Regularization in Directable Environments with Application to Tetris

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

Abstract

Learning from small data sets is difficult in the absence of specific domain knowledge. We present a regularized linear model called STEW that benefits from a generic and prevalent form of prior knowledge: feature directions. STEW shrinks weights toward each other, converging to an equal-weights solution in the limit of infinite regularization. We provide theoretical results on the equal-weights solution that explains how STEW can productively trade-off bias and variance. Across a wide range of learning problems, including Tetris, STEW outperformed existing linear models, including ridge regression, the Lasso, and the non-negative Lasso, when feature directions were known. The model proved to be robust to unreliable (or absent) feature directions, still outperforming alternative models under diverse conditions. Our results in Tetris were obtained by using a novel approach to learning in sequential decision environments based on multinomial logistic regression.

1. Introduction

Domain knowledge can be very useful in machine learning, especially when training data are costly or difficult to obtain. We explore a particular type of domain knowledge in supervised learning: feature directions. The direction of a feature indicates whether the feature is associated positively or negatively with the response variable. In many applications, feature directions are known or can be estimated with ease. For example, in learning how to play the game of Tetris, most players would agree that the *number of holes* is associated negatively with the game score.

How can such domain knowledge be fruitfully incorporated into the learning process? We present an approach based on regularization. Specifically, we propose a model that introduces a bias toward giving all features equal weight. We call this model *Shrinkage Toward Equal Weights* (STEW).

The ordinary least squares (OLS) solution to linear regression is unbiased but can have high variance when the training set is small. Regularization reduces variance by introducing assumptions about the data and anchoring the weights in a way that reflects these assumptions. For example, Lasso-type models (Tibshirani, 1996; Bühlmann & van de Geer, 2011) assume that the features are irrelevant and shrink weights toward zero with increasing regularization strength.

Rather than shrinking the weights toward zero, STEW shrinks them toward each other, converging to equal weights in the limit of infinite regularization. We study properties of the equal-weights model as a source of intuition on when and why STEW can perform well. We provide theoretical evidence that EW has relatively low bias and that this bias is further reduced when directions of features are known.

Our empirical analysis shows that these properties translate from the equal-weights model to STEW. When information on directions is available, STEW routinely outperforms existing models including the non-negative Lasso, which also incorporates feature directions. Unlike methods that are based on non-negativity constraints, we found STEW to be robust when the underlying assumption of known feature directions is violated, that is, when the information on directions is unreliable or absent. Finally, we found STEW to be remarkably useful when learning to play Tetris.

Our results in Tetris were obtained using a novel approach to learning in sequential decision environments. This approach, called *M-learning*, is built around multinomial logistic regression. Here, we describe M-learning and present results when it is used for learning to play Tetris, with or without regularization. A subsequent article will more fully explore the properties and behavior of M-learning.

2. Background

We consider the linear regression problem where the objective is to predict a response $y \in \mathbb{R}$ by

$$\hat{y} = \beta_0 + \sum_{j=1}^{p} \beta_j x_j, \tag{1}$$

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

Proceedings of the 36^{th} International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019. Copyright 2019 by the author(s).

where $x_1, \ldots, x_p$ are feature values and $\beta_0, \ldots, \beta_p$ are feature weights. To estimate feature weights, a training set of n observations, $(y_i, x_{1i}, \ldots, x_{pi}), i = 1, \ldots, n$, is available. Following the standard in the regularization literature (e.g., Friedman et al., 2009), we assume that features and responses are standardized so that $\frac{1}{n}\sum_{i=1}^{n} y_i = 0$, $\frac{1}{n}\sum_{i=1}^{n} x_{ij} = 0$, and $\frac{1}{n}\sum_{i=1}^{n} x_{ij}^2 = 1$, for $j = 1, \ldots, p$. It follows that β_0 is zero and thus can be omitted. We use matrix notation, with $\boldsymbol{y} \in \mathbb{R}^n$ denoting the response vector, $\boldsymbol{X} \in \mathbb{R}^{n \times p}$ the feature matrix, and $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)$ the weight vector. The Ordinary least squares (OLS) estimate is the set of weights that minimizes the residual sum of squares $\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|_2^2$ on the training set.

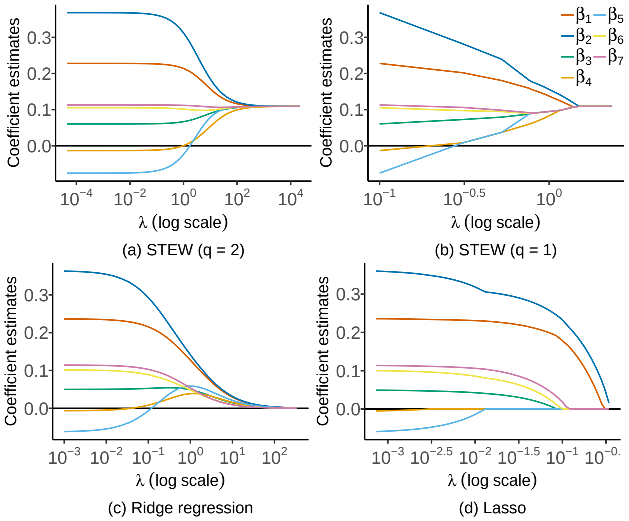
Regularized linear models. Most regularized linear models minimize a penalized residual sum of squares of the form $\mathcal{L}(\boldsymbol{\beta}, \lambda) = \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|_2^2 + \lambda P(\boldsymbol{\beta})$, where P is the penalty function and $\lambda \geq 0$ is the strength of the penalty. Well known penalty functions use the l_q-norm of the weight vector, $\|\boldsymbol{\beta}\|_q$. For example, ridge regression (Hoerl & Kennard, 1970) uses the l_2 penalty, the Lasso (Tibshirani, 1996) uses the l_1 penalty, and the elastic net (Zou & Hastie, 2005) uses a convex combination of the l_1 and the l_2 penalties. These models shrink all weights toward zero as $\lambda \to \infty$. We refer to them as *models that shrink toward zero*.

Equal-weighting models. An equal-weighting model is the linear model of Equation (1) where all feature weights have the same value (γ):

$$\hat{\boldsymbol{y}} = \gamma \sum_{j=1}^{p} \boldsymbol{x}_j. \tag{2}$$

We define *Equal Weights* (EW) as the least-squares estimator of γ and denote the corresponding estimate with γ_{EW}. EW has a long history of use in the social sciences. It appeared in a seminal paper by Dawes & Corrigan (1974) that showed that *even* so-called improper models (such as EW) could outperform human expert judgements. The article also demonstrated that EW can compete well with OLS on real-world data sets, stimulating further work on equal-weighting models in the 1970s, continuing to this day (Einhorn & Hogarth, 1975; Wainer, 1976; Davis-Stober et al., 2010; Graefe, 2015; Lichtenberg & Şimşek, 2017). Equal-weighting models have also been found to be useful in other domains such as paired comparison (Gigerenzer et al., 1999) and portfolio optimization (DeMiguel et al., 2009).

Directability of features. The *direction* of a feature is defined as the sign of the corresponding weight, which can be positive or negative. An environment is said to be *directable* if the directions of the features are known. In a directable environment, features can be "directed" (for example, by recoding any feature with a negative weight by multiplying its values by -1) so that the weights are all positive. Many problems are naturally constrained to have only positive weights (for example mixing problems, see Slawski & Hein, 2013, and references therein). In other problems, features can be directed intuitively by the user, as supported by experimental evidence (Dana & Thomas, 2006; Katsikopoulos et al., 2010). Even without any prior knowledge, directions can be estimated from relatively few training data (Şimşek & Buckmann, 2015).



(a) STEW (q = 2)

(b) STEW (q = 1)

(c) Ridge regression

(d) Lasso

Figure 1. Weight estimates, as a function of the regularization strength λ, for STEW with $q = 1$ and $q = 2$, ridge regression, and the Lasso on the *Rent* data set with seven standardized features.

Notice that EW is a sensible model only if the features are directed so that the true weights have identical signs. The rationale for the use of EW in psychology and decision making is the assumption that people are good in choosing relevant features and know—through intuition or past experience—how to direct them (Einhorn & Hogarth, 1975). The model we propose, STEW, can also use this knowledge fruitfully.

3. Shrinkage Toward Equal Weights

Motivated by the surprisingly high performance of equal-weighting models in the literature—not only in regression but also in classification, paired comparison, and portfolio optimization—we propose to use the equal-weights model as a prior in regularization. In other words, we make the initial assumption that *features have equal impact on the response variable*. This assumption leads to the regularization penalty $\lambda \sum_{i<j} ||\beta_i| - |\beta_j||^q$, for $q > 0$, which penalizes differences in the magnitude of the weights. It leaves the choice of feature directions free. However, the differences of absolute values within the penalty function make the loss function difficult to optimize. We therefore use a penalty function that assumes a directable environment.

We define Shrinkage Toward Equal Weights (STEW) as a regularized linear model which penalizes the l_q-norm of *all* pairwise differences between weights. Specifically, STEW minimizes the loss function below:

$$\mathcal{L}^{STEW}(\boldsymbol{\beta}, \lambda, q) = \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|_2^2 + \lambda \sum_{i<j} |\beta_i - \beta_j|^q, \quad (3)$$

where $q > 0$ and $\lambda \geq 0$ determine the regularization behavior. When $\lambda = 0$, STEW is equivalent to OLS, just like models that shrink weights toward zero. But with increasing regularization strength λ, STEW shrinks weights toward each other rather than toward zero. In the limit as $\lambda \to \infty$, STEW becomes equivalent to EW, with all weights converging to γ_{EW}. Figure 1 illustrates this difference in regularization behavior for STEW with $q = 1$ and $q = 2$ compared to Lasso and ridge regression on the *Rent* data set (described in Section 5.2).

In matrix notation, Equation (3) can be written as follows:

$$\mathcal{L}^{STEW}(\boldsymbol{\beta}, \lambda, q) = \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|_2^2 + \lambda \|\boldsymbol{D}\boldsymbol{\beta}\|_q^q, \quad (4)$$

where $\boldsymbol{D}$ is a pairwise difference matrix with $p(p-1)/2$ rows and p columns. The rows of $\mathcal{D}$ are the unique permutations of the vector $(1, -1, 0, \ldots, 0)$ with p entries such that the entry '1' precedes the entry '-1'. With $q = 1$, Equation (4) has no closed-form solution but can be solved numerically, for example, by using the generalized Lasso framework (Tibshirani & Taylor, 2011). With $q = 2$, minimizing Equation (4) is a Tikhonov regularization problem (Tikhonov et al., 2013) and admits the closed form solution below:

$$\underset{\boldsymbol{\beta}}{\operatorname{argmin}}\, \mathcal{L}^{STEW}(\boldsymbol{\beta}, \lambda, 2) = (\boldsymbol{X}^T\boldsymbol{X} + \lambda \boldsymbol{D}^T\boldsymbol{D})^{-1}\boldsymbol{X}^T\boldsymbol{y}.$$

We use $q = 2$ in the remainder of this article due to the simplicity and computational advantages of its closed-form solution.

Related work. Similar to STEW, non-negative least squares (NNLS) and non-negative Lasso (NNLasso) benefit from positive (or directable) features. NNLS minimizes the residual sum of squares while constraining weights to be positive. Although positivity constraints alone were found to have regularizing properties (Meinshausen, 2013; Slawski & Hein, 2013), NNLS has been combined with l_1-penalty as well (Efron et al., 2004; Slawski & Hein, 2013; Wu et al., 2014). The resulting model is NNLasso and minimizes the loss function $\mathcal{L}^{NNLasso}(\boldsymbol{\beta}, \lambda) = \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|_2^2 + \lambda \|\boldsymbol{\beta}\|_1$, such that $\beta_i \geq 0, \forall i = 1, \ldots p$.

Also related are total variation (TV) models (for example, Chambolle, 2004), which are motivated by environments in which features are spatially or temporarily correlated, such as the pixels of an image or the time points of a time series. TV models estimate smooth functions by penalizing the difference between the weights of adjacent features. In a one-dimensional setting, TV models minimize the loss function $\mathcal{L}^{TV}(\boldsymbol{\beta}, \lambda) = \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|_2^2 + \lambda \sum_{j=2}^{p} |\beta_j - \beta_{j-1}|$. TV models have been developed and used with data sets where a natural adjacency relationship exists. TV models are also used in biostatistics when the data allows a meaningful order of features, for example, in protein mass spectroscopy. The fused Lasso (Tibshirani et al., 2005) considers a Lasso-type l_1-penalty in addition to a TV-type smoothness penalty.

There is a surface similarity between STEW and TV models: both models penalize differences between weights. But the exact form of the penalty differs between the models. TV models penalize the differences between adjacent weights while STEW penalizes all pairwise differences between the weights. This difference is a direct consequence of the different motivations behind the two models and it results in meaningful differences in regularization behavior. Specifically, TV models shrink weights together in patches or clusters that are defined by the adjacency relationships (sample regularization paths are shown in the Supplementary Material), which is quite different than the behavior of STEW (Figure 1). It should be noted that imposing an arbitrary adjacency relationship onto a dataset (to be used with a TV model) is not well justified: different adjacency relationships result in arbitrarily different solutions along the regularization path. The Supplementary Material presents a comparison of regularization paths taken by TV models with different orderings of features of the *Rent* data set.

4. Bias-Variance Analysis of Equal-Weighting Models

Regularized linear models search for a happy medium between OLS, which has low bias but high variance, and a model that has high bias but low variance. For both STEW and models that shrink toward zero, the low-variance model is an equal-weighting model: STEW regularizes toward the EW model ($\gamma = \gamma_{EW}$) while models that shrink toward zero regularize toward what we call the ***0**-model* ($\gamma = 0$).

Theorem 1 shows results on the bias-variance decomposition of mean squared error for equal-weighting models, providing intuition on when and why EW—and therefore STEW—can perform well.

Mean squared error $MSE(\hat{\boldsymbol{\beta}}) = \mathbb{E}||\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}||^2$ can be decomposed into two components, squared bias and the trace of the variance-covariance matrix, $\boldsymbol{\Sigma}_{\hat{\boldsymbol{\beta}}}$, as follows: $MSE(\hat{\boldsymbol{\beta}}) = bias^2 + var = ||\mathbb{E}[\hat{\boldsymbol{\beta}}] - \boldsymbol{\beta}||^2 + tr(\boldsymbol{\Sigma}_{\hat{\boldsymbol{\beta}}})$. Let $\hat{\boldsymbol{\beta}}_{EW}$ and $\hat{\boldsymbol{\beta}}_{\mathbf{0}}$ denote the weight estimates of the EW model and the **0**-model, respectively. Their differences in squared bias and mean squared error are defined as $\Delta\text{bias}^2 := \text{bias}^2(\hat{\boldsymbol{\beta}}_{\mathbf{0}}) - \text{bias}^2(\hat{\boldsymbol{\beta}}_{EW})$ and $\Delta\text{MSE} := \text{MSE}(\hat{\boldsymbol{\beta}}_{\mathbf{0}}) - \text{MSE}(\hat{\boldsymbol{\beta}}_{EW})$, respectively.

Theorem 1. *Let $\boldsymbol{y} \sim (\boldsymbol{X\beta}, \sigma^2 \boldsymbol{I}_{n\times n})$, 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.

The proofs are provided in the Supplementary Material. Result 1 shows that minimum bias is achieved by setting the equal-weighting constant (γ) to the mean of the true weights ($\bar{\boldsymbol{\beta}}^2$). Result 2a shows that, in the special case of an orthonormal data matrix, γ_{EW} is an unbiased estimate of the mean of the true weights and thus attains minimum bias. Result 2b shows that the difference in bias between the **0**-model and EW increases with the square of the mean of true weights, in other words, with increasing distance of the true mean of weights from zero. It follows that EW has lower mean squared error than the **0**-model if the lower bias of EW is not canceled out by the additional variance that results from estimating γ_{EW}. In the case of an orthonormal data matrix, this variance equals the noise parameter σ^2 (Result 2c).

Result 2d examines the impact of knowing feature directions. When feature directions are known, features can be recoded to have the same direction, for example, by multiplying the values of all negative features by -1. This simple operation does not change the biases of the **0**-model, OLS, ridge regression, and the Lasso. It does, however, reduce the bias of the EW model.

5. Empirical Analysis

We present simulation experiments that examine how much the results of Theorem 1 transfer from EW to STEW in a number of challenging environments.

5.1. Simulated Environments

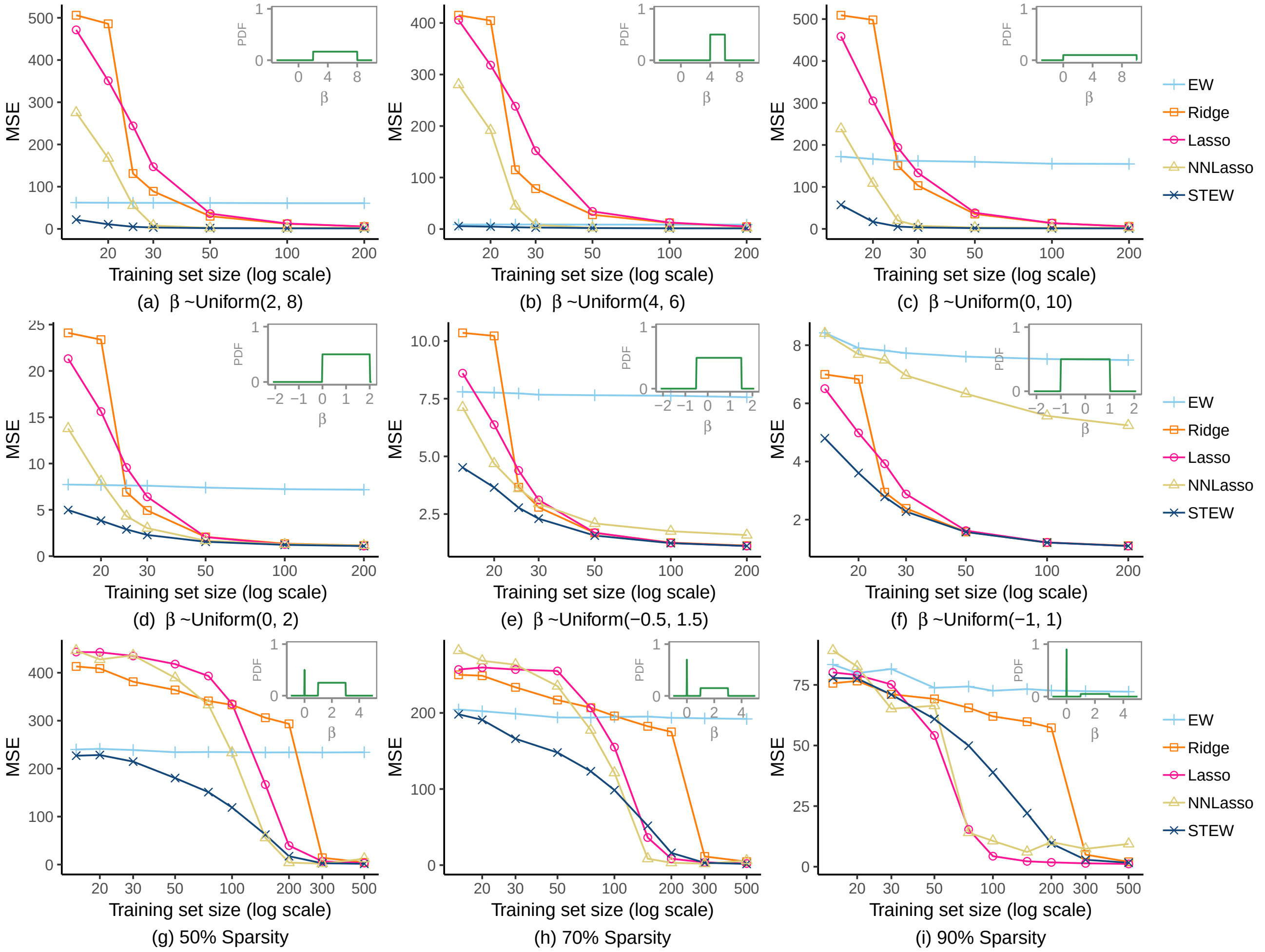
We sampled data from the true model $Y = X_1\beta_1 + \cdots + X_{20}\beta_{20} + \epsilon$, where $X_i \overset{i.i.d}{\sim} \mathcal{N}(0,1)$ and $\epsilon \overset{i.i.d}{\sim} \mathcal{N}(0,1)$. The defining property of each environment was the prior distribution from which the weights $\boldsymbol{\beta} = (\beta_1, \dots, \beta_{20})$ were sampled. For each environment, 400 data sets were sampled to compare the predictive accuracy of STEW, the Lasso, ridge regression, NNLasso, and EW. We also tested NNLS, which is NNLasso without the lasso regularization but omit it from the plots because its performance always lagged behind NNLasso. The regularization strength λ for each model was tuned using cross-validation. Full implementation details are provided in the Supplementary Material. In all environments, when training sets were large enough, STEW, ridge regression, and the Lasso performed equally well, with MSE converging to irreducible error. Our discussion will thus focus on small-to-medium sample sizes.

Directable environments. We first analyze the ideal use case for STEW: when weights are known to be positive (equivalently, if features are directable). Recall that, in such an environment, STEW, EW, and NNLasso are able to directly use the knowledge that the weights are positive. On the other hand, ridge regression and the Lasso cannot incorporate this information directly; they learn it from the data.

Figure 2a shows the predictive performance of various models when $\boldsymbol{\beta} \sim \mathcal{U}(2,8)$. STEW performed best overall. EW performed relatively well when training sets were small—although it was outperformed (as expected) by all adaptively regularizing models for large sample sizes. STEW was able to combine the strengths of different models. For small sample sizes, STEW regularized toward the EW solution and outperformed all competing models, including EW. For large sample sizes, STEW performed as well as the other adaptively regularizing linear models. Notice that, for small sample sizes, NNLasso was far behind STEW, even though it also directly used the knowledge that the weights are positive.

One possible explanation for the superior performance of STEW compared to NNLasso is that the prior distribution of the weights has relatively low variance. When variance is low, weights are relatively close to each other, creating an environment that supports EW, and therefore STEW. We therefore examine two additional environments, $\boldsymbol{\beta} \sim \mathcal{U}(4,6)$ and $\boldsymbol{\beta} \sim \mathcal{U}(0,10)$, that are identical to $\boldsymbol{\beta} \sim \mathcal{U}(2,8)$ in the shape of the distribution and its mean but differ in their variance. The results are shown in panels b and c of Figure 2. STEW remained the best performing model in all three environments but its relative advantage compared to the next best model, NNLasso, decreased with increasing variance. In additional experiments, we increased the variance to unrealistically high levels, up to $\boldsymbol{\beta} \sim \mathcal{U}(0,50)$. The results, provided in the Supplementary Material, remain qualitatively similar.

Effect of directability. Weight priors used in panels d–f of Figure 2 follow a uniform distribution as before. They all have a support of length 2 but differ in the region of support. From panel d to f, the environments decrease in the proportion of weights that are positive. In the $\boldsymbol{\beta} \sim \mathcal{U}(0,2)$ environment, all weights are positive. It therefore represents a fully-directable environment. The slightly shifted $\boldsymbol{\beta} \sim \mathcal{U}(-0.5,1.5)$ environment can be interpreted as a situation in which the user can direct some but not all the



(a) β ~Uniform(2, 8)

(b) β ~Uniform(4, 6)

(c) β ~Uniform(0, 10)

(d) β ~Uniform(0, 2)

(e) β ~Uniform(−0.5, 1.5)

(f) β ~Uniform(−1, 1)

(g) 50% Sparsity

(h) 70% Sparsity

(i) 90% Sparsity

Figure 2. Prediction error in environments defined by uniform weight priors with the same mean but different variance (a–c), with shifting support (d–f), and varying degrees of sparsity (g–i). Probability density functions of the weight priors are shown in green in the top-right corner of each panel.

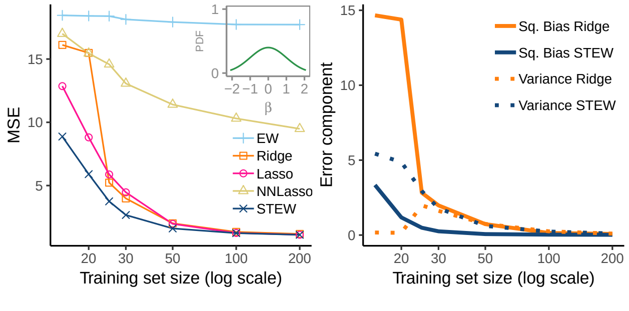
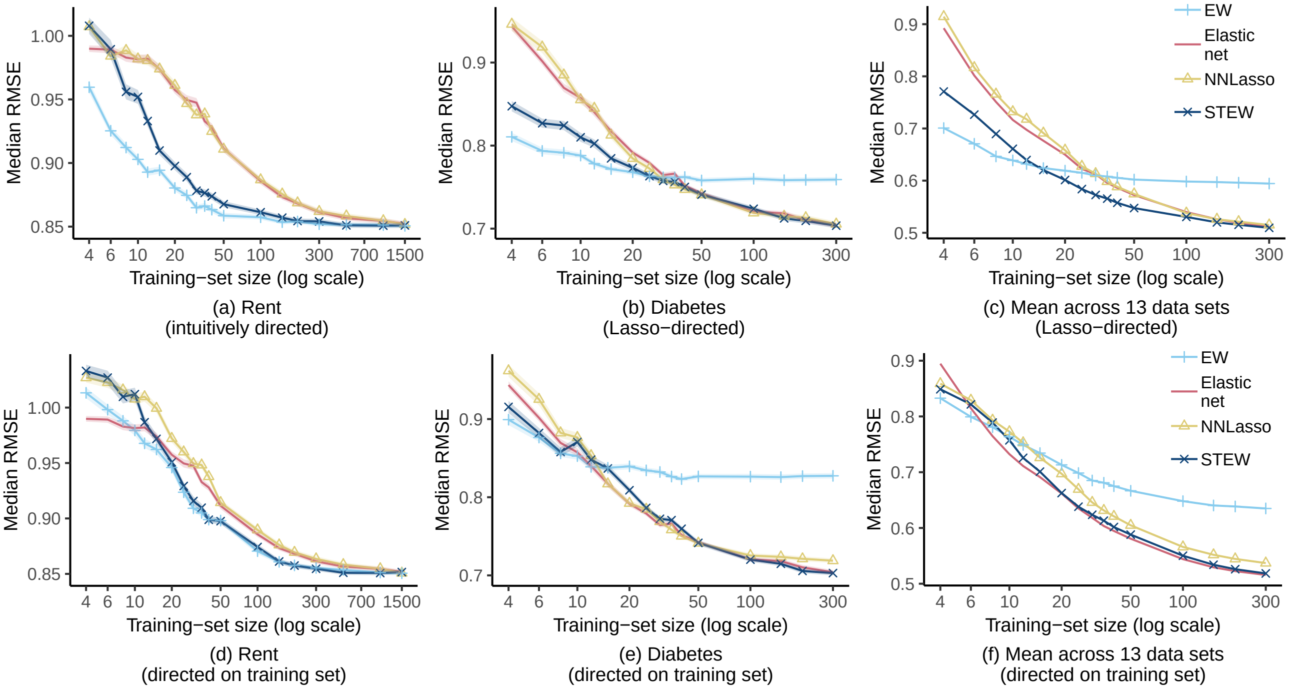


Figure 3. Prediction error (a) and empirical bias-variance decomposition (b) in a Gaussian environment.

weights. Finally, the $\boldsymbol{\beta} \sim \mathcal{U}(-1, 1)$ environment is symmetric around 0; weights cannot be directed. With decreasing directability of weights, the performance of STEW, EW, and NNLasso decreased relative to the performance of models which do not use information about the direction of features. Yet, STEW remained the best performing model even in an undirectable environment. In contrast, NNLasso performed considerably worse than ridge regression and the Lasso.

High-dimensional environments with sparsity. On learning curves presented so far, the early parts of the curves correspond to moderate $p > n$ situations with more features than observations. But p was of the same order of magnitude as n. For the following set of experiments, we increased the number of features to $p = 200$. In addition, we introduced sparsity by setting some proportion of weights to exactly zero. Weights were sampled from a $\mathcal{U}(1, 3)$ distribution and subsequently, conditional on the outcome of a coin flip, set to zero. This coin flip had success probability $\mathbb{P}[\beta = 0] = \omega$, where ω is the expected degree of sparsity in the environment. For example, if $\omega = 0.7$, on average, 70% of the weights have a value of zero, and 30% of the weights



(a) Rent (intuitively directed)

(b) Diabetes (Lasso−directed)

(c) Mean across 13 data sets (Lasso−directed)

(d) Rent (directed on training set)

(e) Diabetes (directed on training set)

(f) Mean across 13 data sets (directed on training set)

Figure 4. Median root mean squared error (RMSE) with standard-error band across 200 repetitions for *Rent*, *Diabetes*, and averaged across 13 data sets. Features were directed based on the intuition of the authors (a), based on a Lasso estimate on the whole data set (b, c), or based on the training set (d–f).

follow a $\mathcal{U}(1, 3)$ distribution. Panels g to i of Figure 2 show the results. With 50% sparsity, STEW outperformed all other models on large parts of the learning curve, especially when $n << p$. With increasing sparsity, Lasso-type models increasingly benefited from their variable selection property. With 90% sparsity, Lasso-type models outperformed STEW across large parts of the learning curve.

Empirical bias-variance analysis. Weights in the environment of Figure 3 follow a Gaussian distribution with zero mean and unit variance, $\boldsymbol{\beta} \sim \mathcal{N}(0, 1)$. This is the ideal environment for ridge regression from a Bayesian perspective (Hoerl & Kennard, 1970). Surprisingly, STEW outperformed all other models including ridge regression across the entire learning curve. The figure also shows the empirical bias-variance decomposition of mean squared error, revealing the different approaches ridge regression and STEW take towards regularization. For small sample sizes, ridge regression reduced variance to almost zero, with error consisting almost entirely of bias. On the other hand, STEW was able to substantially lower bias by allowing some variance.

5.2. Real-World Environments

We compared the prediction performance of STEW, EW, elastic net, and NNLasso on 13 real-world data sets under different conditions regarding how directable features are. The Supplementary Material contains detailed descriptions of each data set.

We first consider the *Rent* data set (Tutz, 2011) where the problem is to estimate the response *rent per m*^{2} for 2053 apartments based on 10 features. In the first stage of our analysis, we directed features based on our intuition. For example, the features *the apartment has warm water (yes = 1, no = 0)* and the *year of construction (in years)* were both expected to be positively associated with the response. Figure 4a shows that both EW and STEW clearly outperformed competing models across the entire learning curve on the intuitively directed *Rent* data set, with EW performing even better than STEW.

Intuitively guessing feature directions is not always easy. In the *Diabetes* data set, in which a quantitative measure of disease progression of 442 diabetes patients needs to be predicted based on *age, sex, body mass index, average blood pressure*, and six blood serum measurements, we could not intuitively guess the directions of most features. However, a physician probably could. We simulated this type of expert knowledge as follows. We estimated a Lasso model on the entire data set and chose the regularization strength that resulted in the lowest cross-validated prediction error. We discarded all features whose Lasso weight was zero and positively directed the remaining features, that is, we multiplied all features with -1 whose Lasso weight was negative. Figure 4b shows learning curves for *Diabetes* obtained in this way. EW performed best until a training set size of 25 but fell behind for training set sizes larger than 40. STEW could not match EW's performance on small training sample sizes. Yet, it clearly outperformed the elastic net and NNLasso on small training set sizes and performed equally well for larger training set sizes.

Average learning curves across all 13 data sets are shown in Figure 4c. EW performed best on very small training sets, STEW on small to medium training sets, and all adaptively-regularized linear models performed equally well on large training set sizes. Individual learning curves (available in the Supplementary Material) show that STEW outperformed both the elastic net and NNLasso on 5 out of 13 data sets, while showing comparable performance in the remaining 8 data sets.

Even when no information about feature directions is available, directions can still be estimated from the training set, for example, from Pearson correlation coefficients between features and the response. Panels d to f of Figure 4 show learning curves when directions were estimated in this way. Averaged across many data sets, STEW did not outperform the competing models but it was robust in the sense that it did not perform worse than the elastic net.

5.3. Tetris

We next present a novel algorithm for learning how to act in sequential decision environments. Multinomial logistic regression plays a central role in our approach. We call the algorithm *M-learning*. The pseudo-code is provided in the Supplementary Material. Below we describe its application to the game of Tetris and evaluate how useful the STEW penalty is within this context.

Tetris. Tetris can be formulated as a Markov decision process, where the state consists of the board configuration and the identity of the falling tetrimino. Available actions are the possible placements of the tetrimino on the board. We denote the set of actions available in state $s \in \mathcal{S}$ by $\mathcal{A}(s)$. In Tetris, the number of actions available in a given state s, denoted by $|\mathcal{A}(s)|$, ranges from 0 to 34. A reward of 1 is received for each cleared line. The game ends when a state allows no legal placement. The objective is to find a policy $\pi : \mathcal{S} \rightarrow \mathcal{A}$ that maximizes the total reward received, in other words, the *game score*.

M-learning in Tetris. The algorithm learns an action-utility function, $U(s,a)$, from which the policy is derived. Specifically, the policy is to choose the action with the highest utility, $\pi(s) = \underset{a \in \mathcal{A}(s)}{\operatorname{argmax}}\, U(s,a)$.

For Tetris, we represent $U(s,a)$ as a linear function of a set of features, $U(s,a) = \boldsymbol{\beta}^T \boldsymbol{\phi}(s,a)$, where $\boldsymbol{\phi}(s,a)$ denote feature values that correspond to selecting action a in state s. As usual, $\boldsymbol{\beta}$ denotes the feature weights. Feature weights are initialized to random values, then periodically updated using multinomial logistic regression. A single training sample for updating feature weights is generated as follows.

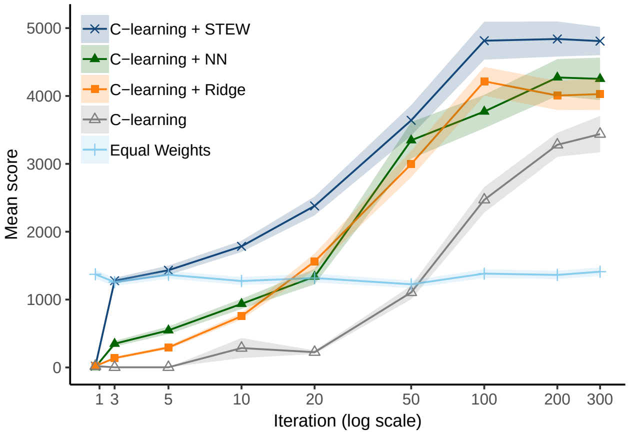


Figure 5. Quality of the policy learned as a function of the iterations of the algorithm. Each learning curve shows means across 100 replications of the algorithm. Quality of the policy is measured by the mean score obtained by the policy in 10 Tetris games.

(1) For every available action in the current state, generate M independent rollouts of length T, where a rollout is a forward simulation of the game beginning at the current state. In this simulation, actions are selected by maximizing $U(s,a)$ unless an immediate clearing of one or more lines is possible, in which the action that clears the highest number of lines is selected.

(2) Execute the action that returned the highest mean total reward in the rollouts. Let $\tilde{a}$ denote this action.

(3) To the training set of multinomial logistic regression, add one new sample, $(\tilde{a}, \boldsymbol{\phi}(s,a_1), ..., \boldsymbol{\phi}(s,a_{|\mathcal{A}(s)|}))$, where the predictors are the feature values of all available actions in state s and the response variable is the identity of the selected action.

Periodically (in our case, after each decision in the game), feature weights are updated through multinomial logistic regression on the accumulated training set. The new set of weights maximizes the likelihood of the selected actions in the training set if the agent were to use action-selection probabilities $\tilde{\pi}(s,a) = \frac{e^{U(s,a)}}{\sum_{a' \in \mathcal{A}(s)} e^{U(s,a')}}$.

Regularized versions of M-learning are easily obtained by adding a penalty term $P(\boldsymbol{\beta})$ to the log-likelihood function, $\log \mathcal{L}(\boldsymbol{\beta}|\mathcal{D}) = \sum_{(s_i,a_i)\in\mathcal{D}} \log(p(s_i,a_i)) - \lambda P(\boldsymbol{\beta})$, where a_i is the action that was selected in state s_i in the training set $\mathcal{D}$ and $\lambda \geq 0$ is the regularization strength. Note that

$\log(p(s_i, a_i))$ depends on $\boldsymbol{\beta}$ because $U(s, a)$ depends on $\boldsymbol{\beta}$. In addition to STEW and ridge penalty, we examine a version of M-learning that maximizes the log-likelihood under non-negativity constraints for all weights.

M-learning is a novel approach to learning in sequential decision environments. In the literature, the most closely-related algorithm is *classification-based reinforcement learning* (Lagoudakis & Parr, 2003), which uses a similar rollout mechanism but does not make use of discrete-choice modeling.

Experiments. We used a board size of 10×10, with rollout parameters $M = 7$, $T = 10$. Given that the number of actions is always smaller than 34, the maximum number of calls to the generative model of Tetris for one iteration of the algorithm was at most $34TM = 2380$. Multinomial logistic regression in iteration k used the most recent $n(k)$ training samples, where $n(k) = min(50, \lfloor \frac{k}{2} \rfloor + 2)$. The regularization strength λ was tuned using cross-validation. Eight features were used to describe a state-action pair: *landing height*, *number of eroded piece cells*, *row 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. Simple and Cumulative dominance filters were applied to all versions of the algorithm as described by Şimşek et al. (2016). Directions for all features were obtained from the weights of the BCTS policy (Thiery & Scherrer, 2009). Features were coded so that they all had positive direction. We present results with four versions of M-learning: STEW penalty, ridge penalty, non-negativity (NN) constraints, and no regularization. In addition, we show the performance of the equal-weights (EW) model as a baseline.

Results. After each iteration of the algorithm, the quality of the learned policy was evaluated by playing 10 games of Tetris. Figure 5 shows mean scores across 100 replications, with shaded areas corresponding to standard error of the mean. STEW led to the highest learning rate, reaching an average score of more than 1000 lines after only 3 iterations, or equivalently, when multinomial logistic regression was trained with only three samples. With alternative penalties, a considerably higher number of samples was needed to achieve comparable play.

While our focus in this article is STEW, it should be noted that the performance obtained by M-learning is remarkable for Tetris. After 100 iterations, the algorithm reached an average score of 4,800 lines. At this point, the algorithm has made no more than 238,000 calls to the generative model. In the literature, the best results for Tetris have been achieved by CBMPI (Gabillon et al., 2013; Scherrer et al., 2015) but these were obtained using a per-iteration budget of 8,000,000 calls to the generative model. When CBMPI is applied with budgets similar to those used for M-learning, CBMPI performance dropped substantially. In the Supplementary Material, we report experimental results comparing M-learning with CBMPI on small budgets.

6. Discussion

One may reasonably assume that STEW would perform well only in environments in which the true feature weights are almost equal. This is clearly not the case. STEW has proven to be useful in a wide range of synthetic and real-world environments, where any assumption of equal weights is clearly violated.

To understand how STEW can outperform models that shrink toward zero, it has been instructive to contrast the two models that are obtained in the limit of infinite regularization: EW and the **0**-model. Our theoretical results show that EW has lower bias than the **0**-model and that this difference increases with increasing directability of features. On data sets that require strong regularization (for example, small data sets), STEW inherits this relatively lower bias.

Sign-constrained models such as NNLS or NNLasso also utilize information about feature directions but showed one main disadvantage compared to STEW. When directions were not available, or were unreliable, these models failed to produce useful estimates, whereas STEW performed on par with other regularized linear models.

STEW showed surprisingly high prediction accuracy across a variety of $p > n$ environments. Yet, unlike Lasso-type models, STEW has no built-in variable-selection mechanism. It is thus clearly not meant to be a model for *sparse recovery*, that is, STEW is not expected to identify the nonzero weights in a sparse environment. It could, however, potentially be developed further to include a sparsity component or used in conjunction with existing methods for variable selection.

One possibility is a two-stage model, similar to *Lasso + OLS* (Efron et al., 2004; Belloni & Chernozhukov, 2013). The first stage of this model consists of fitting a Lasso model on the entire training data and subsequently discarding all features whose Lasso-estimates are zero. The final estimate is then obtained by fitting the second-stage model on the reduced set of features. STEW could prove useful as a second-stage model because the initial Lasso estimate not only takes care of discarding irrelevant features but also provides information about feature directions. STEW and Lasso-type models exploit different types of priors (or information) about the environment. Developing models that can exploit both types of information is a fruitful direction for future research.

Acknowledgements

This work was supported in part by the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 665992 and the UK's EPSRC Centre for Doctoral Training in Digital Entertainment (CDE), EP/L016540/1.

References

Belloni, A. and Chernozhukov, V. Least squares after model selection in high-dimensional sparse models. *Bernoulli*, 19(2):521–547, 2013.

Bühlmann, P. and van de Geer, S. *Statistics for High-Dimensional Data*. Springer, 2011.

Chambolle, A. An algorithm for total variation minimization and applications. *Journal of Mathematical Imaging and Vision*, 20(1):89–97, 2004.

Dana, J. and Thomas, R. In defense of clinical judgment and mechanical prediction. *Journal of Behavioral Decision Making*, 19(5):413–428, 2006.

Davis-Stober, C. P., Dana, J., and Budescu, D. V. A Constrained Linear Estimator for Multiple Regression. *Psychometrika*, 75(3):521–541, 2010.

Dawes, R. M. and Corrigan, B. Linear models in decision making. *Psychological Bulletin*, 81(2):95–106, 1974.

DeMiguel, V., Garlappi, L., and Uppal, R. Optimal versus naive diversification: How inefficient is the 1/N portfolio strategy? *Review of Financial Studies*, 22(5):1915–1953, 2009.

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

Einhorn, H. J. and Hogarth, R. M. Unit weighting schemes for decision making. *Organizational Behavior and Human Performance*, 13(2):171–192, 1975.

Friedman, J., Hastie, T., and Tibshirani, R. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2009.

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.

Gigerenzer, G., Todd, P. M., and the ABC Research Group. *Simple heuristics that make us smart*. Oxford University Press, USA, 1999.

Graefe, A. Improving forecasts using equally weighted predictors. *Journal of Business Research*, 68(8):1792–1799, 2015.

Hoerl, A. E. and Kennard, R. W. Ridge Regression: Biased Estimation for Nonorthogonal Problems. *Technometrics*, 12(1):55, 1970.

Katsikopoulos, K. V., Schooler, L. J., and Hertwig, R. The robust beauty of ordinary information. *Psychological Review*, 117(4):1259, 2010.

Lagoudakis, M. G. and Parr, R. Reinforcement learning as classification: Leveraging modern classifiers. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, pp. 424–431, 2003.

Lichtenberg, J. M. and Şimşek, Ö. Simple regression models. In *Proceedings of the NIPS 2016 Workshop on Imperfect Decision Makers, PMLR*, volume 58, pp. 13–25, 2017.

Meinshausen, N. Sign-constrained least squares estimation for high-dimensional regression. *Electronic Journal of Statistics*, 7:1607–1631, 2013.

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. and Buckmann, M. Learning from small samples: An analysis of simple decision heuristics. In *Advances in Neural Information Processing Systems*, pp. 3141–3149, 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 *International Conference on Machine Learning*, pp. 1757–1765, 2016.

Slawski, M. and Hein, M. Non-negative least squares for high-dimensional linear models: Consistency and sparse recovery without regularization. *Electronic Journal of Statistics*, 7:3004–3056, 2013.

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

Tibshirani, R. Regression Shrinkage and Selection via the Lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288, 1996.

Tibshirani, R., Saunders, M., Rosset, S., Zhu, J., and Knight, K. Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society Series B*, pp. 91–108, 2005.

Tibshirani, R. J. and Taylor, J. The solution path of the generalized lasso. *The Annals of Statistics*, 39(3):1335–1371, 2011.

Tikhonov, A. N., Goncharsky, A. V., Stepanov, V. V., and Yagola, A. G. *Numerical methods for the solution of ill-posed problems*, volume 328. Springer Science & Business Media, 2013.

Tutz, G. *Regression for categorical data*. Cambridge University Press, 2011.

Wainer, H. Estimating coefficients in linear models: It don't make no nevermind. *Psychological Bulletin*, 83(2): 213–217, 1976.

Wu, L., Yang, Y., and Liu, H. Nonnegative-lasso and application in index tracking. *Computational Statistics & Data Analysis*, 70:116–126, 2014.

Zou, H. and Hastie, T. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society, Series B*, 67:301–320, 2005.