Bioinformatics, 2023, pp. 1–15

doi: 10.1093/bioinformatics/XXX
Advance Access Publication Date: Day Month Year
Original Paper

ORIGINAL PAPER

Penalised regression with multiple sources of prior effects

Armin Rauschenberger[1,*] Zied Landoulsi[1] Mark A. van de Wiel[2,†] and Enrico Glaab[1,†]

[1]Luxembourg Centre for Systems Biomedicine (LCSB), University of Luxembourg, Esch-sur-Alzette, Luxembourg and [2]Department of Epidemiology and Data Science (EDS), Amsterdam University Medical Centers (Amsterdam UMC), Amsterdam, The Netherlands

[*]Corresponding author. armin.rauschenberger@uni.lu[†]Mark A. van de Wiel and Enrico Glaab share senior authorship.

FOR PUBLISHER ONLY Received on 17 April 2023; revised on 2 October 2023; accepted on Date Month Year

Abstract

In many high-dimensional prediction or classification tasks, complementary data on the features are available, e.g. prior biological knowledge on (epi)genetic markers. Here we consider tasks with numerical prior information that provide an insight into the importance (weight) and the direction (sign) of the feature effects, e.g. regression coefficients from previous studies. We propose an approach for integrating multiple sources of such prior information into penalised regression. If suitable co-data are available, this improves the predictive performance, as shown by simulation and application. The proposed method is implemented in the R package `transreg` (`https://github.com/lcsb-bds/transreg`, `https://cran.r-project.org/package=transreg`).

Key words: transfer learning, co-data, prior information, ridge regression

Background

For many biomedical prediction or classification studies, there is a previous study with a similar target and a similar high-dimensional feature space, e.g. hundreds of microRNAs (miRNAs), thousands of genes, or millions of single-nucleotide polymorphisms (SNPs). Given a trained model from a previous study, we could use it to obtain predicted values or predicted probabilities for the study of interest, but these predictions are only reliable if the two studies have the same target, the same features, and the same population. However, we expect the feature-target effects from two studies to be strongly correlated in more situations: slightly different targets (e.g. disease status vs disease stage), slightly different features (e.g. imperfectly overlapping feature space, different measurement technique), slightly different populations (e.g. hospitalised vs non-hospitalised patients), or even different modelling approaches (e.g. simple regression vs multiple regression). As it is challenging to estimate feature-target effects in high-dimensional settings, it might be advantageous to use results from previous studies as prior information for the study of interest.

Consider two prediction or classification problems, each one with a target vector and a feature matrix (samples in the rows, features in the columns). Suppose that both feature matrices cover the same features (each column in the first matrix corresponds to a column in the second matrix). In two special cases, the two problems reduce to a single problem: (i) If both problems have the same target and concern samples from the same population, they are in essence one ‘single-target’ problem (combine target vectors and feature matrices by rows, respectively), potentially with batch effects. (ii) If both problems concern the same samples, they are in essence one ‘multi-target’ problem (combine target vectors by columns, feature matrices are the same). In other cases, however, we might be in a transfer learning setting (Table 1).

In such settings - two or more regression problems with related targets and matched features - it might be possible to transfer information from one problem to another. Transfer learning, in contrast to multi-target regression and seemingly unrelated regressions, addresses related regression problems not for the same but for different samples. If the regression problems are sufficiently related to each other, we expect their regression coefficients to be correlated (positively or negatively). When fitting the regression model of interest, we could therefore account for the estimated regression coefficients from the other model. Transferring information on the importance and the direction of the feature effects, we could potentially increase the predictive performance.

Jiang et al. (2016) proposed the prior lasso to account for prior information in high-dimensional predictive modelling. Their method involves a preprocessing step and a weighting step. In the preprocessing step, the prior information is used to predict the target from the features. They present a solution for one set of prior effects from a closely related study (multiplying the feature matrix by the prior effects), but extensions to multiple sets of prior effects or loosely related studies may be feasible. Let $\boldsymbol{y}$ represent the target and let $\hat{\boldsymbol{y}}_{\text{prior}}$ represent the fitted values based on the prior information. In the weighting step, they minimise the penalised combined likelihood $L(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{\beta}) + \eta L(\boldsymbol{x}, \hat{\boldsymbol{y}}_{\text{prior}}; \boldsymbol{\beta}) - \rho(\lambda; \boldsymbol{\beta})$ with respect to the coefficients $\boldsymbol{\beta}$, where $\eta \geq 0$ (balance) and $\lambda \geq 0$

(regularisation). If the balancing hyperparameter η is larger than zero, the prior predictions $\hat{\boldsymbol{y}}_{\text{prior}}$ influence the estimation of the parameters $\boldsymbol{\beta}$.

Dhruba (2021) proposed a transfer learning method based on distribution mapping. Even if features or targets follow different distributions in two data sets, it is possible to build a predictive model using the first data set and make predictions for the second data set. Requiring matched features and targets in the source data set and unmatched features and targets in the target data set, their method transfers (i) features from the target to the source domain and (ii) predictions from the source to the target domain. By contrast, we consider transfer learning settings with matched features and targets in the target data set.

Tian and Feng (2022) proposed and implemented transfer learning for ridge and lasso regression. Their transfer learning algorithm involves two steps: (i) Estimating common coefficients for the target data set and the transferable source data sets ($\hat{\boldsymbol{\omega}}$). (ii) Estimating the deviations from the common coefficients to the target coefficients ($\hat{\boldsymbol{\delta}}$). Both steps together lead to the estimated target coefficients ($\hat{\boldsymbol{\beta}} = \hat{\boldsymbol{\omega}} + \hat{\boldsymbol{\delta}}$). Before applying their transfer learning algorithm, Tian and Feng (2022) apply a transferable source detection algorithm to exclude source data sets that are too different from the target data set. This avoids that non-transferable sources render the common coefficients misleading for the target data set ('negative transfer'). In the case of lasso regularisation in the two steps, there is sparsity in the common estimates as well as in the deviations from the common estimates to the target estimates (and thereby also in the target estimates).

The method from Tian and Feng (2022) requires not only the target data set but also the source data set(s). However, data protection regulations or restrictive data sharing policies might prevent researchers from accessing a source data set, or the available storage or processing capacity might be insufficient for analysing massive source data sets. Although federated transfer learning (Liu et al., 2020) enables the joint analysis of related data sets without sharing them, it is often not practically feasible for biomedical researchers to set up corresponding collaborative analyses with multiple data holders. There is therefore a need for transfer learning methods that require neither direct nor indirect access to the source data but only to the complementary data (co-data) derived from the source data. Such methods allow us to exploit publicly available summary statistics from external studies, e.g. p-values and effect sizes from a genome-wide association study (GWAS), to increase the predictive performance in the study of interest.

We propose a two-step transfer learning method, modelling with and without co-data in the first step and combining different models in the second step. Unless the source and target data sets are very similar, the coefficients from the source data set(s) will not fit well to the target data set. We therefore propose to calibrate these coefficients - preserving their signs and their order - so that they can be transferred from the source data set(s) to the target data set. Additionally, we also estimate the coefficients directly from the target data set, ignoring the co-data. The calibrated coefficients from the source data set(s) as well as the estimated coefficients from the target data set allow us to predict the outcome from the features. Finally, we combine the linear predictors from the models with and without co-data and calculate either predicted values (linear regression) or predicted probabilities (logistic regression).

Kawaguchi et al. (2021) proposed a related transfer learning method. This method penalises (i) the differences between the coefficients and weighted sums of the prior effects (one difference for each feature) and (ii) the weights for these weighted sums (one weight for each set of prior effects), possibly with two different penalties (e.g. ridge and lasso). It not only allows for high-dimensional data (i.e. more features than samples) but also for high-dimensional co-data (i.e. more sets of prior effects than samples). When the unknown feature effects are non-linearly related to an important source of prior effects, however, this method might leave room for improvement.

In a related transfer learning setting, prior information is only available on the importance but not on the direction of the feature effects, i.e. with complementary data consisting of prior weights rather than prior effects. In the generalised linear model framework, the weighted lasso (Bergersen et al., 2011), the feature-weighted elastic net (Tay et al., 2022), and penalised regression with differential shrinkage (Zeng et al., 2021) account for prior weights in the penalty function, through feature-specific penalty factors or feature-specific regularisation parameters. Adaptive group-regularised ridge regression (van de Wiel et al., 2016) is not only applicable to categorical co-data but also to numerical co-data (prior weights), by the means of creating groups of features from numerical co-data and forcing the group-penalties to be monotonically decreasing. An extension from van Nee et al. (2021) makes this approach even more suitable for numerical co-data. For single sources of co-data, it might be possible to extend these methods to prior information on the importance as well as the direction of feature effects by imposing sign constraints on the coefficients. Prior weights have not only been exploited in regression analysis, e.g. co-data moderated random forests (te Beest et al., 2017) adapt the sampling probabilities of the features to the prior weights.

Method

Model

Suppose one target and p features are available for n samples, with many more features than samples ($p \gg n$). We index the samples by i in $\{1, \ldots, n\}$ and the features by j in $\{1, \ldots, p\}$. Our aim is to estimate the generalised linear model

$$\mathbb{E}[y_i] = h^{-1}\left(\beta_0 + \sum_{j=1}^{p} \beta_j x_{ij}\right) .$$

For any sample i, the model expresses the expected value of its target (y_i) as a function of its features ($x_{i1}, \ldots, x_{ip}$). The link function $h(\cdot)$ depends on the family of distributions for the target (Gaussian: identity, binomial: logit, Poisson: log). In the linear predictor, β_0 represents the unknown intercept, and β_j represents the unknown slope of feature j (i.e. the effect of the feature on the linear predictor of the target). Given the estimated intercept $\hat{\beta}_0^\star$ and the estimated slopes $\{\hat{\beta}_1^\star, \ldots, \hat{\beta}_p^\star\}$, we could predict the target of previously unseen samples:

$$\hat{y}_i = h^{-1}\left(\hat{\beta}_0^\star + \sum_{j=1}^{p} \hat{\beta}_j^\star x_{ij}\right) .$$

Co-data

Suppose m sources of co-data are available, indexed by k in $\{1, \ldots, m\}$, with many fewer sources than samples ($m \ll n$). Let z_{jk} indicate the prior effect from source k for feature j. Our method is designed for quantitative co-data that provide an insight into the importance (absolute value) and the direction (sign) of the feature effects. Each set of prior effects ($-1 \leq \boldsymbol{z}_{\circ k} \leq 1$) is assumed to be positively correlated with the true coefficients ($\text{cor}(\boldsymbol{z}_{\circ k}, \boldsymbol{\beta}) > 0$). For any source of co-data, the prior effects may be re-scaled (not re-centred), for example to the interval from -1 to $+1$. In other

Table 1. Abstract representation of the data set of interest (without asterisk, black) and an additional data set (with asterisk, grey). Single-target learning (left): same targets, same features*, different samples (from one population). Multi-target learning (centre): different targets, same features*, same samples. Transfer learning (right): same or different targets, matched features*, different samples (from one or two populations). *Partially overlapping feature spaces are also possible. See, for example, Rauschenberger and Glaab (2021) for multi-target learning or Section 2.8 for transfer learning.

Single-target learning

$$\begin{pmatrix} y_1 \\ \vdots \\ y_n \\ y_1^* \\ \vdots \\ y_m^* \end{pmatrix} \Leftarrow \begin{pmatrix} x_{11} & \cdots & x_{1p} \\ \vdots & & \vdots \\ x_{n1} & \cdots & x_{np} \\ x_{11}^* & \cdots & x_{1p}^* \\ \vdots & & \vdots \\ x_{m1}^* & \cdots & x_{mp}^* \end{pmatrix}$$

Multi-target learning

$$\begin{pmatrix} y_1 & y_1^* \\ \vdots & \vdots \\ y_n & y_n^* \end{pmatrix} \Leftarrow \begin{pmatrix} x_{11} & \cdots & x_{1p} \\ \vdots & & \vdots \\ x_{n1} & \cdots & x_{np} \end{pmatrix}$$

Transfer learning

$$\begin{array}{ccc} \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix} & \Leftarrow & \begin{pmatrix} x_{11} & \cdots & x_{1p} \\ \vdots & & \vdots \\ x_{n1} & \cdots & x_{np} \end{pmatrix} \\ & & \Updownarrow \qquad \Updownarrow \qquad \Updownarrow \\ \begin{pmatrix} y_1^* \\ \vdots \\ y_m^* \end{pmatrix} & \Leftarrow & \begin{pmatrix} x_{11}^* & \cdots & x_{1p}^* \\ \vdots & & \vdots \\ x_{m1}^* & \cdots & x_{mp}^* \end{pmatrix} \end{array}$$

words, the proposed method is invariant under multiplication of the prior effects by a positive scalar ($\boldsymbol{z} \rightarrow c \times \boldsymbol{z}$ where $c > 0$). We explain in the next section why this is important.

It might seem trivial to also allow for co-data that only provide an insight into the importance but not the direction of the feature effects (i.e. prior weights instead of prior effects). Each set of prior weights ($0 \leq \boldsymbol{z}_{\circ k} \leq 1$) is assumed to be positively correlated with the true absolute coefficients ($\text{cor}(\boldsymbol{z}_{\circ k}, |\boldsymbol{\beta}|) > 0$). To obtain prior effects, one might want to assign the signs of the Spearman correlation coefficients between the target and the features to the prior weights. However, marginal effects and conditional effects can have opposite signs. If we wanted to extend our approach to prior weights, we would have to discover the signs inside the calibration procedure (see below), which would be related to high-dimensional regression with binary coefficients (Gamarnik and Zadik, 2017).

Base-learners with co-data

Suppose we are in a transfer learning setting with two prediction or classification problems. For simplicity, we assume that the features do not differ in scale between the two problems. For illustration, we consider two artificial situations (where we would not use transfer learning in practice): (i) If both problems concern the same target on the same scale and the samples come from the same population, we could use the estimated regression coefficients from one problem to make predictions for the other problem. (ii) If the two problems concern the same target on different scales, we could also recycle the estimated regression coefficients, but we would have to adjust for the different scales.

When transferring estimated regression coefficients from one problem to another problem, it might not only be necessary to change their scale but it might also be beneficial to change their shape. For example, it might be that for one problem weak and strong effects matter, while for the other problem only strong effects matter. We should therefore also be able to make differences between small coefficients more or less important than those between large coefficients. We propose two calibration methods, namely exponential and isotonic calibration, to adapt the prior information to the data. For each source of co-data k, both calibration methods estimate the model

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

where the calibrated prior effects $\{\hat{\gamma}_{1k}, \ldots, \hat{\gamma}_{pk}\}$ depend on the initial prior effects $\{z_{1k}, \ldots, z_{pk}\}$. The difference between exponential and isotonic calibration is how the former depend on the latter.

- exponential calibration: Let $\gamma_{jk} = \theta_k \text{sign}(z_{jk})|z_{jk}|^{\tau_k}$, for j in $\{1, \ldots, p\}$, where the factor θ_k and the exponent τ_k are non-negative real numbers ($\theta_k \geq 0$ and $\tau_k \geq 0$). We first fit one simple non-negative regression for different values of τ_k (i.e. estimate α_k and θ_k given τ_k), and then optimise τ_k. Once θ_k and τ_k have been estimated, the initial prior effects z_{jk} determine the final prior effects $\hat{\gamma}_{jk} = \hat{\theta}_k \text{sign}(z_{jk})|z_{jk}|^{\hat{\tau}_k}$, for all j in $\{1, \ldots, p\}$. The estimated factor $\hat{\theta}_k$ and the estimated exponent $\hat{\tau}_k$ allow the model to change the scale and the shape of the prior effects. For example, $\hat{\theta}_k = 0$ sets them to zero, $|\hat{\theta}_k| < 1$ makes them smaller, $|\hat{\theta}_k| > 1$ makes them larger, $\hat{\tau}_k = 0$ sets them to the same value, $\hat{\tau}_k < 1$ makes (absolutely) large ones more similar, and $\hat{\tau}_k > 1$ makes (absolutely) small ones more similar. If one or more sets of prior effects might be negatively associated with the true coefficients, we could remove the non-negativity constraints from the simple regressions (allowing $\hat{\theta}_k < 0$ to invert the signs of the prior effects).
- isotonic calibration: We estimate $\{\gamma_{1k}, \ldots, \gamma_{pk}\}$ under the constraint that the signs of the initial prior effects z_{jk} determine the signs of the final prior effects $\hat{\gamma}_{jk}$ (i.e. $\hat{\gamma}_{jk} = 0 | z_{jk} = 0$, $\hat{\gamma}_{jk} \geq 0 | z_{jk} > 0$, $\hat{\gamma}_{jk} \leq 0 | z_{jk} < 0$) and under the constraint that the order of the initial prior effects determines the order of the final prior effects (i.e. $\hat{\gamma}_{jk} \geq \hat{\gamma}_{lk} | z_{jk} \geq z_{lk}$, $\hat{\gamma}_{jk} \leq \hat{\gamma}_{lk} | z_{jk} \leq z_{lk}$), for all j and l in $\{1, \ldots, p\}$. If one or more sets of prior effects might be negatively associated with the true coefficients, we could fit each model with these constraints and the inverted constraints, and then select the better fit.

To make optimisation more efficient, we rewrite the sign- and order-constrained problem as a sign-constrained problem (see Table 4 in the Appendix). For each source of co-data, we order the columns of the feature matrix by increasing values of the prior effects. Suppose the first q columns correspond to negative prior effects and the last $p - q$ columns correspond to non-negative prior effects. We take the cumulative sum of the feature columns from left to right for the former (columns 1 to q) and from right to left for the latter (columns p to $q + 1$). We then estimate the coefficients on the left under non-positivity constraints, and those on the right under non-negativity constraints. Formally, the model equals

$$\mathbb{E}[y_i] = h^{-1}\left(\alpha_k + \sum_{j=1}^{p} \delta_{jk} w_{ij}\right) ,$$

where $w_{ij} = \sum_{l=1}^{j} x_{i(l)}$ and $\delta_{jk} \leq 0$ for j in $\{1, \ldots, q\}$, and $w_{ij} = \sum_{l=j}^{p} x_{i(l)}$ and $\delta_{jk} \geq 0$ for j in $\{q + 1, \ldots, p\}$, with the subscript within brackets indicating the order of the prior effects. The linear predictor of the sign-constrained model, i.e. $\alpha_k + \sum_{j=1}^{p} \delta_{jk} w_{ij}$, is equivalent to the linear predictor of the order-constrained model, i.e. $\alpha_k + \sum_{j=1}^{p} \gamma_{(j)k} x_{i(j)}$, because

$$\begin{aligned}
\sum_{j=1}^{q} \delta_{jk} w_{ij} &= \sum_{j=1}^{q} \delta_{jk} \left(\sum_{l=1}^{j} x_{i(l)} \right) \\
&= \sum_{j=1}^{q} \left(\sum_{l=j}^{q} \delta_{lk} \right) x_{i(j)} \\
&= \sum_{j=1}^{q} \gamma_{(j)k} x_{i(j)} \ , \\
\sum_{j=q+1}^{p} \delta_{jk} w_{ij} &= \sum_{j=q+1}^{p} \delta_{jk} \left(\sum_{l=j}^{p} x_{i(l)} \right) \\
&= \sum_{j=q+1}^{p} \left(\sum_{l=q+1}^{j} \delta_{lk} \right) x_{i(j)} \\
&= \sum_{j=q+1}^{p} \gamma_{(j)k} x_{i(j)} \ .
\end{aligned}$$

After estimating the coefficients of the sign-constrained model by maximum likelihood, we therefore estimate those of the order-constrained model by $\hat{\gamma}_{(j)k} = \sum_{l=j}^{q} \hat{\delta}_{lk}$ for j in $\{1, \ldots, q\}$ and $\hat{\gamma}_{(j)k} = \sum_{l=q+1}^{j} \hat{\delta}_{lk}$ for j in $\{q+1, \ldots, p\}$.

While exponential calibration involves three unknown parameters, namely the intercept α_k, the factor θ_k and the exponent τ_k, isotonic calibration involves $1+p$ unknown parameters, namely the intercept α_k and the slopes $\boldsymbol{\gamma}_k = \{\gamma_{1k}, \ldots, \gamma_{pk}\}$, for each set of co-data. Figure 1 shows the difference between exponential and isotonic calibration in several empirically assessed scenarios.

After calibration, we pre-assess the utility of each set of co-data. To do this, we calculate the residuals (depending on the family of distributions) between the fitted and the observed targets. We suggest to retain a set of co-data only if the residuals are significantly smaller than those of the intercept-only model (one-sided Wilcoxon signed-rank test) at the nominal 5% level (p-value ≤ 0.05).

Base-learners without co-data

We also fit the model without any co-data. We estimate the coefficients by maximising the penalised likelihood:

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\operatorname{argmax}} \left\{ L(\boldsymbol{x}; \boldsymbol{\beta}) - \rho(\lambda; \boldsymbol{\beta}) \right\} \ ,$$

where $L(\boldsymbol{x}, \boldsymbol{\beta})$ is the likelihood and $\rho(\lambda; \boldsymbol{\beta})$ is the penalty. The likelihood depends on the family of distributions (Gaussian, binomial, Poisson), and the penalty can be the ridge (L_2) or the lasso (L_1) penalty. The penalty shrinks the squared (ridge) or absolute (lasso) slopes $\{\beta_1, \ldots, \beta_p\}$ towards zero (without penalising the intercept β_0). We denote the estimated intercept by $\hat{\beta}_0$ and the estimated slopes by $\{\hat{\beta}_1, \ldots, \hat{\beta}_p\}$.

Cross-validation

We split the samples into ten folds to perform 10-fold internal cross-validation. In each iteration, we fit the models to nine included folds and predict the target for the excluded fold.

Let the $n \times m$ matrix $\hat{\boldsymbol{H}}^{(0,\text{cv})}$ represent the feature-dependent part of the cross-validated linear predictors from the models with co-data. Specifically, the entry in row i (sample) and column k (source of co-data) equals

$$\eta_{ik}^{(0,\text{cv})} = 0 \times \hat{\alpha}_k^{-\kappa(i)} + \sum_{j=1}^{p} \hat{\gamma}_{jk}^{-\kappa(i)} x_{ij} \ ,$$

where the superscript $-\kappa(i)$ indicates that the (ignored) intercept α_k and the slopes γ_{jk} for j in $\{1, \ldots, p\}$ are estimated without using the fold of sample i, as in Rauschenberger and Glaab (2021).

The models without any co-data do not only have $1+p$ unknown parameters, namely the intercept β_0 and the slopes $\{\beta_1, \ldots, \beta_p\}$, but also the unknown hyperparameter λ. In each iteration, we fit this model for a decreasing sequence of 100 values for the regularisation parameter λ, indexed by l in $\{1, \ldots, 100\}$, using the computationally efficient approach from Friedman et al. (2010, `glmnet`).

Accordingly, let the $n \times 100$ matrix $\hat{\boldsymbol{H}}^{(1,\text{cv})}$ represent the cross-validated linear predictors from the model without co-data. Specifically, the entry in row i (sample) and column l (regularisation parameter) equals

$$\eta_{il}^{(1,\text{cv})} = \hat{\beta}_0^{-\kappa(i),l} + \sum_{j=1}^{p} \hat{\beta}_j^{-\kappa(i),l} x_{ij} \ ,$$

where the superscripts $-\kappa(i)$ and l indicate that the intercept β_0 and the slopes $\{\beta_1, \ldots, \beta_p\}$ are estimated without using the fold of sample i and given the regularisation parameter λ_l.

To optimise the predictive performance of the co-data independent model, we would select the λ that minimises the cross-validated loss (λ_{min}). As we base our predictions not only on the co-data independent model but also on the co-data dependent model(s), λ_{min} might be too small. The reason is that the co-data might be informative to the extent that the co-data independent model requires more penalisation. We could let the meta-learner select the optimal λ from the whole sequence, but this might render the inclusion and exclusion of co-data dependent models unstable. Our ad-hoc solution is to include the optimal regularisation parameter for the co-data independent model (λ_{min}) and a slightly larger one (λ_{1se}). The latter is given by the one-standard-error rule, which increases λ until the cross-validated loss equals its minimum plus one standard error.

We concatenate $\hat{\boldsymbol{H}}^{(0,\text{cv})}$ with the columns of $\hat{\boldsymbol{H}}^{(1,\text{cv})}$ that correspond to λ_{min} and λ_{1se} to obtain the $n \times (m+2)$ matrix $\hat{\boldsymbol{H}}^{(\text{cv})}$. The first m columns correspond to the models with co-data, and the last two columns correspond to the model without co-data.

Meta-learner

We combine the base-learners with and without co-data by stacked generalisation (Wolpert, 1992), on the level of the linear predictors (Rauschenberger et al., 2021). In the meta-layer, we regress the target on the cross-validated linear predictors from the base-layer:

$$\mathbb{E}[y_i] = h^{-1} \left(\omega_0 + \sum_{k=1}^{m+2} \omega_k \hat{H}_{ik}^{(\text{cv})} \right) \ .$$

Leaving the intercept unrestricted ($-\infty < \omega_0 < +\infty$) but imposing the lower bound zero on the slopes ($\omega_1 \geq 0, \ldots, \omega_{m+2} \geq 0$), we estimate these coefficients under lasso regularisation. Due to the feature selection property of the lasso, a source of co-data can be excluded ($\hat{\omega}_k = 0$) or included ($\hat{\omega}_k > 0$), where k in $\{1, \ldots, m\}$. Similarly, the models without co-data can be excluded ($\hat{\omega}_k = 0$) or included ($\hat{\omega}_k > 0$), where $k = m+1$ for the model with λ_{min} and $k = m+2$ for the model with λ_{1se}. The estimated slopes function as weights for the co-data dependent models ($\hat{\omega}_1, \ldots, \hat{\omega}_m$) and for the co-data independent models ($\hat{\omega}_{m+1}, \hat{\omega}_{m+2}$). Thus, we do not only select sources but also weight them according to their relevance.

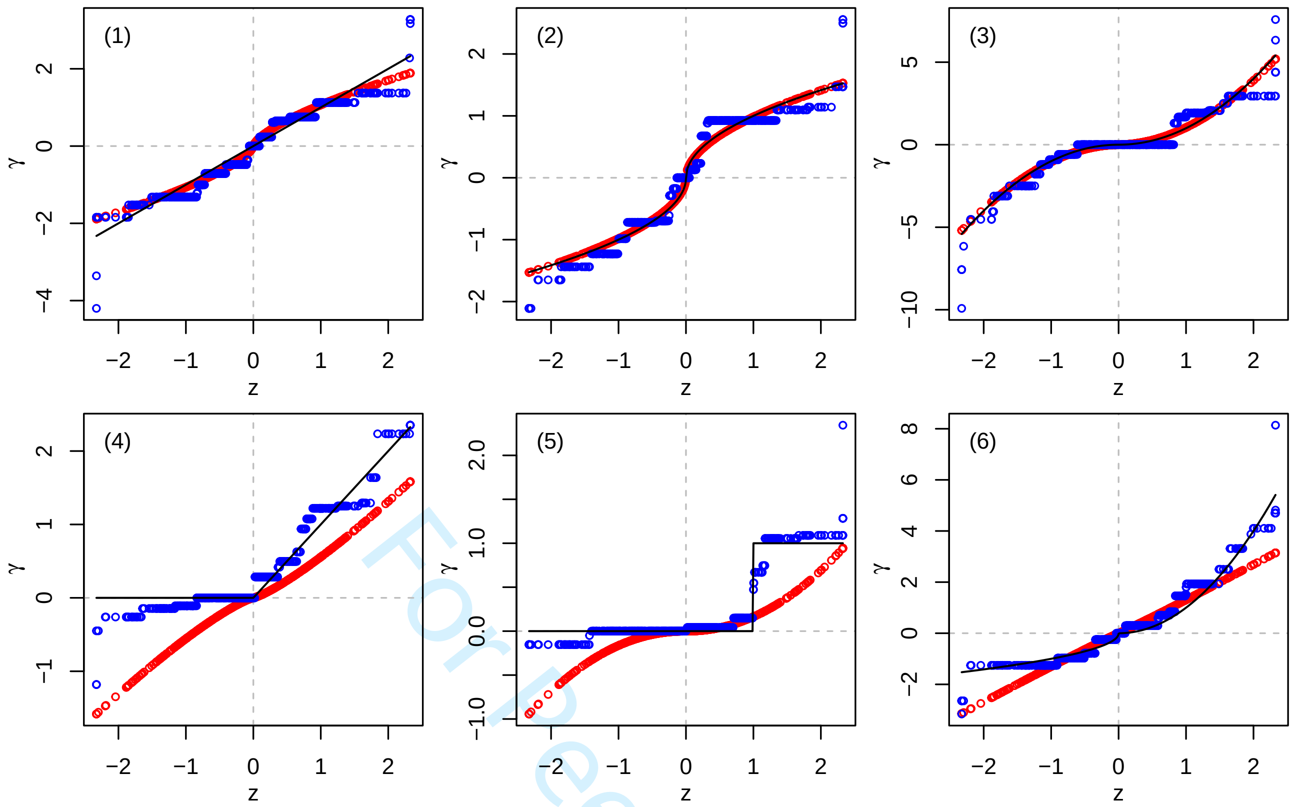

Fig. 1. Final prior effects $\boldsymbol{\gamma}$ (y-axis) against initial prior effects $\boldsymbol{z}$ (x-axis), under exponential calibration (red) and isotonic calibration (blue). The black line corresponds to perfect calibration ($\boldsymbol{\gamma} = \boldsymbol{\beta}$). We simulated the feature matrix $\boldsymbol{X}$ from a standard Gaussian distribution ($n = 200$, $p = 500$) and the initial prior effects $\boldsymbol{z}$ from a trimmed standard Gaussian distribution (trimmed below the 1% and above the 99% quantile). We set the true coefficients to (1) $\boldsymbol{\beta} = \boldsymbol{z}$, (2) $\boldsymbol{\beta} = \text{sign}(\boldsymbol{z})\sqrt{|\boldsymbol{z}|}$, (3) $\boldsymbol{\beta} = \text{sign}(\boldsymbol{z})\boldsymbol{z}^2$, (4) $\boldsymbol{\beta} = \mathbb{I}[\boldsymbol{z} > 0]\boldsymbol{z}$, (5) $\boldsymbol{\beta} = \mathbb{I}[\boldsymbol{z} > 1]$, or (6) $\boldsymbol{\beta} = -\mathbb{I}[\boldsymbol{z} \leq 0]\sqrt{|\boldsymbol{z}|} + \mathbb{I}[\boldsymbol{z} > 0]\boldsymbol{z}^2$. And we simulated the response vector $\boldsymbol{y}$ from Gaussian distributions with the means $\boldsymbol{\eta}$ and the variance Var($\boldsymbol{\eta}$), where $\boldsymbol{\eta} = \boldsymbol{X}\boldsymbol{\beta}$. While exponential calibration performs slightly better in the first three scenarios (top), isotonic calibration performs much better in the last three scenarios (bottom).

Interpretation

The coefficients $\hat{\boldsymbol{\beta}}_{\text{min}}$ and $\hat{\boldsymbol{\beta}}_{\text{1se}}$ give insight into the feature-target effects estimated without co-data, with $\hat{\beta}_{\text{min},j}$ and $\hat{\beta}_{\text{1se},j}$ representing the effect of feature j, where j in $\{1, \ldots, p\}$. The coefficients $\hat{\boldsymbol{\omega}}$ give insight into the importance of the sources of co-data, with $\hat{\omega}_k$ representing the importance of source k, where k in $\{1, \ldots, m\}$. For a previously unseen sample i, the predicted value is:

$$\begin{aligned} \hat{y}_i &= h^{-1}\left(\hat{\omega}_0 + \sum_{k=1}^{m+2} \hat{\omega}_k \hat{H}_{ik}\right) = h^{-1}\left(\hat{\beta}_0^\star + \sum_{j=1}^{p} \hat{\beta}_j^\star x_{ij}\right) \ , \\ \text{where} \quad & \hat{\beta}_0^\star = \hat{\omega}_0 + \hat{\omega}_{m+1}\hat{\beta}_{\text{min},0} + \hat{\omega}_{m+2}\hat{\beta}_{\text{1se},0} \qquad (1) \\ \text{and} \quad & \hat{\beta}_j^\star = \left(\sum_{k=1}^{m} \hat{\omega}_k \hat{\gamma}_{jk}\right) + \hat{\omega}_{m+1}\hat{\beta}_{\text{min},j} + \hat{\omega}_{m+2}\hat{\beta}_{\text{1se},j} \ . \end{aligned}$$

Thus, the estimated effect for a feature ($\hat{\beta}_j^\star$) is a weighted sum of estimated coefficients with co-data ($\hat{\gamma}_{j1}, \ldots, \hat{\gamma}_{jm}$) and the estimated coefficients without co-data ($\hat{\beta}_{\text{min},j}, \hat{\beta}_{\text{1se},j}$).

Sparse models (few non-zero coefficients) are often considered to be more interpretable than dense models (many non-zero coefficients). While the original coefficients are dense ($\sum_{j=1}^{p} \mathbb{I}[\hat{\beta}_j \neq 0] = p$) or sparse ($\sum_{j=1}^{p} \mathbb{I}[\hat{\beta}_j \neq 0] \ll p$) depending on the choice between ridge and lasso regularisation, the weights may contain some zeros due to significance filtering or lasso regularisation ($\sum_{k=1}^{m+2} \mathbb{I}[\hat{\omega}_k \neq 0] \leq m + 2$). As soon as one set of dense prior effects is selected, however, the combined coefficients also become dense ($\sum_{j=1}^{p} \mathbb{I}[\hat{\beta}_j^\star \neq 0] \lesssim p$). This means that the feature selection property of the lasso is not maintained. We should therefore choose between ridge and lasso regularisation (i) to make the model without co-data more predictive or interpretable (ii) or to make the model with co-data more predictive (iii) but not to make the model with co-data more interpretable.

Extension

In some applications, prior information might be available and reliable for some features but missing or unreliable for other features. Note that a missing prior effect can be interpreted as an unreliable prior effect of zero. Although the base-learners with co-data might still be predictive, the meta-learner (weighted average of the base-learners with and without co-data) might be not more predictive than the base-learner without co-data. The reason is that the meta-learner assigns the same weight to all prior effects, rather than more weight to available and reliable prior effects and less weight to missing or unreliable prior effects. We therefore propose an alternative approach for applications with partially informative sources of co-data.

In the following, we use the term ‘meta-features’ for the cross-validated linear predictors from the base learners with co-data. Each meta-feature - one column of the $n \times m$ matrix $\hat{\boldsymbol{H}}^{(0,\text{cv})}$ - corresponds to one source of co-data. In the meta-layer, we regress the target on the meta-features and the base-features:

$$\mathbb{E}[y_i] = h^{-1}\left(\beta_0 + \sum_{k=1}^{m} \omega_k \hat{H}_{ik}^{(0,\text{cv})} + \sum_{j=1}^{p} \beta_j x_{ij}\right) \ ,$$

with non-negativity constraints for the weights for the meta-features ($\omega_1 \geq 0, \ldots, \omega_m \geq 0$) but without constraints for the intercept (β_0) and the slopes for the base-features ($\beta_1, \ldots, \beta_p$).

We estimate the weights for the meta-features and the slopes for the base-features using penalised maximum likelihood:

$$\{\hat{\boldsymbol{\omega}}, \hat{\boldsymbol{\beta}}\} = \underset{\{\boldsymbol{\omega},\boldsymbol{\beta}\}}{\text{argmax}} \left\{ L(\boldsymbol{x}; \boldsymbol{\omega}, \boldsymbol{\beta}) - \rho(\lambda; \boldsymbol{\beta}) \right\} \ ,$$

where $L(\boldsymbol{x}; \boldsymbol{\omega}, \boldsymbol{\beta})$ is the likelihood and $\rho(\lambda; \boldsymbol{\beta})$ is the penalty. We do not penalise the weights for the meta-features ($m \ll n$) but only the slopes for the base-features ($p \gg n$). The more sources of co-data are available, the more it becomes necessary to penalise their weights. But then the weights $\boldsymbol{\omega}$ and the slopes $\boldsymbol{\beta}$ might need differential penalisation, for example a lasso penalty for the meta-features (selection of sources) and a ridge penalty for the base-features (many small effects). To make this computationally efficient, we would need a fast cross-validation procedure for multiple penalties (cf. van de Wiel et al., 2021) with non-negativity constraints (meta-features) and mixed lasso and ridge penalisation (meta-features vs base-features). This extension is therefore only applicable in settings with few sources of co-data.

The predicted value for a previously unseen sample i is

$$\begin{aligned} \hat{y}_i &= h^{-1}\left(\hat{\beta}_0 + \sum_{k=1}^{m} \hat{\omega}_k \hat{H}_{ik}^{(0)} + \sum_{j=1}^{p} \hat{\beta}_j x_{ij}\right) \\ &= h^{-1}\left(\hat{\beta}_0^\star + \sum_{j=1}^{p} \hat{\beta}_j^\star x_{ij}\right) \ , \\ \text{where} \quad & \hat{\beta}_0^\star = \hat{\beta}_0 \\ \text{and} \quad & \hat{\beta}_j^\star = \left(\sum_{k=1}^{m} \hat{\omega}_k \hat{\gamma}_{jk}\right) + \hat{\beta}_j \ . \end{aligned} \tag{2}$$

As the coefficients $\boldsymbol{\beta}$ are shrunk towards zero but the coefficients $\boldsymbol{\omega}$ are not penalised, the combined coefficients $\boldsymbol{\beta}^\star$ are shrunk towards a weighted sum of the calibrated prior effects ($\hat{\boldsymbol{\gamma}}\hat{\boldsymbol{\omega}}$). When the regularisation parameter tends to infinity ($\lambda \to \infty$), the estimated deviations from this weighted sum approach zero ($\hat{\boldsymbol{\beta}} \to 0$) and the combined estimates approach this weighted sum ($\hat{\boldsymbol{\beta}}^\star \to \hat{\boldsymbol{\gamma}}\hat{\boldsymbol{\omega}}$). Lasso regularisation ensures sparsity in the deviations from the calibrated prior effects ($\sum_{j=1}^{p} \mathbb{I}[\hat{\beta}_j \neq 0] \ll p$) - in contrast to ridge regularisation - but not in the combined coefficients ($\sum_{j=1}^{p} \mathbb{I}[\beta_j^\star \neq 0] \lesssim p$). As the combined coefficients may deviate more from unreliable than from reliable calibrated prior effects, this extension is suitable for partially informative co-data. As opposed to 'standard stacking', we refer to this extension as 'simultaneous stacking'. Algorithm 1 includes the pseudo-code for both approaches.

Simulation

We performed two simulation studies to compare the predictive performance between our transfer learning method and the one from Tian and Feng (2022). In contrast to the method from Tian and Feng (2022), which requires the feature-target effects in the target and the source data set(s) to be *positively correlated* and on the *same scale* (i.e. $\boldsymbol{\beta}_{\text{target}} \approx \boldsymbol{\beta}_{\text{source}}$), our method also allows for negatively correlated effects and for effects on different scales (i.e. $\boldsymbol{\beta}_{\text{target}} \approx c \times \boldsymbol{\beta}_{\text{source}}$). Although it is possible to overcome this restriction by inverting the target (Gaussian: $\boldsymbol{y}_{\text{source}} \to -\boldsymbol{y}_{\text{source}}$, binomial: $\boldsymbol{y}_{\text{source}} \to 1-\boldsymbol{y}_{\text{source}}$), by re-scaling the target (Gaussian: $\boldsymbol{y}_{\text{source}} \to 1/c \times \boldsymbol{y}_{\text{source}}$), or by inverting or re-scaling the features ($\boldsymbol{X}_{\text{source}} \to c \times \boldsymbol{X}_{\text{source}}$), we believe it is more user-friendly to directly allow for negative correlations and different scales. To ensure a fair comparison between the two methods, we simulate positively correlated effects on the same scale. Furthermore, although the method from Tian and Feng (2022) is in theory also suitable for mixed response types, the current version of the related `R` package `glmtrans` requires the source data set(s) and the target data set to have the same response type (Gaussian, binomial, Poisson). We therefore always simulate the same response type in the source and target domains. In addition, we also compared our method with the one from Kawaguchi et al. (2021, `xrnet`).

External simulation

We use the simulation approach from Tian and Feng (2022). In each iteration, we call the function `glmtrans::models` with the arguments (1) family of distributions: `family="gaussian"` (default) or `family="binomial"`, (2) source or target data sets: `type="all"` (default), (3) difference between source and target coefficients: `h=5` (default) or `h=250`, (4) number of source data sets: `K=5` (default), (5) sample size for target data set: `n.target=100` (default), (6) sample size for each source data set: `n.source=150` (default), (7) number of non-zero coefficients: `s=15` (default) or `s=50`, (8) number of features: `p=1 000` (default), number of transferable source data sets: `Ka=1`, `Ka=3` or `Ka=K=5` (default).

The simulation from Tian and Feng (2022) involves the following steps:

- Features: The correlation between features i and j is set to $\Sigma_{ij} = 0.5^{|i-j|}$, where i and j in $\{1, \ldots, p\}$. Let $\boldsymbol{\Sigma}$ represent the correlation matrix and let $\boldsymbol{\Sigma} = \boldsymbol{R}^\intercal \boldsymbol{R}$ represent its Cholesky decomposition, where $\boldsymbol{R}$ is an upper triangular matrix. For the target data set ($n_0 = 100$) and each source data set ($n_1 = \ldots = n_5 = 150$), the $n_0 \times p$ matrix $\boldsymbol{X}_0 = \boldsymbol{E}_0 \boldsymbol{R}$ and the $n_k \times p$ matrices $\boldsymbol{X}_k = \boldsymbol{E}_k \boldsymbol{R}$ for k in $\{1, \ldots, 5\}$ represent the features, where the $n_0 \times p$ matrix $\boldsymbol{E}_0$ and the $n_k \times p$ matrices $\boldsymbol{E}_k$ contain Gaussian noise.
- Coefficients: Let β_j represent the effect of feature j, for j in $\{1, \ldots, p\}$, and denote the p-dimensional coefficient vectors by $\boldsymbol{\beta}_0$ for the target data set and $\{\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_5\}$ for the source data sets. For the *target* data set, the first s elements are set to $\beta_j = 0.5$ (causal) and the last $p - s$ elements are set to $\beta_j = 0$ (non-causal). For *transferable source* data sets, the first s elements are set to $\beta_j = 0.5 + (-1)^{z_j} h/p$ and the last $p - s$ elements are set to $\beta_j = (-1)^{z_j} h/p$, where z_j is a realisation of $z_j \sim \text{Bernoulli}(0.5)$. For *non-transferable source* data sets, the first s elements are set to $\beta_j = (-1)^{z_j} 2h/p$ (non-causal), the next s elements are set to $\beta_j = 0.5 + (-1)^{z_j} 2h/p$ (causal), and the last $p - 2s$ elements are a random sample of s causal and $p - 3s$ non-causal elements generated in the same way. To obtain a non-transferable source, it would be sufficient to randomly select causal elements rather than inverting causal and non-causal elements (indices 1 to $2s$).
- Targets: In the *Gaussian* case, the target vector is the n-dimensional vector $\boldsymbol{y}_0 = \boldsymbol{X}_0 \boldsymbol{\beta}_0 + \boldsymbol{\epsilon}_0$ for the target data set, and the n-dimensional vector $\boldsymbol{y}_k = 0.5 + \boldsymbol{X}_k \boldsymbol{\beta}_k + \boldsymbol{\epsilon}_k$ for the source data sets, where the n-dimensional vectors $\{\boldsymbol{\epsilon}_0, \ldots, \boldsymbol{\epsilon}_5\}$ contain Gaussian noise. In the *binomial* case, let $\boldsymbol{p}_0 = 1/(1 + \exp(-\boldsymbol{X}_0 \boldsymbol{\beta}_0))$ for the target data set and $\boldsymbol{p}_k = 1/(1 + \exp(-0.5 - \boldsymbol{X}_k \boldsymbol{\beta}_k))$ for the source data sets.

Algorithm 1 Pseudo-code for the proposed transfer learning method with standard stacking (left) and simultaneous stacking (right). Sub-procedures (italicised) are explained below.

Require: $\boldsymbol{y}$ (target, $n \times 1$ vector), $\boldsymbol{X}$ (features, $n \times p$ matrix), $\boldsymbol{Z}$ (prior effects, $p \times m$ matrix)
Ensure: $\hat{\boldsymbol{\beta}}^\star$ (estimated intercept and slopes, $(p+1) \times 1$ vector)

```
initialise $n \times 1$ vector fold with values from 1 to 10

# base-learners

for $k$ from 1 to $m$ do
    $\hat{\gamma}_k \leftarrow$ Exp/Iso-Calibrate($\boldsymbol{y}, \boldsymbol{X}, \boldsymbol{Z}[, k]$)
end for
$\{\lambda^{\text{base}}_{\text{min}}, \lambda^{\text{base}}_{\text{1se}}\} \leftarrow$ Ridge/Lasso-CV($\boldsymbol{y}, \boldsymbol{X}$, fold)
$\{\hat{\boldsymbol{\beta}}_{\text{min}}, \hat{\boldsymbol{\beta}}_{\text{1se}}\} \leftarrow$ Ridge/Lasso-Fit($\boldsymbol{y}, \boldsymbol{X}, \{\lambda^{\text{base}}_{\text{min}}, \lambda^{\text{base}}_{\text{1se}}\}$)

# cross-validation

initialise empty $n \times (m+2)$ matrix $\hat{\boldsymbol{H}}^{(\text{cv})}$
for all $i$ from 1 to 10 do
    $\boldsymbol{y}_{\text{train}} \leftarrow \boldsymbol{y}[\texttt{fold} \neq i]$
    $\boldsymbol{X}_{\text{train}} \leftarrow \boldsymbol{X}[\texttt{fold} \neq i, ]$
    $\boldsymbol{X}_{\text{test}} \leftarrow \boldsymbol{X}[\texttt{fold} = i, ]$
    for $k$ from 1 to $m$ do
        $\hat{\gamma}_k^{-\kappa(i)} \leftarrow$ Exp/Iso-Calibrate($\boldsymbol{y}_{\text{train}}, \boldsymbol{X}_{\text{train}}, \boldsymbol{Z}[, k]$)
        $\hat{\boldsymbol{H}}^{(\text{cv})}[\texttt{fold} = i, k] \leftarrow \boldsymbol{X}_{\text{test}}\hat{\gamma}_k^{-\kappa(i)}$
    end for
    $\hat{\boldsymbol{\beta}}_{\text{min}}^{-\kappa(i)} \leftarrow$ Ridge/Lasso-Fit($\boldsymbol{y}_{\text{train}}, \boldsymbol{X}_{\text{train}}, \lambda_{\text{min}}$)
    $\hat{\boldsymbol{H}}^{(\text{cv})}[\texttt{fold} = i, k+1] \leftarrow \boldsymbol{X}_{\text{test}}\hat{\boldsymbol{\beta}}_{\text{min}}^{-\kappa(i)}$
    $\hat{\boldsymbol{\beta}}_{\text{1se}}^{-\kappa(i)} \leftarrow$ Ridge/Lasso-Fit($\boldsymbol{y}_{\text{train}}, \boldsymbol{X}_{\text{train}}, \lambda_{\text{1se}}$)
    $\hat{\boldsymbol{H}}^{(\text{cv})}[\texttt{fold} = i, k+2] \leftarrow \boldsymbol{X}_{\text{test}}\hat{\boldsymbol{\beta}}_{\text{1se}}^{-\kappa(i)}$
end for

# meta-learner

$\lambda^{\text{meta}}_{\text{min}} \leftarrow$ Lasso-CV($\boldsymbol{y}, \hat{\boldsymbol{H}}^{(\text{cv})}$, fold)$^*$
$\hat{\boldsymbol{\omega}} \leftarrow$ Lasso-Fit($\boldsymbol{y}, \hat{\boldsymbol{H}}^{(\text{cv})}, \lambda^{\text{meta}}_{\text{min}}$)$^*$
$^*$with $\hat{\boldsymbol{\omega}}$ non-negative

# combination

$\hat{\boldsymbol{\beta}}^\star \leftarrow$ Sta-Combine($\hat{\boldsymbol{\omega}}; \hat{\gamma}_1, \ldots, \hat{\gamma}_m; \hat{\boldsymbol{\beta}}_{\text{min}}, \hat{\boldsymbol{\beta}}_{\text{1se}}$)
```

```
initialise $n \times 1$ vector fold with values from 1 to 10

# base-learners

for $k$ from 1 to $m$ do
    $\hat{\gamma}_k \leftarrow$ Exp/Iso-Calibrate($\boldsymbol{y}, \boldsymbol{X}, \boldsymbol{Z}[, k]$)
end for

# cross-validation

initialise empty $n \times m$ matrix $\hat{\boldsymbol{H}}^{(\text{cv})}$
for all $i$ from 1 to 10 do
    $\boldsymbol{y}_{\text{train}} \leftarrow \boldsymbol{y}[\texttt{fold} \neq i]$
    $\boldsymbol{X}_{\text{train}} \leftarrow \boldsymbol{X}[\texttt{fold} \neq i, ]$
    $\boldsymbol{X}_{\text{test}} \leftarrow \boldsymbol{X}[\texttt{fold} = i, ]$
    for $k$ from 1 to $m$ do
        $\hat{\gamma}_k^{-\kappa(i)} \leftarrow$ Exp/Iso-Calibrate($\boldsymbol{y}_{\text{train}}, \boldsymbol{X}_{\text{train}}, \boldsymbol{Z}[, k]$)
        $\hat{\boldsymbol{H}}^{(\text{cv})}[\texttt{fold} = i, k] \leftarrow \boldsymbol{X}_{\text{test}}\hat{\gamma}_k^{-\kappa(i)}$
    end for
end for

# meta-learner

$\lambda^{\text{meta}}_{\text{min}} \leftarrow$ Ridge/Lasso-CV($\boldsymbol{y}, [\hat{\boldsymbol{H}}^{(\text{cv})}|\boldsymbol{X}]$, fold)$^*$
$[\hat{\boldsymbol{\omega}}|\hat{\boldsymbol{\beta}}] \leftarrow$ Ridge/Lasso-Fit($\boldsymbol{y}, [\hat{\boldsymbol{H}}^{(\text{cv})}|\boldsymbol{X}], \lambda^{\text{meta}}_{\text{min}}$)$^*$
$^*$with $\hat{\boldsymbol{\omega}}$ unpenalised and non-negative

# combination

$\hat{\boldsymbol{\beta}}^\star \leftarrow$ Sim-Combine($\hat{\boldsymbol{\omega}}; \hat{\gamma}_1, \ldots, \hat{\gamma}_m; \hat{\boldsymbol{\beta}}$)
```

- *Exp/Iso-Calibrate*†: Performs exponential or isotonic calibration (Section 2.3). Requires target vector, feature matrix, and prior effects. Ensures calibrated prior effects.
- *Ridge/Lasso-CV*†: Tunes the hyperparameter of ridge or lasso regression by k-fold cross-validation. Requires target vector, feature matrix, and fold identifiers. Ensures optimal regularisation parameter.
- *Ridge/Lasso-Fit*†: Estimates the parameters of ridge or lasso regression. Requires target vector, feature matrix, and regularisation parameter. Ensures estimated coefficients.
- *Sta/Sim-Combine*: Combines estimated parameters from standard stacking (Equation 1) or simultaneous stacking (Equation 2). Requires estimated parameters from base-learners and meta-learner. Ensures combined estimates.

†Note: The procedures *Exp/Iso-Calibrate*, *Ridge/Lasso-CV* and *Ridge/Lasso-Fit* depend on the family of distributions (Gaussian vs binomial), but they also require choices (exponential vs isotonic, ridge vs lasso).

The n-dimensional vectors $\{\boldsymbol{y}_0, \ldots, \boldsymbol{y}_5\}$ are the target vectors, with each element following a Bernoulli distribution with the probability given by $\{\boldsymbol{p}_0, \ldots, \boldsymbol{p}_5\}$.

Internal simulation

The simulation from Tian and Feng (2022) uses the same effect size for all causal features and a decreasing correlation structure with a fixed base. We therefore designed our own data-generating mechanism (i) to simulate different effect sizes for different causal features ($\beta_j \in \mathbb{R}$ instead of $\beta_j \in \{0, 0.5\}$) and (ii) to vary the strength of correlation between features ($\sigma_{ij} = {\rho_x}^{|i-j|}$ instead of $\sigma_{ij} = 0.5^{|i-j|}$).

Our simulation involves the following steps:

- Features: Setting the mean of feature i to $\mu_i = 0$, the variance of feature i to $\sigma_{ii} = 1$, and the covariance between features i and j to $\sigma_{ij} = \rho_x^{|i-j|}$, for all i and j in $\{1, \ldots, p\}$, we simulate multiple feature matrices from the multivariate Gaussian distribution with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$, namely the $n_0 \times p$ feature matrix $\boldsymbol{X}_0$ for the target data set, and the $n_{1/2/3} \times p$ feature matrices $\{\boldsymbol{X}_1, \boldsymbol{X}_2, \boldsymbol{X}_3\}$ for the source data sets ($n_0 = 100, n_1 = n_2 = n_3 = 150, p = 500$).

- Coefficients: Setting the mean and the variance for data set k to $\mu_k = 0$ and $\sigma_{kk} = 1$, for all k in $\{0, 1, 2, 3\}$, and the covariance between data sets k and l to $\sigma_{kl} = 0$ if either k or l equals 1, or to $\sigma_{kl} = \rho_\beta$ if both k and l are in $\{0, 2, 3\}$, we simulate two $p \times 4$ matrices from the multivariate Gaussian distribution with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$, namely $\boldsymbol{B}_1$ and $\boldsymbol{B}_2$. We define the coefficients as $\boldsymbol{B} = \boldsymbol{B}_1\mathbb{I}[\boldsymbol{B}_2 > \phi^{-1}(1-\pi)]$, where ϕ is the Gaussian cumulative distribution function and π equals 0.2 (dense) or 0.05 (sparse), and denote the p-dimensional coefficient vectors by $\boldsymbol{\beta}_0$ for the target data set and by $\{\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \boldsymbol{\beta}_3\}$ for the source data sets. While one set of coefficients is non-transferable ($\boldsymbol{\beta}_1$), we transform the transferable sets of coefficients with $\boldsymbol{\beta}_2 \rightarrow \text{sign}(\boldsymbol{\beta}_2)|\boldsymbol{\beta}_2|^2$ and $\boldsymbol{\beta}_3 \rightarrow \text{sign}(\boldsymbol{\beta}_3)\sqrt{|\boldsymbol{\beta}_3|}$.
- Targets: For the target data set, we compute $\boldsymbol{z}_0 = \boldsymbol{X}_0\boldsymbol{\beta}_0$ and standardise $\boldsymbol{z}_0$ to obtain $\boldsymbol{z}_0^*$. For the source data sets, we proceed similarly to obtain $\{\boldsymbol{z}_1^*, \boldsymbol{z}_2^*, \boldsymbol{z}_3^*\}$. The simulated targets equal $\boldsymbol{y}_k = h^{-1}(\sqrt{w}\boldsymbol{z}_k^* + \sqrt{1-w}\boldsymbol{\epsilon}_k)$, where $h(\cdot)$ is a link function and $\boldsymbol{\epsilon}_k$ follows a standard Gaussian distribution, for k in $\{0, \ldots, 3\}$. Given $0 \leq w \leq 1$, we have $\text{Var}(\sqrt{w}\boldsymbol{z}_k^* + \sqrt{1-w}\boldsymbol{\epsilon}_k) = w\text{Var}(\boldsymbol{z}_k^*) + (1-w)\text{Var}(\boldsymbol{\epsilon}_k) = 1$. While $h(\cdot)$ is the identity link in the Gaussian case, it is the logit link in the binomial case, where the simulated probabilities are rounded to simulated classes. We set the signal-to-noise ratio to 4:1 ($w = 0.8$).

Simulation results

In addition to the target data set, the method from Tian and Feng (2022, `glmtrans`) requires the source data sets, while our method (`transreg`) and the method from Kawaguchi et al. (2021, `xrnet`) require the prior effects derived from the source data sets. Since these methods have different requirements, we first simulate the source data sets (for `glmtrans`) and then derive the prior effects from the simulated source data sets (for `xrnet` and `transreg`). As prior effects, we use the estimated coefficients from penalised regression on the source data sets. We choose the type of regularisation for all methods subject to the simulation setting, namely ridge regularisation for dense settings (`glmtrans`: $\alpha = 0$; `xrnet` and `transreg`: $\alpha_{\text{source}} = \alpha_{\text{target}} = 0$) and lasso or lasso-like elastic net regularisation for sparse settings (`glmtrans`: $\alpha = 1$; `xrnet` and `transreg`: $\alpha_{\text{source}} = 0.95$, $\alpha_{\text{target}} = 1$). The idea of the lasso-like elastic net regularisation is to render the prior information more stable.

In each simulation setting, we simulate 100 training samples and 10 000 testing samples (hold-out) for the target data set. Tables 2 and 3 show the predictive performance for the testing data in the external and internal simulation, under exponential and isotonic calibration. We observe that transfer learning (with `glmtrans`, `xrnet`, or `transreg`) often leads to a significant improvement with respect to standard penalised regression (with `glmnet`). Concerning the proposed method, this holds for the two different calibration approaches and the two different stacking approaches. These simulation studies do not show that the proposed transfer learning method outperforms other transfer learning methods. The advantage of the proposed method as compared to the method from Tian and Feng (2022, `glmtrans`) is that it does not require the source data but only the prior effects derived from the source data, and the advantage as compared to the method from Kawaguchi et al. (2021, `xrnet`) is that it allows for non-linear relationships between the prior effects and the true effects. While the method from Kawaguchi et al. (2021, `xrnet`) shrinks the coefficients towards a linear function of the prior effects, the proposed method adapts the prior effects to the true effects through exponential or isotonic calibration.

Applications

External applications

First, we consider an adapted version of the application on cervical cancer from van de Wiel et al. (2016). The aim is to transfer information from a methylation study with biopsy samples to another methylation study with self-collected samples in order to better discriminate between low-grade and high-grade precursor lesions. Specifically, we transfer the signs of the effect sizes and the p-values from the source data set to the target data set ($n = 44$ samples, $p = 9\,491$ features). We then examine whether this prior information increases the predictive performance of ridge regression, which is more predictive than lasso regression in this application. Next to our transfer learning method (`transreg`) and the one from Kawaguchi et al. (2021, `xrnet`), we consider the co-data learning methods from Tay et al. (2022, `fwelnet`) and van Nee et al. (2021, `ecpc`). While the former exploit information on the importance and direction of the effects (co-data: $-\text{sign}(\text{coef})\log_{10}(p\text{-value})$), the latter only exploit information on their importance (co-data: $-\log_{10}(p\text{-value})$). After 10 repetitions of 10-fold cross-validation, we observe that the proposed method (not with exponential but with isotonic calibration) often increases the predictive performance of ridge regression (`transreg.exp.sta`: 0/10, `transreg.exp.sim`: 4/10, `transreg.iso.sta`: 7/10, `transreg.iso.sim`: 10/10, `fwelnet`: 7/10, `ecpc`: 5/10, `xrnet`: 3/10). We also observe that exploiting information on the importance as well as the direction of the effects (`transreg`) can be more beneficial than exploiting information on the importance of the effects only (`fwelnet`, `ecpc`), as can be seen in the mean change in cross-validated logistic deviance (`transreg.exp.sta`: +6.56%, `transreg.exp.sim`: +0.43%, `transreg.iso.sta`: −2.50%, `transreg.iso.sim`: −9.25%, `fwelnet`: −0.12%, `ecpc`: −1.52%, `xrnet`: +2.59%). The relatively poor performance of the competing transfer learning method (`xrnet`) might be caused by a non-linear relationship between the prior effects (transformed p-values) and the true effects. Here, isotonic calibration outperforms exponential calibration, and simultaneous stacking outperforms standard stacking. A potential explanation for the large difference in performance between exponential and isotonic calibration is that positive effects might be more important than negative effects in this application, for a biological reason (methylation increases the probability of cancer) and a statistical reason (effects of overexpression are easier to detect than those of underexpression). While exponential calibration behaves symmetrically for negative and positive prior effects, isotonic calibration can shrink negative prior effects towards zero.

Second, we consider an adapted version of the application on pre-eclampsia from Tay et al. (2022). Measurements of $p = 1\,125$ plasma proteins are available for $n = 166$ patients at multiple time points ($48 \times 2 + 8 \times 3 + 20 \times 4 + 74 \times 5 + 16 \times 6 = 666$). The aim is to transfer information from late time points (gestational age $>$ 20 weeks) to early time points (gestational age $\leq$ 20 weeks). We repeatedly split the patients into one source data set and one target data set. Patients with only late time points are always in the source data set, and other patients are randomly allocated to the source and the target data set. (Note that this application is somewhat artificial, as it might be better to drop transfer learning in favour of using all earliest time points in the regression of interest.) Using the source data set, we estimate two logistic regression models under ridge regularisation, once using

Table 2. Predictive performance in external simulation. In each setting (row), we simulate 10 data sets, calculate the performance metric (mean-squared error for numerical prediction, logistic deviance for binary classification) for the test sets, express these metrics as percentages of those from prediction by the mean, and show the mean and standard deviation of these percentages. Settings: number of transferable source data sets (K_a), differences between source and target coefficients (h), dense setting with ridge regularisation ($s = 50$, $\alpha = 0$) or sparse setting with lasso regularisation ($s = 15$, $\alpha = 1$), family of distribution ('gaussian' or 'binomial'). These parameters determine (i) the mean Pearson correlation among the features in the target data set ($\bar{\rho}_x$) and (ii) the maximum Pearson correlation between the coefficients in the target data set and the coefficients in the source data sets ($\max(\hat{\rho}_\beta)$). Methods: regularised regression (`glmnet`), competing transfer learning methods (`glmtrans` , `xrnet`), proposed transfer learning method (`transreg`) with exponential/isotonic calibration and standard/simultaneous stacking. In each setting, the colour black (grey) highlights methods that are more (less) predictive than regularised regression without transfer learning (`glmnet`), asterisks (daggers) indicate methods that are *significantly* more (less) predictive at the 5% level (one-sided Wilcoxon signed-rank test), and an underline highlights the most predictive method.

									`transreg`			
K_a	h	α	family	$\bar{\rho}_x$	$\max(\hat{\rho}_\beta)$	`glmnet`	`glmtrans`	`xrnet`	`exp.sta`	`exp.sim`	`iso.sta`	`iso.sim`
1	5	0	gaussian	0.01	1.00	73.2±3.0	43.9±9.2*	47.1±1.8*	32.2±3.2*	30.9±2.8*	24.5±3.6*	23.4±3.3*
3	5	0	gaussian	0.01	1.00	73.2±3.0	33.5±6.5*	29.3±2.4*	18.1±2.5*	16.7±2.0*	13.5±2.0*	12.7±1.6*
5	5	0	gaussian	0.01	1.00	73.2±3.0	24.0±3.7*	22.6±1.5*	14.2±1.7*	13.2±1.7*	10.6±1.0*	10.0±0.9*
1	250	0	gaussian	0.01	0.40	73.2±3.0	31.8±9.4*	63.8±3.6*	54.8±6.0*	57.8±6.5*	49.5±6.9*	50.0±7.3*
3	250	0	gaussian	0.01	0.42	73.2±3.0	33.3±8.2*	51.4±4.7*	46.5±7.9*	47.0±9.2*	39.4±6.5*	37.6±7.5*
5	250	0	gaussian	0.01	0.43	73.2±3.0	33.3±5.0*	43.4±3.4*	40.0±6.3*	39.2±7.3*	32.7±4.4*	30.0±4.7*
1	5	1	gaussian	0.01	1.00	17.3±3.8	12.5±1.4*	14.6±1.7	12.2±1.9*	14.7±1.9*	11.0±1.6*	11.6±1.6*
3	5	1	gaussian	0.01	1.00	17.3±3.8	10.5±0.7*	11.5±0.6*	10.8±0.6*	11.7±1.6*	9.8±0.5*	9.9±0.6*
5	5	1	gaussian	0.01	1.00	17.3±3.8	10.0±0.4*	11.0±0.6*	10.5±0.5*	11.1±1.3*	9.6±0.3*	9.8±0.4*
1	250	1	gaussian	0.01	0.24	17.3±3.8	22.2±13.4†	17.6±4.1	14.7±3.4*	17.8±4.8	14.7±3.4*	18.2±5.0
3	250	1	gaussian	0.01	0.27	17.3±3.8	19.8±6.7†	17.7±3.6†	14.7±3.4*	18.4±4.8	14.8±3.4*	18.7±5.5†
5	250	1	gaussian	0.01	0.27	17.3±3.8	19.8±6.7†	17.5±3.6	14.8±3.3*	19.0±4.6	14.7±3.5*	21.3±6.6†
1	5	0	binomial	0.01	1.00	91.5±1.5	91.0±4.2	84.5±1.0*	82.4±5.6*	82.7±5.4*	77.8±3.7*	78.5±4.7*
3	5	0	binomial	0.01	1.00	91.5±1.5	85.9±3.7*	75.9±2.9*	69.8±3.6*	70.0±3.8*	66.7±3.7*	66.6±4.5*
5	5	0	binomial	0.01	1.00	91.5±1.5	79.9±2.4*	70.8±2.9*	63.1±3.8*	63.2±3.9*	61.2±3.0*	62.2±5.7*
1	250	0	binomial	0.01	0.41	91.5±1.5	92.0±5.0	91.1±2.0	89.4±2.9*	90.2±3.0	88.3±3.2*	88.7±3.0*
3	250	0	binomial	0.01	0.42	91.5±1.5	89.7±4.3	86.5±4.2*	86.3±4.6*	87.8±4.5*	84.0±7.5*	84.2±5.6*
5	250	0	binomial	0.01	0.43	91.5±1.5	85.3±3.5*	83.0±3.2*	84.3±4.8*	85.9±5.5*	79.4±4.4*	79.0±5.0*
1	5	1	binomial	0.01	1.00	80.4±8.2	77.8±12.2	77.1±8.4	71.4±5.8*	73.0±4.0*	70.3±5.2*	72.4±5.0*
3	5	1	binomial	0.01	1.00	80.4±8.2	70.5±12.5*	69.5±11.8*	67.9±5.7*	70.8±10.3*	64.7±4.8*	64.4±5.1*
5	5	1	binomial	0.01	1.00	80.4±8.2	59.5±2.5*	64.4±4.2*	65.7±6.4*	65.5±9.6*	62.7±4.5*	61.8±4.2*
1	250	1	binomial	0.01	0.25	80.4±8.2	82.1±8.7†	82.4±9.9†	81.5±9.2	81.5±9.1	81.2±7.2	82.4±10.1†
3	250	1	binomial	0.01	0.26	80.4±8.2	80.5±9.3	82.0±8.8†	80.7±8.8	81.3±9.5	79.2±8.3	82.7±11.2
5	250	1	binomial	0.01	0.27	80.4±8.2	78.9±9.4*	84.6±15.5	80.3±8.3	83.4±8.8†	79.4±10.7	86.3±16.8†

the early time points and once using all time points. For each patient, all time points are assigned to the same cross-validation fold, and the weight is split evenly among the time points. We then use the two sets of estimated regression coefficients as co-data for the target data set. In the regression for the target data set, we only include the earliest time point of each patient. Using 10-fold cross-validation, we estimate the predictive performance of ridge regression with and without transfer learning. After repeating source-target splitting and cross-validation 10 times, we observe that transfer learning tends to decrease the cross-validated logistic deviance (`transreg.exp.sta`: 8/10, `transreg.exp.sim`: 7/10, `transreg.iso.sta`: 7/10, `transreg.iso.sim`: 9/10, `fwelnet`: 5/10, `ecpc`: 6/10, `xrnet`: 10/10). It is more beneficial to share information not only on the importance but also the direction of the effects, according to the mean change in cross-validated logistic deviance (`transreg.exp.sta`: −2.61%, `transreg.exp.sim`: −4.29%, `transreg.iso.sta`: −3.67%, `transreg.iso.sim`: −8.33%, `fwelnet`: +0.04%, `ecpc`: −6.87%, `xrnet`: −12.12%). Simultaneous stacking again outperforms standard stacking, but exponential and isotonic calibration show a similar performance. In this application, where the prior effects are probablyiot close to the true effects, the competing transfer learning method (`xrnet`) outperforms the proposed one.

For both applications, Figure 2 shows the change in predictive performance from modelling without to modelling with prior information.

Internal application

In this application, we transfer information from a meta-analysis of genome-wide association studies on Parkinson's disease (PD-GWAS, Nalls et al., 2019) to the Luxembourg Parkinson's study (LUXPARK, Hipp et al., 2018). The aim is to classify samples into Parkinson's disease (PD) patients and healthy controls based on single-nucleotide polymorphisms (SNPs).

At the time of our study, the LUXPARK data set included genotyping and clinical data of 790 PD cases and 766 healthy controls. DNA samples were genotyped using the NeuroChip array (Blauwendraat et al., 2017). Quality control steps of genotyping data were conducted according to the standard procedures reported previously (Pavelka et al., 2022). Missing genotyping data were imputed using the reference panel from the Haplotype Reference Consortium (release 1.1) on the Michigan Imputation Server (Das et al., 2016) [RRID:ID_017579], with a filter for imputation quality ($r^2 > 0.3$).

As common SNPs exhibit weak effects on PD, the sample size is likely insufficient to train a highly predictive model. However, publicly available summary statistics from the largest-to-date PD-GWAS (with around 38 000 cases and 1 400 000 controls from European ancestry) (Nalls et al., 2019) might serve as prior information on the SNP effects. For each SNP, these summary statistics are the combined results from simple logistic regression of the PD status on the SNP, namely the estimated slope (logarithmic odds ratio), its standard error, and the associated p-value.

Table 3. Predictive performance in internal simulation. In each setting (row), we simulate 10 data sets, calculate the performance metric (mean-squared error for numerical prediction, logistic deviance for binary classification) for the test sets, express these metrics as percentages of those from prediction by the mean, and show the mean and standard deviation of these percentages. Settings: correlation parameter for features (ρ_x), correlation parameter for coefficients (ρ_β), dense setting with ridge regularisation ($\pi = 30\%$, $\alpha = 0$) or sparse setting with lasso regularisation ($\pi = 5\%$, $\alpha = 1$), family of distribution ('gaussian' or 'binomial'). These parameters determine (i) the mean Pearson correlation among the features in the target data set ($\bar{\rho}_x$) and (ii) the maximum Pearson correlation between the coefficients in the target data set and the coefficients in the source data sets ($\max(\hat{\rho}_\beta)$). Methods: regularised regression (`glmnet`), competing transfer learning methods (`glmtrans` , `xrnet`), proposed transfer learning method (`transreg`) with exponential/isotonic calibration and standard/simultaneous stacking. In each setting, the colour black (grey) highlights methods that are more (less) predictive than regularised regression without transfer learning (`glmnet`), asterisks (daggers) indicate methods that are *significantly* more (less) predictive at the 5% level (one-sided Wilcoxon signed-rank test), and an underline highlights the most predictive method.

									`transreg`			
ρ_x	ρ_β	α	family	$\bar{\rho}_x$	$\max(\hat{\rho}_\beta)$	`glmnet`	`glmtrans`	`xrnet`	`exp.sta`	`exp.sim`	`iso.sta`	`iso.sim`
0.95	0.60	0	gaussian	0.08	0.32	46.1±7.3	48.5±6.3	45.2±7.0*	45.2±7.6*	45.4±7.1	45.1±7.5*	44.5±6.8*
0.99	0.60	0	gaussian	0.32	0.32	33.2±4.8	33.2±6.4	32.6±4.4*	32.2±4.6*	32.5±4.4*	32.1±4.7*	32.0±4.1*
0.95	0.80	0	gaussian	0.08	0.54	45.1±6.7	47.9±11.1	42.0±6.4*	43.7±7.1*	43.3±6.9*	43.8±7.1*	41.9±6.5*
0.99	0.80	0	gaussian	0.32	0.54	32.5±4.8	31.8±4.2	31.1±4.7*	32.0±4.8*	31.6±4.7*	31.6±4.7*	30.4±3.9*
0.95	0.99	0	gaussian	0.08	0.89	44.1±5.1	42.2±5.6	37.3±4.4*	40.9±5.3*	37.6±4.5*	40.1±5.7*	38.8±5.7*
0.99	0.99	0	gaussian	0.32	0.89	31.0±5.1	29.4±2.1	27.6±3.0*	30.4±5.3	28.4±3.8*	29.8±4.8*	29.1±3.9*
0.95	0.60	1	gaussian	0.08	0.27	37.7±6.8	39.1±7.2	37.9±6.6	37.3±7.2	38.0±7.3	37.4±7.0	41.5±6.4†
0.99	0.60	1	gaussian	0.32	0.27	28.5±2.4	29.9±3.5†	29.1±3.4	28.8±2.5	29.0±2.7	28.7±2.7	28.7±2.5
0.95	0.80	1	gaussian	0.08	0.45	37.5±4.8	37.4±5.0	36.9±5.2	35.7±4.2*	36.8±4.7	35.9±4.7*	39.5±5.4†
0.99	0.80	1	gaussian	0.32	0.45	29.9±2.4	29.9±2.8	29.5±3.4	29.1±2.5	29.4±3.3	29.1±2.8	29.6±3.8
0.95	0.99	1	gaussian	0.08	0.87	38.0±6.4	33.0±5.7*	34.5±7.9*	34.8±8.2*	34.1±6.8*	34.0±7.6*	35.2±8.0
0.99	0.99	1	gaussian	0.32	0.87	30.2±4.4	29.5±4.5	29.2±4.6	29.4±4.8	29.1±4.2*	28.6±3.5*	29.4±4.4
0.95	0.60	0	binomial	0.08	0.32	77.1±4.5	81.2±6.1†	76.7±4.4	76.8±4.8	76.1±5.0	77.7±4.8	77.7±5.2
0.99	0.60	0	binomial	0.32	0.32	65.5±4.7	67.0±5.9	65.0±4.7	63.7±4.8*	65.3±4.4	63.8±4.8*	64.0±4.9
0.95	0.80	0	binomial	0.08	0.54	74.9±4.4	81.2±10.7†	73.5±5.1*	75.0±5.7	73.9±5.6	74.7±4.7	73.5±5.4*
0.99	0.80	0	binomial	0.32	0.54	64.3±3.6	64.7±4.6	61.8±4.7*	63.2±4.5*	63.7±5.0	63.0±4.7*	63.6±5.1
0.95	0.99	0	binomial	0.08	0.89	75.4±5.0	74.8±4.7*	69.5±4.2*	72.4±6.5*	71.5±6.3*	71.7±4.9*	71.4±4.8*
0.99	0.99	0	binomial	0.32	0.89	62.4±5.0	61.8±5.0	58.1±4.6*	61.0±6.7	58.9±6.7*	60.4±6.2*	59.6±5.0*
0.95	0.60	1	binomial	0.08	0.27	76.5±6.0	75.8±3.9	75.7±4.1	75.8±5.8	76.5±4.9	75.9±5.4	75.9±2.3
0.99	0.60	1	binomial	0.32	0.27	61.2±4.4	61.5±4.9	63.4±6.8†	63.1±5.6†	62.2±5.7	62.7±5.4†	61.5±4.6
0.95	0.80	1	binomial	0.08	0.45	78.2±10.5	76.6±10.2	76.3±9.8	75.0±6.0	77.1±9.0	75.2±6.1	77.9±6.6
0.99	0.80	1	binomial	0.32	0.45	64.9±4.9	64.8±5.6	65.5±2.9	66.3±6.7	64.4±5.2	65.2±6.0	65.5±4.3
0.95	0.99	1	binomial	0.08	0.87	80.1±6.0	73.3±5.7*	70.7±6.3*	69.2±4.5*	68.8±6.8*	70.1±5.2*	69.3±5.3*
0.99	0.99	1	binomial	0.32	0.87	63.2±5.1	62.9±4.7	61.2±5.6*	62.7±6.1	61.1±5.7*	61.8±5.8	61.2±6.2*

Importantly, the LuxPARK cohort was not part of the PD-GWAS, meaning that the prior information comes from independent data. As the LuxPARK cohort and the PD-GWAS cohorts have a similar ethnic background, the prior information might allow us to increase the predictive performance.

The two lists of SNPs - from the LuxPARK genotyping data (target data set) and the PD-GWAS summary statistics (source data set) - are partially overlapping. SNP data are high-dimensional and strongly correlated. From each block of SNPs in the target data set (250 kb window), we retain the most significant one and those that are in weak pairwise linkage disequilibrium with it ($r^2 < 0.1$). Next, we only retain the SNPs appearing also in the source data set. These two filtering steps together reduce the dimensionality in the target data set from around 18 million SNPs to 196 018 SNPs. We code the SNP data for dominant effects, with 0 meaning no alternate allele (0/0) and 1 meaning one or two alternate alleles (0/1 or 1/1).

It seems that the results from the source data set are informative, because 5.80% of the p-values are nominally significant at the 0.05 level (11 377 out of 196 018), 77 are significant at a false discovery rate of 5% (Benjamini-Hochberg), and 35 are significant at a family-wise error rate of 5% (Holm-Bonferroni). As SNPs with a low minor allele frequency might have large effect sizes but insignificant p-values, we base the prior effects not on the estimated coefficients ($\hat{\beta}$) but on the signed logarithmic p-values ($-\text{sign}(\hat{\beta})\log_{10}(p)$). For each SNP, we compared the reference and the alternate alleles between the two data sets: (i) If both data sets have the same reference allele and the same alternate allele, the signed logarithmic p-value from the source data set becomes the prior effect for the target data set. (ii) If the reference allele of each data set is the alternate allele of the other data set (swapped alleles), we invert the sign of the signed logarithmic p-value. (iii) And if the two data sets have two different sets of alleles (multiallelic SNP), we set the prior effect to zero.

Rather than using the 196 018 SNPs for predictive modelling in the target data set, we also filter them based on their significance in the source data set (which is already a type of transfer learning). For each cut-off in $\{5 \times 10^{-2}, 5 \times 10^{-3}, \ldots, 5 \times 10^{-10}\}$, we exclude all SNPs above and include all SNPs below. This means that for the target data set, we retain a specific number of the most significant SNPs from the source data set. For each significance cut-off, we compare three modelling approaches:

- *Uninformed approach*: We use logistic regression with ridge or lasso penalisation to model the PD status based on the included SNPs. All included SNPs are treated equally, irrespective of their estimated effect in the source data set.
- *Naïve transfer learning*: After calculating for each sample the sum across the signed logarithmic p-values from the source data set multiplied by the SNPs from the target data set, we fit a simple logistic regression of the PD status on this sum.

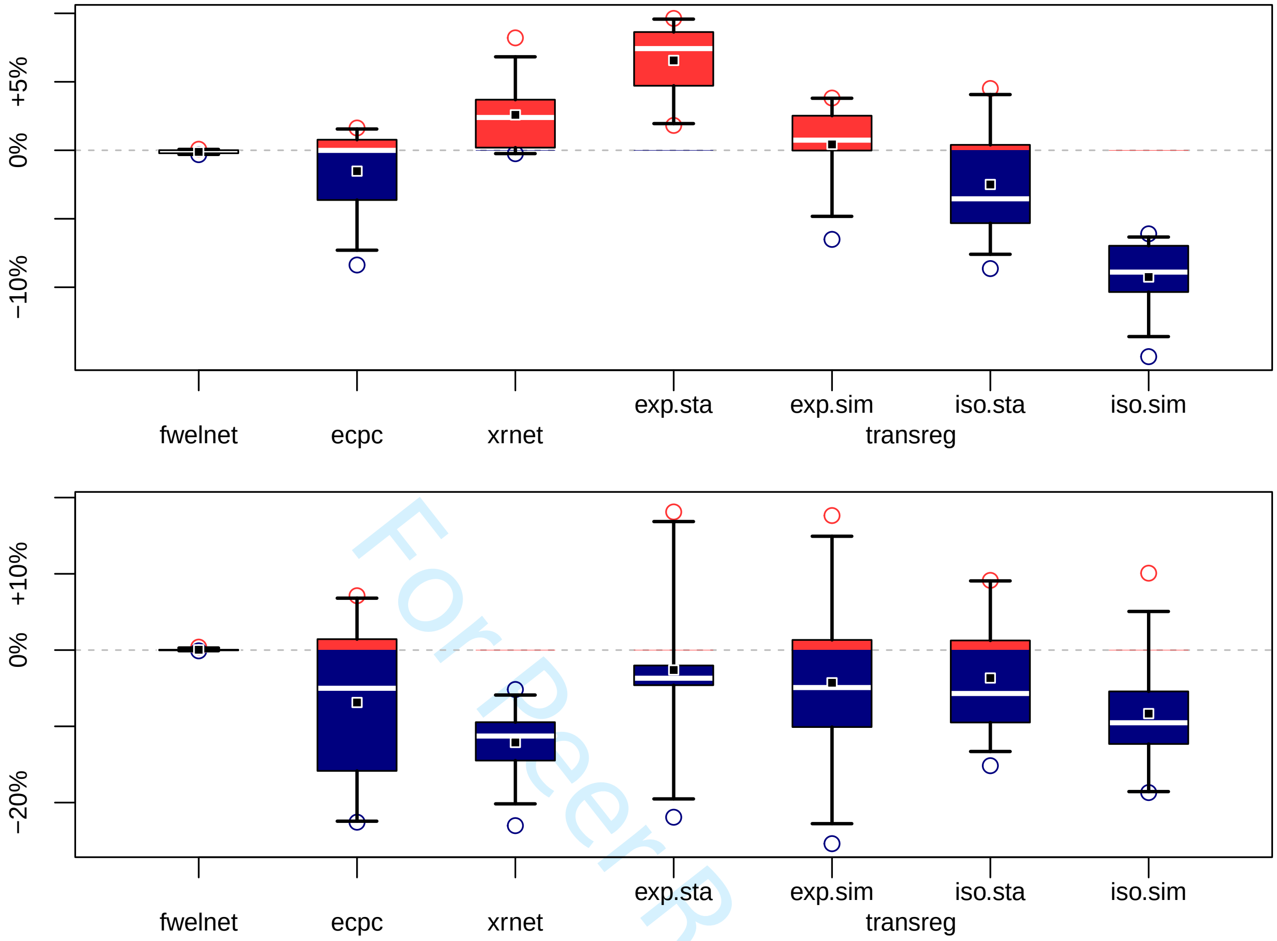

Fig. 2. Change in predictive performance in external applications due to prior information. Percentage change in cross-validated logistic deviance (y-axis) from ridge regression (with `glmnet`) to other methods (x-axis), for 10 repetitions of 10-fold cross-validation. A negative value (blue) indicates an improvement, and a positive value (red) indicates a deterioration, based on the metric 'logistic deviance'. Top: application on cervical cancer. Bottom: application on pre-eclampsia. From left to right: co-data learning methods (`fwelnet`, `ecpc`), competing transfer learning method (`xrnet`), proposed transfer learning method (`transreg`) with exponential/isotonic calibration and standard/simultaneous stacking.

- *Transfer learning*: The proposed transfer learning approach uses the signed logarithmic p-values from the source data set as prior effects for the target data set.

Figure 3 shows the predictive performance of modelling with estimated effects (uninformed approach), with prior effects (naïve transfer learning), or with both (transfer learning). We obtained the results with repeated nested cross-validation (10 repetitions, 10 external folds, 10 internal folds), using the same folds for all methods. If the significance cut-off is very strict, leading to a small number of significant SNPs, transfer learning does not improve the predictive performance of ridge and lasso regression. In these low-dimensional settings with many fewer SNPs than samples, prior information on the SNPs is not helpful. But otherwise transfer learning does improve the predictive performance of ridge and lasso regression. This holds for all four flavours of the proposed transfer learning method (exponential vs isotonic calibration, standard vs simultaneous stacking), but isotonic calibration works considerably better than exponential calibration and simultaneous stacking works marginally better than standard stacking.

Depending on the significance cut-off determining the number of significant SNPs, the performance of naïve transfer learning can be as high as the one of transfer learning with isotonic calibration. In these cases, the prior effects are predictive to the extent that it is not even necessary to estimate any effects. An explanation for the high performance of naïve transfer learning might be (i) the large sample size in the source data set for testing the marginal effects of the SNPs together with (ii) the linkage disequilibrium clumping leading to a selection of relatively independent SNPs.

Discussion

We proposed a two-step transfer learning method for exploiting estimated coefficients from related studies to improve the predictive performance in the study of interest. First, we adapt the prior effects from the source data sets to the target data set, either with exponential or isotonic calibration. While exponential calibration is more robust to outliers (only three free parameters), isotonic calibration is more flexible (only maintains order of prior effects). We expect the former to be superior if the prior effects are close to the true effects, and the latter to be superior if there is no exponential relationship. Second, we combine the calibrated prior effects with information from the observed data, based on two variants of

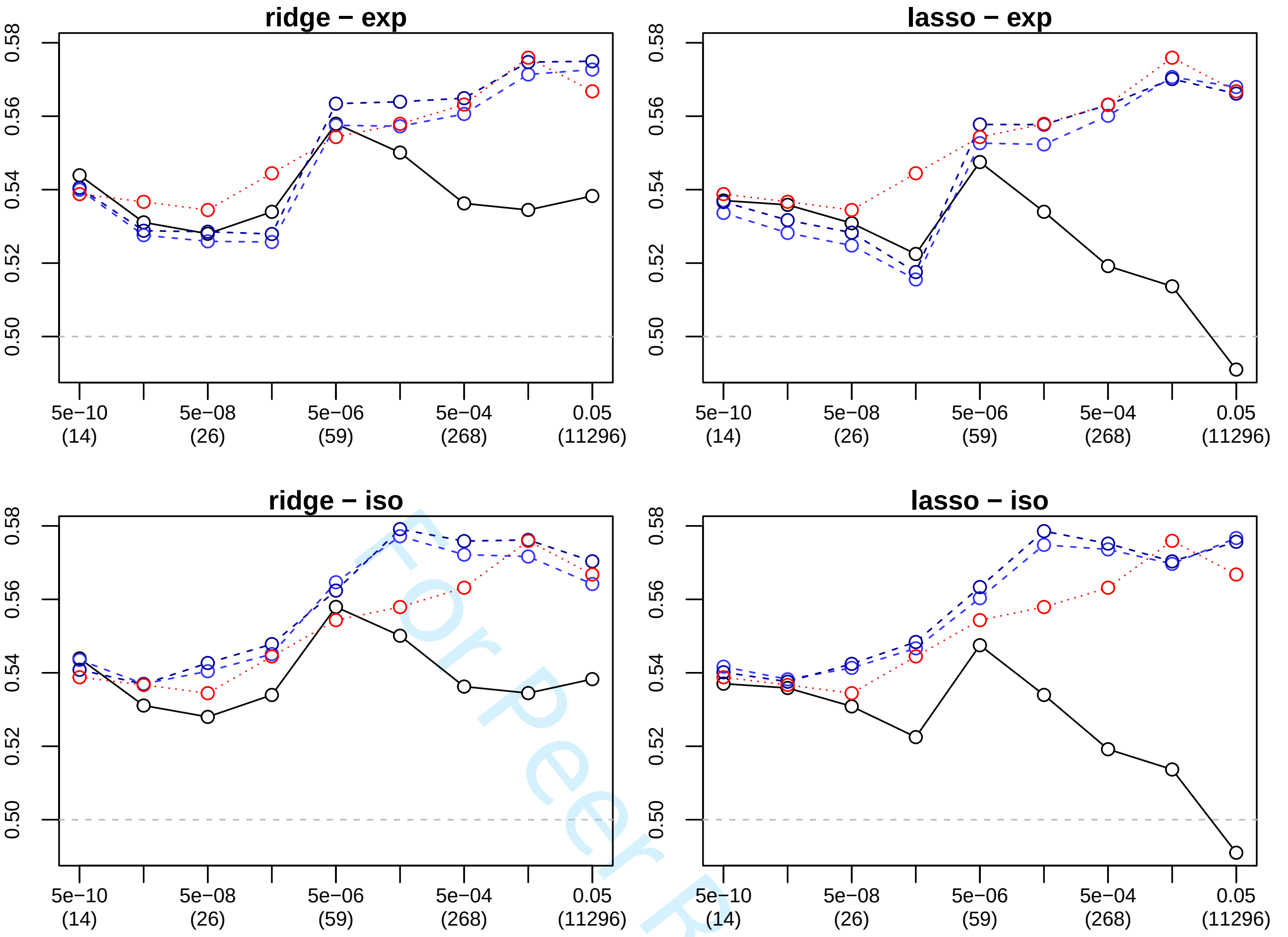

Fig. 3. Predictive performance in internal application. Mean cross-validated area under the ROC curve (AUC) from 10 times 10-fold cross-validation (y-axis) against p-value cutoff (x-axis) for regression without (solid line) and with (dashed lines) transfer learning (bright blue: standard stacking, dark blue: simultaneous stacking), under either ridge (left) or lasso (right) regularisation and either exponential (top) or isotonic (bottom) calibration. The numbers within brackets indicate the dimensionality, and the dotted line is for naïve transfer learning. While AUC = 0.5 represents random classification, AUC = 1.0 represents perfect classification. Given 766 controls and 790 cases, a random classifier achieves an AUC above 0.524 in 5% of the cases.

stacked generalisation. While the first variant (standard stacking) is more suitable if there are many sources of co-data ('averaging calibrated prior effects and estimated effects'), the second variant (simultaneous stacking) is more suitable if there is one source of co-data with partially unreliable or partially missing prior effects ('shrinking combined effects towards calibrated prior effects').

The proposed transfer learning method allows for multiple sources of prior information. It does not require the source data set(s) but only the prior effects derived from the source data set(s). The proposed sequential transfer learning method has a competitive predictive performance with the less flexible parallel transfer learning method from Tian and Feng (2022, `glmtrans`). In the case of closely related tasks, accounting for prior effects with transfer learning seems to be more beneficial than accounting for prior weights with the co-data methods from Tay et al. (2022, `fwelnet`) and van Nee et al. (2021, `ecpc`). And in the case of non-linearly related prior effects, the proposed method seems to outperform the one from Kawaguchi et al. (2021, `xrnet`). We therefore believe that the proposed method could tackle many biomedical predictions problems with one or more sets of prior effects.

In some applications, only one type of prior information derived from the source data sets is available. In other applications, multiple types of prior information are available (or the source data sets themselves). Then we can choose from multiple types of prior information. If the source and target data sets have the same feature space, estimated coefficients from penalised regression might be a reasonable choice. If the feature spaces are different, however, it is problematic that (i) lasso regression erratically selects among correlated features and (ii) ridge regression distributes weight among correlated features. This means that the presence or absence of additional correlated features in the source data sets might change the prior information on the features of interest. The same problem arises under contamination of a subset of features (van de Wiel et al., 2016). We therefore expect that signed logarithmic p-values from pairwise testing ($-\text{sign}(\hat{\beta}) \log_{10}(p)$) will often be more informative than estimated coefficients from multiple regression ($\hat{\beta}$).

Competing interests

No competing interest is declared.

Funding

We acknowledge support by the Luxembourg National Research Fund (FNR) for the project 'Clinnova - a trans-regional digital health effort' [NCER/23/16695277] and for the project DIGIPD [INTER/ERAPerMed 20/14599012] as part of the European Union's Horizon 2020 research and innovation program. The National Centre of Excellence in Research on Parkinson's Disease (NCER-PD) is funded by the FNR [FNR/NCER13/BM/11264123].

Reproducibility

The R package **transreg** is available on GitHub (https://github.com/lcsb-bds/transreg) and CRAN (https://cran.r-project.org/package=transreg), with the code for the simulations and the applications in a vignette (https://lcsb-bds.github.io/transreg/). We obtained our results using R 4.3.0 [RRID:ID_001905] on a physical machine (aarch64-apple-darwin20, macOS Ventura 13.5.2). Data for the application on cervical cancer are available from van de Wiel et al. (2016), in the R package **GRridge** in the data set 'dataVerlaat' (source data: Farkas et al., 2013, target data: van de Wiel et al., 2016). Data for the application on pre-eclampsia are available from Erez et al. (2017), in the supporting file 'pone.0181468.s001.csv'. For the application on Parkinson's disease, the source data are available from Nalls et al. (2019), in the online file 'nallsEtAl2019_excluding23andMe_allVariants.tab', and the target data are available upon request (request.ncer-pd@uni.lu). Information on reproducibility is also available on a frozen page (doi: 10.17881/hczj-3297).

Author contributions statement

EG acquired funding. AR and MvdW developed the method. AR analysed the data and drafted the manuscript. ZL processed the SNP data and critically revised the internal application. MvdW and EG critically revised the manuscript. All authors read and approved the final manuscript.

Acknowledgments

We are grateful to Quentin KLOPFENSTEIN for helpful discussions on differential penalisation, to the responsible and reproducible research (R3) initiative for the pre-publication check, and to the anonymous reviewers for constructive criticism. Data used

Table 4. Isotonic calibration. The aim is to (i) estimate the effects of the features under sign and order constraints determined by q negative (top) and $p-q$ non-negative (bottom) prior effects, i.e. estimate $\gamma_{1k},\ldots,\gamma_{pk}$ for $x_1,\ldots,x_p$ under $\hat{\gamma}_{jk}=0|z_{jk}=0$, $\hat{\gamma}_{jk}\geq 0|z_{jk}>0$, $\hat{\gamma}_{jk}\leq 0|z_{jk}<0$, $\hat{\gamma}_{jk}\geq\hat{\gamma}_{lk}|z_{jk}\geq z_{lk}$, and $\hat{\gamma}_{jk}\leq\hat{\gamma}_{lk}|z_{jk}\leq z_{lk}$. This can be solved by (ii) estimating the effects of the features ordered by the co-data under sign and order constraints, i.e. estimate $\gamma_{(1),k},\ldots,\gamma_{(p),k}$ for $x_{(1)},\ldots,x_{(p)}$ under $\hat{\gamma}_{(j),k}\leq 0|j\leq p$, $\hat{\gamma}_{(j),k}\geq 0|j>p$, and $\hat{\gamma}_{(1),k}\leq\ldots\leq\hat{\gamma}_{(p),k}$. This in turn can be solved by (iii) estimating the effects of the combined features under sign constraints, i.e. estimate $\delta_1,\ldots,\delta_p$ for $w_1,\ldots,w_p$ under $\hat{\delta}_j\leq 0|j\leq p$ and $\hat{\delta}_j\geq 0|j>p$. Our algorithm receives the original features and the prior effects (i), orders the features by the prior effects (ii), combines the features (iii), estimates the effects of the combined features (iii), calculates the estimated effects of the ordered features (ii), and returns the estimated effects of the original features (i).

(i)	$x_{\circ,1}, z_{1,k}$	$x_{\circ,2}, z_{2,k}$	$\cdots$	$x_{\circ,q-1}, z_{q-1,k}$	$x_{\circ,q}, z_{q,k}$
(ii)	$x_{\circ,(1)}$	$x_{\circ,(2)}$	$\cdots$	$x_{\circ,(q-1)}$	$x_{\circ,(q)}$
(iii)	$w_{\circ,1} = x_{\circ,(1)}$	$w_{\circ,2} = x_{\circ,(1)} + x_{\circ,(2)}$	$\cdots$	$w_{\circ,q-1} = x_{\circ,(1)} + \cdots + x_{\circ,(q-1)}$	$w_{\circ,q} = x_{\circ,(1)} + \cdots + x_{\circ,(q)}$
(iii)	$\hat{\delta}_{1,k}$	$\hat{\delta}_{2,k}$	$\cdots$	$\hat{\delta}_{q-1,k}$	$\hat{\delta}_{q,k}$
(ii)	$\hat{\gamma}_{(1),k} = \hat{\delta}_{1,k} + \cdots + \hat{\delta}_{q,k}$	$\hat{\gamma}_{(2),k} = \hat{\delta}_{2,k} + \cdots + \hat{\delta}_{q,k}$	$\cdots$	$\hat{\gamma}_{(q-1),k} = \hat{\delta}_{q-1,k} + \hat{\delta}_{q,k}$	$\hat{\gamma}_{(q),k} = \hat{\delta}_{q,k}$
(i)	$\hat{\gamma}_{1,k}$	$\hat{\gamma}_{2,k}$	$\cdots$	$\hat{\gamma}_{q-1,k}$	$\hat{\gamma}_{q,k}$
(i)	$x_{\circ,q+1}, z_{q+1,k}$	$x_{\circ,q+2}, z_{q+2,k}$	$\cdots$	$x_{\circ,p-1}, z_{p-1,k}$	$x_{\circ,p}, z_{p,k}$
(ii)	$x_{\circ,(q+1)}$	$x_{\circ,(q+2)}$	$\cdots$	$x_{\circ,(p-1)}$	$x_{\circ,(p)}$
(iii)	$w_{\circ,q+1} = x_{\circ,(q+1)} + \cdots + x_{\circ,(p)}$	$w_{\circ,q+2} = x_{\circ,(q+2)} + \ldots + x_{\circ,(p)}$	$\cdots$	$w_{\circ,p-1} = x_{\circ,(p-1)} + x_{\circ,(p)}$	$w_{\circ,p} = x_{\circ,(p)}$
(iii)	$\hat{\delta}_{q+1,k}$	$\hat{\delta}_{q+2,k}$	$\cdots$	$\hat{\delta}_{p-1,k}$	$\hat{\delta}_{p,k}$
(ii)	$\hat{\gamma}_{(q+1),k} = \hat{\delta}_{q+1,k}$	$\hat{\gamma}_{(q+2),k} = \hat{\delta}_{q+1,k} + \hat{\delta}_{q+2,k}$	$\cdots$	$\hat{\gamma}_{(p-1),k} = \hat{\delta}_{q+1,k} + \cdots + \hat{\delta}_{p-1,k}$	$\hat{\gamma}_{(p),k} = \hat{\delta}_{q+1,k} + \cdots + \hat{\delta}_{p,k}$
(i)	$\hat{\gamma}_{q+1,k}$	$\hat{\gamma}_{q+2,k}$	$\cdots$	$\hat{\gamma}_{p-1,k}$	$\hat{\gamma}_{p,k}$

in the preparation of this manuscript were obtained from NCER-PD. All participants provided written informed consent, and the study has been approved by the National Ethics Board (CNER Ref: 201407/13). We would like to thank all participants of the Luxembourg Parkinson's study for their important support of our research. Furthermore, we acknowledge the joint effort of the NCER-PD consortium members from the partner institutions Luxembourg Centre for Systems Biomedicine, Luxembourg Institute of Health, Centre Hospitalier de Luxembourg, and Laboratoire National de Santé generally contributing to the Luxembourg Parkinson's Study as listed below: Alexander HUNDT[2], Alexandre BISDORFF[5], Amir SHARIFY[2], Anne GRÜNEWALD[1], Anne-Marie HANFF[2], Armin RAUSCHENBERGER[1], Beatrice NICOLAI[3], Brit MOLLENHAUER[12], Camille BELLORA[2], Carlos VEGA MORENO[1], Chouaib MEDIOUNI[2], Christophe TREFOIS[1], Claire PAULY[1,3], Clare MACKAY[10], Clarissa GOMES[1], Daniela BERG[11], Daniela ESTEVES[2], Deborah MCINTYRE[2], Dheeraj REDDY BOBBILI[1], Eduardo ROSALES[2], Ekaterina SOBOLEVA[1], Elisa GÓMEZ DE LOPE[1], Elodie THIRY[3], Enrico GLAAB[1], Estelle HENRY[2], Estelle SANDT[2], Evi WOLLSCHEID-LENGELING[1], Francoise MEISCH[1], Friedrich MÜHLSCHLEGEL[4], Gaël HAMMOT[2], Geeta ACHARYA[2], Gelani ZELIMKHANOV[3], Gessica CONTESOTTO[2], Giuseppe ARENA[1], Gloria AGUAYO[2], Guilherme MARQUES[2], Guy BERCHEM[3], Guy FAGHERAZZI[2], Hermann THIEN[2], Ibrahim BOUSSAAD[1], Inga LIEPELT[11], Isabel ROSETY[1], Jacek JAROSLAW LEBIODA[1], Jean-Edouard SCHWEITZER[1], Jean-Paul NICOLAY[19], Jean-Yves FERRAND[2], Jens SCHWAMBORN[1], Jérôme GRAAS[2], Jessica CALMES[2], Jochen KLUCKEN[1,2,3], Johanna TROUET[2], Kate SOKOLOWSKA[2], Kathrin BROCKMANN[11], Katrin MARCUS[13], Katy BEAUMONT[2], Kirsten RUMP[1], Laura LONGHINO[3], Laure PAULY[1], Liliana VILAS BOAS[3], Linda HANSEN[1,3], Lorieza CASTILLO[2], Lukas PAVELKA[1,3], Magali PERQUIN[2], Maharshi VYAS[1], Manon GANTENBEIN[2], Marek OSTASZEWSKI[1], Margaux SCHMITT[2], Mariella GRAZIANO[17], Marijus GIRAITIS[2,3], Maura MINELLI[2], Maxime HANSEN[1,3], Mesele VALENTI[2], Michael HENEKA[1], Michael HEYMANN[2], Michel MITTELBRONN[1,4], Michel VAILLANT[2], Michele BASSIS[1], Michele HU[8], Muhammad ALI[1], Myriam ALEXANDRE[2], Myriam MENSTER[2], Nadine JACOBY[18], Nico DIEDERICH[3], Olena TSURKALENKO[2], Olivier TERWINDT[1,3], Patricia MARTINS CONDE[1], Patrick MAY[1], Paul WILMES[1], Paula Cristina LUPU[2], Pauline LAMBERT[2], Piotr GAWRON[1], Quentin KLOPFENSTEIN[1], Rajesh RAWAL[1], Rebecca TING JIIN LOO[1], Regina BECKER[1], Reinhard SCHNEIDER[1], Rejko KRÜGER[1,2,3], Rene DONDELINGER[5], Richard WADE-MARTINS[9], Robert LISZKA[14], Romain NATI[3], Rosalina RAMOS LIMA[2], Roseline LENTZ[7], Rudi BALLING[1], Sabine SCHMITZ[1], Sarah NICKELS[1], Sascha HERZINGER[1], Sinthuja PACHCHEK[1], Soumyabrata GHOSH[1], Stefano SAPIENZA[1], Sylvia HERBRINK[6], Tainá MARQUES[1], Thomas GASSER[11], Ulf NEHRBASS[2], Valentin GROUES[1], Venkata SATAGOPAM[1], Victoria LORENTZ[2], Walter MAETZLER[15], Wei GU[1], Wim AMMERLANN[2], Yohan JAROZ[1], Zied LANDOULSI[1]. [1]Luxembourg Centre for Systems Biomedicine, University of Luxembourg, Esch-sur-Alzette, Luxembourg, [2]Luxembourg Institute of Health, Strassen, Luxembourg, [3]Centre Hospitalier de Luxembourg, Strassen, Luxembourg, [4]Laboratoire National de Santé, Dudelange, Luxembourg, [5]Centre Hospitalier Emile Mayrisch, Esch-sur-Alzette, Luxembourg, [6]Centre Hospitalier du Nord, Ettelbrück, Luxembourg, [7]Parkinson Luxembourg Association, Leudelange, Luxembourg, [8]Oxford Parkinson's Disease Centre, Nuffield Department of Clinical Neurosciences, University of Oxford, Oxford, UK, [9]Oxford Parkinson's Disease Centre, Department of Physiology, Anatomy and Genetics, University of Oxford, Oxford, UK, [10]Oxford Centre for Human Brain Activity, Wellcome Centre for Integrative Neuroimaging, Department of Psychiatry, University of Oxford, Oxford, UK, [11]Center of Neurology and Hertie Institute for Clinical Brain Research, Department of Neurodegenerative Diseases, University Hospital Tübingen, Tübingen, Germany, [12]Paracelsus-Elena-Klinik, Kassel, Germany, [13]Ruhr-University of Bochum, Bochum, Germany, [14]Westpfalz-Klinikum GmbH, Kaiserslautern, Germany, [15]Department of Neurology, University Medical Center Schleswig-Holstein, Kiel, Germany, [16]Department of Neurology Philipps, University Marburg, Marburg, Germany, [17]Association of Physiotherapists in Parkinson's Disease Europe, Esch-sur-Alzette, Luxembourg, [18]Private practice, Ettelbruck, Luxembourg, [19]Private practice, Luxembourg, Luxembourg.

References

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.

Blauwendraat, C., Faghri, F., Pihlstrom, L., Geiger, J. T., Elbaz, A., Lesage, S., Corvol, J.-C., May, P., Nicolas, A., Abramzon, Y., et al. (2017). NeuroChip, an updated version of the NeuroX genotyping platform to rapidly screen for variants associated with neurological diseases. Neurobiology of Aging, 57:247.e9–247.e13. doi: 10.1016/j.neurobiolaging.2017.05.009.

Das, S., Forer, L., Schönherr, S., Sidore, C., Locke, A. E., Kwong, A., Vrieze, S. I., Chew, E. Y., Levy, S., McGue, M., et al. (2016). Next-generation genotype imputation service and methods. Nature Genetics, 48(10):1284–1287. doi: 10.1038/ng.3656.

Dhruba, S. R. (2021). Application of advanced machine learning based approaches in cancer precision medicine. PhD thesis. `https://hdl.handle.net/2346/88264`. (R package `DMTL`).

Erez, O., Romero, R., Maymon, E., Chaemsaithong, P., Done, B., Pacora, P., Panaitescu, B., Chaiworapongsa, T., Hassan, S. S., and Tarca, A. L. (2017). The prediction of late-onset preeclampsia: Results from a longitudinal proteomics study. PloS one, 12(7):e0181468. doi: 10.1371/journal.pone.0181468.

Farkas, S. A., Milutin-Gašperov, N., Grce, M., and Nilsson, T. K. (2013). Genome-wide DNA methylation assay reveals novel candidate biomarker genes in cervical cancer. Epigenetics, 8(11):1213–1225. doi: 10.4161/epi.26346.

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`).

Gamarnik, D. and Zadik, I. (2017). High-dimensional regression with binary coefficients. Estimating squared error and a phase transtition. In Conference on Learning Theory, volume 65 of Proceedings of Machine Learning Research, pages 948–953. doi: `http://proceedings.mlr.press/v65/david17a`.

Hipp, G., Vaillant, M., Diederich, N. J., Roomp, K., Satagopam, V. P., Banda, P., Sandt, E., Mommaerts, K., Schmitz, S. K., Longhino, L., et al. (2018). The Luxembourg Parkinson's study: a comprehensive approach for stratification and early diagnosis. Frontiers in Aging Neuroscience, 10:326. doi: 10.3389/fnagi.2018.00326.

Jiang, Y., He, Y., and Zhang, H. (2016). Variable selection with prior information for generalized linear models via the prior LASSO method. Journal of the American Statistical Association, 111(513):355–376. doi: 10.1080/01621459.2015.1008363. (R script `pLASSO`).

Kawaguchi, E. S., Li, S., Weaver, G. M., and Lewinger, J. P. (2021). Hierarchical ridge regression for incorporating prior information in genomic studies. Journal of Data Science, 20(1):34–50. doi: 10.6339/21-JDS1030. (R package `xrnet`).

Liu, Y., Kang, Y., Xing, C., Chen, T., and Yang, Q. (2020). A secure federated transfer learning framework. IEEE Intelligent Systems, 35(4):70–82. doi: 10.1109/MIS.2020.2988525.

Nalls, M. A., Blauwendraat, C., Vallerga, C. L., Heilbron, K., Bandres-Ciga, S., Chang, D., Tan, M., Kia, D. A., Noyce, A. J., Xue, A., et al. (2019). Identification of novel risk loci, causal insights, and heritable risk for parkinson's disease: a meta-analysis of genome-wide association studies. The Lancet Neurology, 18(12):1091–1102. doi: 10.1016/S1474-4422(19)30320-5.

Pavelka, L., Rauschenberger, A., Landoulsi, Z., Pachchek, S., May, P., Glaab, E., and Krüger, R. (2022). Age at onset as stratifier in idiopathic parkinson's disease - effect of ageing and polygenic risk score on clinical phenotypes. npj Parkinson's Disease, 8:102. doi: 10.1038/s41531-022-00342-7.

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., Glaab, E., and van de Wiel, M. A. (2021). Predictive and interpretable models via the stacked elastic net. Bioinformatics, 37(14):2012–2016. doi: 10.1093/bioinformatics/btaa535. (R package `starnet`).

Tay, J. K., Aghaeepour, N., Hastie, T., and Tibshirani, R. (2022). Feature-weighted elastic net: using "features of features" for better prediction. Statistica Sinica, in press. doi: 10.5705/ss.202020.0226. (R package `fwelnet`).

te Beest, D. E., Mes, S. W., Wilting, S. M., Brakenhoff, R. H., and van de Wiel, M. A. (2017). Improved high-dimensional prediction with Random Forests by the use of co-data. BMC Bioinformatics, 18:584. doi: 10.1186/s12859-017-1993-1. (R package `CoRF`).

Tian, Y. and Feng, Y. (2022). Transfer learning under high-dimensional generalized linear models. Journal of the American Statistical Association, in press. doi: 10.1080/01621459.2022.2071278. (R package `glmtrans`).

van de Wiel, M. A., Lien, T. G., Verlaat, W., van Wieringen, W. N., and Wilting, S. M. (2016). Better prediction by use of co-data: adaptive group-regularized ridge regression. Statistics in Medicine, 35(3):368–381. doi: 10.1002/sim.6732. (R package `GRridge`).

van de Wiel, M. A., van Nee, M. M., and Rauschenberger, A. (2021). Fast cross-validation for multi-penalty high-dimensional ridge regression. Journal of Computational and Graphical Statistics, 30(4):835–847. doi: 10.1080/10618600.2021.1904962. (R package `multiridge`).

van Nee, M. M., Wessels, L. F., and van de Wiel, M. A. (2021). Flexible co-data learning for high-dimensional prediction. Statistics in Medicine, 40(26):5910–5925. doi: 10.1002/sim.9162. (R package `ecpc`).

Wolpert, D. H. (1992). Stacked generalization. Neural Networks, 5(2):241–259. doi: 10.1016/S0893-6080(05)80023-1.

Zeng, C., Thomas, D. C., and Lewinger, J. P. (2021). Incorporating prior knowledge into regularized regression. Bioinformatics, 37(4):514–521. doi: 10.1093/bioinformatics/btaa776. (R package `xtune`).