# Transfer Performance Evaluation Using Bayesian Methods[*]

Andrés Aradillas Fernández[†]

Fall 2024

**Abstract**

We consider the problem of evaluating the out-of-sample performance, or "transfer performance," of prediction rules. Specifically, we analyze the different performances of ordinary least squares (OLS), a popular econometric method, and least absolute shrinkage and selection operator (Lasso), a regularization method popular in the machine learning (ML) literature. This analysis is done in two parts. First, confidence intervals based on Andrews, Fudenberg, Lei, Liang, and Wu (2024) for transfer performance of both prediction rules are constructed and estimated for a normal linear regression example. Next, confidence intervals constructed within a Bayesian framework are estimated for both prediction rules in the same normal linear regression example, and a framework comparison follows. We find that, while they require assumptions about the distribution of the data and priors on the meta-distribution, the Bayesian framework provides significantly lower computation time. On the other hand, Andrews et al. (2024) offer an entirely nonparametric way of consistently constructing confidence intervals for transfer error.

KEYWORDS: machine learning, transfer performance, out-of-sample prediction, Bayes' rule, confidence intervals, Lasso regression

## 1 Introduction

Two large movements are currently taking place. One is an exponentially-growing interest in the field of machine learning (ML) and other reduced-form methods, with in-sample prediction at the core of goals. The other is a renewed interest in out-of-sample prediction. Recent work, such as

[*] I would like to thank José Luis Montiel Olea for very helpful feedback and comments.

[†] Department of Economics, Cornell University, aa2679@cornell.edu

Andrews et al. (2024), has tackled the question: How do black-box or regularized ML methods perform in out-of-sample prediction compared to traditional economic methods?

The use of machine learning methods has become increasingly popular in the economics literature. For instance, Harris and Yellen (2024) consider the implementation of AI in tandem with human decision-making when deciding truck maintenance. Hansen, McMahon, and Prat (2018) use topic models to make sense of the role transparency plays in Federal Open Market Committee meetings. Gorodnichenko, Pham, and Talavera (2023) later revisit similar data but apply a deep learning algorithm to detect public sentiment. More broadly, Mullainathan and Spiess (2017) give an insightful description into how ML can be intelligently applied to economics research. Athey and Imbens (2019) also build on this literature by describing not only the benefit of ML methods, but also some specific methods and how they may be implemented. There are countless more interesting instances of machine learning methods being utilized in economics (see Kleinberg, Ludwig, Mullainathan, and Obermeyer (2015), Angelico, Marcucci, Miccoli, and Quarta (2022), Davis and Heller (2017), Zheng, Trott, Srinivasa, Parkes, and Socher (2022), Fuster, Goldsmith-Pinkham, Ramadorai, and Walther (2022), Asker, Fershtman, and Pakes (2024), Chernozhukov, Demirer, Duflo, and Fernández-Val (2018), Chernozhukov, Chetverikov, Demirer, Duflo, Hansen, and Newey (2017) for just a few interesting examples).

Despite this abounding interest in ML, something that remains to be better understood is a quantification of the benefits and drawbacks of using novel black-box and regularized ML methods versus standard econometric methods. Andrews et al. (2024) give a broad, nonparametric confidence interval for out-of-sample prediction. However, similar confidence intervals can be constructed within a Bayesian framework, and these confidence intervals may be more appealing to Bayesians, especially due to their much faster rates of computation. The main source of long computation times for the method described in Andrews et al. (2024) lies in the combinatorial nature of their "leave-many-out" procedure, which is necessary for the method to work effectively.

In this paper, we do the following. First, we describe the methods of transfer performance comparison established in Andrews et al. (2024). Next, we reformulate the problem within a Bayesian framework, and define our construction of a confidence interval based on a Bayesian setup. After, we generate data through Monte Carlo simulations and estimate our two types of confidence intervals, comparing two prediction rules, namely ordinary least squares (OLS) and least absolute shrinkage and selection operator (Lasso). We then discuss the differences in the two frameworks, and their respective benefits and drawbacks. We conclude with a summary of our findings and areas for future inquiry.

## 2 An Empirical Distribution Framework

The problem of transfer performance as presented in Andrews et al. (2024) is as follows. Let $\mathcal{X}$ denote set of covariate vectors and $\mathcal{Y}$ denote set of outcomes, with $(x, y) \in \mathcal{X} \times \mathcal{Y}$, and let a *sample* be a set of these observations $S = \{(x_i, y_i)\}_{i=1}^m$. An index $d$ on a sample $S$, written $S_d$, denotes the *domain* to which the sample belongs. The econometrician observes samples from a set of training domains $\mathcal{T}$, with the training samples $S_\mathcal{T} \equiv (S_d)d \in \mathcal{T}$ being used to make predictions about a target domain $d^\star$ with associated target sample $S_{d^\star}$.

The authors of Andrews et al. (2024) refer to $r \equiv |\mathcal{T}|$ as the "parameter of research procedure." For instance, $r = 1$ would denote the problem of single-sample extrapolation, while anything greater could be thought of as a meta-analysis of some sort (e.g., experiments in various different countries being aggregated and utilized to predict experimental results for a new country).

With the notion of domains specified, transfer error $e_{\mathcal{T},d^\star}$ then denotes the transfer error of some prediction rule $f_{S_\mathcal{T}} : \mathcal{X} \to \mathcal{Y}$ that has been trained on the training samples $S_\mathcal{T}$ and predicts values for the target domain $d^\star$. For example, consider the error function for some given prediction rule $f : \mathcal{X} \to \mathcal{Y}$:

$$e(f, S) = \frac{1}{|S|} \sum_{(x,y)\in S} (f(x) - y)^2.$$

Transfer error can then be defined as $e_{\mathcal{T},d^\star} = e(f_{S_\mathcal{T}}, S_{d^\star})$. This will be the form of the transfer error we are interested in for this framework.

The econometrician will not have access to the target sample $S_{d^\star}$. Instead, the econometrician has access to metadata, which will be $n > r$ samples:

$$\mathbf{M} \equiv \{S_1, S_1, ..., S_n\}.$$

Henceforth, we will denote the target sample as $S_{n+1} \equiv S_{d^\star}$.

**Assumption 1** in Andrews et al. (2024) is that there is a fixed, unknown meta-distribution $\mathcal{G} \in \Delta(\mathcal{P}, \mathbb{N})$ over joint distributions $\mathcal{P} \equiv \Delta(\mathcal{X} \times \mathcal{Y})$ and sample sizes $\mathbb{N}$, where each sample $S_d$ is generated by first drawing a distribution and sample size $(P_d, m_d) \sim \mathcal{G}$, then independently drawing $m_d$ observations $(x, y)$ from $P_d$. For simplicity, we will restrict that $m_d = m, \forall d \in \{1, ..., n + 1\}$, as sample size is not of primary interest. Therefore, the revised assumption is now that $\mathcal{G} \in \Delta(\mathcal{P})$ and $P_d \overset{i.i.d.}{\sim} \mathcal{G}$. Notice that this makes an i.i.d. assumption not just for the training samples, but also the target sample. While Andrews et al. (2024) also has results that weaken this i.i.d. assumption, we keep this assumption for simplicity, and also to make the same assumption in the next section.

Since the choice of $r$ training samples out of $n$ possible ones can be thought of as a uniform random variable, we can denote by $\mathbf{T}$ this random variable with realization $\mathcal{T}$. The corresponding random training samples notation is then $S_{\mathbf{T}} \equiv (S_d)_{d \in \mathbf{T}}$, and random transfer error can be denoted $e_{\mathbf{T},n+1}$.

Assumption 1 allows us to use what Andrews et al. (2024) call a *surrogate target sample* $d \in \{1, ..., n\} \setminus \mathcal{T}$ for a realization $\mathcal{T}$ of $\mathbf{T}$. Letting $e^{\mathbf{M}}_{\mathcal{T},d}$ denote the transfer error of a particular surrogate target sample and $\mathbb{T}_{r+1,n}$ denote the set of $\frac{n!}{r!(n-r-1)!}$ unique (up to permutation of the first $r$ terms) pairs $(\mathcal{T}, d)$ satisfying this construction of $d$, the empirical distribution of transfer errors is

$$F_{\mathbf{M}} \equiv \frac{r!(n-r-1)!}{n!} \sum_{(\mathcal{T},d) \in \mathbb{T}_{r+1,n}} \delta_{e^{\mathbf{M}}_{\mathcal{T},d}},$$

where $\delta$ denotes the Dirac delta measure. $F_{\mathbf{M}}$ attains a non-zero value on the support points $\left\{e^{\mathbf{M}}_{\mathcal{T},d} : (\mathcal{T}, d) \in \mathbb{T}_{r+1,n}\right\}$.

Note that this definition of $\mathbb{T}_{r+1,n}$ is a slight divergence from Andrews et al. (2024). While the authors of that paper consider different orderings of the $\{S_1, ..., S_{r+1}\}$ training and target samples to be distinct, we only consider them distinct up to the selection of a target sample and an aggregated training sample. For instance, if $n = 3$ and $r = 2$, then for us, $\{1, 3, 4\}$ and $\{1, 4, 3\}$ are distinct sets, while $\{1, 3, 4\}$ and $\{3, 1, 4\}$ are not. In Andrews et al. (2024), these would all be considered distinct sets. We make this simplification because our prediction rules will not rely on the ordering of the training samples,[1] and thus making this alteration will also reduce computation time (this is demonstrated in Section B.2).

We can now define confidence intervals. With slight abuse of notation, define

$$\overline{e}^{\mathbf{M}}_{\tau} \equiv \inf\{e : F_{\mathbf{M}}((-\infty, e]) \geq \tau\}, \qquad \underline{e}^{\mathbf{M}}_{\tau} \equiv \sup\{e : F_{\mathbf{M}}([e, \infty)) \geq 1 - \tau\}.$$

Therefore, a confidence interval proven to be valid (for the "true" transfer error $e_{\mathbf{T},r+1}$) in Andrews et al. (2024) is $[\underline{e}^{\mathbf{M}}_{\tau}, \overline{e}^{\mathbf{M}}_{\tau}]$. This confidence interval will primarily serve as the "empirical distribution" confidence interval, henceforth the ED confidence interval.

[1] We could use alternate prediction rules that do rely on the ordering of the training samples. Nothing about our methodology breaks if we allow for this, we will just have to revert to the original construction of $\mathbb{T}_{r+1,n}$ presented in Andrews et al. (2024).

# 3 A Bayesian Approach

## 3.1 Description of Method

The desire of Andrews et al. (2024) is to produce a confidence interval that, with a desired probability, will contain the true transfer performance error *without* any parametric assumptions. However, we can instead take a Bayesian approach and see how the two compare.

As before, we will have samples $S_d = \{(x_i, y_i)\}_{i=1}^{m_d}$ each corresponding to a domain $d$. Here, we will have that $x_i \in \mathbb{R}^k$. There is still a meta-distribution as in Assumption 1 of Andrews et al. (2024), with data distribution $P_d$ and corresponding sample size $m_d$ being distributed $(P_d, m_d) \overset{i.i.d.}{\sim} \mathcal{G}$. We will not relax the i.i.d. assumption, but we do simplify that sample sizes are the same, i.e., $m_1 = ... = m_n = m$, same as in the ED framework.

This form gives us a hierarchical model because each $S_d$ can be generated by first sampling $P_d$ from $\mathcal{G}$, then conditional on $P_d$, generating $m$ samples. Under this model, the data we see are independent draws from a mixture model where the mixing distribution is $\mathcal{G}$ and the mixture component is whatever distributional assumption is made about $(x_i, y_i)$.

A common example that one may expect is having a linear model for the data. Consider the model where, for a given domain $d$, we have

$$y_i = \beta_d' x_i + \sigma_d \epsilon_i.$$

We assume $x_i \perp \epsilon_i$ and also that $(x_i, \epsilon_i)$ is multivariate normal with mean $\mathbf{0}$, $Var(x_i) = \mathbb{I}_k$ and make another normalization so that $Var(\epsilon_i) = 1$. Here, each domain has parameters $\theta_d \equiv (\beta_d, \sigma_d)$, and so $\mathcal{G}$ is just a distribution over $\theta_d$.

The loss function based on Andrews et al. (2024) can be thought of as the following. We fix a prediction rule $f : \mathcal{X} \to \mathcal{Y}$, and look first at

$$L_{aux}(f, \mathcal{G}) = \mathbb{E}_{\mathcal{G}}[(y - f(x))^2],$$

where the expectation is taken by assuming $(x, y)$ are drawn according to the previously-described mixture model.

We are interested in using training data to provide an interval in the real line $C \equiv [C_L, C_U]$. We could do this in two ways. One way is that we could aim to balance "not too large" with "contains true error with high probability." These two can be balanced with the following loss function and exogenously-given $\delta$:

$$L_\delta(C_L, C_U, \mathcal{G}) = \delta \mathbf{1}\{L_{aux} \notin [C_L, C_I]\} + (1-\delta)(C_U - C_L).$$

If $\mathcal{G}$ is known, then the oracle rule is trivially $C_U = C_L = L_{aux}(a, \mathcal{G})$, and so we would assume that $\mathcal{G}$ is not known. This would be a statistical decision theory approach to the problem. However, with $\delta$ being exogenously-given, this approach does not allow us to easily determine the confidence level this corresponds with. Therefore, we can instead take an approach similar to the ED framework by choosing $\tau$ and selecting a desired percentile of the generated data (to be clarified in the rest of the section). In this way, we also do not differentiate our two frameworks by too much, allowing for a more controlled comparison of the two. Furthermore, this other form of confidence interval was also estimated, and in our simulated example, the confidence interval was very similar to the other definition we chose to go with. The overall Bayesian approach is as follows.

First, we impose a prior on $\mathcal{G}$. Because of the normality of our problem, a clean solution to updating the prior once data is received can be obtained with the following conjugate priors:

$$\beta | \sigma^2 \sim N(0, \sigma^2 \mathbb{I}_k), \qquad \sigma^2 \sim \text{InvGamma}(\alpha_\pi, \beta_\pi),$$

where InvGamma$(\cdot,\cdot)$ refers to the inverse gamma distribution, $N(0, \sigma^2\mathbb{I}_k)$ refers to a $k$-dimensional normal distribution, and $\mathbb{I}_k$ refers to the $k \times k$ identity matrix. Therefore, our priors are distinct only up to choice of $\alpha_\pi$ and $\beta_\pi$.

Then, once we receive data $\mathbf{X}$ and $\mathbf{y}$, we can use Bayes' rule to update our prior and obtain a posterior distribution. The posterior distribution here is (assuming $\mathbb{X}$ has full column rank[2] with probability 1):

$$\beta | \sigma^2, \mathbf{X}, \mathbf{y} \sim N\left((\mathbf{X}^\top \mathbf{X} + \mathbb{I}_k)^{-1} (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}, \sigma^2 (\mathbf{X}^\top \mathbf{X} + \mathbb{I}_k)^{-1}\right)$$

$$\sigma^2 | \mathbf{X}, \mathbf{y} \sim \text{InvGamma}\left(\alpha_\pi + \frac{n}{2}, \beta_\pi + \frac{1}{2}\left(\mathbf{y}^\top \mathbf{y} - \mathbf{y}^\top \mathbf{X} (\mathbf{X}^\top \mathbf{X} + \mathbb{I}_k)^{-1} \mathbf{X}^\top \mathbf{y}\right)\right).$$

The details of this derivation are in Section A.1.

Letting $\widehat{\mathcal{G}}$ denote the overall posterior distribution here, we can then take $N >> 1$ i.i.d. draws from $\widehat{\mathcal{G}}$, yielding us $\{\widehat{G}_1, ..., \widehat{G}_N\}$ independent draws from the estimated "meta-distribution."

With these $N$ draws from $\widehat{\mathcal{G}}$ in hand, we can then do the following. For each realization $\widehat{G}_j$, gen-

[2] If this does not end up being true, we could nevertheless replaced $(\mathbf{X}^\top \mathbf{X})^{-1}$ with a generalized inverse. In our simulations, we do not run into the problem of rank deficiency, and so we stick to traditional notion of a matrix inverse for the remainder of the paper.

erate $M$ i.i.d. draws of the data according to $\widehat{G}_j$, and then compute a sample analog of $L_{aux}(f, \widehat{G}_j)$, which we collect into a set we shall denote $\{e^B_{\widehat{\mathcal{G}}}\}$.

Finally, with these estimated transfer errors $\{e^B_{\widehat{\mathcal{G}}}\}$, we can then define the confidence intervals. Let $F_{B,\widehat{\mathcal{G}}}$ denote the empirical distribution of $\{e^B_{\widehat{\mathcal{G}}}\}$. Then define (with slight abuse of notation):

$$\overline{e}^B_\tau \equiv \inf\{e : F_{B,\widehat{\mathcal{G}}}((-\infty, e]) \geq \tau\}, \qquad \underline{e}^B_\tau \equiv \sup\{e : F_{B,\widehat{\mathcal{G}}}([e, \infty)) \geq 1 - \tau\}.$$

With large enough choice of $N$ and $M$, a simple central limit theorem argument would guarantee that $[\underline{e}^B_\tau, \overline{e}^B_\tau]$ approaches the actual middle $(2\tau - 1)$-length interval of the transfer error if $\widehat{\mathcal{G}}$ were the true meta-distribution. The smallest $(2\tau - 1)$-level intervals were also generated, but the resulting intervals were nearly the same as the middle $(2\tau - 1)$-level ones. This similarity is likely due to the normality assumptions, so only the middle $(2\tau - 1)$-level intervals are reported. Furthermore, this approach is in line with what is done in Andrews et al. (2024).

## 3.2 Discussion

It is important here to make a methodological distinction between the ED and Bayesian frameworks. In the former, we make absolutely no parametric assumptions about how the data in each sample domain $S_d$ is distributed, as we are not interested in estimating $P_d$. Instead, we wish to construct confidence intervals for transfer performance using a leave-one-out method to attain surrogate target samples. In doing so, though, we do leave out information, as note that $r < n$ is needed. This means that the $r$ we get results for is limited heavily by $n$. Furthermore, to construct reasonable confidence intervals, we need to make $r$ sufficiently lesser than $n$ so that $\frac{r!(n-r-1)!}{n!}$ is a small enough number that $\overline{e}^{\mathbf{M}}_\tau$ and $\underline{e}^{\mathbf{M}}_\tau$ are distinct numbers, and also satisfy their $\tau$ conditions without simply being the highest and lowest values in the support points $\left\{e^{\mathbf{M}}_{\mathcal{T},d} : (\mathcal{T}, d) \in \mathbb{T}_{r+1,n}\right\}$. If we do not do this, our results may be of little insight. For instance, if $r$ close enough to $n$ such that $\frac{r!(n-r-1)!}{n!} > 1 - \tau$, then $\underline{e}^{\mathbf{M}}_\tau = \inf\left\{e^{\mathbf{M}}_{\mathcal{T},d} : (\mathcal{T}, d) \in \mathbb{T}_{r+1,n}\right\}$, which makes for a confidence interval indistinguishable from choice of $\tau + \epsilon$ for small $\epsilon > 0$.

On the other hand, the Bayesian framework ensures that we use all of the available information. This does, however, come at the cost of assuming the parametric distributions $P_d$ of each sample domain $S_d$, something which the ED framework does not suffer from. However, if distributional assumptions are convincing, then this method allows for a far faster computation time, as well as utilizing all of the information available.

Another observation is that the ED framework proposed in Andrews et al. (2024) is reminiscent of a non-parametric bootstrap method and also leave-one-out cross-validation. The ED framework could be thought of as a “leave-many-out” method, wherein prediction is then done on one of the many “left out” domains being used as surrogate target samples.

A feature of the Bayesian framework here is that the information about which domain the observations belong to has been left out. Therefore, this method can be utilized even when the domain of a data is not known. This may serve useful when this information about the data is lost. It can also be used when each domain has a single observation. Examples of this could be insurance claim rates where each claimant is of a different risk type, a problem which is also an example of the poisson example presented in Robbins (1956), though that work estimates the prior directly using a similar but distinct method called empirical Bayes.

# 4 Monte Carlo Simulations

## 4.1 Setup of Normal Linear Model

We wish to generate data for various domains as according to the meta-distribution of $\beta$ and $\sigma^2$. To generate this data, we let the data from each domain be generated from

$$y = \beta' x + \sigma\epsilon,$$

where

$$(x, \epsilon) \sim N(0, \mathbb{I}_{k+1})$$

$$\beta|\sigma \sim N(0, \sigma^2\mathbb{I}_k), \qquad \sigma^2 \sim \text{InvGamma}(3, 1)$$

For tractability of the ED framework, we pick $n = 16$ and $r = 8$. If we have that $m_1 = ... = m_n = m$ and we pick that $m = 100$, then each (empirical) transfer error $e^{\mathbf{M}}_{\mathcal{T},d}$ for a given unique pair $(\mathcal{T}, d \in \mathbb{T}_{r+1,n})$ is

$$e^{\mathbf{M}}_{\mathcal{T},d} = \frac{1}{100} \sum_{(x,y)\in S_d} (f_{\hat{\theta}|S_{\mathcal{T}}}(x) - y)^2,$$

where $f_{\hat{\theta}|S_{\mathcal{T}}}$ denotes estimating a prediction rule $f$ with parameters $\hat{\theta}$ based on a given training sample $S_{\mathcal{T}}$. Collecting all of these values gives a confidence interval, exactly as specified in the protocol given in Andrews et al. (2024). For our confidence interval, we choose $\tau = 0.975$ and generate only the two-sided interval, which corresponds to a 95% confidence interval.

For the Bayesian framework, we first adopt the prior that $\alpha_\pi = \beta_\pi = 1$ (also maintaining the other prior that $\beta|\sigma^2 \sim N(0, \sigma^2\mathbb{I}_k)$). We then apply the method exactly as described in Section 3.1. Again, we choose $\tau = 0.975$, netting us a 95% confidence interval.

For both of these frameworks, the data we generate is actually done for two values of $k = \dim(x)$. We will consider $k \in \{1, 10\}$.

## 4.2 Prediction Rules of Interest

We are primarily interested in two prediction rules: OLS and Lasso. The former is a common form of regression appearing in a vast number of works within the broader economics literature. The objective function here is

$$\min_{\widehat{\beta}} (\mathbf{y} - \mathbf{X}\widehat{\beta})^\top (\mathbf{y} - \mathbf{X}\widehat{\beta}).$$

The latter, Lasso, is an increasingly popular version of OLS that also has a regularization parameter $\lambda$, and solves the objective function

$$\min_{\widehat{\beta}} (\mathbf{y} - \mathbf{X}\widehat{\beta})^\top (\mathbf{y} - \mathbf{X}\widehat{\beta}) + \lambda \sum_{i=1}^{k} |\widehat{\beta}|.$$

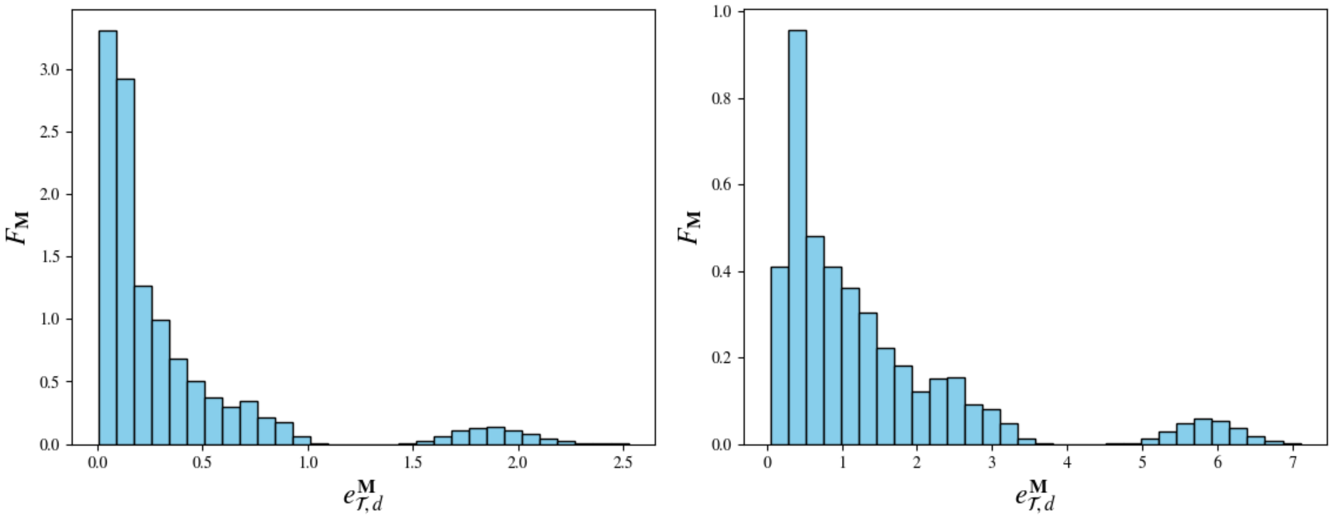
Lasso and related estimation methods have been found to guarantee good out-of-sample prediction error, as shown in Montiel Olea, Rush, Velez, and Wiesel (2022), so long as $\lambda$ is optimally chosen. Therefore, we expect the transfer error to be smaller for Lasso than OLS.

The hyperparameter $\lambda$ of the Lasso regression is $K$-fold cross-validated to ensure more optimal out-of-sample prediction performance. $K$-fold cross-validation consists of dividing up the data into $K$ number of groups or folds, and using each fold as a testing sample in turn, with the rest of the $K - 1$ folds being the (aggregated) training sample. In each of the $K$ turns, $\lambda$ is optimally chosen by trying out many $\lambda$ values, and an overall optimal $\lambda$ is chosen to minimize prediction error across all folds. Section 7.10 of Hastie, Tibshirani, and Friedman (2009) give a more detailed treatment of $K$-fold cross validation and its nice properties.

## 4.3 Results from Simulations

The prediction rules we chose to compare in transfer error via the ED framework were OLS and Lasso with a 5-fold cross-validated regulation parameter chosen for each element of $\mathbb{T}_{r+1,n}$. Lasso was implemented using the off-the-shelf `scikit-learn` and `statsmodels` packages in Python. The choice of 5 folds is a standard one in the literature (see Anguita, Ghelardoni, Ghio, Oneto, Ridella et al. (2012) for a discussion of the reasoning behind this rule-of-thumb).

The distributions of $e^{\mathbf{M}}_{\mathcal{T},d}$ generated by the Monte Carlo simulations for $k = 1$ and $k = 10$ are displayed for OLS in Figure 1 and for Lasso in Figure 2. The shape of the distributions all look



(a) $k = 1$

(b) $k = 10$

Figure 1: Distribution of ED empirical transfer errors for OLS.

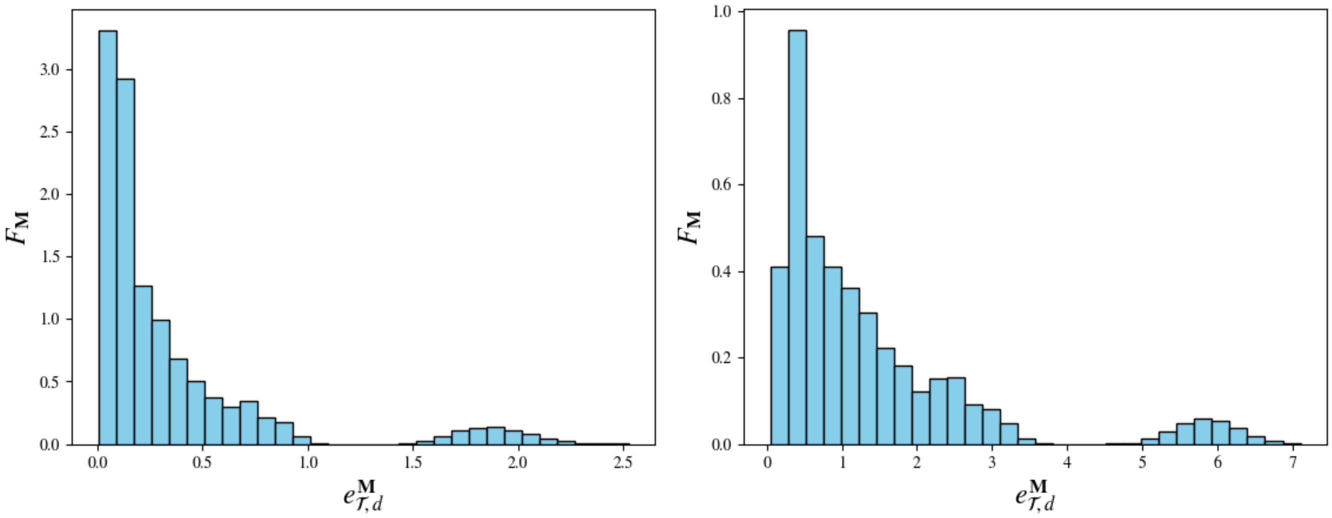
| Prediction Rule | Runtime ($k = 1$) | Runtime ($k = 10$) |
|---|---|---|
| OLS | $1.32 \times 10^2 s$ | $1.16 \times 10^2 s$ |
| Lasso | $5.86 \times 10^3 s$ | $8.03 \times 10^3 s$ |

Table 1: Transfer Error Generation Runtimes.

qualitatively similar, though the magnitude of results when $k = 10$ is larger, which makes sense, as with our current construction, the $k = 10$ case has much more variance than with $k = 1$, and so we expect our transfer errors to enlarge upon increasing $k$. Another interesting thing to notice is that the overall shapes for the OLS and Lasso transfer errors within each value of $k$ look incredibly similar. This fact is likely due to the similar construction of their objective functions. Note, however, that they are not identical, and this will be more apparent in their confidence intervals.

Additionally, the runtimes of generating the transfer errors for OLS and Lasso for $k = 1$ and $k = 10$ are shown in Table 1. These were run on a personal ASUS Vivobook Pro 15 @ 2.5GHz Intel Core Ultra 9 185H. As we can see, generating transfer errors for the Lasso prediction rule in both the $k = 1$ and $k = 10$ case is an order of magnitude larger than for the OLS prediction rule. This observation will factor into our later discussion of what we can take away from all of these transfer error results (see Section 5).

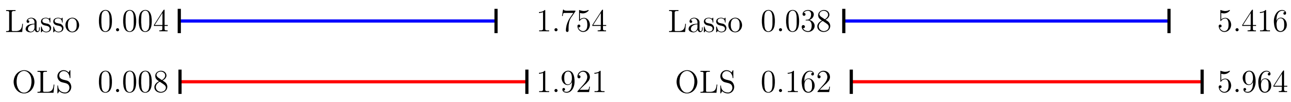
Picking a value of $\tau = 0.975$ yields the two-sided confidence intervals shown in Figure 3. There are two interesting things to perceive here. One is that the transfer error confidence intervals are



(a) $k = 1$

(b) $k = 10$

Figure 2: Distribution of ED empirical transfer errors for Lasso with 5-fold cross-validated $\lambda$.



(a) $k = 1$

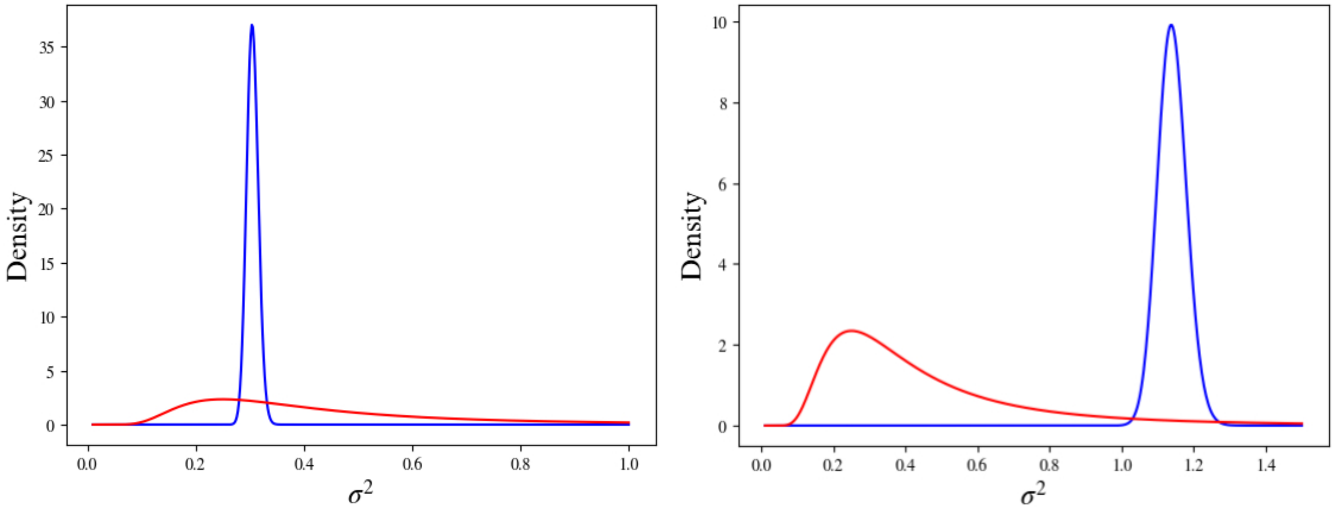
(b) $k = 10$

Figure 3: ED confidence intervals for OLS and Lasso.

larger for the $k = 10$ case. Once again, this larger width was expected, as this case has larger variance by construction than the $k = 1$ case.

Another interesting result from Figure 3 is that the Lasso confidence interval is smaller in both the $k = 1$ and $k = 10$ case. Notice that in the $k = 1$ simulations, the confidence interval generated for Lasso is around 8.5% less in width than for OLS, while for $k = 10$, this decrease is around 7.3%. Montiel Olea et al. (2022) have a result that give a formula for picking an optimal regularization parameter for a method related to Lasso (namely, square root Lasso), and this formula is increasing in $k$ (i.e., more covariates means higher regularization). Therefore, while the difference in width definitely grows in magnitude between the $k = 1$ and $k = 10$ simulations, it appears that the percent difference remains roughly stable, or may even be decreasing in $k$.

For the Bayes framework, we can first see in Figure 4 the prior and posterior distributions of $\sigma^2$ with the data generated. Additionally, the empirical distributions of the transfer errors simulated by the Bayes framework are in Figure 5 for OLS and Figure 6 for Lasso. As was the case with



(a) $k = 1$

(b) $k = 10$

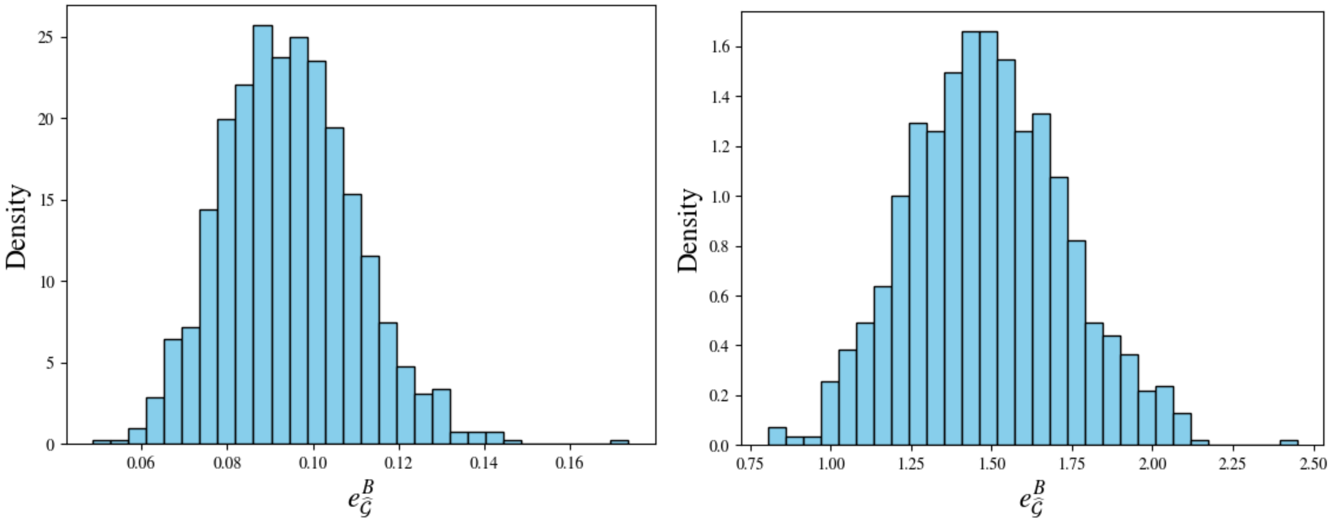
Figure 4: Prior (red) and posterior (blue) distributions of $\sigma^2$.

the ED framework, the $k = 10$ by construction has higher variance, and so while the shapes of all the distributions of $e^B_{\widehat{\mathcal{G}}}$ are qualitatively similar, the $k = 10$ case is of a larger magnitude for both OLS and Lasso. The figures themselves do not tell us much about the differing transfer performances of OLS and Lasso, however, and so we will have to look to the confidence intervals for more information.

A two-sided confidence interval also with $\tau = 0.975$ is shown in Figure 7. Note that, although these confidence intervals fit inside the ED confidence intervals showcased in Figure 3, this is numerical coincidence, and will not necessarily be true all the time. Because we assume a prior, this puts a form on $\mathcal{G}$ and also its posterior $\widehat{\mathcal{G}}$, and therefore the width of the Bayesian confidence intervals is partly due to the choice of this prior. We could have chosen a different prior that may have given a confidence interval wider than that of the ED framework, even if $\tau$ were the same between the two frameworks.

Unlike in the ED framework, these confidence intervals actually show an increase in width difference between $k = 1$ and $k = 10$: a near 0% width reduction turns into a 9.3% reduction when switching from OLS to Lasso. This suggests a different relationship in this framework between covariate dimension $k$ and the OLS/Lasso transfer error reduction. Perhaps one reason for this is the use of a well-specified prior (well-specified in the sense that our prior is within the same class as the true data generating process). It could be that if our prior were misspecified in the sense of Fudenberg, Romanyuk, and Strack (2017), this relationship may not be as clear-cut.

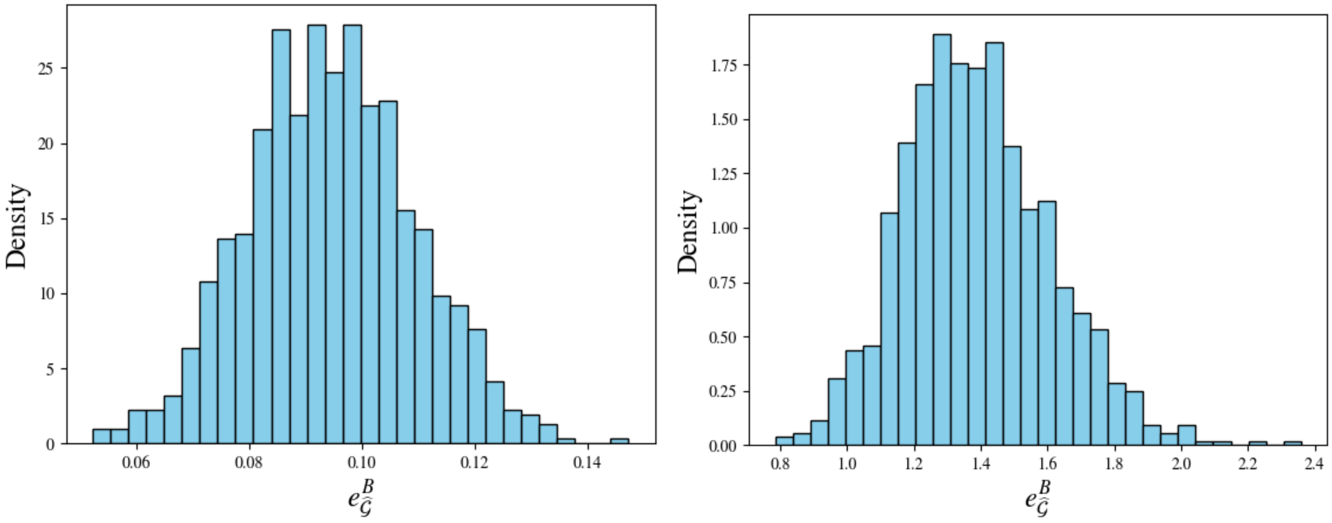
Regardless of the reason for this differing relationship in $k$ and confidence interval narrowing, this



(a) $k = 1$

(b) $k = 10$

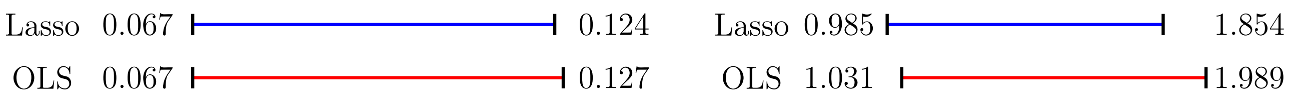
Figure 5: Distribution of transfer errors in Bayesian framework for OLS.



(a) $k = 1$

(b) $k = 10$

Figure 6: Distribution of transfer errors in Bayesian framework for Lasso.



(a) $k = 1$

(b) $k = 10$

Figure 7: Bayesian framework confidence intervals for OLS and Lasso.

observation provides a noticeable difference in the results produced by the two types of confidence intervals. In the ED-based confidence intervals, the difference between Lasso and OLS in terms of transfer performance can be perceived even at the $k = 1$ level, even though works such as Montiel Olea et al. (2022) suggest the optimal use of Lasso in this case may be to keep it close to OLS. On the other hand, the Bayes-based confidence intervals appear to capture a slightly different phenomenon, showing the growing returns in transfer performance when using Lasso over OLS. Nevertheless, while both methods use the same simulated data, what they do with the data is fundamentally different, and is likely a determining factor in this perceivable difference.

### 4.4 Availability of Code

All of the code we utilized in this paper, complete with a detailed README file of instructions, are available at the following GitHub repository: https://github.com/aaf101/transfer-performance-project. It is all in Python, and should allow for replicability of all results, including those generated based on Andrews et al. (2024).

## 5 Frameworks Comparison

One primary issue with the ED framework is the heavy computational cost. Any program will have to run through $|\mathbb{T}_{r+1,n}|$ different possible combinations, and even with the simplification of $\mathbb{T}_{r+1,n}$ we have made, this results in a cost that may be large depending on choice of $n$ and $r$. In fact, this computational cost can be specified by holding $r$ fixed and letting $n \to \infty$. We can show (see derivation in Section B.1) that, if we hold $r$ fixed:

$$|\mathbb{T}_{r+1,n}| = \frac{n!}{r!(n-r-1)!} = O\left(n^{r+1}\right).$$

Furthermore, this computational cost becomes even larger when we instead fix some ratio $\alpha \equiv \frac{r}{n} < 1$. In that case,

$$|\mathbb{T}_{r+1,n}| = O\left(\frac{n}{\alpha^{\alpha n}}\right).$$

The details of the derivation are in Section B.2.

On the other hand, while the Bayesian framework does not suffer from this combinatorial issue when generating the surrogate target samples required for the ED method, it does have the fundamental difference of demanding a prior distribution. The ED framework offers a more frequentist approach to understanding transfer performance, and so the same appeal of frequentist vs. Bayesian comes into play here. Recall that a Bayesian approach necessitates prior beliefs, and so such a confidence interval is limited by the prior placed upon it, while a frequentist confidence interval places no such prior beliefs. It also appears that the two confidence intervals are certainly not the same qualitatively, even in situations where one appears to be contained in the other, despite the intuition that the Bayesian framework simply "held more information" compared to the ED framework being incorrect. This suggests they may capture different phenomena.

# 6 Conclusion

To summarize our findings, we were able to define two fundamentally different confidence intervals for transfer performance evaluation, one based on Andrews et al. (2024) and the other founded in Bayes' rule. With OLS and Lasso as our two prediction rules, our results show a qualitative difference in what these two confidence intervals are able to capture regarding the transfer performance of these two methods.

There are many possible avenues for future research. An immediate one to explore is the following. In the Bayesian framework, we assumed a prior and then generated a posterior to base our transfer performance confidence interval on. However, there also exist empirical Bayes methods that do not assume a prior, only the distribution of the data, and use these data to estimate (often non-parametrically) a prior to then be utilized as in Bayes. It would be interesting to see how such a confidence interval would compare to the current Bayesian one presented in this paper.

One other area for future research we wish to highlight is the use of Bayesian nonparametric methods. To do this, a prior is placed on an infinite dimensional space and a method such as a Dirichlet process could be used to do this. It would be interesting to see whether the same relations between OLS and Lasso would bear out under such a method.

# References

Andrews, I., D. Fudenberg, L. Lei, A. Liang, and C. Wu (2024): “The transfer performance of economic models,” *arXiv preprint arXiv:2202.04796*.

Angelico, C., J. Marcucci, M. Miccoli, and F. Quarta (2022): “Can we measure inflation expectations using Twitter?” *Journal of Econometrics*, 228, 259–277.

Anguita, D., L. Ghelardoni, A. Ghio, L. Oneto, S. Ridella, et al. (2012): “The ’K’ in K-fold Cross Validation.” in *ESANN*, vol. 102, 441–446.

Asker, J., C. Fershtman, and A. Pakes (2024): “The impact of artificial intelligence design on pricing,” *Journal of Economics & Management Strategy*, 33, 276–304.

Athey, S. and G. W. Imbens (2019): “Machine learning methods that economists should know about,” *Annual Review of Economics*, 11, 685–725.

Chernozhukov, V., D. Chetverikov, M. Demirer, E. Duflo, C. Hansen, and W. Newey (2017): “Double/debiased/neyman machine learning of treatment effects,” *American Economic Review*, 107, 261–265.

Chernozhukov, V., M. Demirer, E. Duflo, and I. Fernández-Val (2018): “Generic Machine Learning Inference on Heterogeneous Treatment Effects in Randomized Experiments, with an Application to Immunization in India,” Working Paper 24678, National Bureau of Economic Research.

Davis, J. M. and S. B. Heller (2017): “Using causal forests to predict treatment heterogeneity: An application to summer jobs,” *American Economic Review*, 107, 546–550.

Fahrmeir, L., T. Kneib, S. Lang, and B. Marx (2013): *Regression: Models, Methods and Applications*, Springer Berlin Heidelberg.

Fudenberg, D., G. Romanyuk, and P. Strack (2017): “Active learning with a misspecified prior,” *Theoretical Economics*, 12, 1155–1189.

Fuster, A., P. Goldsmith-Pinkham, T. Ramadorai, and A. Walther (2022): “Predictably unequal? The effects of machine learning on credit markets,” *The Journal of Finance*, 77, 5–47.

Gorodnichenko, Y., T. Pham, and O. Talavera (2023): “The voice of monetary policy,” *American Economic Review*, 113, 548–584.

Hansen, S., M. McMahon, and A. Prat (2018): “Transparency and deliberation within the FOMC: A computational linguistics approach,” *The Quarterly Journal of Economics*, 133, 801–870.

Harris, A. and M. Yellen (2024): “Decision-Making with Machine Prediction: Evidence from Predictive Maintenance in Trucking,” .

Hastie, T., R. Tibshirani, and J. Friedman (2009): *The Elements of Statistical Learning:*

*Data Mining, Inference, and Prediction*, Springer series in statistics, Springer.

Kleinberg, J., J. Ludwig, S. Mullainathan, and Z. Obermeyer (2015): "Prediction policy problems," *American Economic Review*, 105, 491–495.

Montiel Olea, J. L., C. Rush, A. Velez, and J. Wiesel (2022): "The out-of-sample prediction error of the square-root-LASSO and related estimators," *arXiv preprint arXiv:2211.07608*.

Mullainathan, S. and J. Spiess (2017): "Machine learning: an applied econometric approach," *Journal of Economic Perspectives*, 31, 87–106.

Robbins, H. E. (1956): "An empirical Bayes approach to statistics," *Proceedings of the Third Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics*, 3, 157–163.

Zheng, S., A. Trott, S. Srinivasa, D. C. Parkes, and R. Socher (2022): "The AI Economist: Taxation policy design via two-level deep multiagent reinforcement learning," *Science advances*, 8, eabk2607.

# A Bayesian Framework Derivations

## A.1 Derivation of Bayesian Update Formula

Consider the linear model

$$y = \beta' x + \sigma\epsilon,$$

where $x \in \mathbb{R}^k$, $(x, \epsilon) \sim N_{k+1}(0, \mathbb{I}_{k+1})$ and we have conjugate priors

$$\beta|\sigma^2 \sim N(0, \sigma^2\mathbb{I}_k), \qquad \sigma^2 \sim \text{InvGamma}(\alpha_\pi, \beta_\pi).$$

With this construction, we know that

$$p(\mathbf{y}|\mathbf{X}, \beta, \sigma^2) = \prod_{i=1}^{n} p(y_i|\mathbf{X}, \beta, \sigma^2).$$

Therefore, we can write

$$p(\mathbf{y}|\mathbf{X}, \beta, \sigma^2) \propto (\sigma^2)^{-\frac{n}{2}} \exp\left(-\frac{1}{2\sigma^2}(\mathbf{y} - \mathbf{X}\beta)^\top(\mathbf{y} - \mathbf{X}\beta)\right).$$

Note that, by expanding and using orthogonal arguments to simplify some terms, we find that

$$\begin{aligned}(\mathbf{y}-\mathbf{X}\beta)^\top(\mathbf{y}-\mathbf{X}\beta) &= (\mathbf{y}-\mathbf{X}\widehat{\beta}+\mathbf{X}\widehat{\beta}-\mathbf{X}\beta)^\top(\mathbf{y}-\mathbf{X}\widehat{\beta}+\mathbf{X}\widehat{\beta}-\mathbf{X}\beta)\\ &= (\mathbf{y}-\mathbf{X}\widehat{\beta})^\top(\mathbf{y}-\mathbf{X}\widehat{\beta})+(\beta-\widehat{\beta})^\top(\mathbf{X}^\top\mathbf{X})(\beta-\widehat{\beta}),\end{aligned}$$

where $\widehat{\beta}\equiv(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{y}$. Plugging this back in yields a nice separation:

$$\begin{aligned}p(\mathbf{y}|\mathbf{X},\beta,\sigma^2) &\propto (\sigma^2)^{-\frac{n}{2}}\exp\left(-\frac{1}{2\sigma^2}\left((\mathbf{y}-\mathbf{X}\widehat{\beta})^\top(\mathbf{y}-\mathbf{X}\widehat{\beta})+(\beta-\widehat{\beta})^\top(\mathbf{X}^\top\mathbf{X})(\beta-\widehat{\beta})\right)\right)\\ &= (\sigma^2)^{-\frac{n-k}{2}}\exp\left(-\frac{(n-k)s^2}{2\sigma^2}\right)(\sigma^2)^{-\frac{k}{2}}\exp\left(-\frac{1}{2\sigma^2}(\beta-\widehat{\beta})^\top(\mathbf{X}^\top\mathbf{X})(\beta-\widehat{\beta})\right),\end{aligned}$$

where $s^2\equiv\frac{1}{n-k}(\mathbf{y}-\mathbf{X}\widehat{\beta})^\top(\mathbf{y}-\mathbf{X}\widehat{\beta})$. If we are to have that $p(\sigma^2)$ is distributed according to $\text{InvGamma}(\alpha_\pi,\beta_\pi)$, then

$$p(\sigma^2)\propto(\sigma^2)^{-(\alpha_\pi+1)}\exp\left(-\frac{\beta_\pi}{\sigma^2}\right),$$

and with $\beta|\sigma^2\sim N(0,\sigma^2)$, this means

$$p(\beta|\sigma^2)\propto(\sigma^2)^{-\frac{k}{2}}\exp\left(-\frac{1}{2\sigma^2}\beta^\top\beta\right).$$

Therefore, utilizing Bayes' rule,

$$\begin{aligned}p(\beta,\sigma^2|\mathbf{y},\mathbf{X}) &\propto p(\mathbf{y}|\mathbf{X},\beta,\sigma^2)p(\beta|\sigma^2)p(\sigma^2)\\ &= (\sigma^2)^{-\frac{n}{2}}\exp\left(-\frac{1}{2\sigma^2}(\mathbf{y}-\mathbf{X}\beta)^\top(\mathbf{y}-\mathbf{X}\beta)\right)\cdot(\sigma^2)^{-\frac{k}{2}}\exp\left(-\frac{1}{2\sigma^2}\beta^\top\beta\right)\\ &\quad\cdot(\sigma^2)^{-(\alpha_\pi+1)}\exp\left(-\frac{\beta_\pi}{\sigma^2}\right)\\ &= (\sigma^2)^{-\frac{k}{2}}\exp\left(-\frac{\beta_\pi}{\sigma^2}-\frac{1}{2\sigma^2}((\mathbf{y}-\mathbf{X}\beta)^\top(\mathbf{y}-\mathbf{X}\beta)+\beta^\top\beta)\right)(\sigma^2)^{-\left(\alpha_\pi+\frac{n}{2}+1\right)}.\end{aligned}$$

We seek to split the expression within the exponential into two. Roughly following similar steps to those in Fahrmeir, Kneib, Lang, and Marx (2013), we can first do the following:

$$
\begin{aligned}
(\mathbf{y}-\mathbf{X}\beta)^\top(\mathbf{y}-\mathbf{X}\beta)+\beta^\top\beta &= \mathbf{y}^\top\mathbf{y}-\beta^\top\mathbf{X}^\top\mathbf{y}-\mathbf{y}^\top\mathbf{X}\beta+\beta^\top\mathbf{X}^\top\mathbf{X}\beta+\beta^\top\beta \\
&= \beta^\top(\mathbf{X}^\top\mathbf{X}+\mathbb{I}_k)\beta-\mathbf{y}^\top\mathbf{X}\beta-\beta^\top\mathbf{X}^\top\mathbf{y}+\mathbf{y}^\top\mathbf{y} \\
&= \beta^\top(\mathbf{X}^\top\mathbf{X}+\mathbb{I}_k)\beta-\mathbf{y}^\top\mathbf{X}\beta-\beta^\top\mathbf{X}^\top\mathbf{y}+\mathbf{y}^\top\mathbf{X}(\mathbf{X}^\top\mathbf{X}+\mathbb{I}_k)^{-1}\mathbf{X}^\top\mathbf{y} \\
&+\mathbf{y}^\top\mathbf{y}-\mathbf{y}^\top\mathbf{X}(\mathbf{X}^\top\mathbf{X}+\mathbb{I}_k)^{-1}\mathbf{X}^\top\mathbf{y} \\
&= (\beta^\top-\mathbf{y}^\top\mathbf{X}(\mathbf{X}^\top\mathbf{X}+\mathbb{I}_k)^{-1})(\mathbf{X}^\top\mathbf{X}+\mathbb{I}_k)(\beta-(\mathbf{X}^\top\mathbf{X}+\mathbb{I}_k)^{-1}\mathbf{X}^\top\mathbf{y}) \\
&+\mathbf{y}^\top\mathbf{y}-\mathbf{y}^\top\mathbf{X}(\mathbf{X}^\top\mathbf{X}+\mathbb{I}_k)^{-1}\mathbf{X}^\top\mathbf{y}.
\end{aligned}
$$

These final two lines give us the separation we seek. Therefore,

$$
\begin{aligned}
p(\beta,\sigma^2|\mathbf{y},\mathbf{X}) \propto (\sigma^2)^{-\frac{k}{2}}\exp&\left(-\frac{1}{2\sigma^2}\left((\beta-(\mathbf{X}^\top\mathbf{X}+\mathbb{I}_k)^{-1}\mathbf{X}^\top\mathbf{y})^\top(\mathbf{X}^\top\mathbf{X}+\mathbb{I}_k)(\beta-(\mathbf{X}^\top\mathbf{X}+\mathbb{I}_k)^{-1}\mathbf{X}^\top\mathbf{y})\right)\right) \\
&\cdot(\sigma^2)^{-(\alpha_\pi+\frac{n}{2}+1)}\exp\left(-\frac{\beta_\pi+\frac{1}{2}\left(\mathbf{y}^\top\mathbf{y}-\mathbf{y}^\top\mathbf{X}(\mathbf{X}^\top\mathbf{X}+\mathbb{I}_k)^{-1}\mathbf{X}^\top\mathbf{y}\right)}{\sigma^2}\right) \\
&\propto p(\beta|\sigma^2,\mathbf{y}\mathbf{X})\cdot p(\sigma^2|\mathbf{y},\mathbf{X}).
\end{aligned}
$$

These are exactly the expressions we seek that yield

$$
\beta|\sigma^2,\mathbf{X},\mathbf{y}\sim N\left((\mathbf{X}^\top\mathbf{X}+\mathbb{I}_k)^{-1}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{y},\sigma^2(\mathbf{X}^\top\mathbf{X}+\mathbb{I}_k)^{-1}\right)
$$

$$
\sigma^2|\mathbf{X},\mathbf{y}\sim \mathrm{InvGamma}\left(\alpha_\pi+\frac{n}{2},\beta_\pi+\frac{1}{2}\left(\mathbf{y}^\top\mathbf{y}-\mathbf{y}^\top\mathbf{X}(\mathbf{X}^\top\mathbf{X}+\mathbb{I}_k)^{-1}\mathbf{X}^\top\mathbf{y}\right)\right).
$$

# B Computational Costs

## B.1 Derivation of Computational Cost of ED for Fixed $r$

Hold $r$ fixed and let $n\to\infty$. First note that

$$
|\mathbb{T}_{r+1,n}|=\frac{n!}{r!(n-r-1)!}=\frac{n\cdot(n-1)\cdot\ldots\cdot(n-(n-1))}{r!(n-r-1)\cdot(n-r-2)\cdot\ldots\cdot(n-r-1-(n-r-2))}.
$$

The number of terms in the numerator are $n$, while the number of terms involving $n$ in the denominator are $n-r-1$. Looking only at the leading terms (in $n$), we have:

$$\frac{n^n}{n^{n-r-1}} = n^{r+1}.$$

Therefore, we have that

$$|\mathbb{T}_{r+1,n}| = \frac{n!}{r!(n-r-1)!} = O\left(n^{r+1}\right).$$

Note that, since we are keeping $r$ fixed here, the computational cost is (up to coefficients) the same as the original construction of $\mathbb{T}_{r+1,n}$ described in Andrews et al. (2024).

## B.2 Derivation of Computation Cost of ED for Fixed Ratio $\alpha = \frac{r}{n}$

Suppose we fix some ratio $\alpha \equiv \frac{r}{n} < 1$, and let $n \to \infty$ such that $n$ and $r$ are always integers. Now, we have that

$$|\mathbb{T}_{r+1,n}| = \frac{n!}{r!(n-r-1)!} = \frac{n!}{(\alpha n)!(n-\alpha n-1)!}.$$

This can be expanded as

$$\frac{n \cdot (n-1) \cdot \ldots \cdot (n-(n-1))}{(\alpha n) \cdot \ldots \cdot (\alpha n - (\alpha n - 1))(n - \alpha n - 1) \cdot \ldots \cdot (n - \alpha n - 1 - (n - \alpha n - 2))}.$$

Looking at the leading terms in $n$, this yields

$$\frac{n^n}{\alpha^{\alpha n} n^{n-1}} = \frac{n}{\alpha^{\alpha n}} = O\left(\frac{n}{\alpha^{\alpha n}}\right).$$

Note that this is larger than $O(n^{r+1})$ from Section B.1, as $\alpha < 1$. Therefore, this computational cost is even heavier.

Now, consider the original construction of $\mathbb{T}_{r+1,n}$ from Andrews et al. (2024). Mirroring these calculations, we have that

$$|\mathbb{T}_{r+1,n}| = \frac{n!}{(n-r-1)!} = \frac{n!}{(n-\alpha n-1)!}.$$

This can be expanded as

$$\frac{n \cdot (n-1) \cdot \ldots \cdot (n-(n-1))}{(n-\alpha-1) \cdot \ldots \cdot (n-\alpha n-1-(n-\alpha n-2))}.$$

Looking at the leading terms in $n$ yields

$$\frac{n^n}{n^{\alpha n-1}} = O(n^{n-\alpha n+1}).$$

Since $\alpha < 1$, this will grow at an exceedingly large large rate, even more than $O\left(\frac{n}{\alpha^{\alpha n}}\right)$. Therefore, here, this adjustment of $\mathbb{T}_{r+1,n}$ we have made is asymptotically significant.