Regularization in directable environments with application to Tetris

SUPPLEMENTARY MATERIAL

Jan Malte Lichtenberg [1] **Özgür Şimşek** [1]

This supplementary material provides mathematical proofs (Section A), implementation details for the supervised learning experiments (Section B), implementation details for the Tetris experiments and a description of the M-learning algorithm (Section C), additional results and figures (Section D), and detailed information about the real-world data sets (Section E). References to numbered sections and figures refer to sections in the main article.

A. Mathematical Proofs

Theorem 1. *Let $\boldsymbol{y} \sim (\boldsymbol{X}\boldsymbol{\beta}, \sigma^2 \boldsymbol{I}_{n\times n})$, where $||\boldsymbol{\beta}||^2 < \infty$, $\sigma^2 > 0$, and $\boldsymbol{I}_{n\times n}$ is the identity matrix of size n. Let $\bar{\boldsymbol{\beta}} := \frac{1}{p}\sum_{i=1}^{p} \beta_i$ denote the mean of the true weights. Then,*
(1) The minimum-bias equal-weighting estimator of γ is $\bar{\boldsymbol{\beta}}$.
(2) For orthonormal data matrix $\boldsymbol{X}$ (i.e., $\boldsymbol{X}^T\boldsymbol{X} = \boldsymbol{I}_{p\times p}$),
(a) EW is the minimum-bias equal-weighting estimator,
(b) $\Delta bias^2 = p\bar{\boldsymbol{\beta}}^2$,
(c) $\Delta MSE = p\bar{\boldsymbol{\beta}}^2 - \sigma^2$,
(d) The squared mean weight $\bar{\boldsymbol{\beta}}^2$, and thus $\Delta bias^2$ and ΔMSE, is higher on a directed set of weights than on an undirected set of weights.

Proof. **(1)** The bias of an equal-weighting estimator is

$$\begin{aligned} ||\mathbb{E}[\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}]||_2^2 &= ||\mathbb{E}[\gamma \mathbb{1} - \boldsymbol{\beta}]||_2^2 \\ &= \sum_{i=1}^{p} (\gamma - \beta_i)^2 \\ &= \sum_{i=1}^{p} (\gamma^2 - 2\gamma\beta_i + \beta_i^2). \end{aligned}$$

Taking the derivative with respect to γ yields

$$\frac{\partial}{\partial \gamma} \sum_{i=1}^{p} (\gamma^2 - 2\gamma\beta_i + \beta_i^2) = \sum_{i=1}^{p} (2\gamma - 2\beta_i), \tag{1}$$

[1]Department of Computer Science, University of Bath, Bath, United Kingdom. Correspondence to: Jan Malte Lichtenberg <j.m.lichtenberg@bath.ac.uk>.

Supplementary Material for the paper "Regularization in directable environments with application in Tetris" which appears in *Proceedings of the 36^{th} International Conference on Machine Learning*, Long Beach, California, PMLR 97, 2019.

and equating (1) to 0 yields

$$\begin{aligned}
&\sum_{i=1}^{p}(2\gamma - 2\beta_i) = 0 \\
\Leftrightarrow \qquad & 2\gamma p = 2\sum_{i=1}^{p}\beta_i \\
\Leftrightarrow \qquad & \gamma = \frac{\sum_{i=1}^{p}\beta_i}{p} = \bar{\boldsymbol{\beta}}.
\end{aligned}$$

(2a) From here on, we assume $\boldsymbol{X}^T\boldsymbol{X} = \boldsymbol{I}_{p\times p}$. Recall that EW's γ^{EW} is calculated using simple linear regression. Because all variables are assumed to be standardized, the intercept β_0 can be omitted and EW takes the form

$$\hat{\boldsymbol{y}} = \gamma\sum_{i=1}^{p}\boldsymbol{x}_i.$$

Defining $\boldsymbol{c} = \sum_{i=1}^{p}\boldsymbol{x}_i = \boldsymbol{X}\mathbb{1}$, where $\mathbb{1}$ is a p-vector of ones, the simple linear regression estimate is given by

$$\gamma^{EW} = \frac{r_{\boldsymbol{yc}}}{s_{\boldsymbol{c}}}, \tag{2}$$

where $r_{\boldsymbol{yc}}$ is the sample correlation coefficient between $\boldsymbol{y}$ and $\boldsymbol{c}$, and $s_{\boldsymbol{c}}$ is the standard deviation of $\boldsymbol{c}$. For orthogonal $\boldsymbol{X}$ and $\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$ with $\mathbb{E}[\boldsymbol{\epsilon}] = \boldsymbol{0}$, the expected value of EW's γ^{EW} is given by

$$\begin{aligned}
\mathbb{E}[\gamma^{EW}] &= \mathbb{E}[\frac{\mathbb{1}^T\boldsymbol{X}^T\boldsymbol{y}}{\mathbb{1}^T\boldsymbol{X}^T\boldsymbol{X}\mathbb{1}}] \\
&= \mathbb{E}[\frac{\mathbb{1}^T\boldsymbol{X}^T(\boldsymbol{X}\boldsymbol{\beta}+\boldsymbol{\epsilon})}{\mathbb{1}^T\mathbb{1}}] \\
&= \mathbb{E}[\frac{\mathbb{1}^T\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{\beta}}{\mathbb{1}^T\mathbb{1}}] + \mathbb{E}[\frac{\boldsymbol{\epsilon}}{\mathbb{1}^T\mathbb{1}}] \\
&= \frac{\sum_{i=1}^{p}\beta_i}{p} + 0 \\
&= \bar{\boldsymbol{\beta}},
\end{aligned}$$

which was to be shown.

(2b) We compute the squared biases for EW and the **0**-model. Let $\hat{\boldsymbol{\beta}}_{EW} = \gamma^{EW}\mathbb{1}$ and $\hat{\boldsymbol{\beta}}_{\boldsymbol{0}} = \boldsymbol{0}$ the weight estimates for EW and the **0**-model, respectively. Then, the squared biases are given by

$$\begin{aligned}
\text{bias}^2(\hat{\boldsymbol{\beta}}_{EW}) &= ||\mathbb{E}[\hat{\boldsymbol{\beta}}_{EW}] - \boldsymbol{\beta}||^2 \\
&= (\bar{\boldsymbol{\beta}}\mathbb{1} - \boldsymbol{\beta})^T(\bar{\boldsymbol{\beta}}\mathbb{1} - \boldsymbol{\beta}) \\
&= p\left(\bar{\boldsymbol{\beta}}\right)^2 - 2\sum_{i=1}^{p}\bar{\boldsymbol{\beta}}\beta_i + \boldsymbol{\beta}^T\boldsymbol{\beta} \\
&= \boldsymbol{\beta}^T\boldsymbol{\beta} - p\left(\bar{\boldsymbol{\beta}}\right)^2 \\
&= ||\boldsymbol{\beta}||^2 - p\left(\bar{\boldsymbol{\beta}}\right)^2,
\end{aligned}$$

and

$$\begin{aligned}
\text{bias}^2(\hat{\boldsymbol{\beta}}_{\boldsymbol{0}}) &= ||\boldsymbol{0} - \boldsymbol{\beta}||^2 \\
&= ||\boldsymbol{\beta}||^2.
\end{aligned}$$

The difference in bias Δbias^2 directly follows

$$\begin{aligned}\Delta\text{bias}^2 &= \text{bias}^2(\hat{\boldsymbol{\beta}}_{\mathbf{0}}) - \text{bias}^2(\hat{\boldsymbol{\beta}}_{EW}) \\ &= ||\boldsymbol{\beta}||^2 - (||\boldsymbol{\beta}||^2 - p\left(\bar{\boldsymbol{\beta}}\right)^2) \\ &= p\left(\bar{\boldsymbol{\beta}}\right)^2 = p\bar{\boldsymbol{\beta}}^2.\end{aligned}$$

(2c) The variance for the $\mathbf{0} - model$ is clearly 0. For the EW model, we use Equation (2) to note that the EW estimate $\hat{\boldsymbol{\beta}}_{EW} = \mathbb{1}(\mathbb{1}^T\boldsymbol{X}^T\boldsymbol{X}\mathbb{1})^{-1}\mathbb{1}^T\boldsymbol{X}^T\boldsymbol{y}$ is a linear function of $\boldsymbol{y}$. The trace of its variance $tr(Var(\hat{\boldsymbol{\beta}}_{EW}))$ is thus

$$tr(\mathbb{1}(\mathbb{1}^T\boldsymbol{X}^T\boldsymbol{X}\mathbb{1})^{-1}\mathbb{1}^T\boldsymbol{X}^T Var(\boldsymbol{y})\boldsymbol{X}\mathbb{1}(\mathbb{1}^T\boldsymbol{X}^T\boldsymbol{X}\mathbb{1}^T)^{-1}\mathbb{1}^T) = \sigma^2$$

The result follows directly in combination with Result 2b.

(2d) It remains to show that the squared average weight $\left(\bar{\boldsymbol{\beta}}\right)^2$ is larger on a set of directed predictors than on an undirected set of predictors. Let $\boldsymbol{\beta}_{||} = (|\beta_1|, \ldots, |\beta_p|)$ be the weights on a positively directed data set. Then

$$\begin{aligned} & \left(\overline{\boldsymbol{\beta}_{||}}\right)^2 \geq \left(\bar{\boldsymbol{\beta}}\right)^2 \\ \Leftrightarrow\ & \left|\overline{\boldsymbol{\beta}_{||}}\right| \geq \left|\bar{\boldsymbol{\beta}}\right| \\ \Leftrightarrow\ & \overline{\boldsymbol{\beta}_{||}} \geq \left|\bar{\boldsymbol{\beta}}\right|, \end{aligned}$$

where the last line follows directly from Jensen's inequality for the convex function $f(x) = |x|$. □

B. Supervised Learning Experiments

B.1. Algorithms

We used k-fold cross validation to optimize the regularization strength λ. The cross-validation parameter k was set to $min(10, n)$, where n is the training set size.

Shrinkage toward equal weights (STEW). We used the l_2-penalty. Solutions were computed using the closed-form solution given in the main article. The regularization strength λ was optimized on the training set using k-fold cross validation on a log-spaced grid with maximum 100 candidate values (the search stops early when the norm of the difference between actual and previously estimated weights is smaller than some small $\epsilon > 0$).

Equal weights (EW). EW was implemented using univariate linear regression with the only feature being the sum of features.

Nonnegative lasso. We used the R package *nnlasso* (Ma, 2016) to estimate non-negative least squares with lasso penalty (NNLasso). The regularization strength λ was optimized using k-fold cross-validation and using the built-in search path of nnlasso.

Ridge regression, the Lasso, and the elastic net. We used the R package *glmnet* (Friedman et al., 2015). The elastic net has two main parameters. Parameter $\lambda \geq 0$ controls the overall strength of regularization. Parameter $0 \leq \alpha \leq 1$ controls the amount of *ridge* versus *Lasso* characteristics. The parameters α and λ were jointly optimized using k-fold cross-validation on a two-dimensional grid with $\alpha \in \{0, 0.25, 0.5, 0.75, 1\}$ and λ on the built-in search path of glmnet, which uses a log-spaced grid with a maximum of 100 candidate values. For ridge regression and the Lasso, only λ has to be tuned. We used the built-in search path of the glmnet package.

B.2. Method

Simulated environments. The synthetic data sets used in Section 5.1 were generated as follows. For each environment (defined by a weight prior distribution, see Section 5), 400 data sets were generated. Each data set consists of 11000 observations $(y_i, x_{i1}, \ldots, x_{i20})$. Features were sampled according to $X_i \overset{i.i.d}{\sim} \mathcal{N}(0, 1)$ and weights were sampled according to the respective distribution of the environment (e.g., $\beta_j \overset{i.i.d}{\sim} \mathcal{U}(2, 8)$ for Figure 2a). Responses y_i were then generated using the linear model $Y = X_1\beta_1 + \cdots + X_{20}\beta_{20} + \epsilon$, where $\epsilon \overset{i.i.d}{\sim} \mathcal{N}(0, 1)$ is the noise term.

Learning curves on synthetic data sets in Figures 2 and 3 were computed as follows. For each data set we subsampled training sets with varying training set sizes. Model parameters were estimated on these training sets and the models were evaluated on an independent test set of 10000 observations.

Real-world environments. Real-world data sets and their sources are described in detail in Section E. All data sets are publicly available. Responses and predictors of these real-world data sets were standardized to have zero mean and unit variance across the entire data set. Missing predictor values were mean imputed and observations with missing response values were removed from the data set.

Learning curves on real-world data sets (Figure 4) were computed as follows. We set aside a random subset of 10% of the observations as test set. We then progressively sampled training sets of increasing size using the remaining observations to train the different algorithms. Results were averaged across 200 repetitions.

C. Tetris and M-learning

Our implementation of M-learning in Tetris is available under `https://github.com/janmaltel/stew-tetris`. Pseudo-code for M-learning can be found in Algorithm 1. We also provide further details about the algorithm and compare it to related literature.

Function approximation. The utility function $U(s,a)$ is usually represented by a function of a set of features. Let $\boldsymbol{\phi}(s,a)$ denote feature values that correspond to selecting action a in state s. Then $U(s,a) = f(\boldsymbol{\beta}, \boldsymbol{\phi}(s,a))$, where $\boldsymbol{\beta}$ are the weights to be estimated. In particular, we consider linear function approximation, that is, $U(s,a) = \boldsymbol{\beta}^T \boldsymbol{\phi}(s,a)$.

Choice data sets. Discrete choice models are trained using a data set of observed choice decisions made by one or multiple agents. One observation in this data set consists of the set of alternatives available to the agent (also called *choice set*) and the outcome of the decision, that is, the identity of the chosen alternative. Table 1 depicts an example of a typical discrete choice data set including 4 observations (highlighted by alternating shadings). Unlike in typical regression or classification data sets, one observation here corresponds to multiple rows, each describing one alternative in the choice set. Note that the number of alternatives varies between different choice sets. The *choice* variable indicates for each alternative whether it is chosen or not.

choice_set_id	**choice**	$\boldsymbol{\phi_1(s,a)}$	$\boldsymbol{\phi_2(s,a)}$	...	$\boldsymbol{\phi_p(s,a)}$
0	0	$\phi_1(s_0,a_1)$	$\phi_2(s_0,a_1)$	...	$\phi_p(s_0,a_1)$
0	0	$\phi_1(s_0,a_2)$	$\phi_2(s_0,a_2)$	...	...
0	1	$\phi_1(s_0,a_3)$	...		
1	0	$\phi_1(s_1,a_1)$			
1	1	$\phi_1(s_1,a_2)$			
1	0	$\phi_1(s_1,a_3)$			
1	0	$\phi_1(s_1,a_4)$			
2	1	$\phi_1(s_2,a_1)$			
2	0	$\phi_1(s_2,a_2)$			
3	0	$\phi_1(s_3,a_1)$			...
3	0	$\phi_1(s_3,a_2)$	...	...	$\phi_p(s_3,a_2)$
3	0	$\phi_1(s_3,a_3)$	...	...	$\phi_p(s_3,a_3)$
3	0	$\phi_1(s_3,a_4)$	...	...	$\phi_p(s_3,a_4)$
3	1	$\phi_1(s_3,a_5)$	...	...	$\phi_p(s_3,a_5)$
...	...	...	...	...	...

Table 1. Example of a data set used to train multinomial logistic regression in Algorithm 1. Choice sets are of varying size. They are identified by a *choice_set_id*. The *choice* variable denotes the chosen alternative for each choice set. Each alternative in the choice set is characterized by p feature values $\phi_1, \dots, \phi_p$.

Multinomial logistic regression. We used multinomial logistic regression (e.g., Train, 2009) with and without regularization to estimate the action-preference function $U(s,a)$. Multinomial logistic regression maximizes the likelihood of the chosen actions in the training set if the agent were to use action-selection probabilities $p(s,a) = \frac{e^{U(s,a)}}{\sum_{a' \in \mathcal{A}(s)} e^{U(s,a')}}$. The penalized log-likelihood of a weight vector $\boldsymbol{\beta}$ on choice set data $\mathcal{D}$ is given by

$$\log \mathcal{L}(\boldsymbol{\beta}|\mathcal{D}) = \sum_{(s_i,a_i)\in\mathcal{D}} \log(p(s_i,a_i)) - \lambda P(\boldsymbol{\beta}), \tag{3}$$

Algorithm 1 M-learning with multinomial logistic regression and rollouts

Notation:
$p \in \mathbb{N}$ *// number of features*
$\mathcal{A}(s)$ *// set of actions available in state s*
Output:
$\boldsymbol{\beta} \in \mathbb{R}^p$ *// vector of action-preference weights, initialized randomly*
Input:
$U(s,a) = f(\boldsymbol{\beta}, \boldsymbol{\phi}(s,a))$, where *// action-preference function, e.g., linear $U(s,a) = \boldsymbol{\beta}^T \boldsymbol{\phi}(s,a)$.*
  $\boldsymbol{\phi}(s,a) \in \mathbb{R}^p$ *// vector of state-action features*
$\mathcal{D} = \emptyset$ *// data structure to store choice sets*
$M \in \mathbb{N}$ *// number of rollouts*
$T \in \mathbb{N}$ *// rollout length*
$\gamma \in [0,1]$ *// discount factor*
$n(k) : \mathbb{N} \to \mathbb{N}$ *// batch size at step k*
$\pi_r(s, \boldsymbol{\beta}) : \mathcal{S} \times \mathbb{R}^p \to \mathbb{R}$ *// rollout policy (e.g., Equation 4) that returns an action for given s and $\boldsymbol{\beta}$*

$s \leftarrow$ state sampled from initial state distribution
for $k = 0, 1, 2, \ldots$ **do**

  // Rollout action selection
  for all $a \in \mathcal{A}(s)$ **do**
    $\hat{U}(s,a) \leftarrow \textsc{Rollout}(s, a, M, T, \gamma, \pi_r(s, \boldsymbol{\beta}))$ (see Algorithm 2)
  end for
  $\tilde{a} \leftarrow \operatorname*{argmax}_{a \in \mathcal{A}(s)} \hat{U}(s,a)$
  Take action $\tilde{a}$ and observe new state s'
  if s' is not terminal **then**
    $\mathcal{D} \leftarrow \mathcal{D} \cup \{\{\tilde{a}, \boldsymbol{\phi}(s,a_1), \boldsymbol{\phi}(s,a_2), ..., \boldsymbol{\phi}(s, a_{|\mathcal{A}(s)|})\}\}$ *// append choice set to $\mathcal{D}$*
    $s \leftarrow s'$
  else
    $s \leftarrow$ state sampled from initial state distribution *// i.e., reset epsiode*
  end if

  // Learn
  Construct batch $\mathcal{D}_k$ using $n(k)$ most recent choice sets from $\mathcal{D}$
  $\boldsymbol{\beta} \leftarrow \textsc{Train}(\mathcal{D}_k)$ *// e.g., multinomial logistic regression (see Equation 3)*

end for

where s_i and a_i are the state and selected action of observation i, the parameter λ is the regularization strength and P is the regularization penalty. Note that the choice probability $p(s_i, a_i)$ depends on $\boldsymbol{\beta}$ because $U(s,a)$ depends on $\boldsymbol{\beta}$.

Rollout search. Pseudo-code for the rollout procedure is given in Algorithm 2. The rollout search makes use of a rollout policy $\pi_r(s, \boldsymbol{\beta})$ that is informed by the current set of weights. We used a rollout policy that selects the $U(s,a)$-maximizing action unless an intermediate clearing of a line is possible, in which case the action that maximizes immediate reward is selected. This policy can be formalized as follows.

$$\pi_r(s, \boldsymbol{\beta}) = \begin{cases} \operatorname*{argmax}_{a \in \mathcal{A}(s)} r(s,a), & \text{if any } r(s,a) > 0 \\ \operatorname*{argmax}_{a \in \mathcal{A}(s)} U(s,a), & \text{else,} \end{cases} \tag{4}$$

where $r(s,a)$ is the immediate reward obtained upon taking action a in state s. Note that π_r depends on $\boldsymbol{\beta}$ because $U(s,a)$ depends on $\boldsymbol{\beta}$.

Tetris features. Eight features were used to describe a state-action pair: *landing height, number of eroded piece cells, row*

Algorithm 2 Rollout($s, a, M, T, \gamma, \pi_r(s, \boldsymbol{\beta})$)

Notation:
$\mathcal{G}(s, a) : \mathcal{S} \times \mathcal{A}(s) \rightarrow \mathcal{S} \times \mathbb{R}$ *// generative model that returns new state s' and reward r for given s and a*
Input:
$s \in \mathcal{S}$ *// state*
$a \in \mathcal{A}(s)$ *// action to be evaluated*
$M \in \mathbb{N}$ *// number of rollouts*
$T \in \mathbb{N}$ *// rollout length*
$\gamma \in [0, 1]$ *// discount factor*
$\pi_r(s, \boldsymbol{\beta}) : \mathcal{S} \times \mathbb{R}^p \rightarrow \mathbb{R}$ *// rollout policy (e.g., Equation 4) that returns an action for given s and $\boldsymbol{\beta}$*

for all $j = 1, \ldots, M$ **do**
  $(s', r) \leftarrow \mathcal{G}(s, a)$
  $\hat{U}_j \leftarrow r$
  $s \leftarrow s'$
  for all $t = 1, \ldots, T - 1$ **do**
    $(s', r) \leftarrow \mathcal{G}(s, \pi_r(s, \boldsymbol{\beta}))$
    $\hat{U}_j \leftarrow \hat{U}_j + \gamma^t r$
    $s \leftarrow s'$
  end for
end for
return $\hat{U} \leftarrow \frac{1}{M} \sum_{j=1}^{M} \hat{U}_j$

transitions, column transitions, number of holes, number of board wells, hole depth, and number of rows with holes. These features are from earlier work by Thiery & Scherrer (2009) who describe them in detail. Initial directions for all features were obtained from the signs of the weights of the BCTS policy (Thiery & Scherrer, 2009). They were used to direct all features so that they had positive direction. The directions used are given in Table 2.

ID	Feature	Direction
1	Rows with holes	negative
2	Column transitions	negative
3	Holes	negative
4	Landing height	negative
5	Cumulative wells	negative
6	Row transitions	negative
7	Eroded piece cells	positive
8	Hole depth	negative

Table 2. Feature names and directions of the BCTS policy. The features are ordered from top to bottom by decreasing magnitude of the corresponding weight. This is the ordering used for cumulative dominance filters.

Dominance filters. We used *simple dominance* and *cumulative dominance* filters as described by Şimşek et al. (2016). Both filters require information about feature directions. Directions were set to the signs of the weights of the BCTS policy (Thiery & Scherrer, 2009), as shown in Table 2. Cumulative dominance filters also require an ordering of the features. We ordered features by decreasing magnitude of BCTS weights (as shown in Table 2, from top to bottom). The use of these filters reduced considerably the number of actions to be evaluated in a given state $\mathcal{A}(s)$, and thus the computational costs of the rollout procedure. The filters were used during learning but not during testing.

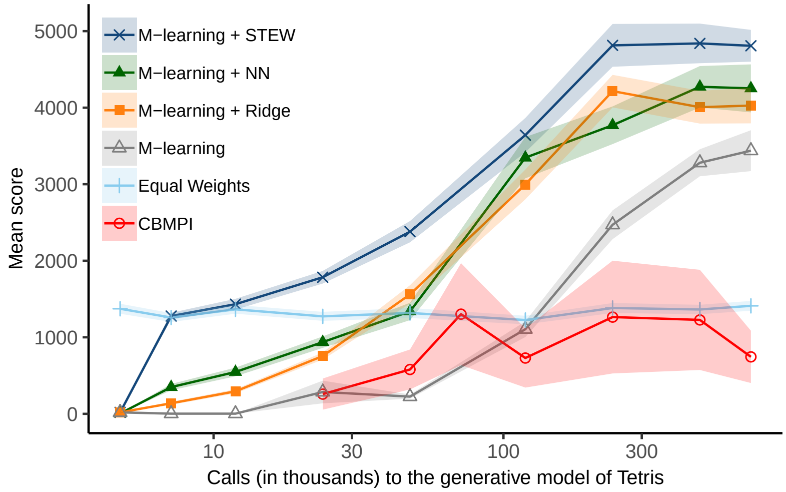

Figure D.1. Quality of the policy learned as a function of calls to the generative model (in thousands). Each learning curve shows means across 100 replications of the algorithm. Quality of the policy is measured by the mean score obtained by the policy in 10 Tetris games.

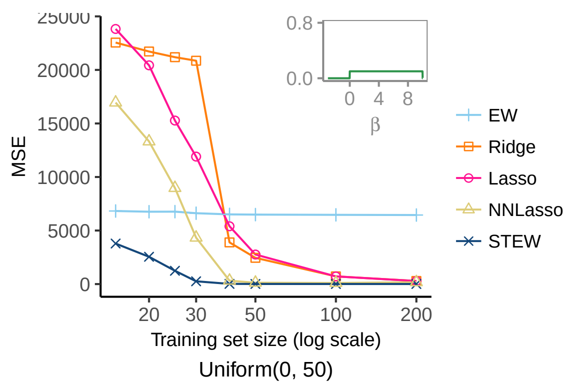

Figure D.2. Average mean squared error (MSE) across 400 repetitions for equal weights (EW), ridge regression, the Lasso, the non-negative Lasso (NNLasso) and shrinkage toward equal weights (STEW) as a function of training set size in a $\boldsymbol{\beta} \sim \mathcal{U}(0, 50)$ environments defined by weight priors shown by shown in the top-right corner of each panel.

D. Additional Results

D.1. CBMPI in Tetris

We examined the performance of CBMPI for budget sizes similar to those used for M-learning. Other CBMPI parameter values were set to those used by the best-performing CBMPI algorithm on the 10×10 board, which uses a rollout length of $m = 5$. With a budget of 2380, value function estimates diverged and the algorithm could not learn a useful policy. Figure D.1 presents results for CBMPI using a budget of 23800 per iteration, that is, 10 times the per-iteration budget we used for M-learning.

D.2. Simulation Analysis

Figure D.2 shows learning curves for equal weights (EW), ridge regression, the Lasso, the non-negative Lasso (NNLasso) and shrinkage toward equal weights (STEW) for an environment, defined by $\boldsymbol{\beta} \sim \mathcal{U}(0, 50)$.

D.3. Ordering of Predictors in Total Variation Models

The regularization behavior of total variation (TV) models depends on the order of the predictors. We compare the regularization paths of TV models with different orders of predictors to those of the l_1 and l_2 versions of STEW. We used

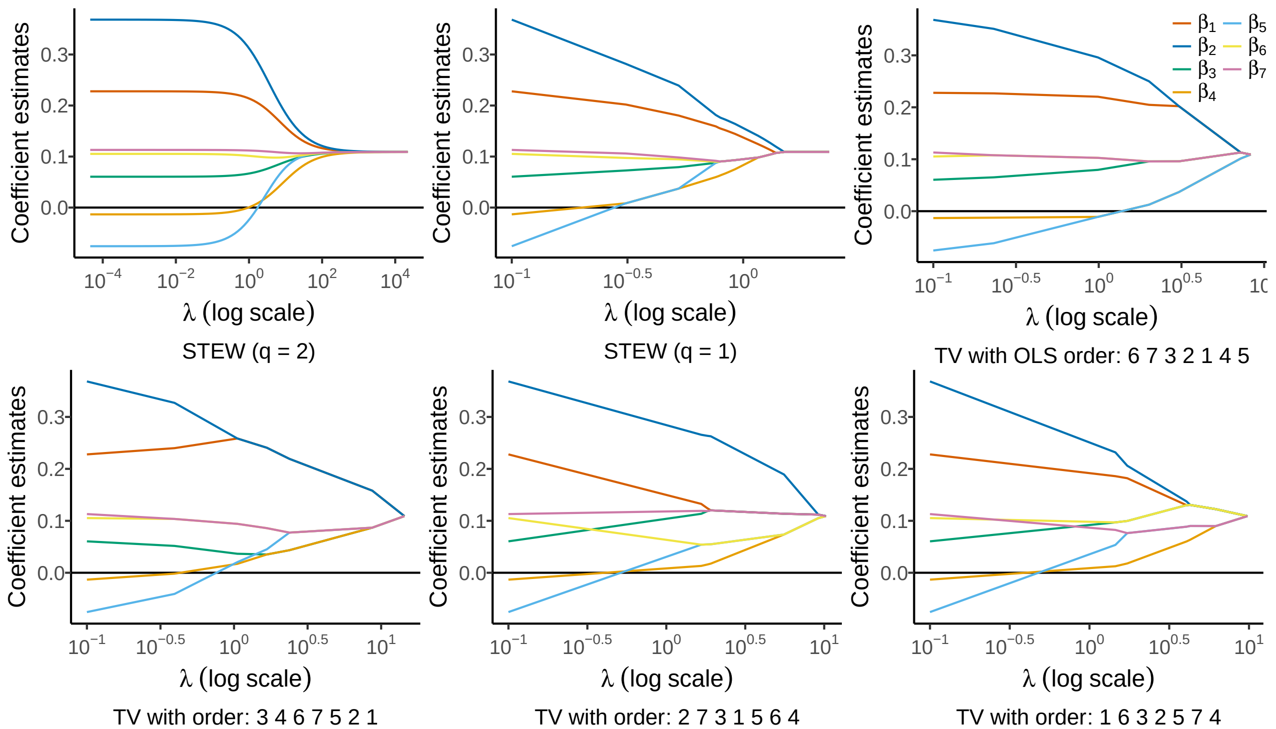

Figure D.3. Weight estimates as a function of regularization strength λ for shrinkage toward equal weights (STEW) with l_1 and l_2 penalties, and total variation (TV) with predictors ordered by OLS estimates (top row), as well as TV models with three random orderings of predictors (bottom row) and the Lasso on the *Rent* data set with seven standardized predictors.

the same *Rent* data set that was used in Figures 1 and 4 of the main article.

The first two panels from the left of the top row in Figure D.3 reproduce the regularization paths for the l_1 and l_2 versions of STEW, shown in Figure 1 of the main article. The rightmost panel of the top row shows the regularization path taken by a TV model, where the predictors have been ordered by the value of the OLS estimates on the training set. The bottom row shows the regularization paths for three TV models with randomly chosen orderings of predictors.

The regularization behavior changes notably for different orderings of predictors. Future work could examine the prediction accuracy of the TV model as a function of different orders of predictors.

D.4. Individual Learning Curves on Real-World Data Sets

Figures 5c and 5f in the paper show average learning curves across 13 data sets. Here we show individual learning curves for Lasso-directed data sets D.4 and data sets that were predicted based on the training set D.5.

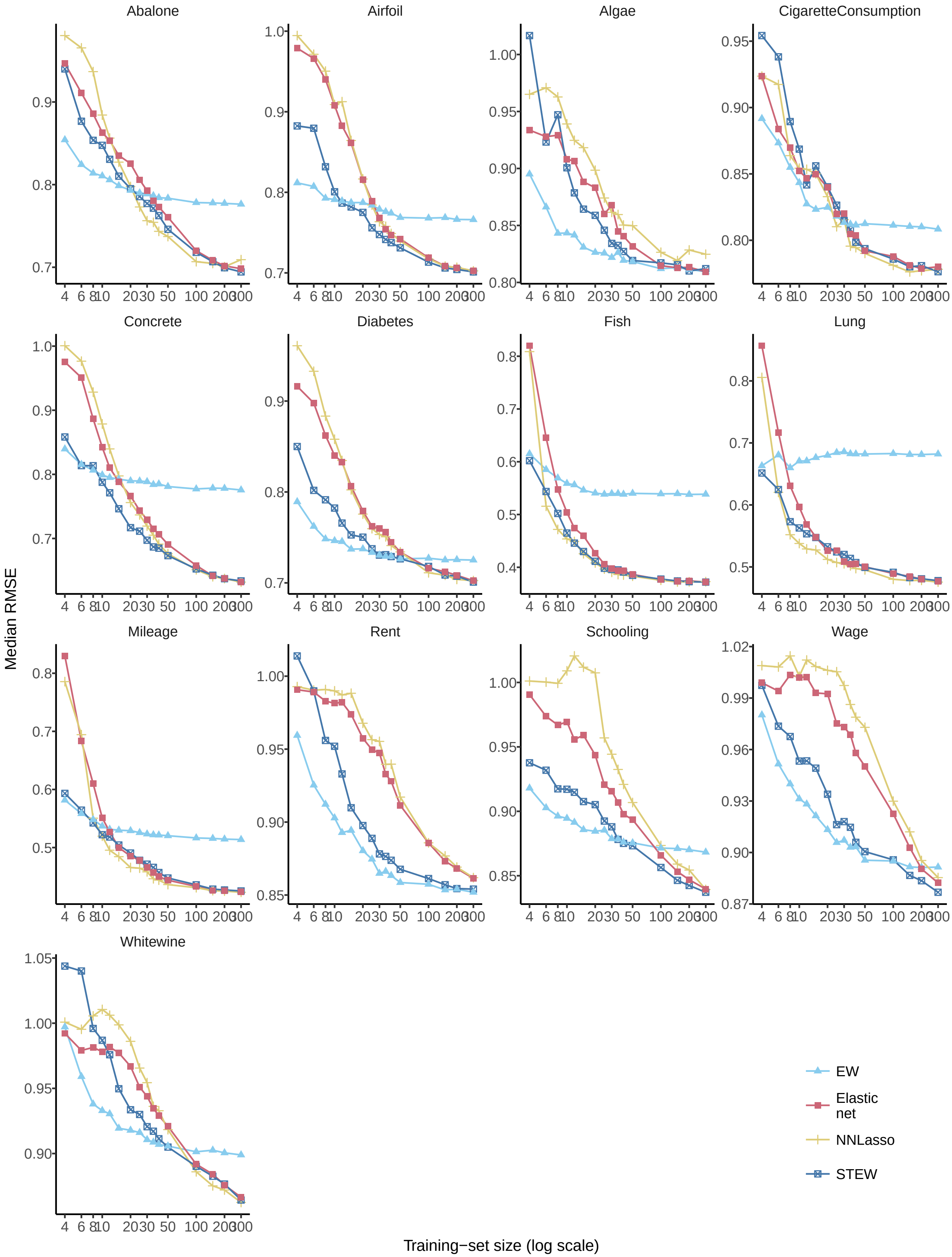

Figure D.4. Median root mean squared error (RMSE) across 200 repetitions for on individual data sets for equal weights (EW), the elastic net (EN), nonnegative Lasso (NNLasso), and shrinkage toward equal weights (STEW). Predictors were directed based on a Lasso estimate on the whole data set.

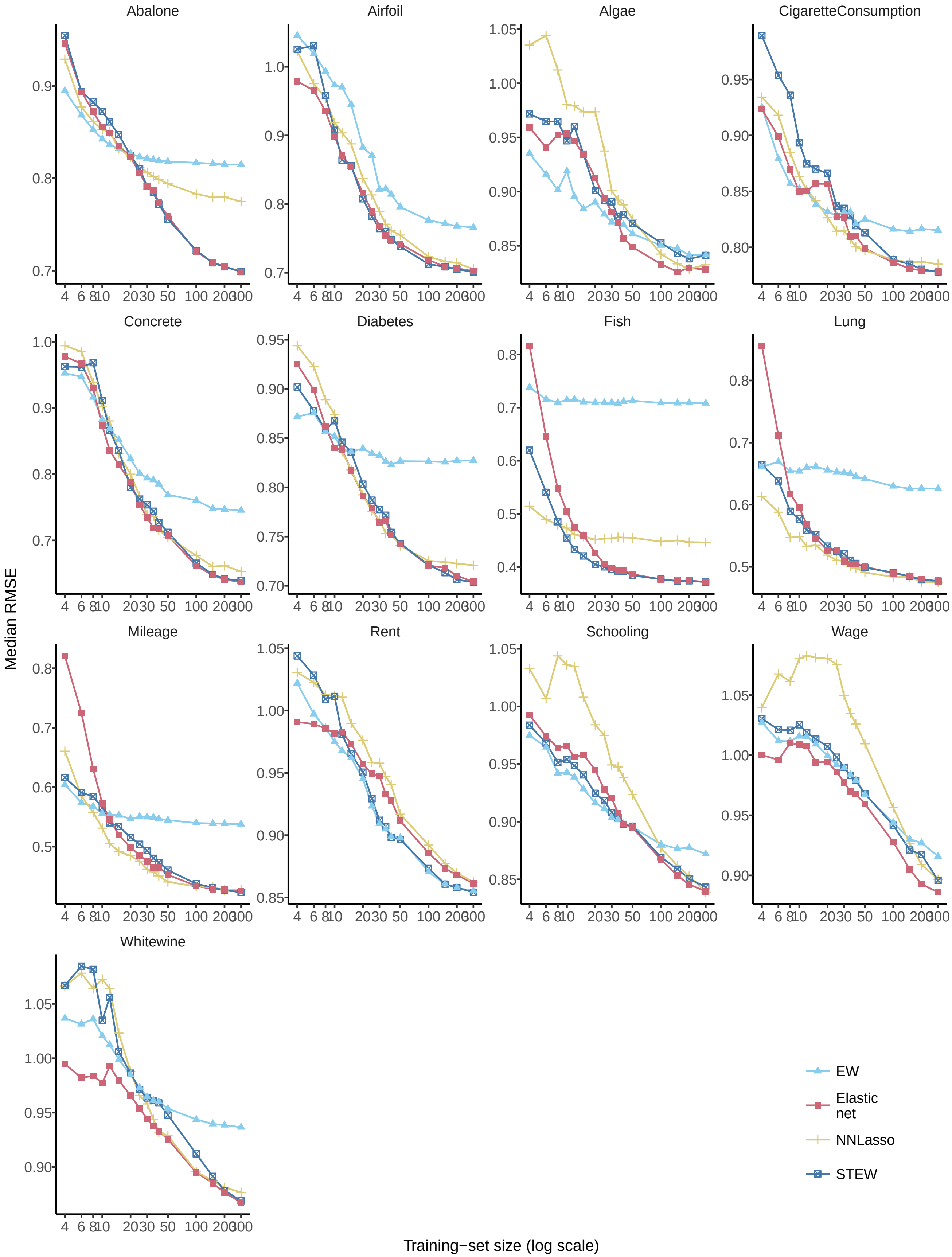

Figure D.5. Median root mean squared error (RMSE) across 200 repetitions for on individual data sets for equal weights (EW), the elastic net (EN), nonnegative Lasso (NNLasso), and shrinkage toward equal weights (STEW). Predictors were directed based on a Lasso estimate on the whole data set.

E. Data sets

This section contains descriptions of all data sets used in the empirical analysis of Section 7 in the main article.

Abalone OBJECTS: 4177 abalones (sea snails). CRITERION: age (measured in visible rings). ATTRIBUTES: sex, length, diameter, height, whole weight, shucked weight, viscera weight, shell weight. SOURCE: This data set comes from a study by Nash et al. (1994). It is available from the UCI Machine Learning Repository (Bache & Lichman, 2013).

Airfoil OBJECTS: 1503 airfoils at various wind tunnel speeds and angles of attack. CRITERION: scaled sound pressure level, in decibels. ATTRIBUTES: frequency, angle of attack, chord length, free-stream velocity, suction side displacement thickness. SOURCE: This data set comes from a study by Brooks et al. (1989). It is available from the UCI Machine Learning Repository (Bache & Lichman, 2013).

Algae OBJECTS: 340 samples from European rivers taken over a period of approximately one year. CRITERION: density of algae type a. ATTRIBUTES: concentrations of eight chemicals, season (fall, winter, spring, summer), river size (small, medium, large), fluid velocity (low, medium, high). SOURCE: The data set is from the 1999 Computational Intelligence and Learning (COIL) competition. It is available from the UCI data repository (Bache & Lichman, 2013), where it is labeled *COIL 1999 competition data*.

Cigarette OBJECTS: 528 states in the USA (in different years). CRITERION: packs per capita. ATTRIBUTES: year, consumer price index, state population, state personal income, average state, federal, and average local excise taxes for fiscal year. SOURCE: The data set was assembled by Professor Jonhatan Gruber, MIT. It has been used in an introductory econometrics textbook (Stock & Watson, 2003). It is available electronically from R package *Ecdat* (Croissant, 2013).

Concrete OBJECTS: 1030 concrete samples. CRITERION: concrete compressive strength. ATTRIBUTES: cement (kg/m^3), blast furnace slag (kg/m^3), fly ash (kg/m^3), water (kg/m^3), superplasticizer (kg/m^3), coarse aggregate (kg/m^3), fine aggregate (kg/m^3), age in days. SOURCE: This data set comes from a study by Yeh (1998). It is available from the UCI Machine Learning Repository (Bache & Lichman, 2013).

Diabetes OBJECTS: 442 diabetes patients. CRITERION: a quantitative measure of disease progression one year after baseline. ATTRIBUTES: age, sex, body mass index, average blood pressure and six blood serum measurements. SOURCE: The data was used in Efron et al. (2004). It is available electronically from R package *lars* (Hastie & Efron, 2013).

Fish OBJECTS: 413 female Arctic charr. CRITERION: number of eggs. ATTRIBUTES: age, weight, mean egg weight. SOURCE: This prediction problem is from a study by Czerlinski et al. (1999). The data were collected by Christian Gillet from the French National Institute for Agricultural Research. The data set used in this study was obtained via personal communication in April 2012.

Lung OBJECTS: 654 children. CRITERION: forced expiratory volume in liters. ATTRIBUTES: age in years, height in inches, gender, exposure to smoking. SOURCE: The data were collected by Tager et al. (1979). The data set is reported in Ekstrom & Sørensen (2010) and is electronically available from associated R package *isdals* (Ekstrom & Sorensen, 2014) where it is labeled *fev*.

Mileage OBJECTS: 398 cars built in 1970–1982. CRITERION: mileage. ATTRIBUTES: number of cylinders, engine displacement, horsepower, vehicle weight, time to accelerate from 0 to 60 mph, model year, origin (American, European, Japanese). SOURCE: The data set was prepared by the Committee on Statistical Graphics of the American Statistical Association for its Second Exposition of Statistical Graphics Technology, held in conjunction with the Annual Meetings in Toronto, August 15–18, 1983. It is electronically available from StatLib (StatLib: Data, software and news from the statistics community), where it is labeled *cars*. The version used in the current work is from the UCI Machine Learning Repository (Bache & Lichman, 2013), named *Auto+MPG*, in which 8 of the original cars were removed because their mileage values were missing.

Rent OBJECTS: 2053 apartments in Munich, Germany. CRITERION: rent per square-meter in euros. ATTRIBUTES: size, number of rooms, year of construction, whether the apartment is located at a good address, whether the apartment is located at the best address, whether the apartment has warm water, whether the apartment has central heating, whether the bathroom has tiles, whether there is special furniture in the bathroom, whether the apartment has an upmarket kitchen. SOURCE: The data set is reported in Fahrmeir et al. (2010) and is electronically available from R package *catdata* (Schauberger & Tutz, 2014).

Schooling OBJECTS: 3010 individuals in the US. CRITERION: log of wage. ATTRIBUTES: lived in smsa 1966, lived in smsa in 1976, grew up near 2-yr college, grew up near 4-yr college, grew up near 4-year public college, grew up near 4-year private college, education in 1976, education in 1966, age in 1976, lived with mom and dad at age 14, single mom at 14, step parent at 14, lived in south 1966, lived in south in 1976, mom-dad education class (1-9), black, enrolled in 1976, the kww score, normed IQ score, married in 1976, library card in home at age 14, experience in 1976. SOURCE: The data set comes from the National Longitudinal Survey of Young Men (NLSYM) and has been used by Card (1993). It is available electronically from R package *Ecdat* (Croissant, 2013).

Wages OBJECTS: 4360 males in the US (from 1980 to 1987). CRITERION: log of wage. ATTRIBUTES: year, years of schooling, years of experience, whether the wage has been set by collective bargaining, ethnicity, whether married, whether health problem, industry (12 levels), occupation (9 levels), residence (rural area, north east, nothern central, south). SOURCE: The data set comes from the National Longitudinal Survey (NLS Youth Sample) and has been used by Vella et al. (1998). It is available electronically from R package *Ecdat* (Croissant, 2013) where it is called *Males*.

White wine OBJECTS: 4898 white wines. CRITERION: quality score (between 0 and 10). ATTRIBUTES: fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, density, pH, sulphates, alcohol. SOURCE: This data set comes from a study by Cortez et al. (2009). It is available from the UCI Machine Learning Repository (Bache & Lichman, 2013).

References

Bache, K. and Lichman, M. UCI machine learning repository, 2013. URL http://archive.ics.uci.edu/ml.

Brooks, T. F., Pope, D. S., and Marcolini, M. A. *Airfoil self-noise and prediction*, volume 1218. National Aeronautics and Space Administration, Office of Management, Scientific and Technical Information Division, 1989.

Card, D. Using Geographic Variation in College Proximity to Estimate the Return to Schooling. Working Paper 4483, National Bureau of Economic Research, 1993.

Cortez, P., Cerdeira, A., Almeida, F., Matos, T., and Reis, J. Modeling wine preferences by data mining from physicochemical properties. *Decision Support Systems*, 47(4):547–553, 2009.

Croissant, Y. *Ecdat: Data sets for econometrics*, 2013. URL http://CRAN.R-project.org/package=Ecdat. R package version 0.2-2.

Czerlinski, J., Gigerenzer, G., and Goldstein, D. G. How good are simple heuristics? In Gigerenzer, G., Todd, P. M., and the ABC Research Group (eds.), *Simple heuristics that make us smart*, pp. 97–118. Oxford University Press, New York, 1999.

Efron, B., Hastie, T., Johnstone, I., and Tibshirani, R. Least angle regression. *The Annals of Statistics*, 32(2):407–499, 2004.

Ekstrom, C. and Sorensen, H. *isdals: Provides datasets for Introduction to Statistical Data Analysis for the Life Sciences*, 2014. URL http://CRAN.R-project.org/package=isdals. R package version 2.0-4.

Ekstrom, C. T. and Sørensen, H. *Introduction to statistical data analysis for the life sciences*. CRC Press, 2010.

Fahrmeir, L., Künstler, R., Pigeot, I., and Tutz, G. *Statistik: Der Weg zur Datenanalyse*. Springer Berlin Heidelberg, 2010. ISBN 9783642019388.

Friedman, J., Hastie, T., Simon, N., and Tibshirani, R. glmnet: Lasso and Elastic-Net Regularized Generalized Linear Models, 2015.

Gabillon, V., Ghavamzadeh, M., and Scherrer, B. Approximate dynamic programming finally performs well in the game of Tetris. In *Advances in neural information processing systems*, pp. 1754–1762, 2013.

Hastie, T. and Efron, B. lars: Least Angle Regression, Lasso and Forward Stagewise, April 2013.

Ma, B. N. M. a. J. nnlasso: Non-Negative Lasso and Elastic Net Penalized Generalized Linear Models, 2016.

Nash, W. J., Sellers, T. L., Talbot, S. R., Cawthorn, A. J., and Ford, W. B. The population biology of abalone (haliotis species) in Tasmania. I. Blacklip Abalone (h. rubra) from the north coast and islands of Bass Strait. *Sea Fisheries Division, Technical Report*, (48), 1994.

Schauberger, G. and Tutz, G. *catdata: Categorical Data*, 2014. URL http://CRAN.R-project.org/package=catdata. R package version 1.2.1.

Scherrer, B., Ghavamzadeh, M., Gabillon, V., Lesner, B., and Geist, M. Approximate modified policy iteration and its application to the game of Tetris. *Journal of Machine Learning Research*, 16:1629–1676, 2015. URL http://www.jmlr.org/papers/volume16/scherrer15a/scherrer15a.pdf.

Şimşek, Ö., Algorta, S., and Kothiyal, A. Why most decisions are easy in tetrisAnd perhaps in other sequential decision problems, as well. In *International Conference on Machine Learning*, pp. 1757–1765, 2016. URL http://proceedings.mlr.press/v48/simsek16.html.

StatLib: Data, software and news from the statistics community, 2013. URL http://lib.stat.cmu.edu/.

Stock, J. H. and Watson, M. W. *Introduction to econometrics*, volume 104. Addison Wesley Boston, 2003.

Tager, I. B., Weiss, S. T., Rosner, B., and Speizer, F. E. Effect of parental cigarette smoking on the pulmonary function of children. *American Journal of Epidemiology*, 110(1):15–26, 1979. ISSN 0002-9262.

Thiery, C. and Scherrer, B. Building controllers for Tetris. *Icga Journal*, 32(1):3–11, 2009.

Train, K. E. *Discrete Choice Methods with Simulation*. Cambridge University Press, Cambridge ; New York, 2 edition edition, September 2009. ISBN 978-0-521-74738-7.

Vella, F., Verbeek, M., and others. Whose wages do unions raise? A dynamic model of unionism and wage rate determination for young men. *Journal of Applied Econometrics*, 13(2):163–183, 1998.

Yeh, I.-C. Modeling of strength of high-performance concrete using artificial neural networks. *Cement and Concrete research*, 28(12): 1797–1808, 1998.