Regularization in Directable Environments with Application to Tetris: Supplementary Material

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

This supplementary material contains mathematical proofs (Section A), implementation details for the algorithms used in the supervised learning experiments (Section B), pseudo-code for the M-learning algorithm (Section C), implementation details for the Tetris experiments (Section D), additional results (Section E), and descriptions of the real-world data sets (Section F).

A. Mathematical Proofs

Theorem 1. *Let $\boldsymbol{y}$ be distributed with mean $\boldsymbol{X\beta}$ and variance $\sigma^2 \boldsymbol{I}_{n\times n}$, 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) $\Delta MSE = p\bar{\boldsymbol{\beta}}^2 - p\sigma^2$,
(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 can be written as follows:

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

The derivative of the bias with respect to γ is then given by the following:

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

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

This document is a supplement to: Lichtenberg, J. M. and Şimşek, Ö. Regularization in Directable Environments with Application to Tetris. *Proceedings of the 36^{th} International Conference on Machine Learning*, PMLR 97:3953-3962, 2019. This document was last updated on June 5, 2019.

Equating this derivative to 0 yields the following:

$$\sum_{i=1}^{p}(2\gamma - 2\beta_i) = 0$$

$$2\gamma p = 2\sum_{i=1}^{p}\beta_i$$

$$\gamma = \frac{\sum_{i=1}^{p}\beta_i}{p} = \bar{\boldsymbol{\beta}}.$$

(2a) From here on, we assume that $\boldsymbol{X}$ is orthonormal, that is, $\boldsymbol{X}^T\boldsymbol{X} = \boldsymbol{I}_{p\times p}$. Recall that γ_{EW} is calculated using simple (univariate) linear regression on the model

$$\hat{\boldsymbol{y}} = \gamma \boldsymbol{X}\mathbb{1}, \tag{6}$$

where γ is the equal-weighting parameter to be estimated and $\mathbb{1}$ is a column-vector of ones of length p. Defining $\boldsymbol{c} := \boldsymbol{X}\mathbb{1}$, the simple linear regression estimate of Equation (6) is given by

$$\gamma_{EW} = \frac{r_{\boldsymbol{yc}}}{s_{\boldsymbol{c}}}, \tag{7}$$

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 $\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$ with $\mathbb{E}[\boldsymbol{\epsilon}] = \boldsymbol{0}$, the expected value of EW's γ_{EW} is given by

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

(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

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

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

The difference in biases, Δbias^2, can be computed directly as follows:

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

(2c) The variance for the **0**-model is clearly 0. For the EW model, we use Equation (7) 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 written as follows:

$$
\begin{aligned}
& tr(\mathbb{1}(\mathbb{1}^T\boldsymbol{X}^T\boldsymbol{X}\mathbb{1})^{-1}\mathbb{1}^T\boldsymbol{X}^T Var(\boldsymbol{y})\boldsymbol{X}\mathbb{1}(\mathbb{1}^T\boldsymbol{X}^T\boldsymbol{X}\mathbb{1}^T)^{-1}\mathbb{1}^T) \\
= & \sigma^2 tr(\mathbb{1}(\mathbb{1}^T\boldsymbol{X}^T\boldsymbol{X}\mathbb{1})^{-1}\mathbb{1}^T\boldsymbol{X}^T\boldsymbol{I}\boldsymbol{X}\mathbb{1}(\mathbb{1}^T\boldsymbol{X}^T\boldsymbol{X}\mathbb{1}^T)^{-1}\mathbb{1}^T) \\
= & \sigma^2 p.
\end{aligned}
$$

The result follows directly by adding these variances to the squared biases from 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 $\boldsymbol{\beta}_{||} = (|\beta_1|, \dots, |\beta_p|)$ denote the weights on a positively directed data set. Then the following holds:

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

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

B. Supervised Learning Experiments

The regularization strength λ of all regularized linear models was chosen using k-fold cross validation. 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 STEW with $q = 2$ for the experiments. Solutions were computed using the closed-form solution described 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 $\epsilon > 0$).

Nonnegative lasso (NNLasso). We used the R package *nnlasso* (Mandal & 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 the nnlasso package.

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. We tested $\alpha \in \{0, 0.25, 0.5, 0.75, 1\}$ and used the built-in search path of glmnet for λ. For ridge regression and the Lasso, only λ has to be tuned.

Learning curves on real-world data sets (Figure 4) were computed as follows. We pre-processed each real-world data by standardizing responses and predictors to have zero mean and unit variance. Missing predictor values were mean imputed and observations with missing response values were removed from the data set. 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. Results were averaged across 200 repetitions, each corresponding to a different train/test-split of the data.

C. Pseudo-Code for M-learning

Pseudo-code for M-learning can be found in Algorithm 1. The rollout procedure that is used within M-learning is given in Algorithm 2.

Algorithm 1 M-learning

Notation: $p \in \mathbb{N}$ is the number of features, $\mathcal{A}(s)$ is the set of actions available in state s.
Output: $\boldsymbol{\beta} \in \mathbb{R}^p$, a vector of action-utility weights, initialized randomly.
Input:
$U(s,a) = f(\boldsymbol{\beta}, \boldsymbol{\phi}(s,a))$, where *// action-utility 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 (e.g., Table 2)*
$M \in \mathbb{N}$ *// 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 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**
  for all $a \in \mathcal{A}(s)$ **do**
    $\hat{U}(s,a) \leftarrow$ Rollout$(s, a, M, T, \gamma, \pi_r(s, \boldsymbol{\beta}))$
  end for
  $\tilde{a} \leftarrow \underset{a \in \mathcal{A}(s)}{\text{argmax}}\ \hat{U}(s,a)$
  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 *// reset episode*
  end if
  Construct batch $\mathcal{D}_k$ using $n(k)$ most recent choice sets from $\mathcal{D}$
  Update $\boldsymbol{\beta}$ using multinomial logistic regression on batch $\mathcal{D}_k$
end for

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

Notation: $\mathcal{G}(s,a) : \mathcal{S} \times \mathcal{A}(s) \to \mathcal{S} \times \mathbb{R}$ is a 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*
$\pi_r(s, \boldsymbol{\beta}) : \mathcal{S} \times \mathbb{R}^p \to \mathbb{R}$ *// rollout policy 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 $\hat{U} \leftarrow \frac{1}{M} \sum_{j=1}^{M} \hat{U}_j$

D. Tetris Experiments

Our implementation of M-learning in Tetris is available at `https://github.com/janmaltel/stew-tetris`.

Dominance filters. We used simple and cumulative dominance filters as described by Şimşek et al. (2016). The use of these filters considerably reduced the number of actions to be evaluated in a given state, and thus the computational cost of the rollout procedure. The filters were used during learning but not during testing. Both filters require information about feature directions. We set feature directions to the signs of the weights of the BCTS policy (Thiery & Scherrer, 2009). Cumulative dominance filters also require an ordering of the features. We ordered features by decreasing magnitude of BCTS weights. Feature directions and order are shown in Table 1

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

Table 1. Feature directions and order of the BCTS policy. The features are ordered from top to bottom by decreasing magnitude of the corresponding weight.

Choice data sets. Discrete choice models are trained using a data set of observed choice decisions made by one or multiple agents (e.g., Train, 2009). 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. Each alternative is described by a set of features $\phi_1, \ldots, \phi_p$. Table 2 shows an example choice data set with 4 observations. Unlike standard classification or regression data sets, one observation here corresponds to multiple rows, each describing one alternative in the choice set. Note that the number of alternatives varies among different choice sets. In Tetris, the different alternatives are the different placements of the falling tetrimino on the current board. The *choice* variable indicates for each placement whether it was chosen or not.

choice set id	choice	rows with holes (ϕ_1)	...	hole depth (ϕ_8)
0	0	$\phi_1(s_0, a_1)$	...	$\phi_8(s_0, a_1)$
0	0	$\phi_1(s_0, a_2)$	...	...
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	$\phi_1(s_3, a_2)$	...	...
3	0	$\phi_1(s_3, a_3)$	...	...
3	0	$\phi_1(s_3, a_4)$	...	...
3	1	$\phi_1(s_3, a_5)$	...	$\phi_8(s_3, a_5)$

Table 2. Sample choice data set with 4 observations. Choice sets are of varying size. The *choice* variable denotes the chosen alternative for each choice set. Each alternative in a choice set is characterized by the 8 feature values $\phi_1, \ldots, \phi_8$.

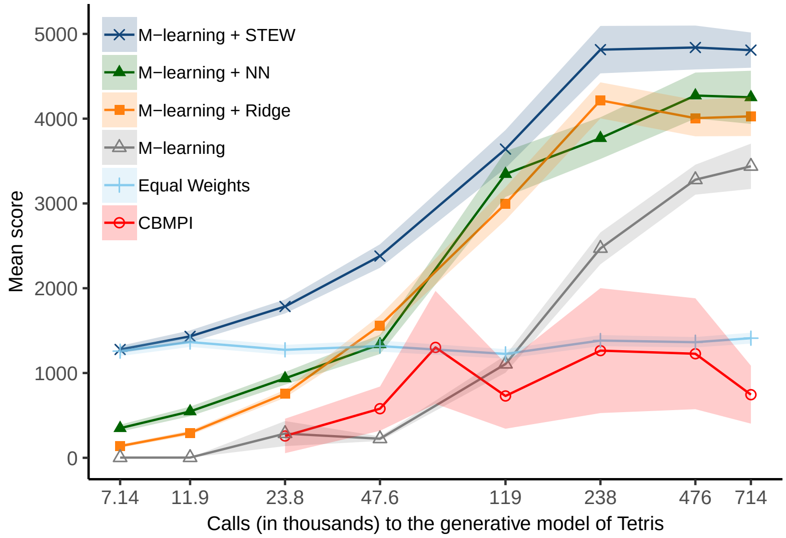

Figure 6. Quality of the policy learned as a function of the number of calls to the generative model (in thousands). Quality of the policy is measured by the mean score obtained in 10 Tetris games. The learning curves show mean score across 100 replications of the algorithm. M-learning algorithms used a per-iteration budget of 2,380 calls to the generative model and were tested after 3, 5, 10, 20, 50, 100, 200, and 300 iterations. CBMPI used a per-iteration budget of 23,800 calls to the generative model and was tested after 1, 2, 3, 5, 10, 20, and 30 iterations. See Section E for further details.

E. Additional Results

CBMPI in Tetris. We compared the performance of M-learning to *classification-based modified policy iteration* (CBMPI, Scherrer et al., 2015), the best-performing reinforcement learning algorithm for Tetris reported in the literature. On the 10×10 board, the results reported by Scherrer et al. used a per-iteration budget of 8,000,000 calls to the generative model of Tetris. In comparison, the per-iteration budget we used for M-learning in Figure 5 was 2,380. In order to compare the algorithms meaningfully, we experimented with CBMPI using budgets in the same range as those used for M-learning. CBMPI's per-iteration budget is a function of the number of states from which rollouts are started (called the rollout set size N), the number of rollouts from each start state, M, and the rollout length, m. The best-performing version reported in Scherrer et al. (2015) used $N = 42,000$, $M = 1$, and $m = 5$, resulting in a budget of $B = (m+1)MN|\mathcal{A}| = 8,000,000$. In order to obtain smaller per-iteration budgets, we decreased the rollout set size N while keeping all other parameters fixed. CBMPI using a budget of 2,380 (obtained by setting $N = 13$) resulted in highly unstable value function estimates. The algorithm could not learn a useful policy. We then gradually increased CBMPI's budget until value function estimates stabilized. Figure 6 shows results for CBMPI using a per-iteration budget of $23,800$ (obtained by setting $N = 124$). CBMPI achieved lower mean score than regularized versions of M-learning across the entire learning curve.

STEW in environments with high variance of weights. Figure 7 shows learning curves for STEW in an environment in which the true weights are highly non-equal. We used the experimental setup of Section 5.1 with prior weight distribution $\boldsymbol{\beta} \sim \mathcal{U}(0, 50)$.

Ordering of predictors in total variation models. The regularization behavior of total variation (TV) models depends on the order of the predictors. We compared the regularization paths of TV models with different orders of predictors to those of the l_1 and l_2 versions of STEW on the *Rent* data set. The first two panels from the left of the top row in Figure 8 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.

Individual learning curves on real-world data sets. We show learning curves on the individual data sets that were used to compute the average learning curves shown in figures 5c and 5f in the main article. Figure 9 corresponds to Figure 5c: feature directions were estimated using an initial Lasso estimate on the entire data set. Figure 10 corresponds to Figure 5f: feature directions were estimated on the training set using Pearson correlation coefficients.

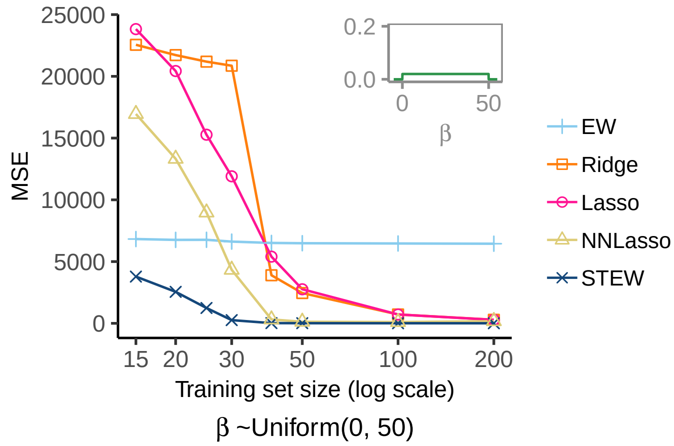

Figure 7. Mean squared error (MSE) across 400 repetitions for equal weights (EW), ridge regression, the Lasso, the non-negative Lasso (NNLasso) and shrinkage toward equal weights (STEW) as a function of training set size in an environment where the true weights are sampled from $\boldsymbol{\beta} \sim \mathcal{U}(0, 50)$.

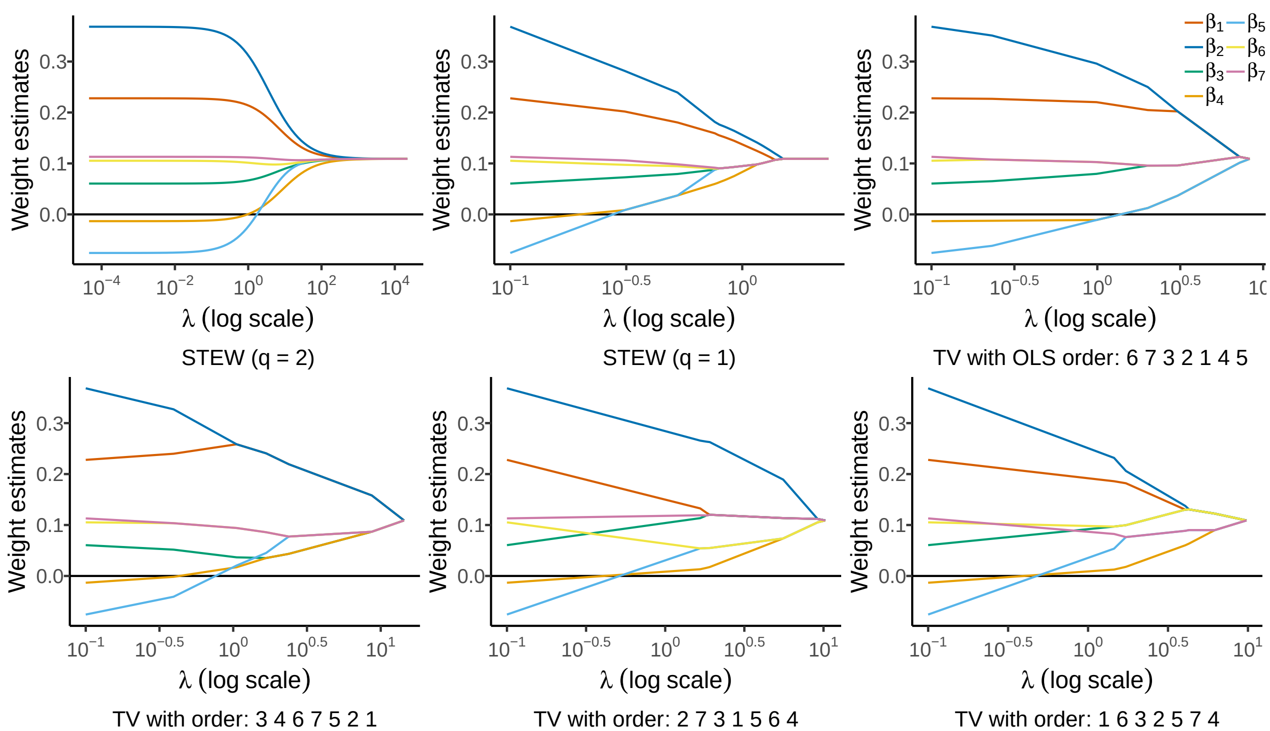

Figure 8. Weight estimates on the *Rent* data set as a function of regularization strength λ for STEW with l_1 and l_2 penalties, and total variation (TV) with various orders of the predictors.

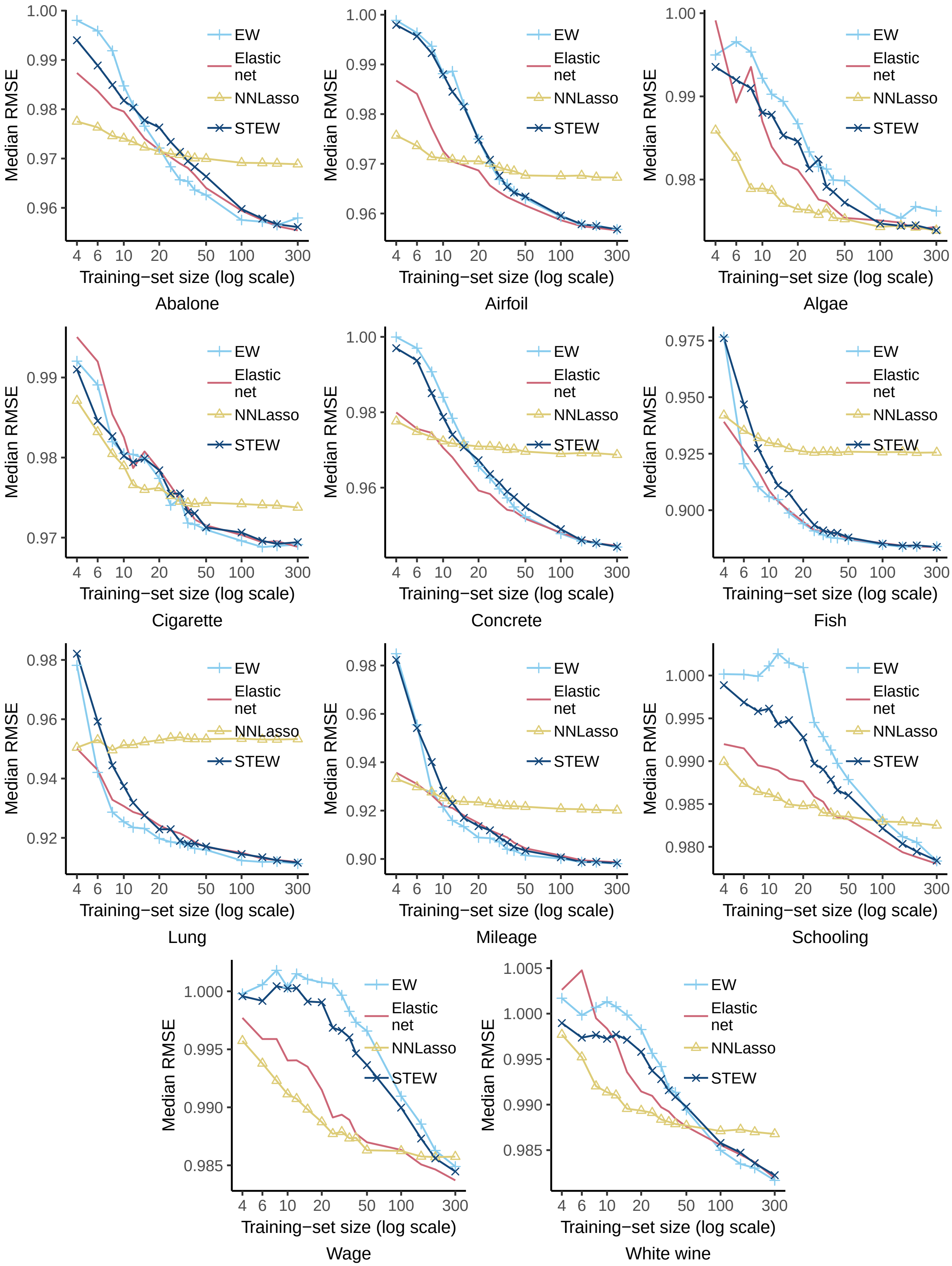

Figure 9. Median root mean squared error (RMSE) across 200 repetitions on individual data sets. Predictors were directed based on a Lasso estimate on the entire data set.

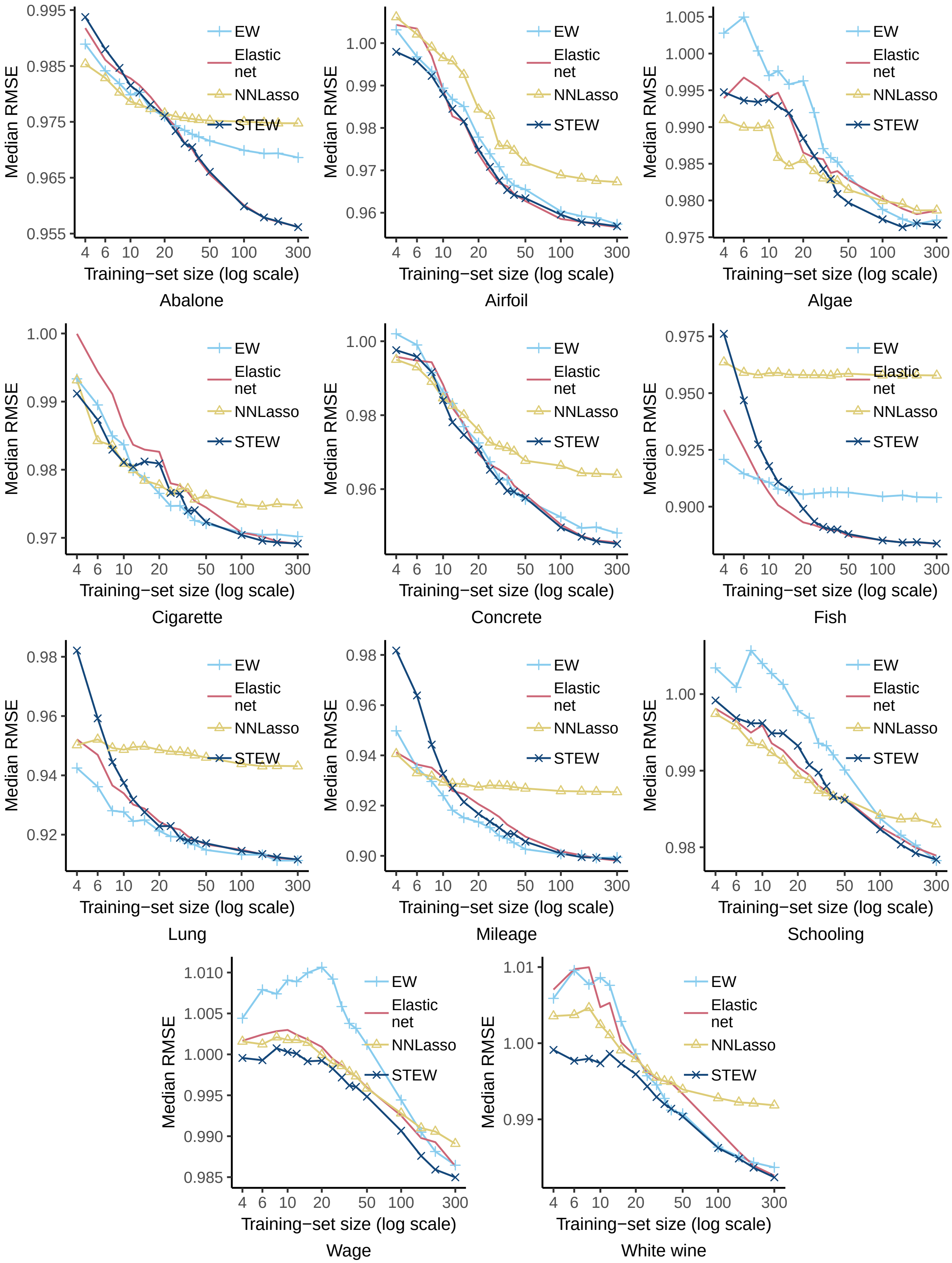

Figure 10. Median root mean squared error (RMSE) across 200 repetitions on individual data sets. Predictors were directed based on the training set using Pearson correlation coefficients between features and response.

F. Real-World Data Sets

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 (Dua & Graff, 2019).

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 (Dua & Graff, 2019).

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 (Dua & Graff, 2019), 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 (Dua & Graff, 2019).

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 (Meyer & Vlachos, 1989), where it is labeled *cars*. The version used in the current work is from the UCI Machine Learning Repository (Dua & Graff, 2019), 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 (Dua & Graff, 2019).

References

Brooks, T. F., Pope, D. S., and Marcolini, M. A. *Airfoil self-noise and prediction*. 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.

Dua, D. and Graff, C. UCI machine learning repository, 2019. URL `http://archive.ics.uci.edu/ml`.

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.

Friedman, J., Hastie, T., Simon, N., and Tibshirani, R. glmnet: Lasso and elastic-net regularized generalized linear models, 2015. URL `https://cran.r-project.org/web/packages/glmnet/index.html`.

Hastie, T. and Efron, B. lars: Least angle regression, lasso and forward stagewise, 2013. URL `https://cran.r-project.org/web/packages/lars/index.html`.

Mandal, B. N. and Ma, J. nnlasso: Non-negative lasso and elastic net penalized generalized linear models, 2016. URL `https://cran.r-project.org/web/packages/nnlasso/index.html`.

Meyer, M. and Vlachos, P. StatLib. `http://lib.stat.cmu.edu/`, 1989.

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.

Şimşek, Ö., Algorta, S., and Kothiyal, A. Why most decisions are easy in Tetris—and perhaps in other sequential decision problems, as well. In *Proceedings of the 33rd International Conference on Machine Learning*, pp. 1757–1765, 2016.

Stock, J. H. and Watson, M. W. *Introduction to Econometrics*. 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.

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, 2 edition, 2009.

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.