Online Appendix for: Estimating Heterogeneous Treatment Effects and the Effects of Heterogeneous Treatments with Ensemble Methods

Justin Grimmer * Solomon Messing † Sean J. Westwood ‡

April 14, 2017

1 Constructing the Ensemble via Ensemble Bayesian Model Averaging

A closely related ensemble creation procedure is Ensemble Bayesian Model Averaging (EBMA) (Raftery et al., 2005; Montgomery, Hollenbach and Ward, 2012). EBMA draws on an analogy to Bayesian Model Averaging (BMA) to generate a weighted ensemble to generate predictions. To do this, EBMA utilizes a predictive posterior that is a mixture of component predictions. Given our focus on dichotomous dependent variables, we note that estimates of $E[Y(k)|\boldsymbol{x}]$, $g(k,\boldsymbol{x})$ are also estimates of $P(Y(k)=1|\boldsymbol{x})$. In this case, then, we can write out predictive posterior as,

$$\begin{aligned} p(Y(k)=1|\boldsymbol{x},\boldsymbol{Y}) &= \sum_{m=1}^{M}\int w_m P(Y(k)=1|\boldsymbol{x})p(w_m|\boldsymbol{x},\boldsymbol{Y})dw_m \\ &= \sum_{m=1}^{M}\int w_m g_m(k,\boldsymbol{x})p(w_m|\boldsymbol{x},\boldsymbol{Y})dw_m \end{aligned}$$

And if we assume that weights are point masses at the maximum a posterior (MAP) estimate—as is commonly done in the literature (Raftery et al., 2005; Montgomery, Hollenbach and Ward, 2012)—then this reduces to $p(Y(k)=1|g_1,g_2,\ldots,g_m,\boldsymbol{x}) = \sum_{m=1}^{M} w_m g_m(k,\boldsymbol{x})$. Our estimate of the CATE for treatment conditions k and $k^{'}$ with covariates $\boldsymbol{x}$ is

$$\widehat{\phi}(k,k^{'},\boldsymbol{x}) = \sum_{m=1}^{M} w_m g_m(k,\boldsymbol{x}) - \sum_{m=1}^{M} w_m g_m(k^{'},\boldsymbol{x}). \tag{1.1}$$

*Associate Professor, Department of Political Science, Stanford University; Encina Hall West 616 Serra St., Stanford, CA, 94305

†Director, Data Labs, Pew Research Center 1615 L Street NW, Washington, DC

‡Assistant Professor, Department of Government, Dartmouth College

This is, of course, equivalent to Equation **??**, or the formula used to compute our ensemble for estimating heterogeneous treatment effects previously proposed.

Super learning and EBMA share a methodology focused on accurate combinations of component methods. The two methods differ (as presented here) in how the weights are estimated. In Appendix 1.1 we provide three ways to estimate the weights for EBMA, including the maximum a posteriori (MAP) methods used in the prior literature (Raftery et al., 2005; Montgomery, Hollenbach and Ward, 2012) and two ways to obtain the posterior distribution on the weights—Gibbs sampling and a variational approximation (Jordan et al., 1999). While distinct, the methods presented in Appendix 1.1 share the same intuition as the regression in Step 2 of the super learner algorithm: the out of sample predictions are used to identify the methods that provide accurate out of sample predictions of individual values.

1.1 Estimating Weights for EBMA

In this appendix we describe the posterior distribution for EBMA and provide three ways to estimate the weights. Following prior literature (Raftery et al., 2005; Montgomery, Hollenbach and Ward, 2012) we assume that our predictive posterior is a mixture of the component methods. We will suppose that the weights are drawn from a uniform distribution (or a Dirichlet($\mathbf{1}$)). We will suppose that each observation i is drawn from one of the M component models. Denote the model with a $M \times 1$ indicator vector $\boldsymbol{\tau}_i$ where $\tau_{im} = 1$ when observation i is drawn from model m and all other entries are zero. We will suppose that $\boldsymbol{\tau}_i \sim \text{Multinomial}(\boldsymbol{w})$. Finally, given a realization of $\boldsymbol{\tau}_i$ with $\tau_{im} = 1$ we will suppose that the out of sample prediction for observation i assigned to treatment k $Y(k)_i$ is drawn from a Bernoulli distribution, with chance of success $\pi = g_{im}(k, \boldsymbol{x})$ or $\widehat{Y}_{im}(k)$ in the notation above.

Together this implies the following model

$$\begin{aligned} \boldsymbol{w} &\sim \text{Dirichlet}(\mathbf{1}) \\ \boldsymbol{\tau} &\sim \text{Multinomial}(\boldsymbol{w}) \\ Y_i(k)|\tau_{im} = 1, \boldsymbol{x} &\sim \text{Bernoulli}\left(\widehat{Y}_{im}(k)\right) \end{aligned}$$

and the following posterior distribution for the weights,

$$p(\boldsymbol{w}, \boldsymbol{\tau}|\widehat{\boldsymbol{Y}}, \boldsymbol{x}, \boldsymbol{Y}) \propto \prod_{i=1}^{N}\prod_{m=1}^{M}\left[w_m \times \left(\widehat{Y}_{im}(k)^{Y_i(k)} \times (1 - \widehat{Y}_{im}(k))^{1-Y_i(k)}\right)\right]^{\tau_{im}}$$

We provide three ways to estimate weights with this posterior: an Expectation-Maximization (EM) algorithm, a Gibbs sampler, and a variational approximation. Each derivation is straightforward and available in previous work on estimation in mixture models.

1.2 EM Algorithm

The EM algorithm proceeds in two steps. To begin, initialize estimates for the weights w_m^t where t will index the iteration. Then, we compute the E-step. For each observation i and

each model m compute $\widehat{\tau}_{im}$ which is equal to

$$\widehat{\tau}_{im}^{t} \quad = \quad \frac{w_m^t \left[\widehat{Y}_{im}(k)^{Y_i(k)} \times (1-\widehat{Y}_{im}(k))^{1-Y_i(k)}\right]}{\sum_{l=1}^{M} w_l^t \left[\widehat{Y}_{im}(k)^{Y_i(k)} \times (1-\widehat{Y}_{im}(k))^{1-Y_i(k)}\right]}$$

Computing the M step is straightforward, with the new estimates of the weight for model m, w_m^{t+1} given by

$$w_m^{t+1} \quad \propto \quad 1 + \sum_{i=1}^{N} \widehat{\tau}_{im}^{t}$$

Estimation of the EM-algorithm proceeds until the change in the parameters (or other summary of changes) drops below a predetermined threshold. The EM estimates,

1.3 Gibbs Sampler

A Gibbs sampler provides estimates of the posterior. This facilitates estimation of the uncertainty in the weights when calcuating ATEs and CATEs. Like the EM algorithm, the steps of the Gibbs sampler are well established. Again, initialize weights w_m^t where t tracks the iteration of the sampler. We then sample in two stages. First, we draw $\widehat{\boldsymbol{\tau}}_i^t$,

$$\widehat{\boldsymbol{\tau}}_i^t \quad \sim \quad \text{Multinomial}(1, \boldsymbol{\theta}_i)$$

where $\boldsymbol{\theta}_i = (\theta_{i1}, \theta_{i2}, \ldots, \theta_{iM})$ and

$$\theta_{im}^{t} \quad = \quad \frac{w_m^t \left[\widehat{Y}_{im}(k)^{Y_i(k)} \times (1-\widehat{Y}_{im}(k))^{1-Y_i(k)}\right]}{\sum_{l=1}^{M} w_l^t \left[\widehat{Y}_{im}(k)^{Y_i(k)} \times (1-\widehat{Y}_{im}(k))^{1-Y_i(k)}\right]}$$

Conditional on the drawn indicator vectors, $\widehat{\boldsymbol{\tau}}_i$, we draw the weights, $\boldsymbol{w}^t$,

$$\boldsymbol{w}^{t+1} \quad \sim \quad \text{Dirichlet}(\boldsymbol{\eta})$$

where $\boldsymbol{\eta} = (\eta_1, \eta_2, \ldots, \eta_M)$ and

$$\eta_m \quad = \quad 1 + \sum_{i=1}^{N} \widehat{\tau}_{im}^{t}.$$

After a burn in period and convergence is diagnosed, the sampler is run to approximate the posterior distribution of weights. These weights can then be used to estimate the ATEs and CATEs.

1.4 Variational Approximation

A third method for estimating the posterior on the weights is with a variational approximation, a deterministic method for approximating the full posterior. Variational approximations make a simplifying assumption about the posterior and then finds the member of this simpler, though still general, functional family that provides the closest approximation to the full posterior, as measured by the Kullback-Leibler divergence. We will approximate the posterior distribution for $\boldsymbol{w}$ and $\boldsymbol{\tau}$ with the simpler functional form $q(\boldsymbol{w}, \boldsymbol{t}) = q(\boldsymbol{w})q(\boldsymbol{\tau})$. By the independence assumptions in our data, this implies that we can write the approximating function as $q(\boldsymbol{w})q(\boldsymbol{\tau}) = q(\boldsymbol{w}) \prod_{i=1}^{N} q(\boldsymbol{\tau}_i)$.

Standard arguments for variational approximations of exponential family distributions (see Jordan et al. (1999); Bishop (2006)) leads to the form of the posterior approximations and the update steps. A standard derivation shows that $q(\boldsymbol{\tau}_i)$ is a Multinomial distribution, with parameter $\boldsymbol{\theta}_i$ where

$$\theta_{im} \quad \propto \quad \exp\left(\mathrm{E}[\log w_m] + \log\left[\widehat{Y}_{im}(k)^{Y_i(k)} \times (1 - \widehat{Y}_{im}(k))^{1-Y_i(k)}\right]\right)$$

where $\mathrm{E}[\log w_m]$ is taken over the approximating distribution and dependent on $q(\boldsymbol{w})$. A second standard calculation shows that $q(\boldsymbol{w})$ is a Dirichlet($\boldsymbol{\eta}$) distribution with η_m equal to
,

$$\eta_m = 1 + \sum_{i=1}^{N} \theta_{im}$$

This implies that $\mathrm{E}[\log w_m] = \psi(\eta_m) - \psi\left(\sum_{l=1}^{M} \eta_l\right)$ where $\psi(\cdot)$ is the digamma function. After initializing values of $\boldsymbol{\eta}^t$ the formulas are applied iteratively to update the parameters until the change in the parameters (or change in a lower bound) drops below a sufficient level for convergence. The approximating posterior distribution on the weights with the converged parameter estimates can then be used to reflect posterior uncertainty in the weights.

2 Details on Monte Carlo Simulations

We specify four data generating processes for our Monte Carlo simulations. Each of the Monte Carlo simulations build off the simulations in Imai and Ratkovic (2013).

Monte Carlo 1 For this simulation we have a sparse data generating process with discrete covariates. Specifically, we suppose that for all 2500 observations that,

$$\begin{aligned} Y_i &\sim \text{Bernoulli}(\pi_i) \\ \pi_i &= \Phi\left(\boldsymbol{\beta}\boldsymbol{T}_i + \gamma \boldsymbol{X}_i + \sum_{k=1}^{46}\sum_{j=1}^{2} \eta_{jk} X_{ij} \times T_{ik}\right) \end{aligned} \tag{2.1}$$

where:

- Φ is the standard Normal CDF

- $\boldsymbol{T}_i$ is a 46-element treatment indicator vector. Suppose that $\boldsymbol{p}$ is a 47 element long vector equal to $(\frac{1}{47}, \frac{1}{47}, \ldots, \frac{1}{47})$. Then we draw $T_i \sim \text{Multinomial}(\boldsymbol{p})$ and if all elements of $\boldsymbol{T}_i$ are equal to zero then this corresponds with a control condition.

- $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_{46})$ are coefficients for T_i. We set $\beta_1 = 2$, $\beta_2 = 1, \beta_3 = 0.5, \beta_4 = -1, \beta_5 - 2$. For k from 6 to 46 we draw $\beta_k \sim \text{Uniform}(-0.07, 0.07)$.

- $\boldsymbol{X}_i$ is a 2-element long vector of covariates, with $X_{i1} \sim$ Bernoulli(0.4) and $X_{i2} \sim$ Bernoulli(0.6) .

- $\boldsymbol{\eta}$ is a vector of interaction terms for each treatment and covariate. We suppose that the first five treatments have systematic interactions with the covariates. The remaining eta values are assumed to be drawn from a $\text{Uniform}(-0.1, 0.1)$ distribution.

We then assess the RMSE by generating all possible treatment and covariate combinations and comparing to the actual estimated effects.

Monte Carlo 2 For this simulation we maintain the same basic structure as in Monte Carlo 1, but change the discrete covariates to continuous covariates. Specifically, we suppose that in Equation 2.2 that for each i we generate $a_i \sim \text{Normal}(0, 1)$, $b_i \sim \text{Normal}(0, 1)$, and $c_i \sim \text{Normal}(0, 1)$. We then compute,

- $X_{i1} = \sin(a_i) \times b_i + \cos(c_i) * a_i$

- $X_{i2} = \exp\left(\frac{a_i}{10}\right) \times (b_i^2 + \sin(c_i))$

Because the continuous covariates don't allow us to exactly estimate the treatment effects for every possible valuable, we vary across a range of each variable to compare the actual and estimate treatment effects.

Monte Carlo 3 Monte Carlo 3 provides a *dense* data generating process, with many more treatments having a systematic and large effect—and many more having hetergeneous treatment effects. We suppose again the basic structure

$$\begin{aligned} Y_i &\sim \text{Bernoulli}(\pi_i) \\ \pi_i &= \Phi\left(\boldsymbol{\beta}\boldsymbol{T}_i + \gamma \boldsymbol{X}_i + \sum_{k=1}^{46}\sum_{j=1}^{2} \eta_{jk} X_{ij} \times T_{ik}\right) \end{aligned} \tag{2.2}$$

where:

- Φ is the standard Normal CDF

- $\boldsymbol{T}_i$ is a 46-element treatment indicator vector. Suppose that $\boldsymbol{p}$ is a 47 element long vector equal to $(\frac{1}{47}, \frac{1}{47}, \ldots, \frac{1}{47})$. Then we draw $T_i \sim \text{Multinomial}(\boldsymbol{p})$ and if all elements of $\boldsymbol{T}_i$ are equal to zero then this corresponds with a control condition.

- But now we suppose that many more of the treatments have systematic effects. Specifically we suppose for each k ($k = 1, \ldots, 46$) that we draw $n_k \sim$ Bernoulli(0.5). And then we draw the coefficients,

$$\beta_k \sim \begin{cases} \text{Normal } (-1, 0.1) \text{ If } n_k = 1 \\ \text{Normal } (1, 0.1) \text{ If } n_k = 0 \end{cases}$$

- And we suppose that there are interactions between covariates and the treatments for all the covariate and treatment pairs. We suppose each for each j and k we draw $n_{jk} \sim$ Bernoulli(0.5). And then for each η_{jk} we draw,

$$\eta_{ij} \sim \begin{cases} \text{Uniform}(-1, -0.5) \text{ If } n_{jk} = 1 \\ \text{Uniform}(0.5, 1) \text{ If } n_{jk} = 0 \end{cases}$$

Monte Carlo 4 This Monte Carlo simulation generates the covariates as in Monte Carlo 2 and coefficients as in Monte Carlo 3.

3 Details on Simulation Results

This section provides the results for the individual iterations of the monte carlo simulation. The tables contain the root mean square errors for the estimating the heterogeneous treatment effect across the synthetic data sets.

Table 1: Monte Carlo Simulation 1

Methods	Iter 1	Iter 2	Iter 3	Iter 4	Iter 5
LASSO	0.04	0.05	0.04	0.04	0.05
Elastic Net 0.5	0.04	0.05	0.05	0.05	0.05
Elastic Net 0.25	0.07	0.08	0.07	0.07	0.07
Find It	0.05	0.05	0.05	0.06	0.05
Bayesian GLM	0.08	0.09	0.09	0.08	0.09
BART	0.04	0.05	0.05	0.05	0.05
Random Forest	0.23	0.23	0.25	0.23	0.23
KRLS	0.06	0.07	0.08	0.07	0.07
SVM-SMO	0.11	0.12	0.12	0.13	0.12
Weighted Ensemble	0.04	0.05	0.04	0.05	0.05
Naive Average	0.06	0.07	0.07	0.06	0.07

4 Details of Ensemble Creation

We apply seven methods to estimate the heterogeneous treatment effects.

1) LASSO: We estimate the LASSO using the `glmnet` (Friedman, Hastie and Tibshirani, 2010). We use cross validation to determine the penalty parameter, using mean square

Table 2: Monte Carlo Simulation 2

Methods	Iter 1	Iter 2	Iter 3	Iter 4	Iter 5
LASSO	0.14	0.15	0.12	0.26	0.15
Elastic Net 0.5	0.13	0.15	0.13	0.26	0.15
Elastic Net 0.25	0.17	0.17	0.19	0.24	0.18
Find It	3.03	2.89	2.51	2.47	2.47
Bayesian GLM	0.13	0.14	0.14	0.21	0.15
BART	0.48	0.46	0.5	0.49	0.46
Random Forest	0.33	0.26	0.3	0.3	0.36
KRLS	0.45	0.42	0.38	0.44	0.38
SVM-SMO	0.26	0.27	0.28	0.39	0.38
Weighted Ensemble	0.13	0.13	0.13	0.21	0.15
Naive Average	0.31	0.29	0.24	0.3	0.25

Table 3: Monte Carlo Simulation 3

Methods	Iter 1	Iter 2	Iter 3	Iter 4	Iter 5
LASSO	0.09	0.14	0.17	0.1	0.12
Elastic Net 0.5	0.08	0.14	0.16	0.08	0.11
Elastic Net 0.25	0.07	0.1	0.1	0.08	0.1
Find It	0.15	0.16	0.21	0.14	0.16
Bayesian GLM	0.09	0.13	0.13	0.07	0.1
BART	0.13	0.13	0.12	0.12	0.13
Random Forest	0.26	0.31	0.27	0.25	0.31
KRLS	0.07	0.13	0.14	0.08	0.09
SVM-SMO	0.14	0.2	0.18	0.14	0.18
Weighted Ensemble	0.07	0.11	0.11	0.08	0.1
Naive Average	0.08	0.09	0.13	0.08	0.12

error, and the binomial family. We predict values with the penalty parameter that minimizes the mean square error.

2) Elastic-Net $\alpha = 0.5$: We estimate the elastic net using the `glmnet` (Friedman, Hastie and Tibshirani, 2010). We use cross validation to determine the penalty parameter, using mean square error, and the binomial family. We predict values with the penalty parameter that minimizes the mean square error.

3) Elastic-Net $\alpha = 0.25$: We estimate the elastic net using the `glmnet` (Friedman, Hastie and Tibshirani, 2010). We use cross validation to determine the penalty parameter, using mean square error, and the binomial family. We predict values with the penalty parameter that minimizes the mean square error.

4) Bayesian GLM: We use the logit link in the binomial family in the `arm` package (Gelman and Hill, 2007)

Table 4: Monte Carlo Simulation 4

Methods	Iter 1	Iter 2	Iter 3	Iter 4	Iter 5
LASSO	0.2	0.12	0.14	0.13	0.21
Elastic Net 0.5	0.16	0.12	0.14	0.13	0.17
Elastic Net 0.25	0.14	0.15	0.15	0.14	0.22
Find It	2.81	4	3.18	3.63	2.61
Bayesian GLM	0.13	0.12	0.14	0.12	0.13
BART	0.42	0.51	0.41	0.49	0.5
Random Forest	0.26	0.28	0.27	0.29	0.34
KRLS	0.41	0.44	0.41	0.42	0.43
SVM-SMO	0.27	0.27	0.24	0.32	0.32
Weighted Ensemble	0.12	0.12	0.13	0.12	0.14
Naive Average	0.29	0.38	0.32	0.34	0.28

5) Find It: we use the `FindIt` package (Imai and Ratkovic, 2013). We search for the lambda parameters and use the glmnet option.

6) KRLS: we use the `KRLS` package, using a gaussian kernel with default settings for the σ parameter.

7) SVM: we use the `RWeka` and `rJava` packages using the SMO command, with the polynomial kernels.

8) BART: we use the `BayesTree` package and the `bart` function to the BART models, where we have potential cut values chosen based on the empirical data distribution, with 500 draws for burnin and 1000 draws for posterior summary.

References

Bishop, Christopher. 2006. *Pattern Recognition and Machine Learning*. Springer.

Chen, Jowei. 2010. "Electoral Geography's Effect on Pork Barreling in Legislatures." *American Journal of Political Science* 54(2):301–322.

Friedman, Jerome, Trevor Hastie and Rob Tibshirani. 2010. "Regularization Paths for Generalized Linear Models via Coordinate Descent." *Journal of Statistical Software* 33(1):1.

Gelman, Andrew and Jennifer Hill. 2007. *Data Analysis Using Regression and Multilevel/Hierarchical Models*. Cambridge University Press.

Grimmer, Justin. 2013. *Representational Style: What Legislators Say and Why It Matters*. Cambridge University Press.

Grimmer, Justin, Solomon Messing and Sean J. Westwood. 2012. "How Words and Money Cultivate a Personal Vote: The Effect of Legislator Credit Claiming on Constituent Credit Allocation." *American Political Science Review* 106.

Imai, Kosuke and Marc Ratkovic. 2013. "Estimating Treatment Effect Heterogeneity in Randomized Program Evaluation." *The Annals of Applied Statistics* 7(1):443–470.

Jordan, Michael et al. 1999. "An Introduction to Variational Methods for Graphical Models." *Machine Learning* 37:183–233.

Lazarus, Jeffrey and Shauna Reiley. 2010. "The Electoral Benefits of Distributive Spending." *Political Research Quarterly* 63(2):343–355.

Montgomery, Jacob M., Florian M. Hollenbach and Michael D. Ward. 2012. "Improving Predictions Using Ensemble Bayesian Model Averaging." *Political Analysis* 20(3):271–291.

Raftery, Adrian E., Tilmann Gneiting, Fadoua Balabdaoui and Michael Polakowski. 2005. "Using Bayesian Model Averaging to Calibrate Forecast Ensembles." *Monthly Weather Review* 133:1155–1174.

Shepsle, Kenneth A. et al. 2009. "The Senate Electoral Cycle and Bicameral Appropriations Politics." *American Journal of Political Science* 53(2):343–359.