Data and text mining

ORIGINAL PAPER

Estimating sparse regression models in multi-task learning and transfer learning through adaptive penalisation

Armin Rauschenberger[1,2,*] Petr N. Nazarov[1,†] and Enrico Glaab[2,†]

[1] Bioinformatics and Artificial Intelligence, Department of Medical Informatics, Luxembourg Institute of Health (LIH), 1 A-B Rue Thomas Edison, 1445 Strassen, Luxembourg and [2] Biomedical Data Science, Luxembourg Centre for Systems Biomedicine (LCSB), University of Luxembourg, 7 Avenue des Hauts-Fourneaux, 4362 Esch-sur-Alzette, Luxembourg

[*] Corresponding author. armin.rauschenberger@lih.lu

[†] Petr N. Nazarov and Enrico Glaab share senior authorship.

FOR PUBLISHER ONLY Received on 6 January 2025; revised on 6 June 2025; accepted on X Month 2025

Abstract

Method: Here we propose a simple two-stage procedure for sharing information between related high-dimensional prediction or classification problems. In both stages, we perform sparse regression separately for each problem. While this is done without prior information in the first stage, we use the coefficients from the first stage as prior information for the second stage. Specifically, we designed feature-specific and sign-specific adaptive weights to share information on feature selection, effect directions and effect sizes between different problems.
Results: The proposed approach is applicable to multi-task learning as well as transfer learning. It provides sparse models (i.e., with few non-zero coefficients for each problem) that are easy to interpret. We show by simulation and application that it tends to select fewer features while achieving a similar predictive performance as compared to available methods.
Availability: An implementation is available in the R package “sparselink” (`https://github.com/rauschenberger/sparselink`, `https://cran.r-project.org/package=sparselink`).

Key words: multi-task learning, transfer learning, sparse regression, feature selection, adaptive penalisation

Background

Here we are concerned with related high-dimensional regression problems, for numerical prediction or binary classification. In high-dimensional settings, where the number of features is much larger than the sample size, it can be difficult to estimate accurate predictive models. If it is not possible to increase the sample size available for model training, it might be beneficial to share information among related problems to increase the predictive performance. In this work, we use the term *multi-task learning* to mean “solving related problems about the same features and the same samples” and the term *transfer learning* to mean “solving related problems about the same features but different samples”.

To illustrate the difference between these definitions of multi-task and transfer learning, we consider two sets, each consisting of two related regression problems. For simplicity, we assume the targets are numerical and centred (i.e., linear regression with zero intercept). Each target is then modelled as a linear combination of the features plus an error term. In the case of multi-task learning, we have

$$\begin{pmatrix} y_{1,1} & y_{1,2} \\ \vdots & \vdots \\ y_{n,1} & y_{n,2} \end{pmatrix} = \begin{pmatrix} x_{1,1} & \cdots & x_{1,p} \\ \vdots & & \vdots \\ x_{n,1} & \cdots & x_{n,p} \end{pmatrix} \times \begin{pmatrix} \beta_{1,1} & \beta_{1,2} \\ \vdots & \vdots \\ \beta_{p,1} & \beta_{p,2} \end{pmatrix} + \begin{pmatrix} \epsilon_{1,1} & \epsilon_{1,2} \\ \vdots & \vdots \\ \epsilon_{n,1} & \epsilon_{n,2} \end{pmatrix} ,$$

where both problems concern the same n samples and the same p features. While each problem has its own vector for the target ($\boldsymbol{y}_{\circ,1}$ or $\boldsymbol{y}_{\circ,2}$), the unknown slopes ($\boldsymbol{\beta}_{\circ,1}$ or $\boldsymbol{\beta}_{\circ,2}$), and the unknown errors ($\boldsymbol{\epsilon}_{\circ,1}$ or $\boldsymbol{\epsilon}_{\circ,2}$), both models have a common feature matrix ($\boldsymbol{X}$). By contrast, in the case of transfer learning, we have

$$\begin{pmatrix} y_1^{(1)} \\ \vdots \\ y_n^{(1)} \end{pmatrix} = \begin{pmatrix} x_{1,1}^{(1)} & \cdots & x_{1,p}^{(1)} \\ \vdots & & \vdots \\ x_{n,1}^{(1)} & \cdots & x_{n,p}^{(1)} \end{pmatrix} \times \begin{pmatrix} \beta_{1,1} \\ \vdots \\ \beta_{p,1} \end{pmatrix} + \begin{pmatrix} \epsilon_{1,1}^{(1)} \\ \vdots \\ \epsilon_{n,1}^{(1)} \end{pmatrix}$$

and

$$\begin{pmatrix} y_1^{(2)} \\ \vdots \\ y_m^{(2)} \end{pmatrix} = \begin{pmatrix} x_{1,1}^{(2)} & \cdots & x_{1,p}^{(2)} \\ \vdots & & \vdots \\ x_{m,1}^{(2)} & \cdots & x_{m,p}^{(2)} \end{pmatrix} \times \begin{pmatrix} \beta_{1,2} \\ \vdots \\ \beta_{p,2} \end{pmatrix} + \begin{pmatrix} \epsilon_{1,2}^{(2)} \\ \vdots \\ \epsilon_{m,2}^{(2)} \end{pmatrix} ,$$

where the two problems concern the same p features but different n or m samples, respectively. Each problem has not only its own vector for the target ($\boldsymbol{y}^{(1)}$ or $\boldsymbol{y}^{(2)}$), the unknown slopes ($\boldsymbol{\beta}_{\circ,1}$ or $\boldsymbol{\beta}_{\circ,2}$), and the unknown errors ($\boldsymbol{\epsilon}^{(1)}$ or $\boldsymbol{\epsilon}^{(2)}$), but also its own feature matrix ($\boldsymbol{X}^{(1)}$ or $\boldsymbol{X}^{(2)}$). For both multi-task and transfer learning, however, we expect that the unknown slopes ($\boldsymbol{\beta}_{\circ,1}$ and $\boldsymbol{\beta}_{\circ,2}$) are positively correlated, provided the problems are sufficiently related. When fitting these models, it can therefore be beneficial to share information between the two problems (e.g., by shrinking each pair of slopes towards a common value).

We previously proposed methods for multi-task learning (Rauschenberger and Glaab, 2021) and transfer learning (Rauschenberger et al., 2023) that increase the predictive performance of penalised regression by sharing information between related prediction or classification problems. Here we are also concerned with multi-task and transfer learning, but our focus is on sparse models. We want to estimate models with few non-zero coefficients, which are easy to interpret. For example, if a method selects 10 out of 20 000 genes for predicting an outcome, interpreting the estimated effects of the genes on the outcome is straightforward. A researcher can then examine the signs and sizes of the estimated effects for all selected genes one-by-one.

Here we propose a simple two-stage procedure for estimating sparse models in high-dimensional multi-task or transfer learning settings. In both stages, we perform penalised regression separately for each problem. While this is done with standard elastic net regularisation (Zou and Hastie, 2005) in the first stage, we use the initial coefficients from the first stage to construct feature-specific and sign-specific penalty factors for the second stage. These penalty factors depend on the same problem (internal weights), as for the adaptive lasso (Zou, 2006), and on the other problems (external weights), as for the weighted lasso (Bergersen et al., 2011). If the problems are sufficiently related, this information exchange should increase the predictive performance of the models while maintaining their sparsity.

Methods

Notation

Suppose there are q datasets, indexed by $k \in \{1, \ldots, q\}$, with the same set of p features, indexed by $j \in \{1, \ldots, p\}$. In each dataset k, the samples are indexed by $i \in \{1, \ldots, n_k\}$. We assume that all datasets include the same samples for multi-task learning (i.e., the total sample size is $n = n_1 = \ldots = n_k$) and that each dataset includes different samples for transfer learning (i.e., the total sample size is $n = \sum_{k=1}^{q} n_k$). Let $y_i^{(k)}$ be the value of the target for sample i in dataset k, and let $x_{i,j}^{(k)}$ be the value of feature j for sample i in dataset k. For each dataset, this leads to an $n_k \times 1$ target vector $\boldsymbol{y}^{(k)}$, with one entry for each sample, and an $n_k \times p$ feature matrix $\boldsymbol{X}^{(k)}$, with samples in the rows and features in the columns. In the case of multi-task learning, we could use a more compact notation with a common $n \times q$ outcome matrix, namely $\boldsymbol{Y} = (\boldsymbol{y}^{(1)} \| \ldots \| \boldsymbol{y}^{(q)})$, and a common $n \times p$ feature matrix, namely $\boldsymbol{X} = \boldsymbol{X}^{(1)} = \ldots = \boldsymbol{X}^{(q)}$.

Model

For each dataset k, the aim is to fit the generalised linear model

$$h\left(\mathbb{E}\left[y_i^{(k)}\right]\right) = \beta_{0,k} + \beta_{j,k} x_{i,j}^{(k)} \ ,$$

where h denotes a link function (e.g., identity for Gaussian family or logit for binomial family), $\beta_{0,k}$ denotes the unknown intercept, and $\beta_{j,k}$ denotes the unknown slope of feature j. In low-dimensional settings ($p \ll n_k$), we could estimate the unknown parameters (intercept and slopes) by maximising the family-dependent likelihood $L(\boldsymbol{y}^{(k)}; \boldsymbol{\beta}_{\circ,k})$, but high-dimensional settings ($p \gg n_k$) require penalisation.

Stage 1 - Estimation without prior information

In the first stage, we estimate the unknown parameters without sharing information between different problems. This involves penalising the likelihood with the sum of the absolute values (L_1-norm, lasso) and the sum of the squared values (L_2-norm, ridge) of the slopes:

$$\rho_{\text{init}}(\lambda_{1,k}; \boldsymbol{\beta}_{\circ,k}) = \lambda_{1,k} \sum_{j=1}^{p} \alpha\, |\beta_{j,k}| + (1-\alpha) \frac{\beta_{j,k}^2}{2} \ ,$$

where the hyperparameter $\lambda_{1,k}$ is restricted to be non-negative ($\lambda_{1,k} \geq 0$) and determines the shrinkage, with $\lambda_{1,k} = 0$ leading to an unpenalised model and $\lambda_{1,k} = \infty$ leading to an intercept-only model (i.e., $\hat{\beta}_{1,k} = \ldots = \hat{\beta}_{p,k} = 0$). We set the elastic net mixing parameter not to $\alpha = 1$ (lasso) but to $\alpha = 0.95$ (lasso-like elastic net) to allow the model to select more features than there are samples and to select strongly correlated features. If lasso-like elastic net regularisation sets all coefficients equal to zero—such as under small sample sizes or many weak effects—ridge regularisation ($\alpha = 0$) might be a better choice. We estimate the parameters by optimising the penalised likelihood $L(\boldsymbol{y}^{(k)}; \boldsymbol{\beta}_{\circ,k}) - \rho_{\text{init}}^{(k)}(\lambda_{1,k}; \boldsymbol{\beta}_{\circ,k})$ with respect to the vector $\boldsymbol{\beta}_{\circ,k}$. Let $\hat{\beta}_{0,k}$ denote the estimated intercept in problem k, and let $\hat{\beta}_{j,k}$ denote the estimated slope for feature j in problem k.

Prior information

We use the coefficients from the initial regressions to construct penalty factors for the final regressions. For each problem k, we consider the initial coefficients from the same problem (k) and those from the other problems ($l \neq k$) separately. As an initial positive coefficient is evidence for a positive effect and evidence against a negative effect, and the other way round for initial negative coefficients, we derive different weights for positive effects and negative effects in the final regressions. For the internal information (i.e., obtained from the same problem), the idea is that the initial coefficient determines the weight for a final coefficient of the same sign. Let $w_{j,k}^{\text{int}}$ and $w_{p+j,k}^{\text{int}}$ indicate the internal weights for a final positive or negative coefficient, respectively, for feature j in problem k. For the external information (i.e., obtained from the other problems), the idea is that all positive and all negative initial coefficients determine the weight for a final positive or negative coefficient, respectively. Let $w_{j,k}^{\text{ext}}$ and $w_{p+j,k}^{\text{ext}}$ indicate the weights for the final positive or negative coefficient, respectively, for feature j in problem k. The internal and external weights (columns) for estimating a positive or a negative effect (rows) are

	internal	external
$+$	$w_{j,k}^{\text{int}} = \max(0, \hat{\beta}_{j,k})$	$w_{j,k}^{\text{ext}} = \sum_{l \neq k} \max(0, \hat{\beta}_{j,l})$
$-$	$w_{p+j,k}^{\text{int}} = \lvert \min(0, \hat{\beta}_{j,k}) \rvert$	$w_{p+j,k}^{\text{ext}} = \lvert \sum_{l \neq k} \min(0, \hat{\beta}_{j,l}) \rvert$

.

By construction, for feature j in problem k, only one internal weight can be non-zero ($\mathbb{I}[w_{j,k}^{\text{int}} \neq 0] + \mathbb{I}[w_{p+j,k}^{\text{int}} \neq 0] \in \{0,1\}$) but both external weights can be non-zero ($\mathbb{I}[w_{j,k}^{\text{ext}} \neq 0] + \mathbb{I}[w_{p+j,k}^{\text{ext}} \neq 0] \in \{0,1,2\}$). Specifically, if a feature obtains a positive or negative initial coefficient in the same problem, one of its internal weights is non-zero, and if a feature obtains at least one positive and at least one negative initial coefficient in the other problems, both external weights are non-zero.

Stage 2 - Estimation with prior information

In the second stage, we re-estimate the unknown parameters (intercepts and slopes) while sharing information between problems. We do this by decomposing each slope into a positive and a negative part ($\beta_{j,k} = \gamma_{j,k} - \gamma_{p+j,k}$), where both parts are constrained to be non-negative ($\gamma_{j,k} \geq 0$ and $\gamma_{p+j,k} \geq 0$), and by penalising each part differentially.

The model is represented by

$$h\left(\mathbb{E}\left[y_i^{(k)}\right]\right) = \gamma_{0,k} + \sum_{j=1}^{p} \left(\gamma_{j,k} - \gamma_{p+j,k}\right) x_{i,j}^{(k)} \ ,$$

with the non-negativity constraint

$$\gamma_{j,k} \geq 0 \quad \forall j \in \{1, \ldots, 2p\} \ ,$$

where $\gamma_{j,k}$ and $\gamma_{p+j,k}$ denote the effects of feature j. Again, estimating the family-dependent likelihood $L(\boldsymbol{y}^{(k)}; \boldsymbol{\gamma}_{\circ,k})$ requires penalisation.

We penalise the likelihood with a weighted sum of the non-negative slopes:

$$\rho_{\text{final}}(\lambda_{2,k}, \delta_k^{\text{int}}, \delta_k^{\text{ext}}; \boldsymbol{\gamma}_{\circ,k}) = \lambda_{2,k} \sum_{j=1}^{2p} \frac{\gamma_{j,k}}{\left(w_{j,k}^{\text{int}}\right)^{\delta_k^{\text{int}}} + \left(w_{j,k}^{\text{ext}}\right)^{\delta_k^{\text{ext}}}} \ ,$$

where the hyperparameter $\lambda_{2,k}(\geq 0)$ determines the total amount of shrinkage, and the hyperparameters $\delta_k^{\text{int}}(\geq 0)$ and $\delta_k^{\text{ext}}(\geq 0)$ determine the scaling of the internal and the external weights, respectively, as in the adaptive lasso (Zou, 2006) and the weighted lasso (Bergersen et al., 2011), respectively. As $\lambda_{2,k}$ increases, more and more coefficients are set to zero, until reaching an intercept-only model ($\gamma_{1,j} = \ldots = \gamma_{2p,k} = 0$). The numerator $(w_{j,k}^{\text{int}})^{\delta_k^{\text{int}}} + (w_{j,k}^{\text{ext}})^{\delta_k^{\text{ext}}}$ represents the prior weight of feature j in problem k. If the exponent δ_k^{int} or δ_k^{ext} equals zero, the internal or external weights, respectively, have no impact on the prior weights, as any base raised to the power of zero equals one. The model can thereby ignore the internal weights from the same problem or the external weights from the other problems. Exponents less than one dampen the influence of large internal or external weights more than that of smaller ones.

We estimate the parameters by optimising the penalised likelihood $L(\boldsymbol{y}^{(k)}; \boldsymbol{\gamma}_{\circ,k}) - \rho_{\text{init}}^{(k)}(\lambda_{2,k}; \boldsymbol{\gamma}_{\circ,k})$ with respect to the vector $\boldsymbol{\gamma}_{\circ,k}$. For problem k, the final estimated intercept is $\hat{\gamma}_{0,k}$ and the final estimated slope for feature j is $\hat{\gamma}_{j,k} - \hat{\gamma}_{p+j,k}$. As $\hat{\gamma}_{j,k}(\geq 0)$ and $\hat{\gamma}_{p+j,k}(\geq 0)$ are estimated for the same feature (perfect multicollinearity) under penalisation, at least one of $\hat{\gamma}_{j,k}$ and $\hat{\gamma}_{p+j,k}$ will equal zero ($\mathbb{I}[\hat{\gamma}_{j,k} \neq 0] + \mathbb{I}[\hat{\gamma}_{p+j,k} \neq 0] \leq 1$). Figure 1 shows the flow of information from the first stage to the second stage.

Hyperparameter optimisation

The proposed approach includes the hyperparameters $\lambda_{1,k}$ and $\lambda_{2,k}$ for determining the shrinkage in the first and the second stage, respectively, and the hyperparameters δ_k^{int} and δ_k^{ext} for determining the scaling of the internal and external weights, respectively. We use sequences of up to 100 candidate values for the regularisation parameters $\lambda_{1,k}$ and $\lambda_{2,k}$ (Friedman et al., 2010, R package `glmnet`) and the candidate values $\{0.0, 0.2, 0.4, 0.6, 0.8, 1.0\}$ for the scaling parameters δ_k^{int} and δ_k^{ext}. We use grid search with 10-fold internal cross-validation for optimising these four hyperparameters, for each problem k, although other approaches (e.g., random search) or other numbers of folds (e.g., leave-one-out cross-validation) are also possible. While cross-validation involves excluding a fold for all problems (targets) in the case of multi-task learning, it involves excluding a fold for one problem (dataset) in the case of transfer learning. This is because multi-task learning involves multiple problems for the same samples but transfer learning involves multiple problems for different samples. Hyperparameter optimisation is relatively fast due cyclical coordinate descent along the regularisation path (Friedman et al., 2010, R package `glmnet`). Table 1 includes the pseudocode for (hyper)parameter optimisation.

Simulation

Data generating process

We repeatedly simulate data for multi-task learning and transfer learning to test the proposed method.

For multi-task learning, we repeatedly simulate $q = 3$ targets and $p = 200$ features for a small number of training samples ($n = 100$) and a large number of testing samples ($m = 10\,000$), where the targets are indexed by $k \in \{1, \ldots, q\}$, the features by $j \in \{1, \ldots, p\}$, and the samples by $i \in \{1, \ldots, n+m\}$. In each iteration, we simulate (i) the rows of the $(n+m) \times q$ feature matrix $\boldsymbol{X}$ from a multivariate Gaussian distribution with a decreasing correlation structure (i.e., $X_{i,\circ} \sim \text{N}(0, \boldsymbol{R})$, where $R_{j,l} = \rho^{|j-l|}$ and $\rho = 0.5$), (ii) the $p \times q$ effect matrix $\boldsymbol{B}$ as shown below, and (iii) the $(n+m) \times q$ target matrix $\boldsymbol{Y} = \boldsymbol{XB} + \boldsymbol{E}$, where $\boldsymbol{E}$ is Gaussian noise ($E_{i,k} \sim \text{N}(0,1)$).

For transfer learning, we repeatedly simulate $q = 3$ datasets with one target and $p = 200$ features for small numbers of training samples ($n_1 = 50$, $n_2 = 100$ and $n_3 = 200$) and large numbers of testing samples ($m_1 = m_2 = m_3 = 10\,000$), where the datasets are indexed by $k \in \{1, \ldots, q\}$, the features by $j \in \{1, \ldots, p\}$, and the samples by $i \in \{1, \ldots, n_k + m_k\}$. In each iteration, we simulate (i) the rows of the $(n_k + m_k) \times q$ feature matrices $\boldsymbol{X}^{(k)}$ from multivariate Gaussian distributions with decreasing correlation structures (i.e., $X_{i,\circ}^{(k)} \sim \text{N}(0, \boldsymbol{R}^{(k)})$, where $R_{j,l}^{(k)} = \rho_k^{|j-l|}$ and $\rho_k = 0.5$), (ii) the $p \times 1$ effect vectors $\boldsymbol{B}_{\circ,k}$ as shown below, and (iii) the $(n_k + m_k) \times 1$ target vectors $\boldsymbol{y}^{(k)} = \boldsymbol{X}^{(k)} \boldsymbol{B}_{\circ,k} + \boldsymbol{\epsilon}^{(k)}$, where $\boldsymbol{\epsilon}^{(k)}$ is Gaussian noise ($\epsilon_i^{(k)} \sim N(0,1)$).

For multi-task learning and transfer learning, each problem obtains its own $p \times 1$ effect vector $\boldsymbol{B}_{\circ,k} = \boldsymbol{\theta} + \boldsymbol{\Delta}_{\circ,k}$, where $\boldsymbol{\theta}$ contains the effects shared by all problems and $\boldsymbol{\Delta}_{\circ,k}$ contains the problem-specific effects. Given the probabilities of success for the shared effects (π_θ) and the specific effects (π_δ), we simulate the effects from mixture distributions of Bernoulli trials and standard Gaussian distributions (i.e., $\theta_j \sim \text{Bern}(\pi_\theta) \times \text{N}(0,1)$ and $\delta_{k,j} \sim \text{Bern}(\pi_\delta) \times \text{N}(0,1)$). If there are many shared and few specific effects (large π_θ, small π_δ), the problems are close to each other. But if there are few shared and many specific effects (small π_θ, large π_δ), the problems are far from each other. We consider all combinations of π_θ and π_δ in $\{0\%, 2.5\%, 5\%\}$.

In short, we simulate related problems about the same features and the same samples (multi-task learning) or about the same features but different samples (transfer learning). While all problems share the same feature matrix in multi-task learning, each problem has its own feature matrix in transfer learning. Here, each prediction problem is supported by information from two other prediction problems. Table 2 shows the two data-generating processes.

Predictive and selection performance

While we compare the proposed approach to multivariate Gaussian lasso regression from Simon et al. (2013, R package `glmnet` with `family="mgaussian"`) and to sparse partial least squares (Chung and Keleş, 2010, R package `spls`) in the case of multi-task learning, we compare it to transfer learning Gaussian lasso regression from Tian

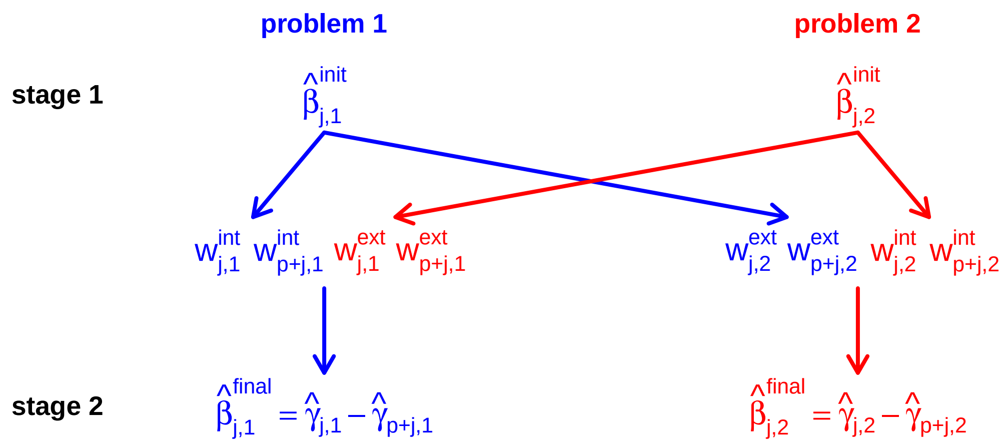
problem 1
problem 2
stage 1
$\hat{\beta}^{\text{init}}_{j,1}$
$\hat{\beta}^{\text{init}}_{j,2}$
$w^{\text{int}}_{j,1}$ $w^{\text{int}}_{p+j,1}$ $w^{\text{ext}}_{j,1}$ $w^{\text{ext}}_{p+j,1}$
$w^{\text{ext}}_{j,2}$ $w^{\text{ext}}_{p+j,2}$ $w^{\text{int}}_{j,2}$ $w^{\text{int}}_{p+j,2}$
stage 2
$\hat{\beta}^{\text{final}}_{j,1} = \hat{\gamma}_{j,1} - \hat{\gamma}_{p+j,1}$
$\hat{\beta}^{\text{final}}_{j,2} = \hat{\gamma}_{j,2} - \hat{\gamma}_{p+j,2}$

Fig. 1. Information flow. The initial coefficients from the first-stage regressions (top) determine the internal and external weights for the second-stage regressions (centre) and thereby influence the final coefficients from the second-stage regressions (bottom). This example is for two problems (blue, red). If there are three (or more) problems, the external weights are based on the other two (or more) problems. In each problem k, each feature j obtains two internal and two external weights, namely for a positive effect (indexed by j) and for a negative effect (indexed by $p + j$).

and Feng (2023, R package `glmtrans`) and to hierarchical regularised regression (Weaver and Lewinger, 2019, R package `xrnet`) in the case of transfer learning. For sparse partial least squares, we specify the candidate values $\{3, \ldots, 10\}$ for the hidden components and $\{0, 0.1, \ldots, 0.9\}$ for the thresholding parameter, and for hierarchical regularised regression, we provide the coefficients from lasso-like elastic net regression ($\alpha = 0.95$) as external data. Figures 2 and 3 show the results for multi-task learning and transfer learning, respectively.

Concerning the predictive performance, we make similar observations for multi-task learning and transfer learning. We choose the mean squared error as a performance metric, calculated on the testing data (hold-out method). The mean squared error decreases as the predictive performance improves. If there are neither shared effects ($\pi_\theta = 0\%$) nor specific effects ($\pi_\delta = 0\%$), the proposed and the available methods have a similar predictive performance as standard lasso regression. (Note that the proposed method reduces to the standard lasso if $\delta_k^{\text{int}} = \delta_k^{\text{ext}} = 0$.) If there are some shared effects ($\pi_\theta = 2.5\%$) but no specific effects ($\pi_\delta = 0\%$), the proposed and the available methods outperform standard lasso regression, meaning that sharing information between problems is beneficial. This benefit is greater if there are many shared effects ($\pi_\theta = 5\%$), which makes sense as multi-task learning and transfer learning should be more beneficial for closely related problems than for loosely related problems. If there are no shared effects ($\pi_\theta = 0\%$) but specific effects ($\pi_\delta = 2.5\%$ or $\pi_\delta = 5\%$), the proposed method is more predictive than the available methods. It seems that adaptive penalisation, even without sharing information between problems, improves the predictive performance in these sparse simulation settings. (Note that the proposed method reduces to the adaptive lasso with weights obtained from the same dataset if $\delta_k^{\text{ext}} = 0$.) Given a proportion of shared effects ($\pi_\theta = 0\%$, $\pi_\theta = 2.5\%$ or $\pi_\theta = 5\%$), the performance of the proposed method relative to the available methods increases with the proportion of specific effects ($\pi_\delta = 0\%$, $\pi_\delta = 2.5\%$, or $\pi_\delta = 5\%$). If there are many specific effects ($\pi_\delta = 5\%$), only the proposed method shows a benefit with respect to the standard lasso.

Concerning the selection performance, we make different observations for multi-task learning and transfer learning. We assess two aspects of the selection performance, namely the sparsity (measured by number of non-zero coefficients) and the precision (measured by the number of coefficients with correct signs divided by number of non-zero coefficients). As the selection performance improves, the number of non-zero coefficients decreases and the proportion of correct signs increases. For multi-task learning, except if there are neither shared nor specific effects ($\pi_\theta = 0\%$ and $\pi_\delta = 0\%$), the proposed method and the method from Chung and Keleş (2010) estimate sparser and more precise models than the method from Simon et al. (2013). A possible explanation is that the multivariate method from Simon et al. (2013) by construction selects the same features for all tasks. For transfer learning, the proposed method estimates sparser and more precise models than the available methods if there are specific effects ($\pi_\theta = 2.5\%$ or $\pi_\theta = 5\%$). A possible explanation is that the transfer learning method from Tian and Feng (2023) first selects a shared set of features for all datasets and then adds a specific set of features for each dataset, and that the method from Weaver and Lewinger (2019) shrinks the coefficients towards a linear combination of the prior effects..

Computation time

Sharing information between problems increases the computation time. In the multi-task learning simulation, the proposed method is almost twice as slow as the slower one of the two available methods (computation time relative to standard lasso: 28.6; Simon et al., 2013, R package `glmnet`: 1.4; Chung and Keleş, 2010, R package `spls`: 16.6). And in the transfer learning simulation, the proposed method is as slow as the slower one of the two available methods (computation time relative to standard lasso: 22.3; Tian and Feng, 2023, R package `glmtrans`: 21.9; Weaver and Lewinger, 2019, R package `xrnet`: 5.7). The relative computation times depend on the number of problems, number of features, and sample sizes.

Impact of sample size

Fixing the proportion of shared effects at $\pi_\theta = 5\%$ and the one of specific effects at $\pi_\theta = 2.5\%$, we repeat the simulation study under different sample sizes. Figure 4 shows how the out-of-sample predictive performance depends on the sample size. In the case of transfer learning, we distinguish between the sample sizes of the source datasets (i.e., the supporting datasets) and the target

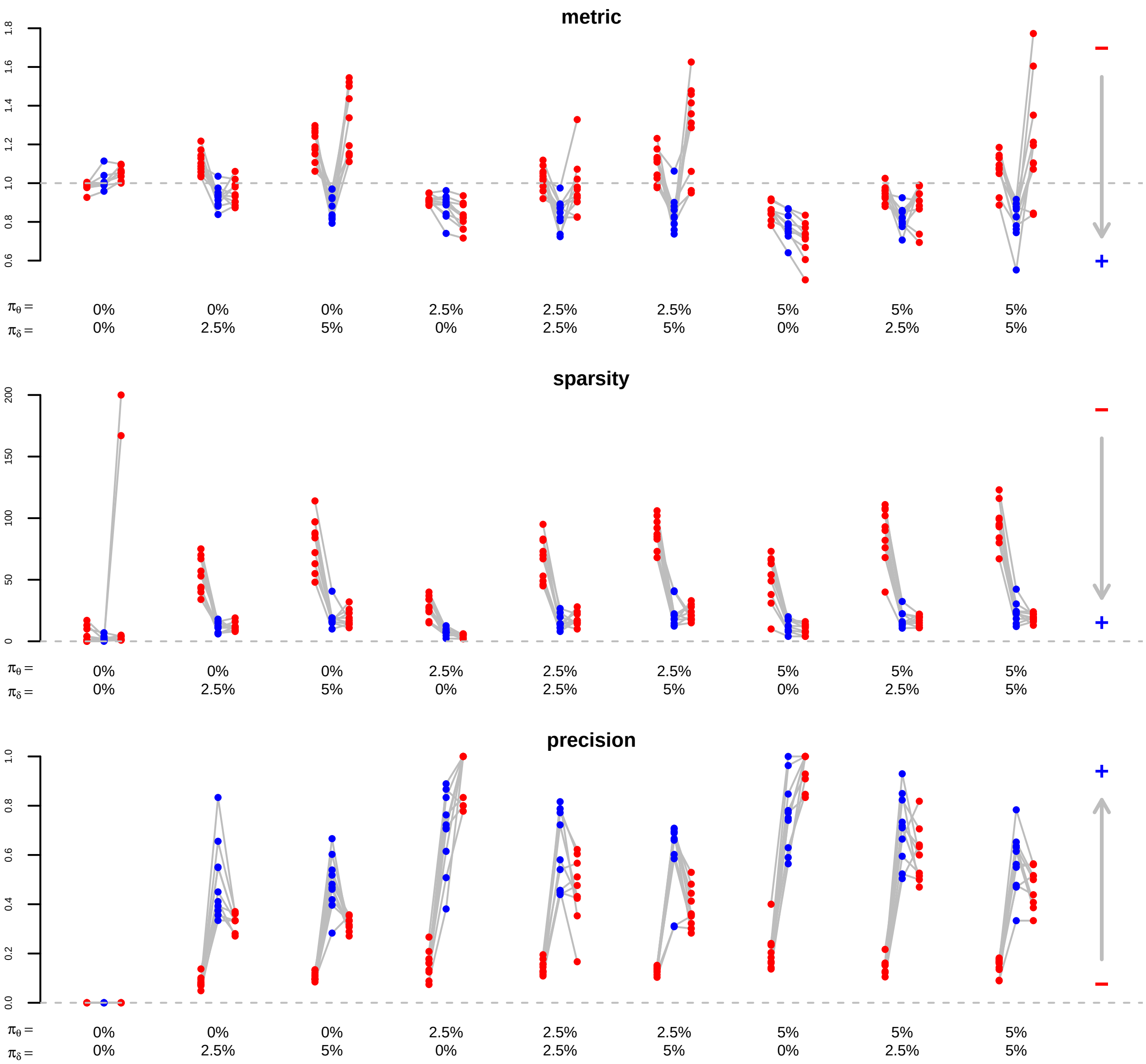

Fig. 2. Multi-task learning. Comparison of different measures (rows) between the proposed method (blue points) and two available methods (Simon et al., 2013, R package `glmnet`: red points on the left; Chung and Keleş, 2010, R package `spls`: red points on the right) in different simulation settings (columns), based on the average of three problems (tasks) for each repetition out of ten. Measures: performance metric (mean squared error on hold-out data, as a fraction of the one from standard lasso regression; a point below the dashed line means that multi-task learning improves predictions), sparsity (number of non-zero coefficients), precision (number of coefficients with correct signs divided by number of non-zero coefficients). The arrows point in the direction of improvement. Settings: percentage of features with a shared effect for all problems (π_θ), percentage of features with a specific effect for each problem (π_δ).

dataset (i.e., the dataset of interest). The proposed method is more predictive than the available methods, with the following exceptions: (i) In the case of multi-task learning under a small sample size, the method from Simon et al. (2013, R package `glmnet`) is more predictive. This method selects a common set of features for all tasks. (ii) In the case of transfer learning under a small target sample size, the methods from Tian and Feng (2023, R package `glmtrans`) and Weaver and Lewinger (2019, R package `xrnet`) are more predictive. The former method first estimates a common model for all datasets and then the deviations for the target dataset, and the latter method penalises the differences between the models for the source dataset and the one for the target dataset. We therefore conclude that these available methods are more suitable for small sample sizes (multi-task learning) or small target sample sizes (transfer learning). By contrast, the proposed method needs a sufficiently large (target) sample size for model training.

Application

Data preparation

We searched for datasets on treatment response in three autoimmune diseases, namely inflammatory bowel disease (IBD), rheumatoid arthritis (RA), and multiple sclerosis (MS), on a public resource of

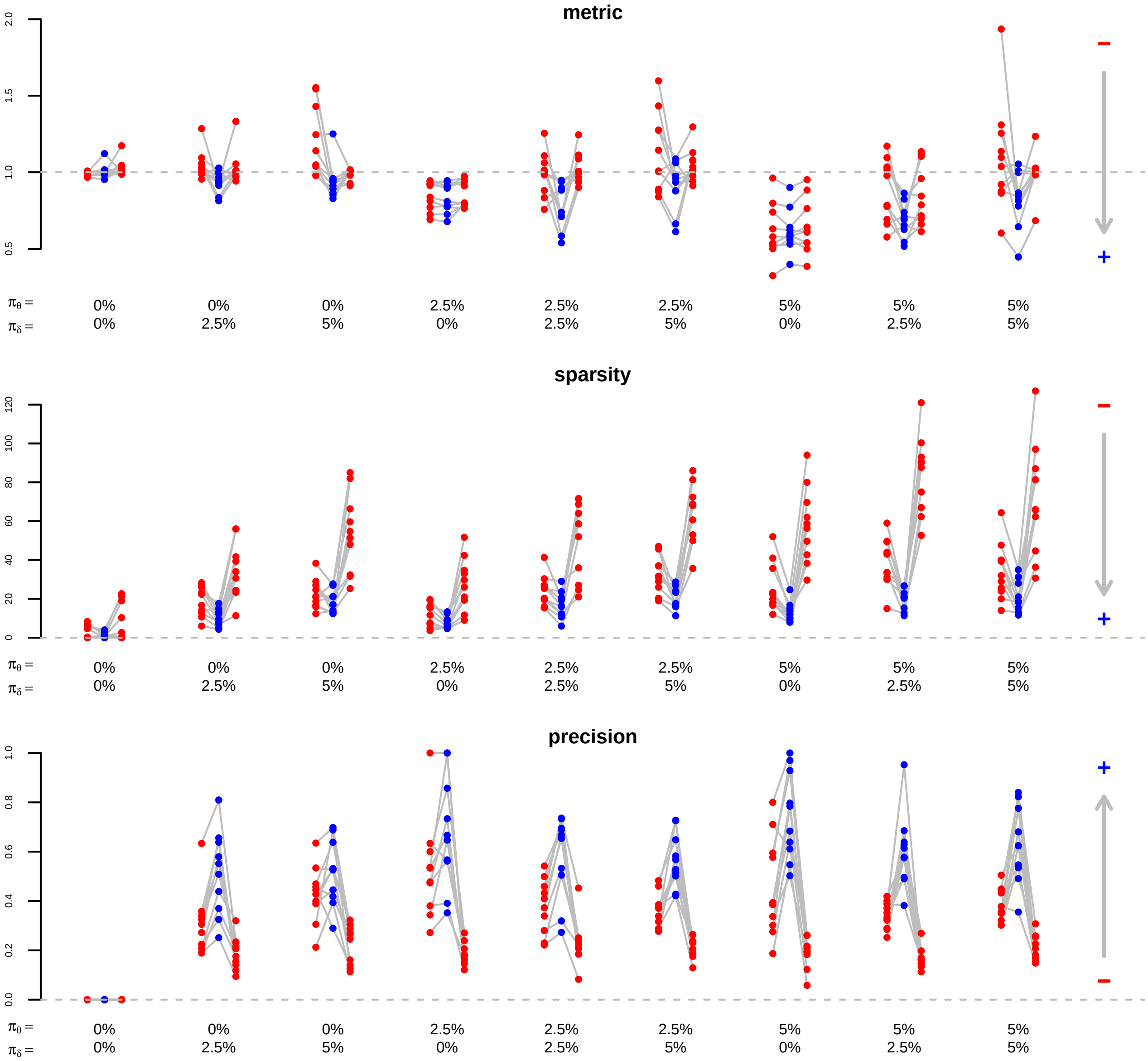

Fig. 3. Transfer learning. Comparison of different measures (rows) between the proposed method (blue points) and two available methods (Tian and Feng, 2023, R package `glmtrans`: red points on the left; Weaver and Lewinger, 2019, R package `xrnet`: red points on the right) in different simulation settings (columns), based on the average of three problems (datasets) for each repetition out of ten. Measures: performance metric (mean squared error on hold-out data, as a fraction of the one from standard lasso regression; a point below the dashed line means that transfer learning improves predictions), sparsity (number of non-zero coefficients), precision (number of coefficients with correct signs divided by number of non-zero coefficients). The arrows point in the direction of improvement. Settings: percentage of features with a shared effect for all problems (π_θ), percentage of features with a specific effect for each problem (π_δ).

uniformly processed RNA-seq data (Wilks et al., 2021, R package **`recount3`**). Restricting the search to prospective datasets with at least 40 samples, we found four datasets on treatment response in IBD (Tew et al., 2016; Haberman et al., 2019; Verstockt et al., 2019, 2020). We also included four related datasets, namely a cross-sectional dataset on disease activity in IBD (Boyd et al., 2018), a prospective dataset on drug-free remission in RA (Baker et al., 2019), a prospective dataset on treatment response in juvenile idiopathic arthritis (Moncrieffe et al., 2017), and a cross-sectional dataset on disease activity in RA (Goldberg et al., 2018). Table 3 includes some details on the five datasets related to treatment response in IBD and the three datasets related to treatment response in RA. The aim is to apply transfer learning to these datasets, separately for IBD and RA. This means that each classification problem is supported by information from four or two other classification problems, respectively.

Restricting the analysis to the 22 321 protein-coding genes from the 63 856 transcripts, we select the 5 000 most variably expressed protein-coding genes across all datasets. Specifically, in each dataset, we rank the genes in ascending order of their variances, take the mean of these ranks across all datasets, and retain the genes with the top 5 000 highest mean ranks. For each dataset, we calculate the library sizes by summing the expression values across all transcripts (not only the selected protein-coding genes)

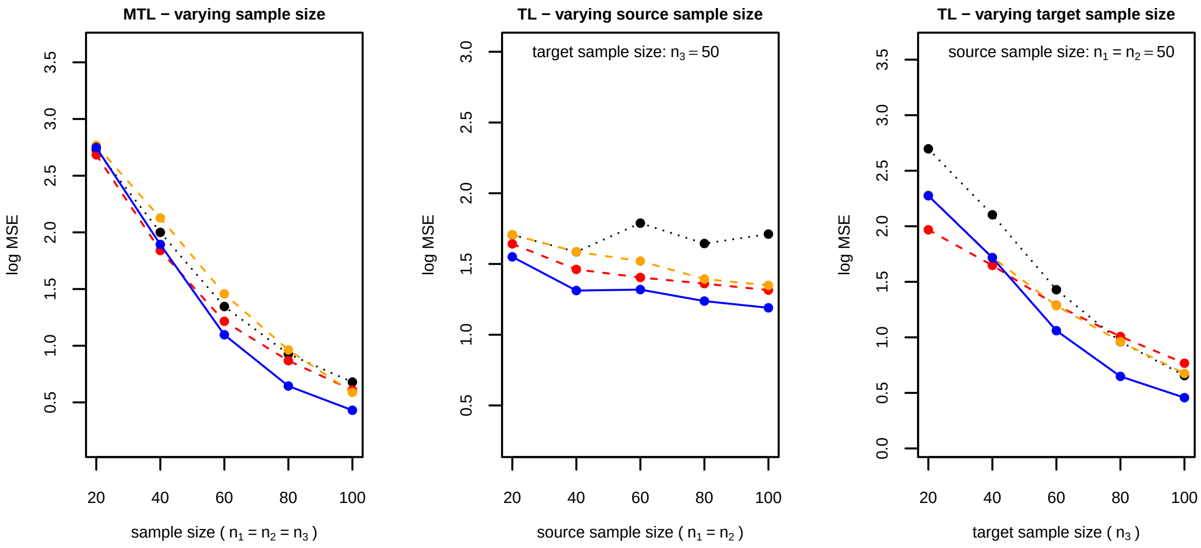

Fig. 4. Sample size. Logarithmic transformation of mean squared error calculated on hold-out data (y-axis) against source or target sample size (x-axis), for the standard lasso (dotted black line), available methods (dashed red and orange lines), and the proposed method (solid blue line), based on 10 simulated datasets for each sample size. The available methods are those from Simon et al. (2013, R package `glmnet`, red) and Chung and Keleş (2010, R package `spls`, orange) for multi-task learning (MTL, left) and those from Tian and Feng (2023, R package `glmtrans`, red) and Weaver and Lewinger (2019, R package `xrnet`, orange) for transfer-learning (TL, centre and right). While the mean squared error is averaged across all three targets in the case of multi-task learning, it is only reported for the target dataset ($k = 3$) in the case of transfer learning.

for each sample, normalise the samples for different library sizes and compositional biases using the trimmed mean of M-values method (Robinson and Oshlack, 2010, R package **edgeR**), stabilise the variance with the Anscombe transform ($x \rightarrow 2\sqrt{x + 3/8}$), and standardise the features (zero mean, unit variance).

Data exploration

To explore the relationships between the different datasets, we trained logistic ridge regression on each dataset. We use ridge rather than lasso regularisation to facilitate comparisons across datasets, because it leads to non-zero coefficients for all features. Note that correlated ridge regression coefficients (dense models) are neither necessary nor sufficient for successful transfer learning in lasso regression (sparse models).

First, we compared the regression coefficients from the 8 datasets with Spearman's correlation test, maintaining the family-wise error rate for the $\binom{8}{2} = 28$ pairwise combinations at 5% with the Bonferroni-correction. We observe that the regression coefficients from the 5 IBD datasets are significantly correlated (p-value $\leq 0.05/28$ for all 10 pairwise combinations). While the correlation is significantly *positive* for all 6 pairwise combinations not including the dataset from Verstockt et al. (2020), it is significantly *negative* for all 4 pairwise combinations including the dataset from Verstockt et al. (2020). We therefore inverted the target from Verstockt et al. (2020), by exchanging the class labels for responders and non-responders. Table 4 shows the correlation coefficients between the regression coefficients from the different datasets, after this inversion. The regression coefficients are significantly positively correlated in 10 out of 10 comparisons between IBD datasets, 1 out of 3 comparisons between RA datasets, and 6 out of 15 comparisons between IBD and RA datasets (p-value $\leq 0.05/28$). It is therefore more promising to share information among the IBD datasetsthan among the RA datasets or between the IBD and the RA datasets.

Second, we assessed the cross-dataset predictive performance, i.e., by training the logistic regression on a dataset and testing it on the other datasets. Based on the out-of-sample predicted probabilities, we calculated the area under the receiver operating characteristic curve (ROC-AUC) for each pairwise permutation of datasets. To examine whether a classifier significantly outperforms a random classifier, we used the one-sided Mann-Whitney U test to test whether the ranks of the predicted probabilities of treatment response are significantly higher for the responders than for the non-responders. Table 5 shows the ROC-AUC for each pairwise permutation of datasets (off-diagonal entries). While models trained on an IBD dataset and tested on another IBD dataset tend to significantly outperform a random classifier at the nominal 5% level (p-value $\leq$ 0.05 for 16 out of 20 pairwise permutations), this is rarely the case for models that are trained or tested on an RA dataset (p-value $\leq$ 0.05 for 2 out of 36 pairwise permutations). This suggests that transfer learning might be beneficial for the IBD datasets.

In addition, we trained and tested logistic lasso regression on each dataset using 10-fold external cross-validation. Table 5 shows the cross-validated ROC-AUC for each dataset (diagonal entries). For the datasets from Tew et al. (2016) and Boyd et al. (2018), the predictions are significantly better than those from a random classifier at the nominal 5% level (p-value $\leq$ 0.05). For the other datasets, standard lasso regression fails to capture any predictive signal.

Transfer learning

We compared the proposed method to standard lasso regression, the transfer learning method from Tian and Feng (2023, R package **glmtrans**), and the transfer learning method from Weaver and Lewinger (2019, R package **xrnet**), in the application on IBD and in the application on RA. To estimate the predictive performance of different methods, we used 5 times 10-fold external cross-validation.

This involves splitting each dataset 5 times into 10 external folds, and using 9 external folds for training and 1 external fold for testing in each repetition. To assess the model sparsity of different methods, we also refit each model 5 times on each full dataset, using different internal folds in each repetition. The repetitions introduce variability in hyperparameter optimisation due to different splits of the samples into folds. For each dataset and each model, we thereby obtained 5 times a cross-validated ROC-AUC and 5 times a number of non-zero coefficients.

Figure 5 shows the performance of the proposed and the available transfer learning methods. For the IBD application, the proposed method reaches not only a higher mean cross-validated ROC-AUC (across 5 datasets and 5 repetitions) than the available methods (`glmtrans`: 0.62, `sparselink`: 0.66, `xrnet`: 0.59) but also a lower mean number of non-zero coefficients (`glmtrans`: 8.56, `sparselink`: 4.52, `xrnet`: 13.36). For the RA application, none of the three transfer learning methods reaches a mean ROC-AUC (across 3 datasets and 5 repetitions) above 0.50 (i.e., the performance of a random classifier), meaning that transfer learning cannot compensate for the weak predictive signal in the individual datasets.

In the IBD application, the proposed method increases the mean ROC-AUC (across 5 datasets and 5 repetitions) of logistic lasso regression from 0.58 to 0.66, with important differences between datasets. The increase in mean ROC-AUC (across 5 repetitions) is largest for the three prospective datasets on treatment response with less than 25 samples in the minority class, namely from 0.58 to 0.69 for Tew et al. (2016), from 0.44 to 0.59 for Verstockt et al. (2019), and from 0.48 to 0.60 for Verstockt et al. (2020). The proposed method only slightly increases the mean ROC-AUC from 0.53 to 0.58 for the prospective dataset on treatment response with more than 100 samples in the minority class (Haberman et al., 2019) and leaves it unchanged at 0.86 for the cross-sectional dataset on disease activity (Boyd et al., 2018). Information from problems with a strong signal appears to be useful for addressing problems with a weak signal, but not vice versa.

Figure 6 shows the coefficients of the selected genes for the IBD datasets. Genes that are selected in at least 3 out of 5 repetitions in any of the prospective dataset on treatment response (excluding the cross-sectional dataset on disease activity) are *SORBS3*, *EIF4H*, *SLC5A3*, *SOS2*, *MLXIP*, *EPC1*, and *PGM1* with positive coefficients, and *SIK1* and *CR2* with negative coefficients. For none of these genes, however, we could find evidence in the scientific literature of direct effects on IBD treatment response.

Discussion

We designed the proposed method to estimate sparse but predictive models for each prediction or classification problem. Our adaptive penalisation scheme encourages the method to select the same features as well as to estimate the same signs and the same sizes for their effects in all problems. As we do not impose any strict constraints on the models, however, the method might select different features as well as estimate different signs and different sizes for their effects.

This low degree of harmonisation is only one possible compromise between estimating a common model for all problems and estimating a separate model for each problem. A higher degree of harmonisation might involve a common set of non-zero coefficients, a common set of non-positivity and non-negativity constraints on the coefficients for all problems, or penalised differences in coefficients between problems. This might not only be beneficial for interpretation in some applications but also for predictivity under low signal-to-noise ratios (i.e., due to small sample sizes or many weak effects).

While all problems might be of interest in some applications, only some problems might be of interest in other applications. We can then distinguish between problems providing prior information (i.e., source datasets in transfer learning or secondary outcomes in multi-task learning) and problems receiving prior information (i.e., target datasets in transfer learning or primary outcomes in multi-task learning). We expect that sharing information between problems is useful when the supporting problems are relatively straightforward and the supported problems are relatively difficult (e.g., due to small sample sizes or outlier contamination).

The proposed approach involves two hyperparameters for determining the influence of the internal and external weights in each problem. If there are three (or more) problems, however, two (or more) problems influence the external weights. Instead of giving equal weight to each external problem, it might be advantageous to give more weight to informative ones and less weight to uninformative ones. A computationally expensive solution would be to tune one weight for each combination of models. Alternatively, it might be possible to construct weights based on the predictivity of the initial coefficients from each problem for the targets from the other problems.

Federated learning allows researchers to build models based on multiple datasets without sharing the datasets. The proposed method could be implemented for federated transfer learning settings, because each first-stage regression and each second-stage regression only exploits a single dataset while information is transferred between datasets by means of the initial coefficients from the first-stage regressions. This is suitable for a decentralised federated setting, where each node first trains its local model based on its dataset, shares this model with the other nodes, and finally retrains its local model. This procedure shares information between datasets not to obtain a global model for all datasets together but to improve the local model for each dataset separately.

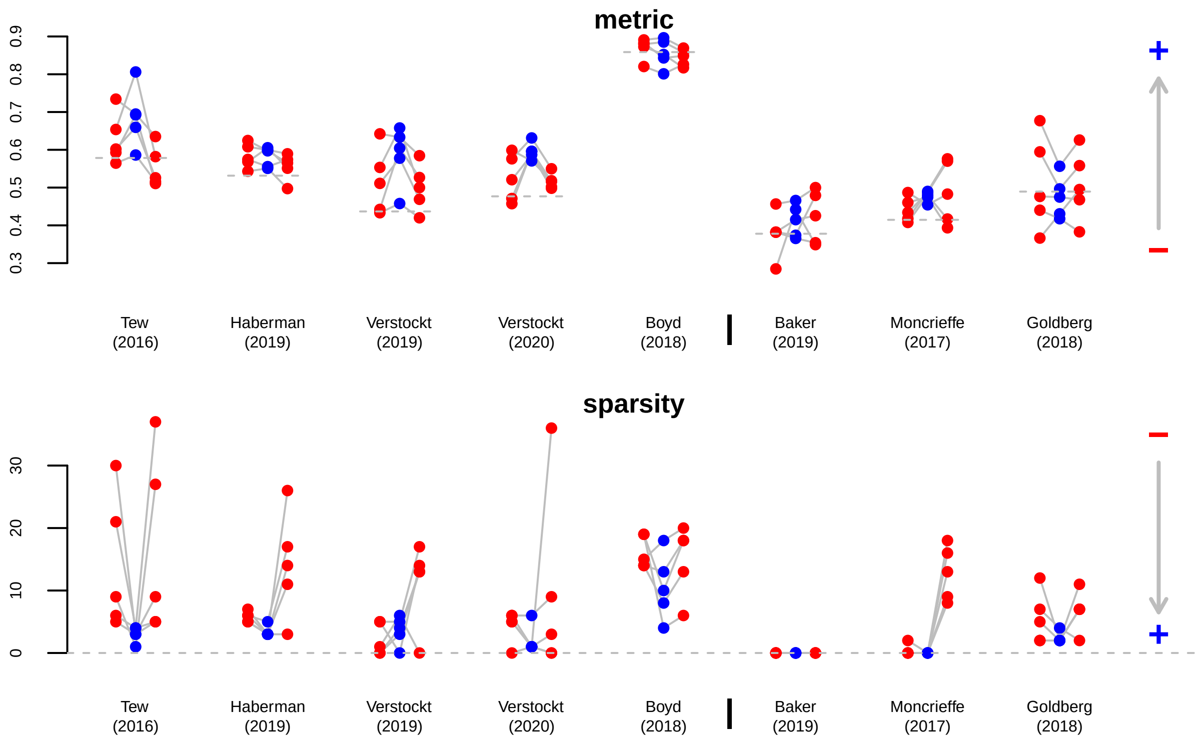

Fig. 5. Comparison of different measures (rows) between the proposed method (blue points) and two available methods (Tian and Feng, 2023, R package `glmtrans`: red points on the left; Weaver and Lewinger, 2019, R package `xrnet`: red points on the right) in different applications (blocks of columns), based on 5 repetitions of 10-fold cross-validation. Measures: performance metric (cross-validated AUC, the dashed lines indicate the mean from standard lasso regression), sparsity (number of non-zero coefficients). The arrows point in the direction of improvement. Applications: datasets related to treatment response in IBD (left), datasets related to treatment response in RA (right).

Table 1. Pseudocode for (hyper)parameter optimisation in the proposed approach to multi-task learning (left) and transfer learning (right).

multi-task learning

```
Require:
$n \times q$ matrix $\boldsymbol{Y}$ (targets)
$n \times p$ matrix $\boldsymbol{X}$ (features)
Ensure:
$q \times 1$ vectors $\boldsymbol{\lambda_1}$, $\boldsymbol{\lambda_2}$, $\boldsymbol{\delta}^{\text{int}}$, $\boldsymbol{\delta}^{\text{ext}}$ (hyperparameters)
$(p+1) \times q$ matrix $\hat{\boldsymbol{B}}$ (coefficients)

set $n \times 1$ vector of fold identifiers fold,
with values from 1 to 10
initialise various empty $n \times q$ matrices $\hat{\boldsymbol{Y}}$

Hyperparameter optimisation

for $k$ in 1 to $q$ do
    $\lambda_{1,k} \leftarrow \mathit{ElasticNetCV}(\boldsymbol{Y}_{\circ,k}, \boldsymbol{X}, \texttt{fold})$
end for

for $i$ in 1 to 10 do
    for $k$ in 1 to $q$ do
        $\hat{\boldsymbol{B}}_{\circ,k}^{-\kappa(i)} \leftarrow \mathit{ElasticNet}(\boldsymbol{Y}_{\circ,k}^{-\kappa(i)}, \boldsymbol{X}^{-\kappa(i)}, \lambda_{1,k})$
    end for
    for $k$ in 1 to $q$ do
        $\{\boldsymbol{w}_k^{\text{int}}, \boldsymbol{w}_k^{\text{ext}}\} \leftarrow \mathit{Weights}(\hat{\boldsymbol{B}}^{-\kappa(i)}, k)$
        for various $\lambda_{2,k}$, $\delta_k^{\text{int}}$, $\delta_k^{\text{ext}}$ do
            $\boldsymbol{z}_k \leftarrow \mathit{PenaltyFactors}(\boldsymbol{w}_k^{\text{int}}, \boldsymbol{w}_k^{\text{ext}}, \delta_k^{\text{int}}, \delta_k^{\text{ext}})$
            $\hat{\boldsymbol{\Gamma}}_{\circ,k}^{-\kappa(i)} \leftarrow \mathit{AdCoLasso}(\boldsymbol{Y}_{\circ,k}^{-\kappa(i)}, \boldsymbol{X}^{-\kappa(i)}, \lambda_{2,k}, \boldsymbol{z}_k)$
            $\hat{\boldsymbol{Y}}_{\lambda_{2,k},\delta_k^{\text{int}},\delta_k^{\text{ext}};k}^{\kappa(i)} \leftarrow \mathit{InverseLink}(\boldsymbol{X}^{\kappa(i)}\hat{\boldsymbol{\Gamma}}_{\circ,k}^{-\kappa(i)})$
        end for
    end for
end for
for $k$ in 1 to $q$ do
    $\{\lambda_{2,k}, \delta_k^{\text{int}}, \delta_k^{\text{ext}}\} \leftarrow \mathit{MinimiseMetric}(\boldsymbol{Y}_{\circ,k}, \hat{\boldsymbol{Y}}_{\circ,\circ,\circ;k})$
end for

Parameter estimation

for $k$ in 1 to $q$ do
    $\hat{B}_{\circ,k} \leftarrow \mathit{ElasticNet}(\boldsymbol{Y}_{\circ,k}, \boldsymbol{X}, \lambda_{1,k})$
end for
for $k$ in 1 to $q$ do
    $\{\boldsymbol{w}_k^{\text{int}}, \boldsymbol{w}_k^{\text{ext}}\} \leftarrow \mathit{Weights}(\hat{\boldsymbol{B}}, k)$
    $\boldsymbol{z}_k \leftarrow \mathit{PenaltyFactors}(\boldsymbol{w}_k^{\text{int}}, \boldsymbol{w}_k^{\text{ext}}, \delta_k^{\text{int}}, \delta_k^{\text{ext}})$
end for
for $k$ in 1 to $q$ do
    $\hat{\boldsymbol{\Gamma}}_{\circ,k} \leftarrow \mathit{AdCoLasso}(\boldsymbol{y}_{\circ,k}, \boldsymbol{X}, \lambda_{2,k}, \boldsymbol{z}_k)$
end for
```

transfer learning

```
Require:
$n_k \times 1$ vectors $\boldsymbol{y}^{(k)}$ (targets), $\forall k \in \{1, \ldots, q\}$
$n_k \times p$ matrices $\boldsymbol{X}^{(k)}$ (features), $\forall k \in \{1, \ldots, q\}$
Ensure:
$q \times 1$ vectors $\boldsymbol{\lambda_1}$, $\boldsymbol{\lambda_2}$, $\boldsymbol{\delta}^{\text{int}}$, $\boldsymbol{\delta}^{\text{ext}}$ (hyperparameters)
$(p+1) \times q$ matrix $\hat{\boldsymbol{B}}$ (coefficients)

set $n_k \times 1$ vector of fold identifiers $\texttt{fold}_k$,
with values from 1 to 10, $\forall k \in \{1, \ldots, q\}$
initialise various empty $n_k \times 1$ vectors $\hat{\boldsymbol{y}}^{(k)}$, $\forall k \in \{1, \ldots, q\}$

Hyperparameter optimisation

for $k$ in 1 to $q$ do
    $\lambda_{1,k} \leftarrow \mathit{ElasticNetCV}(\boldsymbol{y}^{(k)}, \boldsymbol{X}^{(k)}, \texttt{fold}_k)$
end for

for $k$ in 1 to $q$ do
    $\hat{\boldsymbol{B}}_{\circ,k} \leftarrow \mathit{ElasticNet}(\boldsymbol{y}^{(k)}, \boldsymbol{X}^{(k)}, \lambda_{1,k})$
end for
for $i$ in 1 to 10 do
    for $k$ in 1 to $q$ do
        $\hat{\boldsymbol{B}}^{-\kappa(i)} \leftarrow \hat{\boldsymbol{B}}$
        $\hat{\boldsymbol{B}}_{\circ,k}^{-\kappa(i)} \leftarrow \mathit{ElasticNet}(\boldsymbol{y}^{(k),-\kappa(i)}, \boldsymbol{X}^{(k),-\kappa(i)}, \lambda_{1,k})$
        $\{\boldsymbol{w}_k^{\text{int}}, \boldsymbol{w}_k^{\text{ext}}\} \leftarrow \mathit{Weights}(\hat{\boldsymbol{B}}^{-\kappa(i)}, k)$
        for various $\lambda_{2,k}$, $\delta_k^{\text{int}}$, $\delta_k^{\text{ext}}$ do
            $\boldsymbol{z}_k \leftarrow \mathit{PenaltyFactors}(\boldsymbol{w}_k^{\text{int}}, \boldsymbol{w}_k^{\text{ext}}, \delta_k^{\text{int}}, \delta_k^{\text{ext}})$
            $\hat{\boldsymbol{\Gamma}}_{\circ,k}^{-\kappa(i)} \leftarrow \mathit{AdCoLasso}(\boldsymbol{y}^{(k),-\kappa(i)}, \boldsymbol{X}^{(k),-\kappa(i)}, \lambda_{2,k}, \boldsymbol{z}_k)$
            $\hat{\boldsymbol{y}}_{\lambda_{2,k},\delta_k^{\text{int}},\delta_k^{\text{ext}}}^{(k),\kappa(i)} \leftarrow \mathit{InverseLink}(\boldsymbol{X}^{(k),\kappa(i)}\hat{\boldsymbol{\Gamma}}_{\circ,k}^{-\kappa(i)})$
        end for
    end for
end for
for $k$ in 1 to $q$ do
    $\{\lambda_{2,k}, \delta_k^{\text{int}}, \delta_k^{\text{ext}}\} \leftarrow \mathit{MinimiseMetric}(\boldsymbol{y}^{(k)}, \hat{\boldsymbol{y}}_{\circ,\circ,\circ}^{(k)})$
end for

Parameter estimation

for $k$ in 1 to $q$ do
    $\hat{B}_{\circ,k} \leftarrow \mathit{ElasticNet}(\boldsymbol{y}^{(k)}, \boldsymbol{X}^{(k)}, \lambda_{1,k})$
end for
for $k$ in 1 to $q$ do
    $\{\boldsymbol{w}_k^{\text{int}}, \boldsymbol{w}_k^{\text{ext}}\} \leftarrow \mathit{Weights}(\hat{\boldsymbol{B}}, k)$
    $\boldsymbol{z}_k \leftarrow \mathit{PenaltyFactors}(\boldsymbol{w}_k^{\text{int}}, \boldsymbol{w}_k^{\text{ext}}, \delta_k^{\text{int}}, \delta_k^{\text{ext}})$
end for
for $k$ in 1 to $q$ do
    $\hat{\boldsymbol{\Gamma}}_{\circ,k} \leftarrow \mathit{AdCoLasso}(\boldsymbol{y}_{\circ,k}, \boldsymbol{X}^{(k)}, \lambda_{2,k}, \boldsymbol{z}_k)$
end for
```

Standard subprocedures (details omitted): *ElasticNetCV** (optimising regularisation parameter), *ElasticNet** (estimating regression coefficients), *InverseLink** (transforming linear predictor to predicted value), *MinimiseMetric** (selecting optimal hyperparameters). Subprocedures described in Section 2: *Weights* (calculating internal and external weights), *PenaltyFactors* (calculating penalty factors), *AdCoLasso** (fitting sign-aware adaptive lasso regression). The subprocedures marked with an asterisk differ between linear regression (numerical outcome) and logistic regression (binary outcome). The inclusion or exclusion of all samples from fold i is indicated by $\kappa(i)$ and $-\kappa(i)$, respectively.

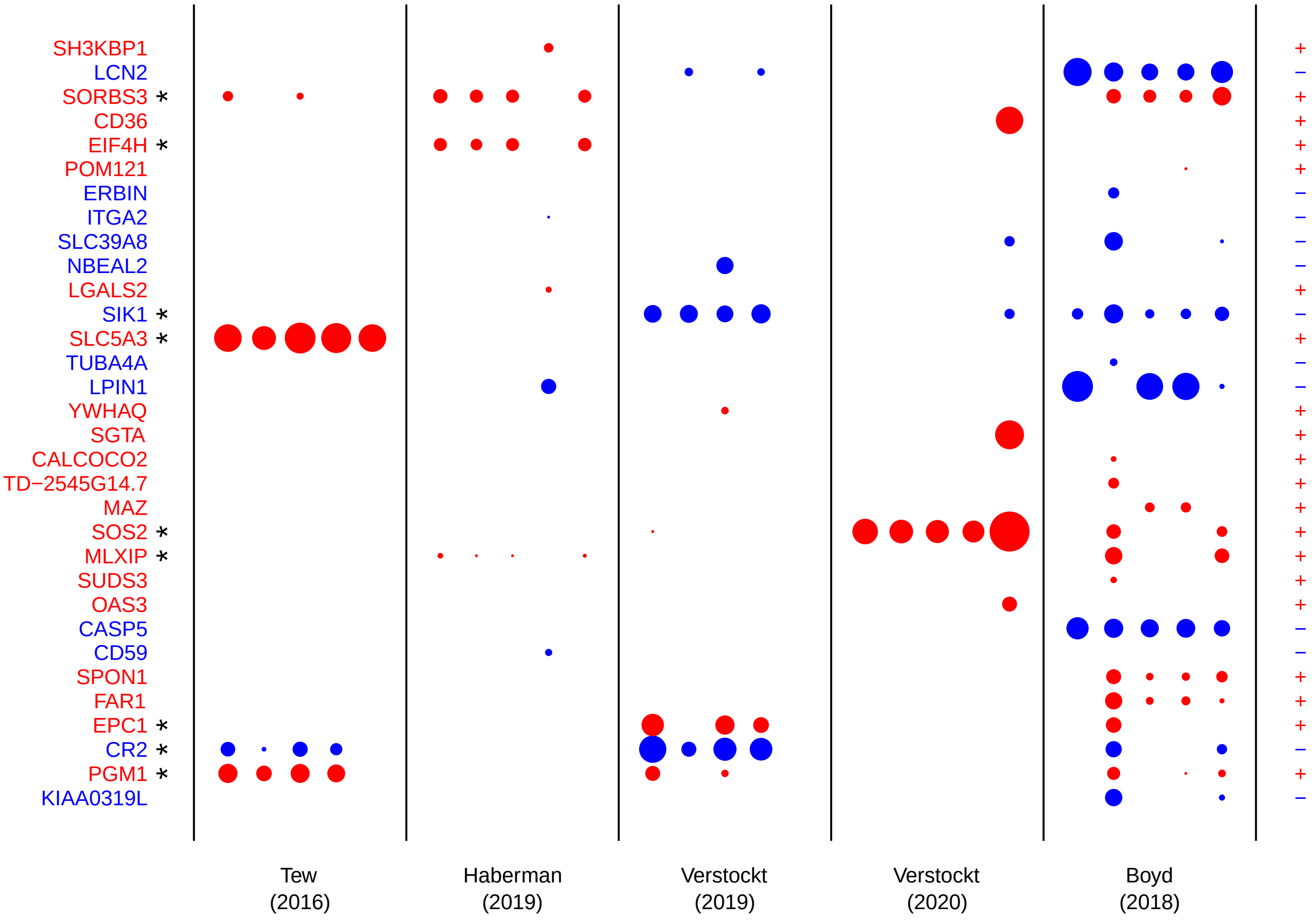

Fig. 6. Standardised coefficients from the proposed transfer learning method for the datasets related to treatment response in IBD, based on five repetitions with different internal folds. Only genes selected for at least one dataset are shown. Blue (red) colour indicates that a gene is estimated to decrease (increase) the probability of treatment response (datasets 1-4) or disease inactivity (dataset 5), with the area of the circles proportional to the absolute value of the estimated effects. An asterisk indicates that a gene is selected in at least three out of five repetitions for at least one prospective dataset on treatment response (datasets 1-4). A plus (minus) sign indicates that a gene receives non-negative (non-positive) mean coefficients for all datasets.

Table 2. Data-generating processes for the simulation studies on multi-task learning (left) and transfer learning (right).

multi-task learning	**transfer learning**
q problems (targets), indexed by $k \in \{1,\ldots,q\}$ p features, indexed by $j,l \in \{1,\ldots,p\}$ $n+m$ samples, indexed by $i \in \{1,\ldots,n+m\}$	q problems (datasets), indexed by $k \in \{1,\ldots,q\}$ p features, indexed by $j,l \in \{1,\ldots,p\}$ n_k+m_k samples, indexed by $i \in \{1,\ldots,n_k+m_k\}$
features	**features**
$(n+m)\times p$ matrix $\boldsymbol{X}$ $X_{i,\circ} \sim \mathrm{N}(0,\boldsymbol{R})$, where $R_{j,l} = \rho^{\lvert j-l\rvert}$	$(n_k+m_k)\times p$ matrices $\boldsymbol{X}^{(k)}$ $X^{(k)}_{i,\circ} \sim \mathrm{N}(0,\boldsymbol{R}^{(k)})$, where $R^{(k)}_{j,l} = \rho_k^{\lvert j-l\rvert}$
effects	**effects**
$p\times q$ matrix $\boldsymbol{B}$ $\boldsymbol{B}_{\circ,k} = \boldsymbol{\theta} + \boldsymbol{\Delta}_{\circ,k}$ $\theta_j \sim \mathrm{Bern}(\pi_\theta)\times\mathrm{N}(0,1)$ $\Delta_{j,k} \sim \mathrm{Bern}(\pi_\delta)\times\mathrm{N}(0,1)$	$p\times q$ matrix $\boldsymbol{B}$ $\boldsymbol{B}_{\circ,k} = \boldsymbol{\theta} + \boldsymbol{\Delta}_{\circ,k}$ $\theta_j \sim \mathrm{Bern}(\pi_\theta)\times\mathrm{N}(0,1)$ $\Delta_{j,k} \sim \mathrm{Bern}(\pi_\delta)\times\mathrm{N}(0,1)$
targets	**targets**
$(n+m)\times q$ matrix $\boldsymbol{Y}$ $\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{B} + \boldsymbol{E}$ $E_{i,k} \sim N(0,1)$	$(n_k+m_k)\times 1$ vectors $\boldsymbol{y}^{(k)}$ $\boldsymbol{y}^{(k)} = \boldsymbol{X}^{(k)}\boldsymbol{B}_{\circ,k} + \boldsymbol{\epsilon}^{(k)}$ $\epsilon^{(k)}_i \sim N(0,1)$

Table 3. Datasets related to treatment response in IBD (1-5) and to treatment response in RA (6-8).

Tew et al. (2016, SRP063496)	70 UC patients were treated with etrolizumab. Gene expression was measured in colonic biopsies at baseline. During follow-up at week 10, 12 remissions and 58 non-remissions were observed.
Haberman et al. (2019, SRP129004)	206 new-onset pediatric UC patients were treated with mesalamine or corticosteroids. Gene expression was measured in rectal mucosal biopsies at baseline. During follow-up at week 4, 105 remissions and 101 non-remissions were observed.
Verstockt et al. (2019, ERP113396)	43 IBD patients were treated with two different tumour necrosis factor inhibitors, namely 27 with adalimumab and 16 with infliximab. Mucosal gene expression was measured in inflamed biopsies. Treatment response was endoscopic remission during follow-up, with 18 responders and 25 non-responders.
Verstockt et al. (2020, ERP114636)	47 IBD patients, namely 20 with Crohn's disease (CD) and 27 with ulcerative colitis (UC), were treated with vedolizumab. Gene expression was measured in biopsies from inflamed colon. Treatment response is endoscopic remission during follow-up, with 24 responders and 23 non-responders.
Boyd et al. (2018, SRP100787)*	64 IBD patients, namely 22 with CD and 42 with UC, were assessed for disease activity. Gene expression was measured in pinch biopsies from the descending colon at baseline. Disease activity was defined based on two different scores for CD and UC, leading to 20 patients with an inactive and 44 patients with an active disease at baseline.
Baker et al. (2019, SRP169062)†	44 patients with RA were observed for drug-free remission. Gene expression was measured in CD4+ T cells from peripheral blood samples. Drug-free remission was defined based on a disease activity score in multiple joints, leading to 21 patients without flares and 23 patients with flares.
Moncrieffe et al. (2017, SRP074736)†	46 patients with juvenile idiopathic arthritis, which is different from RA, were treated with methotrexate. Gene expression was measured in peripheral blood mononuclear cells. Treatment response was defined based on six scores, with a large increase in at least three scores and a large decrease in at most one score, leading to 29 responders and 17 non-responders.
Goldberg et al. (2018, SRP155483)*	48 patients with RA were observed. Gene expression was measured in white blood cells from peripheral blood. 24 patients under remission are to be compared to $3 + 15 + 6 = 24$ patients with low, moderate or high disease activity.

*No prospective dataset on treatment response but cross-sectional dataset on disease activity.

†No dataset on treatment response in IBD or RA but closely related.

Table 4. Spearman correlation coefficients between the ridge regression coefficients from different datasets. Pairwise combinations of datasets with significantly correlated regression coefficients are highlighted, with black colour for nominal significance (p-value ≤ 0.05) and stars for adjusted significance (p-value $\leq 0.05/28$). We expect a correlation coefficient close to 0 for unrelated problems and close to 1 for identical problems.

	Tew (2016)	Haberman (2019)	Verstockt (2019)	Verstockt (2020)	Boyd (2018)	Baker (2019)	Moncrieffe (2017)	Goldberg (2018)
Tew (2016)	-	$0.23^\star$	$0.34^\star$	$0.22^\star$	$0.18^\star$	-0.02	-0.01	-0.01
Haberman (2019)	$0.23^\star$	-	$0.26^\star$	$0.29^\star$	$0.22^\star$	$0.08^\star$	$0.06^\star$	-0.01
Verstockt (2019)	$0.34^\star$	$0.26^\star$	-	$0.35^\star$	$0.15^\star$	-0.02	$0.08^\star$	0.04
Verstockt (2020)	$0.22^\star$	$0.29^\star$	$0.35^\star$	-	$0.15^\star$	$0.05^\star$	$0.07^\star$	-0.04
Boyd (2018)	$0.18^\star$	$0.22^\star$	$0.15^\star$	$0.15^\star$	-	$0.10^\star$	-0.04	-0.04
Baker (2019)	-0.02	$0.08^\star$	-0.02	$0.05^\star$	$0.10^\star$	-	0.02	0.02
Moncrieffe (2017)	-0.01	$0.06^\star$	$0.08^\star$	$0.07^\star$	-0.04	0.02	-	$-0.05^\star$
Goldberg (2018)	-0.01	-0.01	0.04	-0.04	-0.04	0.02	$-0.05^\star$	-

Table 5. Out-of-sample area under the receiver operating characteristic curve (ROC-AUC) from logistic ridge regression trained on the dataset in the row and tested on the dataset in the column (off-diagonal entries), or cross-validated ROC-AUC from logistic lasso regression trained and tested on the same dataset by 10-fold external cross-validation (diagonal entries, between brackets). The ROC-AUC of a random classifier is 0.5, while that of a perfect classifier is 1.0. Entries on and off the diagonal are not comparable. Predictions that are significantly better than random predictions (according to the one-sided Mann-Whitney U test for testing whether the ranks of the predicted probabilities are significantly higher for the cases than for the controls) are highlighted, with black colour for nominal significance (p-value ≤ 0.05) and stars for adjusted significance (p-value $\leq 0.05/64$).

	Tew (2016)	Haberman (2019)	Verstockt (2019)	Verstockt (2020)	Boyd (2018)	Baker (2019)	Moncrieffe (2017)	Goldberg (2018)
Tew (2016)	(0.69)	0.61	0.71	0.66	0.78*	0.48	0.51	0.43
Haberman (2019)	0.55	(0.54)	0.66	0.72	0.79*	0.58	0.56	0.40
Verstockt (2019)	0.62	0.59	(0.44)	0.72	0.72	0.44	0.52	0.56
Verstockt (2020)	0.57	0.59	0.65	(0.49)	0.74	0.53	0.57	0.48
Boyd (2018)	0.58	0.65*	0.67	0.67	(0.87)*	0.65	0.49	0.46
Baker (2019)	0.48	0.54	0.46	0.53	0.76*	(0.46)	0.50	0.51
Moncrieffe (2017)	0.44	0.50	0.59	0.62	0.42	0.52	(0.44)	0.45
Goldberg (2018)	0.48	0.52	0.58	0.44	0.41	0.46	0.46	(0.54)

Funding

This research was funded by the Luxembourg National Research Fund (FNR) for the project "Clinnova - Federating Digital Medicine in Europe" [NCER/23/16695277]. For the purpose of open access, and in fulfilment of the obligations arising from the grant agreement, the authors have applied a Creative Commons Attribution 4.0 International (CC BY 4.0) license to any author accepted manuscript version arising from this work.

Author contributions

EG acquired funding. AR developed and implemented the method. AR analysed the data and drafted the manuscript. PVN and EG critically revised the manuscript. All authors read and approved the final version of the manuscript.

Acknowledgments

The authors are grateful to Miroslav Kratochvil for troubleshooting in LaTeX and Git, and to two anonymous reviewers for constructive criticism.

Data and code availability

The R package `sparselink` is available on GitHub (`https://github.com/rauschenberger/sparselink`) and CRAN (`https://cran.r-project.org/package=sparselink`) under the MIT license and includes the analysis code in a vignette. Data are accessible via the R package `recount3` (Wilks et al., 2021), available on Bioconductor (`https://bioconductor.org/packages/recount3/`), under the project identifiers (INSDC accession numbers) SRP063496 (Tew et al., 2016), SRP129004 (Haberman et al., 2019), ERP113396 (Verstockt et al., 2019), ERP114636 (Verstockt et al., 2020), SRP100787 (Boyd et al., 2018), SRP074736 (Moncrieffe et al., 2017), SRP169062 (Baker et al., 2019), and SRP155483 (Goldberg et al., 2018).

References

Baker, K. F. et al. (2019). Predicting drug-free remission in rheumatoid arthritis: A prospective interventional cohort study. Journal of Autoimmunity, 105:102298. doi: 10.1016/j.jaut.2019.06.009 (INSDC accession number: SRP169062).

Bergersen, L. C., Glad, I. K., and Lyng, H. (2011). Weighted lasso with data integration. Statistical Applications in Genetics and Molecular Biology, 10(1):39. doi: 10.2202/1544-6115.1703.

Boyd, M. et al. (2018). Characterization of the enhancer and promoter landscape of inflammatory bowel disease from human colon biopsies. Nature Communications, 9(1):1661. doi: 10.1038/s41467-018-03766-z (INSDC accession number: SRP100787).

Chung, D. and Keleş, S. (2010). Sparse partial least squares classification for high dimensional data. Statistical Applications in Genetics and Molecular Biology, 9(1):17. doi: 10.2202/1544-6115.1492 (`spls`).

Friedman, J. H., Hastie, T., and Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. Journal of Statistical Software, 33(1):1–22. doi: 10.18637/jss.v033.i01 (R package `glmnet`).

Goldberg, G. et al. (2018). White blood cells from rheumatoid arthritis patients and matched healthy donors. Available from `https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE117769`. (INSDC accession number: SRP155483).

Haberman, Y. et al. (2019). Ulcerative colitis mucosal transcriptomes reveal mitochondriopathy and personalized mechanisms underlying disease severity and treatment response. Nature Communications, 10(1):38. doi: 10.1038/s41467-018-07841-3 (INSDC accession number: SRP129004).

Moncrieffe, H. et al. (2017). Transcriptional profiles of JIA patient blood with subsequent poor response to methotrexate. Rheumatology, 56(9):1542–1551. doi: 10.1093/rheumatology/kex206 (INSDC accession number: SRP074736).

Rauschenberger, A. and Glaab, E. (2021). Predicting correlated outcomes from molecular data. Bioinformatics, 37(21):3889–3895. doi: 10.1093/bioinformatics/btab576. (R package `joinet`).

Rauschenberger, A., Landoulsi, Z., van de Wiel, M. A., and Glaab, E. (2023). Penalized regression with multiple sources of prior effects. Bioinformatics, 39(12):btad680. doi: 10.1093/bioinformatics/btad680 (R/CRAN package: transreg).

Robinson, M. D. and Oshlack, A. (2010). A scaling normalization method for differential expression analysis of RNA-seq data. Genome Biology, 11:R25. doi: 10.1186/gb-2010-11-3-r25 (R/Bioconductor package: edgeR).

Simon, N., Friedman, J. H., and Hastie, T. (2013). A blockwise descent algorithm for group-penalized multiresponse and multinomial regression. arXiv (preprint). doi: 10.48550/arXiv.1311.6529 (R/CRAN package: glmnet).

Tew, G. W. et al. (2016). Association between response to etrolizumab and expression of integrin αe and granzyme a in colon biopsies of patients with ulcerative colitis. Gastroenterology, 150(2):477–487.e9. doi: 10.1053/j.gastro.2015.10.041 (INSDC access number: SRP063496).

Tian, Y. and Feng, Y. (2023). Transfer learning under high-dimensional generalized linear models. Journal of the American Statistical Association, 118(544):2684–2697. doi: 10.1080/01621459.2022.2071278. (R package `glmtrans`).

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society, Series B (Statistical Methodology), 58(1):267–288. doi: 10.1111/j.2517-6161.1996.tb02080.x.

Verstockt, B. et al. (2019). Low TREM1 expression in whole blood predicts anti-TNF response in inflammatory bowel disease. EBioMedicine, 40:733–742. doi: 10.1016/j.ebiom.2019.01.027 (INSDC accession number: ERP113396).

Verstockt, B. et al. (2020). Expression levels of 4 genes in colon tissue might be used to predict which patients will enter endoscopic remission after vedolizumab therapy for inflammatory bowel diseases. Clinical Gastroenterology and Hepatology, 18(5):1142–1151.e10. doi: 10.1016/j.cgh.2019.08.030 (INSDC accession number: ERP114636).

Weaver, G. M. and Lewinger, J. P. (2019). xrnet: Hierarchical regularized regression to incorporate external data. Journal of Open Source Software, 4(44):1761. doi: 10.21105/joss.01761 (R/CRAN package: xrnet).

Wilks, C. et al. (2021). recount3: summaries and queries for large-scale RNA-seq expression and splicing. Genome Biology, 22:323. doi: 10.1186/s13059-021-02533-6 (R/Bioconductor package: recount3).

Zou, H. (2006). The adaptive lasso and its oracle properties. Journal of the American Statistical Association, 101(476):1418–1429. doi: 10.1198/016214506000000735.

Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 67(2):301–320. doi: 10.1111/j.1467-9868.2005.00503.x.