambda J(\boldsymbol{\beta})+\gamma\|\boldsymbol{\alpha}\|_2^2\right\}, \tag{2}$$

where $\mathbf{X}$ is the matrix containing the labeled and unlabeled observations, $\boldsymbol{y}_{\alpha}$ is the vector obtained concatenating the labels $\boldsymbol{y}_L$ with the vector $\mathbf{X}_{\mathrm{U}}\boldsymbol{\alpha}$ ($\mathbf{X}_{\mathrm{U}}$ being the matrix containing the unlabeled observations), and $J(\boldsymbol{\beta})$ is any penalization function, such the elastic net, lasso or ridge.It is important to notice that, from a computational point of view, JT is not a novel algorithm. Its solution is computed using the supervised elastic net (specifically, the glmnet package for R), so it can exploit the properties of the elastic net implementation, such as regularization paths [27] and the safe rules [28].

It is important to highlight that Culp showed that the above formulation is equivalent to solving the optimization problem [20]:

$$\underset{\boldsymbol{\beta}}{\operatorname{argmin}}\left\{\|\boldsymbol{y}_{\mathrm{L}}-\mathbf{X}_{\mathrm{L}}\boldsymbol{\beta}\|_2^2+\left\|\mathbf{0}-\mathbf{X}_{\mathrm{U}}^{(\gamma)}\boldsymbol{\beta}\right\|_2^2+\lambda_1\|\boldsymbol{\beta}\|_1+\lambda_2\|\boldsymbol{\beta}\|_2^2\right\}, \tag{3}$$

with

$$\mathbf{X}_{\mathrm{U}}^{(\gamma)}=\sqrt{\gamma}(\boldsymbol{\Sigma}^2+\gamma\mathbb{I})^{-1/2}\boldsymbol{U}^{\top}\mathbf{X}_{\mathrm{U}}, \tag{4}$$

where $\mathbf{X}_{\mathrm{U}}=\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\top}$ is the singular value decomposition of $\mathbf{X}_{\mathrm{U}}$.

The above work was extended by Ryan and Culp to include scenarios where covariate shift might occur [21], that is, situations where the feature distributions of labeled and unlabeled data may differ. One possible scenario, indicated by the authors, is drug discovery. When a new drug is developed, obtaining covariates is straightforward (e.g., measurements of its components). However, responses such as side effects or the overall effect can take years to be obtained. For that reason, labeled data would come from similar drugs that were previously analyzed and for which both covariates and response are available. To handle those scenarios, Ryan and Culp introduced an extra parameter in the previous semi-supervised problem formulation that allow a better control of the importance given to the unlabeled part:

$$\underset{\boldsymbol{\beta}}{\operatorname{argmin}}\left\{\|\boldsymbol{y}_{\mathrm{L}}-\mathbf{X}_{\mathrm{L}}\boldsymbol{\beta}\|_2^2+\gamma_1\left\|\mathbf{X}_{\mathrm{U}}^{(\gamma_2)}\boldsymbol{\beta}\right\|_2^2+\lambda_1\|\boldsymbol{\beta}\|_1+\lambda_2\|\boldsymbol{\beta}\|_2^2\right\}, \tag{5}$$

where

$$\mathbf{X}_{\mathrm{U}}^{(\gamma_2)}=\sqrt{\gamma_2}(\boldsymbol{\Sigma}^2+\gamma_2\mathbb{I})^{-1/2}\mathrm{U}^{\top}\mathbf{X}_{\mathrm{U}}, \tag{6}$$

This formulation, in addition of being more flexible, includes the supervised least-squares problem as a particular case when $\gamma_1=0$.

Recently, Larsen et al. proposed the extended linear joint trained framework (ExtJT). It adds a methodological improvement to Ryan's semi-supervised formulation, which takes into account the shift in the expected value of the response variable in the unlabeled data with respect to the labeled data [22]. This improvement is introduced through an extra term in the objective function

$$\underset{\boldsymbol{\beta}}{\operatorname{argmin}}\left\{\|\boldsymbol{y}_{\mathrm{L}}-\mathbf{X}_{\mathrm{L}}\boldsymbol{\beta}\|_2^2+\gamma_1\gamma_2\left\|\mathbf{X}_{\mathrm{U}}^{(\gamma_2)}\boldsymbol{\beta}\right\|_2^2+\gamma_3\frac{n_{\mathrm{L}}n_{\mathrm{U}}}{n_{\mathrm{U}}+n_{\mathrm{L}}}\left\|\boldsymbol{\mu}^\top\boldsymbol{\beta}\right\|_2^2\right\}, \tag{7}$$

where n_{U} and n_{L} are the number of unlabeled and labeled observations, respectively, and $\boldsymbol{\mu}$ is the vector mean of the columns of $\mathbf{X}_{\mathrm{U}}$. However, Larsen et al. focused only on the linear response case. Among the conducted experiments, it was observed the excellent performance that ExtJT achieved in predicting the weight in percentage of the active ingredient in pharmaceutical tables when the measurements were collected with two similar spectrometers.

The s^{2}net proposed in this article integrates the core ideas of ExtJT, including the elastic-net regularization to deal with high-dimensional data, and a generalization to both regression and classification problems. Thus, our framework also provides semi-supervised logistic regression models with elastic-net penalizations. In addition, unlike the previous techniques, it does not rely on other implementations. The s^{2}net methodology is described in the following section.

3. Methodology

In the following, we present the analytical derivations leading to the creation of our s^{2}net technique. For this purpose, we first take the ExtJT formulation, including the elastic net regularization,

$$\underset{\boldsymbol{\beta}}{\operatorname{argmin}}\left\{\|\boldsymbol{y}_{\mathrm{L}}-\mathbf{X}_{\mathrm{L}}\boldsymbol{\beta}\|_2^2+\gamma_1\gamma_2\left\|\mathbf{X}_{\mathrm{U}}^{(\gamma_2)}\boldsymbol{\beta}\right\|_2^2+\gamma_3\frac{n_{\mathrm{L}}n_{\mathrm{U}}}{n_{\mathrm{U}}+n_{\mathrm{L}}}\left\|\boldsymbol{\mu}^\top\boldsymbol{\beta}\right\|_2^2+\lambda_1\|\boldsymbol{\beta}\|_1+\lambda_2\|\boldsymbol{\beta}\|_2^2\right\}. \tag{8}$$

Using a reparameterization of γ_1, γ_2 and γ_3, one can show that the terms $\gamma_1\gamma_2\left\|\mathbf{X}_{\mathrm{U}}^{(\gamma_2)}\boldsymbol{\beta}\right\|_2^2+\gamma_3 n_{\mathrm{L}}n_{\mathrm{U}}/(n_{\mathrm{U}}+n_{\mathrm{L}})\|\boldsymbol{\mu}^\top\boldsymbol{\beta}\|_2^2$ are equivalent to $\gamma_1\|\boldsymbol{T}(\gamma_2,\gamma_3)\boldsymbol{\beta}\|_2^2$, where $\boldsymbol{T}(\gamma_2,\gamma_3)$ is a transformation of the unlabeled data that captures both the covariance structure and the shift with respect to the labeled data, given by

$$\boldsymbol{T}(\gamma_2,\gamma_3)=\sqrt{\gamma_2}\boldsymbol{U}(\boldsymbol{\Sigma}^2+\gamma_2\mathbb{I})^{-1/2}\boldsymbol{\Sigma}\boldsymbol{V}^\top+\gamma_3\mathbb{1}\boldsymbol{\mu}^\top. \tag{9}$$

After this reparameterization, the objective function is given by

$$\underset{\boldsymbol{\beta}}{\operatorname{argmin}}\left\{\|\boldsymbol{y}_L-\mathbf{X}_L\boldsymbol{\beta}\|_2^2+\lambda_1\|\boldsymbol{\beta}\|_1+\lambda_2\|\boldsymbol{\beta}\|_2^2+\gamma_1\|\bar{\boldsymbol{y}}_L\mathbb{1}-\boldsymbol{T}(\gamma_2,\gamma_3)\boldsymbol{\beta}\|_2^2\right\}. \tag{10}$$

Notice that in the previous formulation, we included a transformation to center the data.

We now turn our attention to an extension of (10). The choice of square error norm for the error term $\|\boldsymbol{y}_{\mathrm{L}}-\mathbf{X}_{\mathrm{L}}\boldsymbol{\beta}\|_2^2$ is justified when the underlying model is linear. However, in other scenarios (for instance, binary response) it makes more sense to use other risk functions. With that in mind, we propose to write (10) in a more general form, letting $\mathcal{R}(\cdot\,|\,\boldsymbol{y},\mathbf{X}):\mathbb{R}^p\to\mathbb{R})$ be any (continuously differentiable and convex) risk function.

$$\underset{\boldsymbol{\beta}}{\operatorname{argmin}}\left\{\mathcal{R}(\boldsymbol{\beta}\,|\,\boldsymbol{y}_{\mathrm{L}},\mathbf{X}_{\mathrm{L}})+\lambda_1\|\boldsymbol{\beta}\|_1+\lambda_2\|\boldsymbol{\beta}\|_2^2+\gamma_1\mathcal{R}(\boldsymbol{\beta}\,|\,\bar{\boldsymbol{y}}_{\mathrm{L}},\boldsymbol{T}(\gamma_2,\gamma_3))\right\}. \tag{11}$$

Notice that both the input data matrices and the hyper-parameters are fixed, and therefore, (without loss of generality) problem (11) can be reparameterized as the s^{2}net formulation

$$\underset{\boldsymbol{\beta}\in\mathbb{R}^p}{\operatorname{argmin}}\left\{L(\boldsymbol{\beta})+\lambda_1\|\boldsymbol{\beta}\|_1+\lambda_2\|\boldsymbol{\beta}\|_2^2\right\}, \tag{12}$$

where $L(\boldsymbol{\beta}\,|\,\boldsymbol{y}_{\mathrm{L}},\mathbf{X}_{\mathrm{L}},\mathbf{X}_{\mathrm{U}},\gamma_1,\gamma_2,\gamma_3)$ is given by

$$L(\boldsymbol{\beta})=\mathcal{R}(\boldsymbol{\beta}\,|\,\boldsymbol{y}_{\mathrm{L}},\mathbf{X}_{\mathrm{L}})+\gamma_1\mathcal{R}(\boldsymbol{\beta}\,|\,\bar{\boldsymbol{y}}_{\mathrm{L}},\boldsymbol{T}(\gamma_2,\gamma_3)). \tag{13}$$

Before showing the s^{2}net optimization, we present a few remarks.

Remark 1. *Problem* (12) *is a generalized elastic net problem with a custom loss function. If* $\gamma_1 = 0$, *then* (12) *is the (naïve) supervised elastic net problem [29].*

Remark 2. *If we let* $\boldsymbol{T}(\gamma_2) = \sqrt{\gamma_2}\boldsymbol{U}(\boldsymbol{\Sigma}^2 + \gamma_2\mathbb{I})^{-1/2}\boldsymbol{U}^\top\mathbf{X}_U$, *with* $\mathbf{X}_U = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top$ *the singular value decomposition of* $\mathbf{X}_U$ *(without centering), and* $\mathcal{R}(\cdot\,|\,\boldsymbol{y}, \mathbf{X})$ *the norm-2 squared error, then* (11) *is the linear joint trained framework (JT) [20].*

Remark 3. *In* (11), *the hyper-parameter* γ_2 *regulates the covariance structure, whereas* γ_3 *controls the shift between the center of the labeled data and the center of the unlabeled data. Figure 1 provides insights into the intuition behind* $\boldsymbol{T}(\gamma_2, \gamma_3)$, *when the hyper-parameters* γ_2 *and* γ_3 *are changed.*

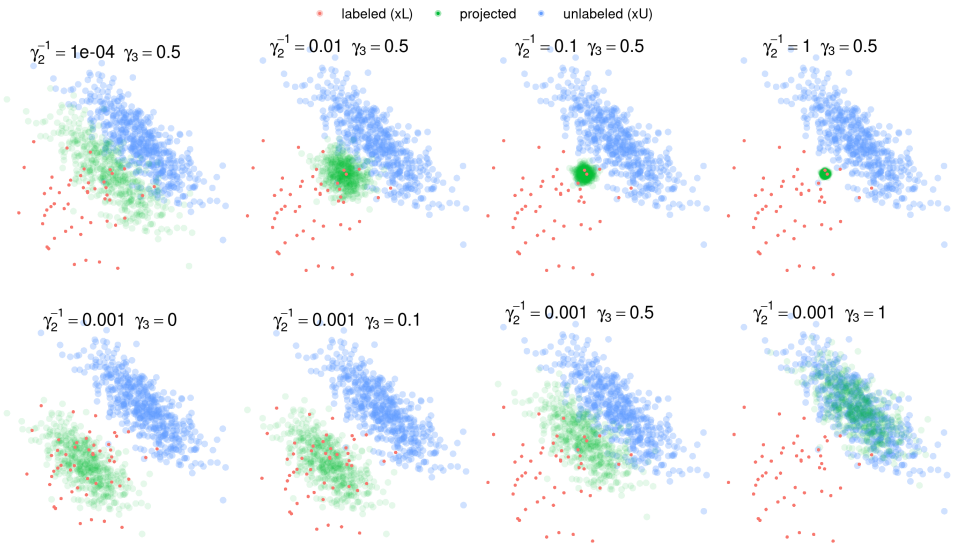

Figure 1. Simulated two-dimensional data that illustrate how varying the parameters γ_2 and γ_3 affects the projected "null" data $\boldsymbol{T}(\gamma_2, \gamma_3)$.

Previous remarks highlight that s^{2}net generalizes other approaches, and therefore, with a strong algorithm to optimize the objective function and an appropriate selection of the hyper-parameters, s^{2}net can outperform (or at least emulate) other popular methods' results.

Before concluding this section, we would like to provide a few notes about the intuition behind the method. As shown in panel A of Figure 2, we have a set of labeled source observations composed of two independent variables, and a set of unlabeled target observations. Initially, the method performs a transformation of the mean and the covariance matrix of the unlabeled target observations (panel B). This is achieved by fixing the values γ_2 y γ_3 as explained in Remark 3 and shown in Figure 1. Using the labeled source observations and the transformed target observations, the regression plane is updated (panel C). This regression plane is used to obtain the prediction of the unlabeled target observations (panel D).

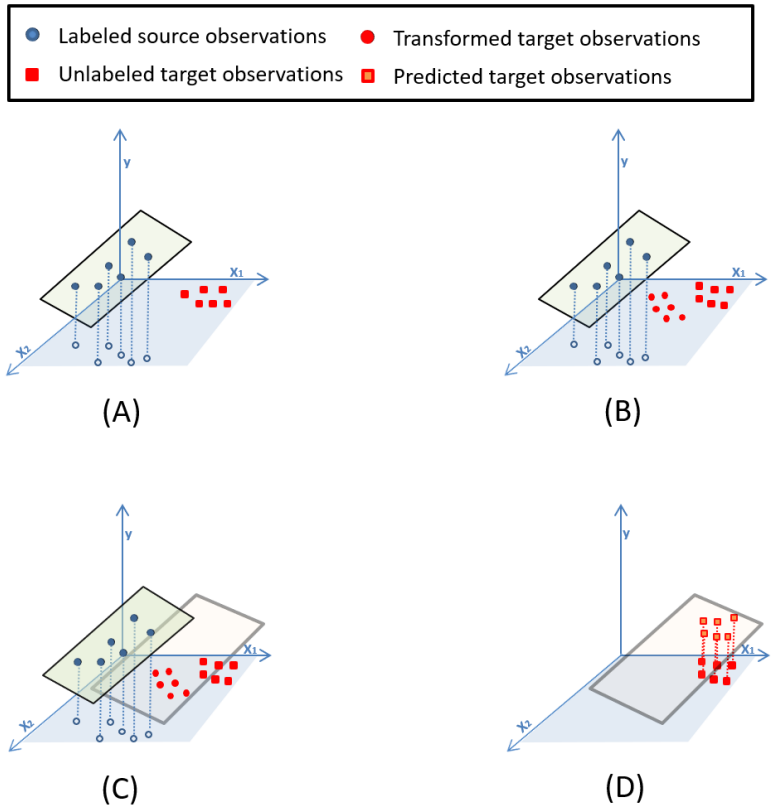

Figure 2. Intuition behind the proposed technique.

4. Algorithm

Remark 1 suggests that the solution of (12) can be found by solving an elastic net problem with a general error term. To solve it, we prefer the fast iterative shrinkage-thresholding algorithm (FISTA) [30], which is an accelerated gradient descent approach with backtracking. In each step, given an initial $\boldsymbol{\beta}_0 \in \mathbb{R}^p$, we minimize the surrogate function

$$M_{\mathrm{t}}(\boldsymbol{\beta}) = \frac{1}{2\mathrm{t}}\|\boldsymbol{\beta} - \boldsymbol{\beta}_0 + \mathrm{t}\nabla L(\boldsymbol{\beta}_0)\|_2^2 + \lambda_1\|\boldsymbol{\beta}\|_1 + \lambda_2\|\boldsymbol{\beta}\|_2^2, \tag{14}$$

where $\mathrm{t} > 0$ is some step-size (chosen using backtracking).

Proposition 1.

$$U_t(\boldsymbol{\beta}) := \underset{\boldsymbol{\beta}\in\mathbb{R}^p}{\operatorname{argmin}}\{M_t(\boldsymbol{\beta})\} = \underbrace{(1+2t\lambda_2)^{-1}}_{\textit{ridge}}\underbrace{S(\boldsymbol{\beta}_0 - t\nabla L(\boldsymbol{\beta}_0),\, t\lambda_1)}_{\textit{lasso shrinkage}}, \tag{15}$$

where S is the coordinate-wise soft-thresholding operator,

$$S(\mathbf{z},\lambda)_i = \operatorname{sign}(z_i)(|z_i| - \lambda)_+.$$

The proof of Proposition 1 can be found in Appendix A. Proposition 1 suggests a gradient descent procedure to minimize (14). In addition, after each iteration k, we apply the FISTA update, given by

$$\boldsymbol{\beta}_{(k+1)} \leftarrow U_{t_k}(\boldsymbol{\beta}_{(k)}) + \frac{l_k - 1}{l_{k+1}}(U_{t_k}(\boldsymbol{\beta}_{(k)}) - U_{t_{k-1}}(\boldsymbol{\beta}_{(k-1)})), \tag{16}$$

where $l_{k+1} = (1 + \sqrt{1 + 4l_k^2})/2$, $l_1 = 1$.

The choice for the function R in (13) depends on the type of response variable. For instance, if the response is continuous (linear regression) then $\mathcal{R}(\boldsymbol{\beta} \mid \boldsymbol{y}, \mathbf{X}) = \|\boldsymbol{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2$ is probably the best choice. However, if the response is binary (logistic regression), then the logit loss is more appropriate,

$$\mathcal{R}(\boldsymbol{\beta} \mid \boldsymbol{y}, \mathbf{X}) = \sum_{i=1}^{n} \Big(\log(1 + \exp(\mathbf{x}_i^\top \boldsymbol{\beta})) - y_i \mathbf{x}_i^\top \boldsymbol{\beta} \Big) \tag{17}$$

Removing the Shift in the Unlabeled Data

When the direction of the mean shift of the unlabeled data $\mathbf{X}_\mathrm{U}$ with respect to the labeled data $\mathbf{X}_\mathrm{L}$ is in the same direction as $\boldsymbol{\beta}$ (or close), then $\mathbb{E}\boldsymbol{y}_\mathrm{L} \neq \mathbb{E}\boldsymbol{y}_\mathrm{U}$. This, as Larsen et al. noticed, forces the optimal hyper-parameter γ_3 to be zero [22]. One strategy that they propose is to remove the effect of $\boldsymbol{\beta}$ in $\boldsymbol{\mu}$ (which is the mean shift of $\mathbf{X}_\mathrm{U}$ with respect to $\mathbf{X}_\mathrm{L}$) by updating $\mathbf{X}_\mathrm{U}$ with

$$\tilde{\mathbf{X}}_\mathrm{U} = \mathbf{X}_\mathrm{U} - \mathbb{1}\boldsymbol{\mu}^\top \boldsymbol{p}\boldsymbol{p}^\top, \tag{18}$$

where

$$\boldsymbol{p} = \frac{\mathbf{X}_\mathrm{L}^\top \boldsymbol{y}_\mathrm{L}}{\|\mathbf{X}_\mathrm{L}^\top \boldsymbol{y}_\mathrm{L}\|_2}. \tag{19}$$

We instead propose to use

$$\boldsymbol{p} = -\frac{\nabla \mathcal{R}(\mathbf{0} \mid \boldsymbol{y}_\mathrm{L}, \mathbf{X}_\mathrm{L})}{\|\nabla \mathcal{R}(\mathbf{0} \mid \boldsymbol{y}_\mathrm{L}, \mathbf{X}_\mathrm{L})\|_2} \tag{20}$$

thus extending this idea to general loss functions. However, the update in (18) is not necessary (and may introduce unwanted noise) if the angle between $\boldsymbol{\mu}$ and $\boldsymbol{\beta}$ is too big [22]. Figure 3 illustrates update (18) with a two-dimensional example. The unlabeled data $\mathbf{X}_\mathrm{U}$ (blue) are shifted (green) toward the center of $\mathbf{X}_\mathrm{L}$ (red) in the direction of $\nabla \mathcal{R}(\mathbf{0})$ after evaluating if $|\cos(\theta)| < 1/\sqrt{2}$.

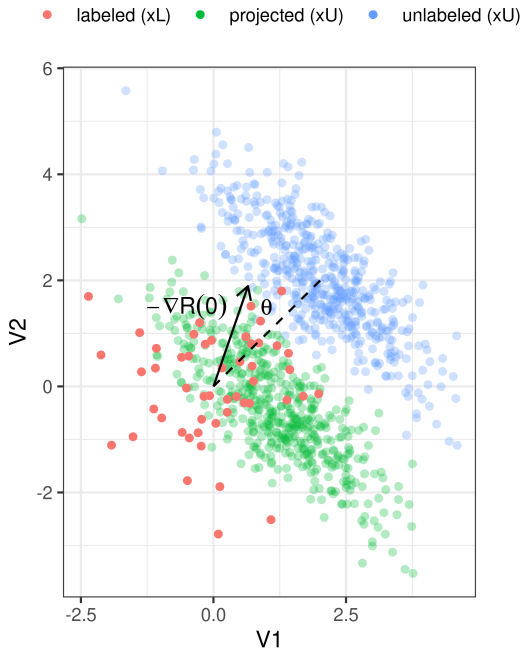

Figure 3. Example update of the unlabeled data in the direction of $-\nabla R(0)$ prior to computing the s^{2}net solution.

5. Simulations

In this section, we will investigate our proposed method s^{2}net as a semi-supervised alternative to the elastic net when the underlying model is linear and sparse.

To introduce the simulations and analysis in the rest of the paper, we make the following assumptions on the problem.

1. There are labeled samples $\mathbf{X}_{\mathrm{L}}^{\mathrm{s}}, \boldsymbol{y}_{\mathrm{L}}^{\mathrm{s}}$ from a source domain (e.g., measurements taken with an old instrument).
2. There are (some) labeled samples $\mathbf{X}_{\mathrm{L}}^{\mathrm{t}}, \boldsymbol{y}_{\mathrm{L}}^{\mathrm{t}}$ from a target domain (e.g., measurements taken with a new instrument or with different raw materials going into the production).
3. There are unlabeled samples $\mathbf{X}_{\mathrm{U}}^{\mathrm{t}}$ from a target domain (e.g., measurements taken with a new instrument, which are very expensive to label).
4. The objective is to construct a model that predicts the labels from the target domain.

Moreover, in a recent article, Oliver et al. established some guidelines for comparing semi-supervised methods [31]. Some of them can be adapted to our framework of study as follows.

- *High-quality supervised baseline*. The goal is to obtain better performance using $\mathbf{X}_{\mathrm{U}}^{\mathrm{t}}$ and $\mathbf{X}_{\mathrm{L}}^{\mathrm{s}}$ than what would be obtained using $\mathbf{X}_{\mathrm{L}}^{\mathrm{s}}$ alone. In our case, a natural baseline to compare against is s^{2}net with $\gamma_1 = 0$ (as mentioned in Remark 1). We denote this supervised method as baseline. In addition, we also include the elastic net (glmnet) from the R package glmnet [27] to compare the naïve estimation of baseline with the actual elastic-net solution. The hyper-parameters of each method were selected using random search, which has been shown to be superior to grid search [32], with a total of 1000 random points. The hyper-parameters that minimized the loss in the validation data set were selected as the best combination.
- *Varying the amount of labeled and unlabeled data.* To cover different scenarios in the simulations, we vary the number of unlabeled target samples n^{t}, in addition to the number of variables p.
- *Realistically small validation dataset*. This is related to the assumption 2 above, which is very important in order to have validation data. Without it, there is no clear and realistic way to select the hyper-parameters of the methods. It is possible to select the hyper-parameters using test data, but this would contradict the fact that in a real semi-supervised scenario, these labels are unknown. To make it feasible, we assume that the number of available samples for validation is small (in the rest of the simulations and data analyses, we fix it at 20).

Additionally, the following semi-supervised methods were included in the simulations: the safe semi-supervised semi-parametric model (s4pm) and fast anchor graph approximation (agraph) from Culp and Ryan [24], available in the R package SemiSupervised, the implicitly constrained semi-supervised least squares classifier (ICLS) [25], available in the R package RSSL, and the joint trained linear framework (JT) from Culp [20].

5.1. Two-Group Design

The simulation design for the first experiment is the following. Let

$$\boldsymbol{\Sigma}_{\rho}^{\sigma^2} = \begin{bmatrix} \sigma^2 & \rho & \dots & \rho \\ \rho & \sigma^2 & \dots & \rho \\ \vdots & \vdots & \ddots & \vdots \\ \rho & \rho & \dots & \sigma^2 \end{bmatrix}_{p/2 \times p/2}, \quad \boldsymbol{\Sigma}_{\rho_1,\rho_2}^{\sigma_1^2,\sigma_2^2} = \begin{bmatrix} \boldsymbol{\Sigma}_{\rho_1}^{\sigma_1^2} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Sigma}_{\rho_2}^{\sigma_2^2} \end{bmatrix}_{p \times p}.$$

The source and target data rows are independent and identically distributed, given by

$$\mathbf{x}^{\mathrm{s}} \sim \mathrm{N}\left(\mathbf{0}, \boldsymbol{\Sigma}_{.8,.01}^{1,.05}\right), \quad \mathbf{x}^{\mathrm{t}} \sim \mathrm{N}\left(\mathbf{0}, \boldsymbol{\Sigma}_{.01,.5}^{.1,1}\right). \tag{21}$$

Figure 4 illustrates this simulation design using an example data set, with $p = 200$ variables, and 50 source and 200 target observations, respectively.

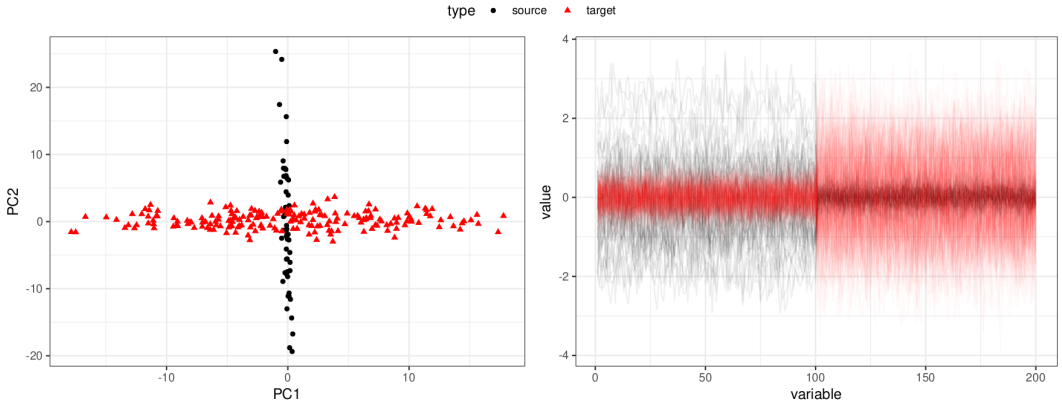

Figure 4. Example of simulated source/target data structure. Left panel shows the projected data on the first two principal components. Right panel compares the rows of $\mathbf{X}^{\mathrm{s}}$ (black) and $\mathbf{X}^{\mathrm{t}}$ (red).

To generate the responses for the source data $\mathbf{X}^{\mathrm{s}}$, we used a sparse coefficient vector, given by

$$\boldsymbol{\beta}_j = \begin{cases} 0 & j \notin I \\ 1 & j \in I \end{cases},$$

where I is the included variables' index set that contains 5 random indexes between 1 and $p/2 - 1$ and 5 random indexes between $p/2$ and p. Therefore, there are 10 out of p "true" variables in the model. The target model's coefficients, however, are given by

$$\boldsymbol{\beta}_j^t = \mathrm{U}_j \boldsymbol{\beta}_j, \text{ where } \mathrm{U}_j \sim \mathrm{U}[0.9, 1.1] \text{ for } j \in I. \tag{22}$$

This introduces additional uncertainty in the target data, and models the case of a small change in the underlying coefficient vector for the new data.

The training set consists of labeled source data $\mathbf{X}^{\mathrm{s}}_{\mathrm{train}}, \boldsymbol{y}^{\mathrm{s}}_{\mathrm{train}}$ ($\mathrm{n}^{\mathrm{s}} = 50$ rows) and unlabeled target data $\mathbf{X}^{\mathrm{t}}_{\mathrm{train}}$ (n^{t} rows), whereas the validation set consists of labeled target samples $\mathbf{X}^{\mathrm{t}}_{\mathrm{valid}}, \boldsymbol{y}^{\mathrm{t}}_{\mathrm{valid}}$ (20 rows). A test data set $\mathbf{X}^{\mathrm{t}}_{\mathrm{test}}, \boldsymbol{y}^{\mathrm{t}}_{\mathrm{test}}$ (800 rows) is used to evaluate the performance of both methods for each of the 100 repetitions. The set-up for this and the following experiments is given in Algorithm 1.

Algorithm 1 Experimental set-up.

Let $\mathcal{C}$ be a given classifier.
Let $\mathcal{H}$ be the set that contains the different hyperparameters combinations.
Let n_{rep} be the number of repeated validations.
for i **do** in 1:n_{rep}
Obtain $\boldsymbol{X}^{\mathrm{s}}_{\mathrm{train}}, \boldsymbol{y}^{\mathrm{s}}_{\mathrm{train}}, \boldsymbol{X}^{\mathrm{t}}_{\mathrm{train}}, \boldsymbol{X}^{\mathrm{t}}_{\mathrm{valid}}, \boldsymbol{y}^{\mathrm{t}}_{\mathrm{valid}}, \boldsymbol{X}^{\mathrm{t}}_{\mathrm{test}}, \boldsymbol{y}^{\mathrm{t}}_{\mathrm{test}}$.
Find h *in* H that maximizes a performance metric in $\boldsymbol{X}^{\mathrm{t}}_{\mathrm{valid}}, \boldsymbol{y}^{\mathrm{t}}_{\mathrm{valid}}$ using $\boldsymbol{X}^{\mathrm{s}}_{\mathrm{train}}, \boldsymbol{y}^{\mathrm{s}}_{\mathrm{train}}, \boldsymbol{X}^{\mathrm{t}}_{\mathrm{train}}$
Get the performance measure $P(i)$ of $\mathcal{C}$ in $\boldsymbol{X}^{\mathrm{t}}_{\mathrm{test}}, \boldsymbol{y}^{\mathrm{t}}_{\mathrm{test}}$ using h
end for
Return the performance measures Ps

Logistic response

For the classification case, to simulate the source data labels $\boldsymbol{y}^{\mathrm{s}}$, we used a logistic model,

$$y^{\mathrm{s}} \mid \mathbf{x}^{\mathrm{s}} \sim \mathrm{Ber}(p), \text{ with } p = \left(1 + \exp(-\boldsymbol{\beta}^{\top} \mathbf{x}^{\mathrm{s}})\right)^{-1}. \tag{23}$$

The target labels $\boldsymbol{y}^{\mathrm{t}}$ were generated analogously, but using $\boldsymbol{\beta}^{\mathrm{t}}$ instead—the noisy version of $\boldsymbol{\beta}$ given in (22).

Tables 1 and 2 summarize the simulation results for linear and logistic responses, respectively. To evaluate the statistical significance of the difference between each method and baseline, we performed a Friedman rank test, followed by paired post-hoc tests [33]. Significant improvements ($\alpha = 0.05$) with respect to baseline are highlighted with an asterisk. In these simulations, s^{2}net achieves the best result in every scenario.

Table 1. Average test mean squared error (MSE) of the different methods (two-group design, linear response), over 100 simulations for each scenario. Significant improvements ($\alpha = 0.05$) with respect to the baseline are indicated by an asterisk.

	$n^t = 50$			$n^t = 250$		
	$p = 50$	$p = 100$	$p = 200$	$p = 50$	$p = 100$	$p = 200$
baseline	0.59	0.58	0.69	0.56	0.53	0.64
glmnet	0.61	0.60	0.71	0.58	0.56	0.66
s^{2}net	0.55 *	0.54 *	0.65 *	0.53 *	0.51 *	0.62 *
s4pm	0.71	0.71	0.75	0.64	0.57	0.65
agraph	0.86	0.88	0.99	0.77	0.76	0.91
JT	0.62	0.61	0.72	0.56	0.53 *	0.63 *

Table 2. Average test area under the receiver operating characteristic curve (AUC, %) of the different methods (two-group design, logistic response), over 100 simulations for each scenario. Significant improvements ($\alpha = 0.05$) with respect to the baseline are indicated by an asterisk.

	$n^t = 50$			$n^t = 250$		
	$p = 50$	$p = 100$	$p = 200$	$p = 50$	$p = 100$	$p = 200$
baseline	75.3	70.2	78.4	74.8	73.7	72.1
glmnet	75.9	71.8	78.3	73.6	74.9	71.7
s^{2}net	79.4 *	73.8 *	79.4 *	78.6 *	75.8 *	76.6 *
s4pm	71.1	68.5	77.0	75.0 *	74.8 *	75.8 *
agraph	68.7	65.3	73.5	68.8	67.0	70.8
ICLS	60.4	54.2	57.6	60.4	55.8	53.6

5.2. Extrapolation Design

This simulation design is based on the one described by Ryan and Culp [21], but we varied the number of variables and unlabeled target samples, the shift, and included the logistic response case. The source data are simulated with independent and identically distributed rows given by

$$\mathbf{x}^{\mathrm{s}} \sim \mathrm{N}(\mathbf{0}, 0.4\mathbb{I}) \tag{24}$$

Two possible coefficient patterns are considered:

$$\boldsymbol{\beta}^{(\mathrm{lucky})} = (\underbrace{1 \ldots 1}_{5}\ \underbrace{-1 \ldots -1}_{5}\ \underbrace{0 \ldots 0}_{p-10}) \quad \text{and} \quad \boldsymbol{\beta}^{(\mathrm{unlucky})} = (\underbrace{1 \ldots 1}_{10}\ \underbrace{0 \ldots 0}_{p-10}) \tag{25}$$

There are three scenarios for the target data:

Same: $\mathbf{x}^{\mathrm{t}} \sim N(\mathbf{0}, 0.4\mathbb{I})$ and $\boldsymbol{\beta} = 5/\sqrt{10}\boldsymbol{\beta}^{(\mathrm{lucky})}$

Lucky: $\mathbf{x}^{\mathrm{t}} \sim N(\delta\boldsymbol{\beta}^{(\mathrm{unlucky})}, 0.4\mathbb{I})$, and $\boldsymbol{\beta} = 5/\sqrt{10}\boldsymbol{\beta}^{(\mathrm{lucky})}$

Unlucky: $\mathbf{x}^{\mathrm{t}} \sim N(\delta\boldsymbol{\beta}^{(\mathrm{unlucky})}, 0.4\mathbb{I})$, and $\boldsymbol{\beta} = 5/\sqrt{10}\boldsymbol{\beta}^{(\mathrm{unlucky})}$

With δ as the shift of the target with respect to the source domain, Figure 5 displays the three possible configurations for the data, projected in X_1 and X_6. In the "same" scenario, the source and target data follow the same distribution, and thus the direction of $\boldsymbol{\beta}$ is not important. In the "lucky" case, $\boldsymbol{\beta}$ is orthogonal to the shift (the source and target domains

are different, but the response is less affected by the shift). In the "unlucky" case, however, $\boldsymbol{\beta}$ is parallel to the shift, and thus we expect the responses to be shifted as well. This "unlucky" scenario is more challenging, especially in the linear response case, where the bias in the estimation of $\boldsymbol{\beta}$ will impact the extrapolation.

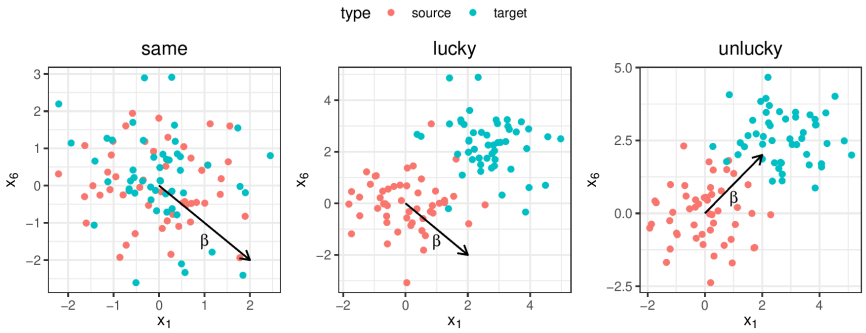

Figure 5. Simulated source/target data structure: extrapolation design.

For each repetition, the training data consist of $n^s = 50$ rows of labeled $\mathbf{X}^s_{\text{train}}$, $\boldsymbol{y}^s_{\text{train}}$, and varying n^t rows of unlabeled target data $\mathbf{X}^t_{\text{train}}$. The validation and test sets consist of 20 and 100 observations, respectively, from the target domain.

Linear response

The labels (for the source and target data, respectively) are simulated as $\boldsymbol{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$, with $\epsilon_i \sim N(0, 2.5)$, for $i = 1, 2 \ldots n$. The number of features $p = 100$ and the shift $\delta = 1$.

Logistic response

The labels (source and target) are generated following a logistic response model,

$$y \mid \mathbf{x} \sim \text{Ber}(p), \text{ with } p = \left(1 + \exp(-\boldsymbol{\beta}^\top \mathbf{x})\right)^{-1}. \tag{26}$$

The number of features $p = 20$ and the shift $\delta = 0.1$.

Tables 3 and 4 compare the simulations for linear and logistic responses, respectively. Table 4 displays better performance for baseline, and s^{2}net, suggesting that there is improvement when choosing the semi-supervised elastic net framework. However, in the "unlucky" scenario of Table 3 (where the shift δ is in a direction parallel to the response direction of the labeled data), glmnet outperforms the other alternatives by a weak margin. The implementation of JT estimates the coefficients using glmnet, so they are expected to yield similar estimations when the supervised model prevails. However, glmnet and baseline are (in theory) solving the same optimization problem. We believe such differences are due to the way coefficients are actually estimated: baseline uses a block gradient descent optimization with soft-threshold, whereas glmnet is optimized using coordinate-gradient descent, with rules to discard predictors [28], and a correction factor in the $\boldsymbol{\beta}$ estimations. A detailed description of the differences between the naive and the elastic-net solution can be found in the work of Bühlmann and Van De Geer [34].

Table 3. Average test MSE of the different methods (extrapolation design, linear response), over 100 simulations for each scenario. Significant improvements ($\alpha = 0.05$) with respect to baseline are highlighted with an asterisk.

	"Same"		**"Lucky"**		**"Unlucky"**	
	$n^t = 50$	$n^t = 250$	$n^t = 50$	$n^t = 250$	$n^t = 50$	$n^t = 250$
baseline	5.58	5.71	5.85	5.74	61.6	48.0
glmnet	5.66	5.82	6.03	5.97	56.5 *	46.1 *
s^{2}net	5.56 *	5.70	5.75 *	5.73	62.1	48.1
s4pm	6.23	6.21	5.76 *	5.81	120	86.7
agraph	6.21	6.39	6.09	6.06	56.6 *	71.6
JT	5.79	5.74	5.58 *	5.69 *	59.1 *	47.7 *

Table 4. Average test area under the ROC curve (AUC, %) of the different methods (extrapolation design and logistic response), over 100 simulations for each scenario. Significant improvements ($\alpha = 0.05$) with respect to baseline are highlighted with an asterisk.

	"Same"		**"Lucky"**		**"Unlucky"**	
	$n^t = 50$	$n^t = 250$	$n^t = 50$	$n^t = 250$	$n^t = 50$	$n^t = 250$
baseline	74.7	74.9	76.2	74.0	77.5	75.5
glmnet	74.7 *	75.1	76.2	74.0	77.3	75.5
s^{2}net	76.3 *	74.9	76.3 *	74.1 *	77.5	75.6 *
s4pm	74.2	74.4	74.6	74.1	73.6	74.2
agraph	74.4	73.0	74.3	72.8	75.7	72.9
ICLS	69.0	68.1	68.3	68.1	68.2	67.0

6. Application to Real Data

The purpose of this section is to evaluate the performance of s^{2}net on real data-based examples, and compare it with glmnet, s4pm, agraph, JT, ICLS, and the baseline (s^{2}net with $\gamma_1 = 0$) in regression and classification tasks. An overview of the datasets used in this section is given in Table 5.

Table 5. Description of the data used in the analysis.

Dataset	**Labeled n^s (Train)**	**Unlabeled n^t (Train)**	**Regression**	**Classification**	p
shootout	50	50	✓		575
auto-mpg (P1)	149	100	✓		9
auto-mpg (P2)	208	100	✓		7
spambase	100	500		✓	52

6.1. IDRC 2002 "Shootout" Data

This data set was published in the International Diffuse Reflectance Conference in 2002, and it is currently available online (http://eigenvector.com/data/tablets, last access on 21 October 2019). It consists of the spectra from 655 pharmaceutical tablets measured with two spectrometers. The response variable is the proportion of the active ingredient. As shown in Figure 6, there are differences in both instruments' measures ranging from 0.6 to 0.7 µm and 1.7 to 1.8 µm.

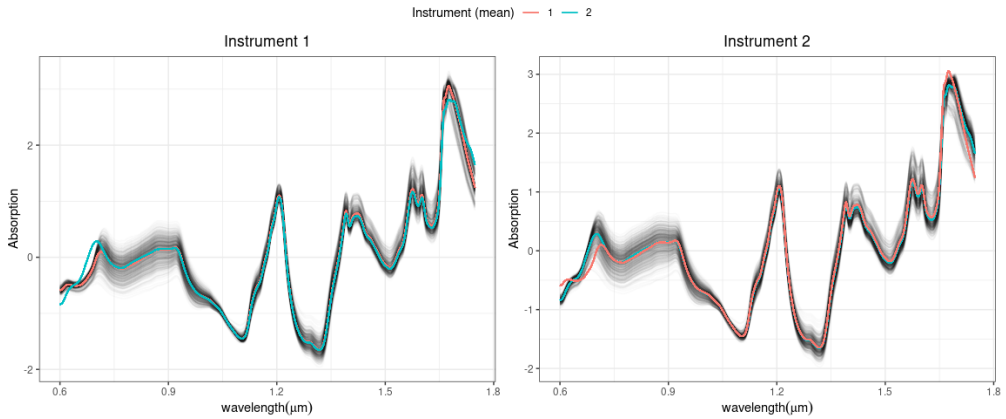

Figure 6. Spectra from 655 tablets (IDRC 2002 "Shootout" data) measured with two different instruments (**left–right**).

To illustrate the s^{2}net methodology, we assume that labels associated with measures from Instrument 1 are known, and we investigate how predictions are affected when labels are predicted using measures from Instrument 2. For this purpose, the original data are randomly divided up into training, validation and test data sets, and this process is repeated

100 times. A total of 50 tablets are used as training labeled samples from Instrument 1 (source), whereas 50 measures from Instrument 2 (target) are used as training unlabeled samples. To select the best hyper-parameters for the methods, we separated a sample of 20 labeled measurements from Instrument 2 (target). The remaining tablets (unknown during the training process) are used as test samples from Instrument 2, in addition to the (already known) 50 measures used as training unlabeled samples. The response variable in the test data is used to compute prediction errors.

Figure 7 compares the distributions of the MSE obtained by the different algorithms in the test data set, for 100 repetitions. Notice that s^{2}net is the one that achieves the smallest error mean and variance, but all the methods are very similar.

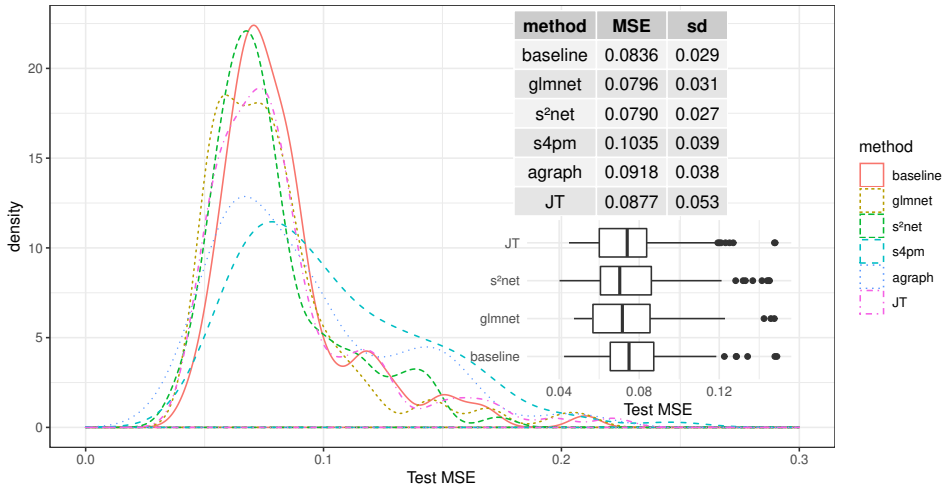

Figure 7. Density estimation of the (test) MSE of each method for 100 repetitions (*shootout* data).

6.2. Auto MPG Dataset

This data set is available in the UCI repositories [35], and the original data were published by Quinlan [36]. We processed these data for the semi-supervised setting following the paper by Ryan and Culp [21]. The first set-up (P1) separates source and target domains by variable `Domestic`, whereas the second set-up (P2) splits the data by variable `Cylinder <= 4`.

Figures 8 and 9 display the results for 100 repetitions (varying the validation and training target samples). As indicated by the distribution of the test error, and its mean in Figure 8, s^{2}net clearly outperforms the other methods in the *auto-mpg (P1)* data. However, for the *auto-mpg (P2)* setting, the supervised glmnet is the one minimizing the test error.

6.3. Spambase Data

This data set was collected by Hewlett-Packard Labs, and it is available at the UCI Repository of Machine Learning Databases [35]. It classifies 4601 e-mails as *spam* or *non-spam*. There are 57 explanatory variables indicating the frequency of certain words and characters in the e-mail. This data set was also studied by Kawakita and Kanamori [37] in a semi-supervised context. To adapt it to our semi-supervised set-up, we split the data according to variable `Internet` (e-mails from the source domain containing the word internet in the body of the message). This partition yields different balances of the response variable in the source and target domains, which suggests an additional complexity for the prediction.

Figure 10 displays the empirical distribution of the accuracy in the test set for the *spambase* data. We notice that s^{2}net outperforms glmnet by a margin close to 10%. However—and this is why it is important to have a baseline method to compare—the supervised version of s^{2}net performs very similarly (slightly better). In this case, there is no advantage

in using the unlabeled data, but the optimization method itself that computes the coefficient estimations for s^{2}net and baseline shows good performance.

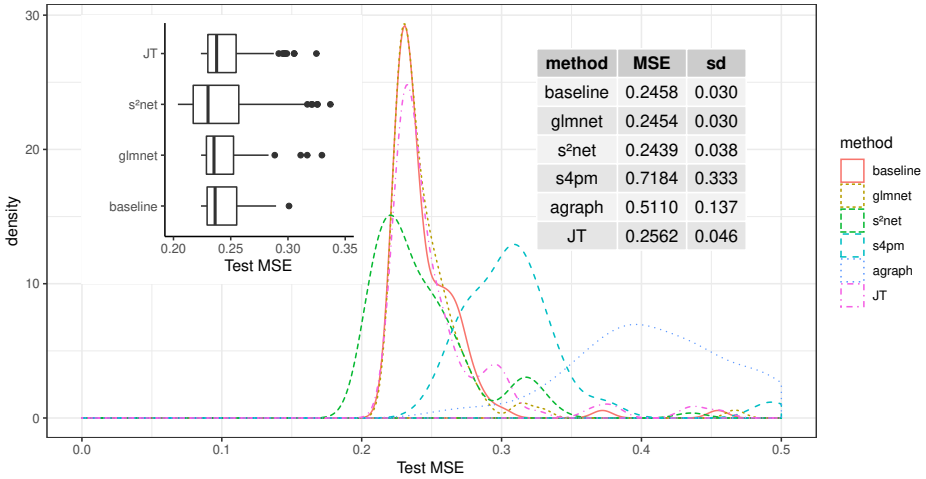

Figure 8. Density estimation of the (test) MSE of each method for 100 repetitions (*auto-mpg-P1* data).

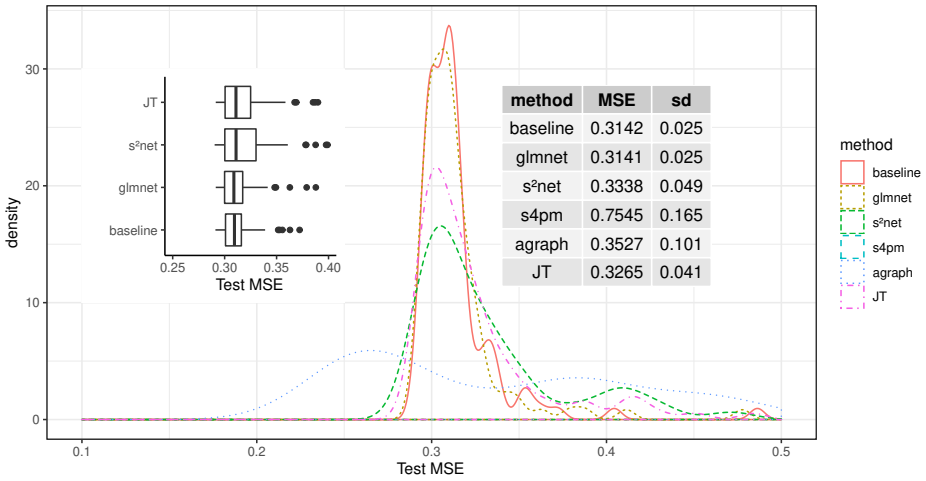

Figure 9. Density estimation of the (test) MSE of each method for 100 repetitions (*auto-mpg-P2* data).

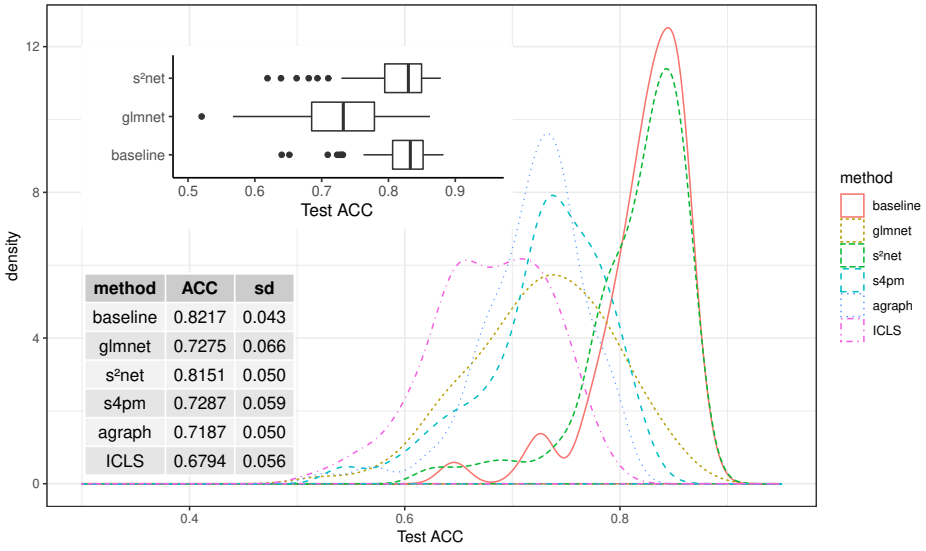

Figure 10. Density estimation of the (test) accuracy of each method for 100 repetitions (*spambase* data).

7. Conclusions, Limitations, and Future Research

In this article, we introduced s^{2}net, a semi-supervised elastic net for generalized linear models. It is shown that s^{2}net generalizes the semi-supervised techniques JT and ExtJT. It is also exhibited that, if the unlabeled information is not relevant, s^{2}net chooses the hyper-parameters so that it adopts the traditional supervised elastic network. Our method was tested using both real and synthetic data sets, and the experiments confirmed our approach as a good alternative to the elastic net in the semi-supervised context.

We introduced a general optimization framework that implements the FISTA algorithm to solve the elastic-net for a generic loss function. As feature works, the implementation can be easily adapted to solve other extensions of lasso, such as the group-lasso and the sparse-group lasso. In addition, we observed a relative improvement of using gradient-descent to optimize (12) with respect to coordinate-descent, demonstrated by the fact that our elastic net baseline sometimes outperforms glmnet (Tables 1–3, and Figure 10). A limitation of the proposed s^{2}net is that it is not suitable for moderate/large databases. The fact that it requires computing the singular value decomposition of the centered unlabeled data $\mathbf{X}_U$ makes it impossible to be applied to huge/big data sets. Adapting the proposed technique to the latter databases is currently another possibility for future work. Another limitation is that the proposed technique has its focus in the context of the generalized linear model. Extension of the technique to nonlinear models will be considered in the future.

The simulation design studied in Section 5.1 highlighted a scenario where s^{2}net clearly outperforms all the other methods. The increased performance might be a consequence of the fact that the underlying model's coefficient is different for the source and target domains. Since s^{2}net uses the information in the unlabeled data (in contrast to the elastic net), it can learn that change and adapt. Compared to other semi-supervised methods, s^{2}net has the advantage of separating the shift from the covariance information, which adds flexibility to the model. Additionally, s^{2}net brings desirable properties of elastic net to the semi-supervised framework, such as the sparsity in the solution.

To conclude, the excellent results obtained, together with the fact that the developed software is freely offered to the scientific community, make it possible to solve several current problems in various research areas. Among these are the social and behavioral sciences. In these areas, researchers do not usually have the training to develop mathematical software, but they have a multitude of open problems. As an example, and based on our previous experience, the developed technique could be used to identify suicidal behavior in one country from data obtained in another.

8. Computational Details

All the experiments in Sections 5 and 6 were conducted in the same HPC cluster (www.hpc.dtu.dk, accessed on 1 June 2022), specifically 8 nodes with Intel(R) Xeon(R) CPUs E5-2680 v2, 128G RAM, running Linux 3.10.0 and R (3.6.1—platform x86_64-conda_cos6-linux-gnu (64-bit)—Anaconda Inc. (Austin, TX, USA)).

To select the hyper-parameters of all the methods, we used random search with 1000 iterations. For s^{2}net and baseline, we took $\lambda_1, \lambda_2 \sim 2^{U[-8,1]}$, and $\gamma_1, \gamma_3 \sim 2^{U[-8,1]}, \gamma_2 \sim 2^{U[-1,10]}$ (s^{2}net). For glmnet and JT, $\alpha \sim U[0,1]$, $\lambda \sim 2^{U[-8,1]}$, and $\gamma_1(\tau) \sim 2^{U[-8,1]}, \gamma_2(\gamma) \sim 2^{U[-1,10]}$ (JT). For s4pm and agraph, $lams, gams, hs \sim 2^{U[-8,1]}$, and for ICLS, $\lambda_1, \lambda_2 \in 2^{U[-8,1]}$. The code for the simulations and data analyses is available online (https://github.com/jlaria/s2net-paper, accessed on 1 June 2022), and the implementation of s^{2}net is available in CRAN (https://cran.r-project.org/package=s2net, accessed on 1 June 2022).

9. Code Availability

Code is available at https://github.com/jlaria/s2net (accessed on 1 June 2022).

Author Contributions: Conceptualization, J.C.L., L.H.C.; Methodology, J.C.L., L.H.C.; Software, J.C.L., L.H.C.; Resources, D.D.-G., B.K.E.; Writing, D.D.-G., B.K.E., L.H.C., J.C.L. All authors have read and agreed to the published version of the manuscript.

Funding: We acknowledge Ministerio de Ciencia e Innovación (Grant No. RTI2018-101857-B-432 I00),Ministerio de Universidades (Grant for the requalification of permanent lectures, David Delgado-433 Gómez), and Instituto Salud Carlos III (Grant No. AES2021, DTS21/00091). We also acknowledge the Action financed by the Community of Madrid within the framework of the multi-year agreement with Universidad Carlos III Madrid in its line of action "Excellence for University Faculty". Established within the framework of the V Regional Plan for Scientific Research and Technological Innovation 2016–2020.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: All data sets are publicly available and are available from the s^{2}net package.

Acknowledgments: We gratefully acknowledge the help provided by Mark Culp, who gave us access to the source code of the methods JT, s4pm and agraph, compared in our simulations and data analyses.

Conflicts of Interest: The authors declare that they have no competing interest.

Abbreviations

The following abbreviations are used in this manuscript:

DNA	Deoxyribonucleic acid
JT	Joint trained elastic net
ExtJT	Extended linear joint trained framework
FISTA	Fast iterative shrinkage-thresholding algorithm
MSE	Average test mean squared error
AUC	Area under the receiver operating characteristic curve

Appendix A. Proof of Proposition 1

Let $\boldsymbol{\beta}$ be the minimizer of $M_t(\boldsymbol{\beta})$, and let $\boldsymbol{B_0} = \boldsymbol{\beta}_0 - t\nabla L(\boldsymbol{\beta}_0)$. By the subgradient conditions, $\partial_j M_t(\boldsymbol{\beta}) \ni 0$, and notice that

- $$\partial_j\left(\frac{1}{2}\|\boldsymbol{\beta} - \boldsymbol{B_0}\|_2^2\right) = \beta_j - (\boldsymbol{B_0})_j.$$
- $$\partial_j\|\boldsymbol{\beta}\|_1 = v_j = \begin{cases} \text{sign}(\beta_j), & \beta_j \neq 0 \\ \in [0,1], & \beta_j = 0 \end{cases}$$
- $$\partial_j\|\boldsymbol{\beta}\|_2^2 = 2\beta_j$$

Then,

$$0 = \partial_j M_t(\boldsymbol{\beta}) = \beta_j - (\boldsymbol{B_0})_j + t\lambda_1 v_j + 2t\lambda_2\beta_j = \beta_j(1 + 2t\lambda_2) - (\boldsymbol{B_0})_j + t\lambda_1 v_j.$$

Separating by cases,

- Case $\beta_j > 0$ $(v_j = 1)$

$$\begin{aligned} 0 &= \beta_j(1 + 2t\lambda_2) - (\boldsymbol{B_0})_j + t\lambda_1 \\ \beta_j &= (1 + 2t\lambda_2)^{-1}[(\boldsymbol{B_0})_j - t\lambda_1] \Leftrightarrow (\boldsymbol{B_0})_j > t\lambda_1. \end{aligned}$$

- Case $\beta_j < 0$ $(v_j = -1)$

$$\begin{aligned} 0 &= \beta_j(1 + 2t\lambda_2) - (\boldsymbol{B_0})_j - t\lambda_1 \\ \beta_j &= (1 + 2t\lambda_2)^{-1}[(\boldsymbol{B_0})_j + t\lambda_1] \Leftrightarrow (\boldsymbol{B_0})_j < -t\lambda_1. \end{aligned}$$

- Case $\beta_j = 0$ ($|v_j| \leq 1$)

$$0 = -(\boldsymbol{B_0})_j + t\lambda_1 v_j \Leftrightarrow |(\boldsymbol{B_0})_j| < t\lambda_1.$$

Putting the three cases together and taking $j = 1, 2 \ldots p$, the result follows.

$$\boldsymbol{\beta} = (1 + 2t\lambda_2)^{-1} S(\boldsymbol{B_0}, t\lambda_1).$$

References

1. Cheplygina, V.; de Bruijne, M.; Pluim, J.P. Not-so-supervised: A survey of semi-supervised, multi-instance, and transfer learning in medical image analysis. *Med. Image Anal.* **2019**, *54*, 280–296. [CrossRef] [PubMed]
2. Zhu, X.; Goldberg, A.B. Introduction to semi-supervised learning. *Synth. Lect. Artif. Intell. Mach. Learn.* **2009**, *3*, 1–130.
3. Liu, C.L.; Hsaio, W.H.; Lee, C.H.; Chang, T.H.; Kuo, T.H. Semi-supervised text classification with universum learning. *IEEE Trans. Cybern.* **2015**, *46*, 462–473. [CrossRef] [PubMed]
4. Wu, H.; Prasad, S. Semi-supervised deep learning using pseudo labels for hyperspectral image classification. *IEEE Trans. Image Process.* **2017**, *27*, 1259–1270. [CrossRef]
5. Kuo, C.F.J.; Liu, S.C. Fully Automatic Segmentation, Identification and Preoperative Planning for Nasal Surgery of Sinuses Using Semi-Supervised Learning and Volumetric Reconstruction. *Mathematics* **2022**, *10*, 1189. [CrossRef]
6. Zhang, X.; Zhang, X.; Xiao, Y.; Liu, G. Theme-Aware Semi-Supervised Image Aesthetic Quality Assessment. *Mathematics* **2022**, *10*, 2609. [CrossRef]
7. Rozza, A.; Manzo, M.; Petrosino, A. A novel graph-based fisher kernel method for semi-supervised learning. In Proceedings of the 2014 22nd International Conference on Pattern Recognition, Stockholm, Sweden, 24–28 August 2014; pp. 3786–3791.
8. Zheng, H.; Zhou, Y.; Huang, X. Improving Cancer Metastasis Detection via Effective Contrastive Learning. *Mathematics* **2022**, *10*, 2404. [CrossRef]
9. Hussain, A.; Cambria, E. Semi-supervised learning for big social data analysis. *Neurocomputing* **2018**, *275*, 1662–1673. [CrossRef]
10. Zhu, X.J. Semi-Supervised Learning Literature Survey. 2005. Available online: https://minds.wisconsin.edu/handle/1793/60444 (accessed on 1 June 2022).
11. Van Engelen, J.E.; Hoos, H.H. A survey on semi-supervised learning. *Mach. Learn.* **2020**, *109*, 373–440. [CrossRef]
12. Tanha, J.; van Someren, M.; Afsarmanesh, H. Semi-supervised self-training for decision tree classifiers. *Int. J. Mach. Learn. Cybern.* **2017**, *8*, 355–370. [CrossRef]
13. Ding, S.; Zhu, Z.; Zhang, X. An overview on semi-supervised support vector machine. *Neural Comput. Appl.* **2017**, *28*, 969–978. [CrossRef]
14. Kipf, T.N.; Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv* **2016**, arXiv:1609.02907.
15. Nie, F.; Xiang, S.; Jia, Y.; Zhang, C. Semi-supervised orthogonal discriminant analysis via label propagation. *Pattern Recognit.* **2009**, *42*, 2615–2627. [CrossRef]
16. Kostopoulos, G.; Karlos, S.; Kotsiantis, S.; Ragos, O. Semi-supervised regression: A recent review. *J. Intell. Fuzzy Syst.* **2018**, *35*, 1483–1500. [CrossRef]
17. Zou, H.; Hastie, T. Regularization and variable selection via the elastic net. *J. R. Stat. Soc. Ser. B (Stat. Methodol.)* **2005**, *67*, 301–320. [CrossRef]
18. Araveeporn, A. The Higher-Order of Adaptive Lasso and Elastic Net Methods for Classification on High Dimensional Data. *Mathematics* **2021**, *9*, 1091. [CrossRef]
19. Cubilla-Montilla, M.; Nieto-Librero, A.B.; Galindo-Villardón, M.P.; Torres-Cubilla, C.A. Sparse HJ biplot: A new methodology via elastic net. *Mathematics* **2021**, *9*, 1298. [CrossRef]
20. Culp, M. On the Semisupervised Joint Trained Elastic Net. *J. Comput. Graph. Stat.* **2013**, *22*, 300–318. [CrossRef]
21. Ryan, K.J.; Culp, M.V. On semi-supervised linear regression in covariate shift problems. *J. Mach. Learn. Res.* **2015**, *16*, 3183–3217.
22. Larsen, J.S.; Clemmensen, L.; Stockmarr, A.; Skov, T.; Larsen, A.; Ersbøll, B.K. Semi-supervised covariate shift modelling of spectroscopic data. *J. Chemom.* **2020**, *34*, e3204. [CrossRef]
23. Amini, M.R.; Gallinari, P. Semi-supervised logistic regression. *ECAI* **2002**, *2*, 390–394.
24. Culp, M.V.; Ryan, K.J. SemiSupervised: Scalable Semi-Supervised Routines for Real Data Problems. 2018. Available online: https://rdrr.io/cran/SemiSupervised/f/inst/doc/SemiSupervised.pdf (accessed on 1 June 2022).
25. Krijthe, J.H.; Loog, M. Implicitly constrained semi-supervised least squares classification. In *Advances in Intelligent Data Analysis XIV. IDA 2015*; Lecture Notes in Computer Science; De Bie, T., van Leeuwen, M., Eds.; Springer: Cham, Switzerland, 2015; Volume 9385, pp. 158–169._14. [CrossRef]
26. Friedman, J.; Hastie, T.; Tibshirani, R. A note on the group lasso and a sparse group lasso. *arXiv* **2010**, arXiv:1001.0736.
27. Friedman, J.; Hastie, T.; Tibshirani, R. Regularization paths for generalized linear models via coordinate descent. *J. Stat. Softw.* **2010**, *33*, 1. [CrossRef]
28. Tibshirani, R.; Bien, J.; Friedman, J.; Hastie, T.; Simon, N.; Taylor, J.; Tibshirani, R.J. Strong rules for discarding predictors in lasso-type problems. *J. R. Stat. Soc. Ser. B (Stat. Methodol.)* **2012**, *74*, 245–266. [CrossRef]

29. Zou, H.; Hastie, T. Regression shrinkage and selection via the elastic net, with applications to microarrays. *J. R. Stat. Soc. Ser. B* **2003**, *67*, 301–320. [CrossRef]
30. Beck, A.; Teboulle, M. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imaging Sci.* **2009**, *2*, 183–202. [CrossRef]
31. Oliver, A.; Odena, A.; Raffel, C.A.; Cubuk, E.D.; Goodfellow, I. Realistic evaluation of deep semi-supervised learning algorithms. *Adv. Neural Inf. Process. Syst.* **2018**, *31*, 3235–3246.
32. Bergstra, J.; Bengio, Y. Random search for hyper-parameter optimization. *J. Mach. Learn. Res.* **2012**, *13*, 281–305.
33. Pohlert, T. PMCMRplus: Calculate Pairwise Multiple Comparisons of Mean Rank Sums Extended. R Package Version 1.4.2. 2019. Available online: https://cran.r-project.org/web/packages/PMCMRplus/index.html (accessed on 1 June 2022).
34. Bühlmann, P.; Van De Geer, S. *Statistics for High-Dimensional Data: Methods, Theory and Applications*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2011.
35. Dua, D.; Graff, C. *UCI Machine Learning Repository*; University of California, School of Information and Computer Science: Irvine, CA, USA, 2019.
36. Quinlan, J.R. Combining instance-based and model-based learning. In Proceedings of the Tenth International Conference on Machine Learning, Amherst, MA, USA, 27–29 July 1993; pp. 236–243.
37. Kawakita, M.; Kanamori, T. Semi-supervised learning with density-ratio estimation. *Mach. Learn.* **2013**, *91*, 189–209. [CrossRef]