On Deconstructing Ensemble Models

William D Heavlin, Google, Inc., P O Box 2846, El Granada, CA 94018

October 1, 2015

Abstract

Consider a prediction problem with correlated predictors. In such a case, the best model specification, that is, the best subset of active predictors, can be ambiguous. In spite of this ambiguity, a forecast that informs a high-stakes decision warrants a compact, informative description of the model that produces it. For forecasts based on ensemble models, such descriptions are not straightforward.

Our example considers searches on google.com; each observation consists of one experiment changing the details in how the system responds to user queries. Our predictors measure the changes, relative to a contemporaneous control, of short-term metrics. Our response measures a shift in user behavior observable only after a longer term, also calculated relative to the control.

Our ensemble of models comes from a spike-and-slab regression. We represent each ensemble — each model — by its specification, a vector of booleans denoting the active predictors. For each such model we calculate its goodness of fit statistic. Applying logic regression to predict goodness of fit as a function of the specification booleans, we obtain a metamodel. As a weighted sum of boolean expressions, the metamodel provides a description that is both parsimonious and illuminating.

key words: collinearity, factor analysis, logic regression, model deconstruction, spike-and-slab regression, variance function

1 Introduction

Over the last three decades, statistical learning technology has advanced considerably (Jordan and Mitchell, 2015). Improved computational infrastructure enables model fitting on unprecedented scales, by computing in parallel and over large data volumes (Dean and Ghemawat, 2004). Key advances have come from nonparametric links (Friedman and Tibshirani, 1984), expansive feature sets (Schapire and Singer, 1999), graph-based data structures (Malewicz et al, 2010), richer model forms and flexible bases (Friedman, 1991), and adaptive coefficient estimates (Hastie and Tibshirani, 1993). Further progress derives from combining multiple models — ensemble models (Breiman, 2001; George and McCulloch, 1993; Chipman et al, 2010) — some of which exploit the incremental predictive power of weakly contributing features.

Statistical learning's greatest advances have come from commercial applications — spam filters, recommendation engines, speech recognition, text and handwriting recognition, and fraud detection come to mind (Mitchell, 2009). For these applications, achieving reasonably accurate predictions is sufficient. For some applications, high prediction accuracy is not the only criterion — the statistical model needs in some sense to be understood. Such models can be assessed directly by prediction accuracy and indirectly by prospective experiments.

For a subset of successful statistical models, their very success requires a more detailed description. Models with substantial financial implications require some form of fiscal due diligence; models that shape health care treatment likewise warrant some form of expert validation; models that act in a commercial regime need to conform to appropriate laws and regulations. Finally, models that aspire to scientific insight ought to provide some avenue for scientific scrutiny (Waltz and Buchanan, 2009).

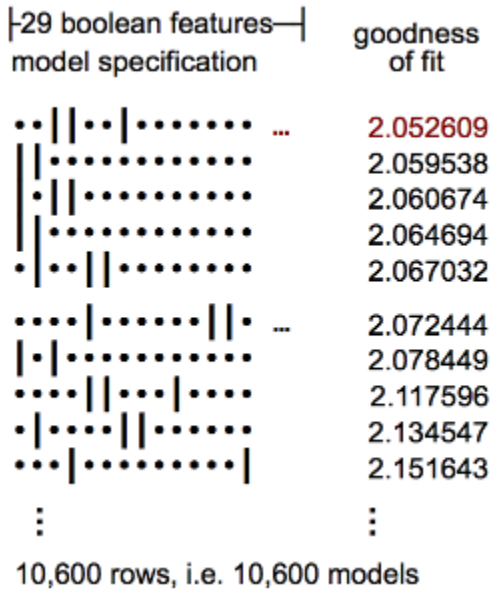

Figure 1: The data structure used for fitting a metamodel. Vertical bars (TRUE) and bullets (FALSE) denote which variables are included in the model that achieves the corresponding goodness of fit value.

The need to better describe statistical models dates from at least the computer experiment literature, and the recognized need to provide summaries and visualizations suitable for human consumption (Fox and Hendler, 2011). Approaches to model description have included restricting models to additive curves and surfaces (Friedman, 1991), mandating constraints like monotonicity (Garcia and Gupta, 2009), aligning model families to compatible narrative forms (Breiman et al, 1993), providing user interfaces for human interaction (Buja et al, 1995), and aggressively reducing complexity to support first-order narratives (Hohnhold et al, 2015; Friedman and Popescu, 2008).

This work represents an additional attempt to describe statistical models. It places special priority on black-box predictive models with measured responses, the so-called supervised learning models. Our general philosophy, which we encapsulate by the term *deconstruction*, seeks validations from within the modeling approach and associated training data. These efforts succeed to the extent that they furnish *narratives* that suffice as critical summaries: deconstruction seeks scientifically founded, compelling narratives that describe models.

Heavlin (2014) focused deconstruction on visualizing the links between training data and coefficient estimates, a formulation tied implicitly to linear models and least squares. Here, the approach is more ambitious, presuming the model in question to result in a black box function of its input features. The principal idea is to form an interpretable predictive metamodel of the associated goodness-of-fit with respect to which features employed. Since the inclusion of any feature is naturally represented by a boolean, we turn to the semantics of logic regression to fit this metamodel.

In section 2, we describe our particular motivating application. In section 3, we introduce metamodels, first by sketching the metamodel data structure, then by providing a formal definition of the three-component metamodel problem. Addressing each of these components forms the body of section 4. Section 5 addresses the intrinsic correlation among models, and section 6 presents the metamodel results for our application. We conclude with section 7, which offers some perspectives based on our example.

2 Application

The application we present consists of a series of search engine result page (SERP) experiments on google.com. Each experiment X_t has a concurrent control group C_t; at the conclusion of experiment X_t, the relative changes observed between X_t and C_t are calculated across a variety of standard metrics. Many of these metrics are calculated based on observed user behavior (clickstream data); Kohavi et al (2009) and Tang et al (2010) describe such clickstream experiments further. The remainder involve opinions gathered of SERPs from third parties (human evaluation data); these estimate relevance from the perspective of a neutral party, a librarian, say. The human evaluation data can be collated, modeled, and imputed to expand the range of queries to which it applies. In this way, each experiment can be represented by the relative changes in metrics it produces, Δx_t. If one can wait a few weeks, one can assess, relative to its control, the relatively persistent or long term changes Δy_t. Our dataset therefore consists of the long-term, observed response Δy_t, associated with the short-term predictor covariates Δx_t, $t = 1, 2, ..., n$. In the particular context of injecting ads into SERPs, Hohnhold et al (2015) report on a dataset of similar construction.

Because the response-covariate pairs $(\Delta y_t, \Delta x_t)$ represent the aggregate statistics of experiment (X_t, C_t), the set $\{(\Delta y_t, \Delta x_t) : t = 1, 2, ..., n\}$ invites an ecological regression model in the sense of King et al (2004). In common with Hohnhold et al (2015) and ecological regression, the covariates are correlated, so the correct model specification is rather ambiguous. As Leamer (1978) observes, and as a close reading of Hohnhold et al reveals, searching for an acceptable model specification is rather an artisan's endeavor: a balanced combination of empirical criteria such as model goodness of fit and more subjective criteria such as clean theoretical interpretation. This paper attempts to provide a framework by which specification searches become more objective, theoretically aligned, and scientifically transparent; an important side-benefit is that such methods produce a supporting narrative that flows more naturally from actual empirical results.

At its most basic level, a specification search tries out different specifications. Table 1 sketches a record of such attempts for the application we explore in this paper. The left-hand columns record as booleans which variables are included in any given specification; the right-hand column, goodness of fit, records some measure of how well each specification performs. In the metamodel approach we investigate here, Table 1 represents our primary data structure. Our metamodel approach takes as its response the goodness-of-fit criterion, and when sufficiently interpretable, provides an overview of how alternate specifications influence model goodness of fit.

As Table 1 suggests, the SERP application has $J = 29$ covariate predictors. Since $2^{29} \approx 5 \times 10^8$ is too large, assessing all possible specifications is not feasible. A sampled subset of the better fitting models is therefore suggested; the specification set we on which we focus has size 10^4 and consists of the last iterates from a spike-and-slab (George and McCullough, 1993) Monte Carlo Markov chain; the calculation uses the R package BoomSpikeSlab of Scott (2015).

To summarize, our application has source data consisting of a corpus of SERP experiments on google.com. Each experiment is represented by aggregate changes of the experiment relative to its contemporaneous control, an ecological regression. Our predictors consists of changes Δx_t observable at the conclusion of the experiment, while our response consists of changes observed in longer term behavior, Δy_t. Exactly which metrics in Δx_t best predict longer term behavior Δy_t is ambiguous; the later MCMC iterates of a spike-and-slab regression, sketched in Table 1, both reflect the ambiguity of the model specification and favor the better specification candidates.

3 Metamodel Formalism

For our SERP application, we limit our scope to linear models. Let y denote an $n \times 1$ response vector and X the $n \times J$ matrix of predictors/covariates/features. The standard

linear model has $y = \mathbb{E}\{y|X\} + \sigma\epsilon$, where $\sigma > 0$ is a positive scalar and ϵ is an $n \times 1$ vector such that $\mathbb{E}\{\epsilon\} = 0$ and $\mathbb{E}\{\epsilon\epsilon^T\} = I_n$, the $n \times n$ identity matrix.

For a given specification or model m, we represent the inclusion of the jth column of X by $m[j] = 1$ and exclusion by $m[j] = 0$. By this representation, m is a J-vector of booleans designating the covariates to be included by model m. Given m, the corresponding mean function is $\mathbb{E}\{y|X, m\} = X[, m]\beta[m]$, with estimate $X[, m]\hat{\beta}[m] \equiv \hat{y}(m)$. A model is good when a model using features $X[, m]$ predict well the observed response y.

With this as background, we define the space of all specifications, $\mathbb{M}$, consisting of all 2^J possible models. Since it is not always feasible to observe all elements $\mathbb{M}$, we denote the observable subset by $\mathbb{M}_0 \subseteq \mathbb{M}$.

For any given specification m, we can calculate the goodness of fit. Define $g : \mathbb{M} \to \mathbb{R}$ as the function that calculates model goodness-of-fit. Let us denote by the $N_0 \times J$ matrix Z whose rows consist of the elements of $\mathbb{M}_0$. The metamodel data structure consists of the response-predictor pair, and $N_0 \times (J + 1)$ matrix $(g(Z), Z)$.

In this paper, a metamodel consists of a function $\hat{g} : \mathbb{M} \to \mathbb{R}$ calculated from $(g(Z), Z)$. We want $\hat{g}$ to have two properties: (1) it approximates g well and (2) it offers a clear interpretation.

4 Metamodel Components

The foregoing invites a few tactical questions — from where does $\mathbb{M}_0$ come?, which choice of g?, how does one obtain $\hat{g}$? — and one strategic question — *why metamodels?* This latter question we address first.

4.1 Why metamodels?

Our target is predictive models, and the goal is to create for such models empirically grounded descriptions, or *narratives*. But why develop narratives about predictive models at all? One answer applies to successful models: sometimes their very success induces extra scrutiny, from executives who becoming curious, doing due diligence, or ensuring regulatory compliance. Another answer matters for candidate models: they need some narrative to articulate their value or worth over incumbent methods, and to describe how (and therefore hint at *why*) they predict better. A third class of answers applies to models under active development: these simply benefit from identifying directions for further improvement in a way that consolidates a strategic consensus. A fourth reason recognizes that scientific review standards want more than good predictions—the expectation is that models be reviewable, reasonably transparent, and objectively founded, all in order to comply with the spirit of the peer review process.

Two themes run through these rationales: (1) the model narratives are intended for human consumption, and (2) they aspire to offer fact-based insights sufficient to satisfy human curiosity.

Ensemble models, which include random forests (Breiman, 2001), spike-and-slab models (George and McCulloch, 1993, 1997; Chipman et al, 2001), and Bayesian additive regression trees (Chipman et al, 2010), all build internal structures not amenable to detailed examination. Therefore we adopt an external point of view, treating such modeling methods as resulting from black-box operations, and contemplate changes in the space of the least common denominator, that of the input features.

4.2 From whence $\mathbb{M}_0$? From MCMCs

When the number of features J is less than 20 or so, it becomes computationally feasible to take $\mathbb{M}_0 = \mathbb{M}$, essentially enumerating all 2^J model specifications.

When J is moderate or large, *and* when there is a measure of feature importance available, one can partition the features into 10 deciles or 20 vintiles. This allows the model space

to be recast as including each decile separately from each other. In the author's personal experience, in the presence of highly important, yet correlated features, the more important deciles are not guaranteed to participate in the best model.

In the present case, we exploit a data structure internal to spike-and-slab, the recorded iterations of the Monte Carlo Markov chain. Each such iterate has coefficients $\{\tilde{\beta}_J : j = 1, 2, ..., J\}$, zero or otherwise, for each feature; the corresponding model specification is defined by the boolean vector $(1\{\tilde{\beta}_j \neq 0\})_j$. This approach to $\mathbb{M}_0$ is available for all MCMC-based methods. The fact that MCMCs produce naturally the metamodel data structure is quite interesting. In some sense, this should be unsurprising: histograms of the MCMC iterations are routinely constructed to estimate posterior distributions; it is therefore plausible that there exist applications that use this MCMC data structure for a non-histogram-based statistical inferences. Here we supplement each MCMC iterate with an additional metric, a measure of goodness of fit, g, for which an approximate model, $\hat{g}$, is constructed.

4.3 Which $g : \mathbb{M} \to \mathbb{R}$? log precision

For continuous responses, goodness-of-fit measures typically have two elements, (a) closeness of the fit and (b) complexity of the model. In the notation of section 3, sum of squared errors, $SSE(m) \equiv (y - \hat{y}(m))^T(y - \hat{y}(m))$ is an example of (a), as if $R^2(m) = 1 - SSE(m)/SSE(0)$, where the model $m = 0$ is understood to include no features but the intercept. In contrast, the mean squared error, $MSE(m) \equiv SSE(m)/(n-1-m^T1)$ now has in the denominator a factor reflecting model complexity. Likewise, the so-called adjusted R^2, $1 - MSE(m)/MSE(0)$, also combines elements (a) and (b).

However, in the face of active and post hoc model selection, it is widely recognized that MSE does not adequately penalize better fitting models for their complexity. Akaike (1974) proposed an information criterion (AIC) that explicitly discounts model complexity, while Schwartz (1978) formulated BIC, a variant that penalizes model complexity more strongly.

In the analysis we present below, the results we report below make use of $g(m) = -\log(MSE(m)) = \log(1/MSE(m))$, that is, log-model precision. Logarithms of variances are common enough in the variance function literature (Davidian and Carroll, 1987); the leading minus sign makes $g(m)$ a higher-is-better function. For two models, m and m' such that $g(m') - g(m) = 0.05$, we can say that m' fits 5 percent better than m.

Although in our opinion AIC and BIC are viable candidates, we choose g =log precision for its more straightforward semantics. We note in passing that AIC, BIC, and log precision differ in how they discount (b), model complexity; on (a), closeness of fit, all do essentially the same thing.

4.4 How to fit $\hat{g}$? logic regression

In the metamodel data structure, the predictors consist of the models and boolean vectors $m \in \mathbb{M}_0$, and the response consists of the calculated goodness-of-fit values $g(m) = -log(MSE(m))$, for $m \in \mathbb{M}_0$.

We want to make statements like this: (i) When we include feature A, then the fit improves by 5 percent. (ii) When we include either feature B or feature C, then the fit improves by 6 percent; in that sense, features B and C can substitute for one another. (iii) When we include both features D and E, then the fit improves by 7 percent.

Statement (i) is recognizable as linear regression term for the boolean feature A, with coefficient 0.05. By analogy, statement (ii) has the same form, except that the boolean feature calculated by the logical expression (B or C) and a coefficient of 0.06. Likewise, statement (iii) is similar, but involves the calculated boolean expression (D *and* E) and coefficient=0.07. In other words, additive logic expressions of the booleans m form a natural basis by which to describe globally, that is, *ceteris paribus*, the form of g.

Motivated by single nucleotide protein (SNP) data, Ruczinski et al (2003) formulated logic regression to build out additive bases such as (i), (ii), and (iii) in a deliberately general

way. The terms, called logic expression trees, are combined by adding them together, one coefficient for each tree. And each term or logic expression tree defines an logical expression of the boolean features, involving *and*s, *or*s, and *not*s; the result in a new boolean term, for which a linear coefficient in an additive model is calculated, one coefficient for one logic expression tree. The process of identifying the set of boolean expression trees is guided by simulated annealing.

Note that including a logical-*not* corresponds to changing the sign of that term's coefficient, and that injecting a logical-*and* is isomorphic to multiplying the two booleans together — the classical means for forming an interaction term for two two-level factors. A logical-*or* reduces to combining these two: $m[,1]$ *or* $m[,2] = not(\, not(m[,1]$ *and* $not(m[,2]))$ $= -(-m[,1] \times -m[,2])$ (modulo a concurrent change in the intercept term).

Practitioners of logic regression typically use the R package LogicReg by Kooperberg and Ruczinski (2015). A rule of thumb is that the *or*-based expressions of logic regression are the most useful. Results discussed in section 6 reinforce this point.

5 Metamodel Correlation

Implicit in the logic regression implementation of Kooperberg and Ruzcinski (2015) is the independence among the observations, that is, that the response-feature pairs that comprise the N_0 rows of $(g,\, Z)$, are independent. For our application this is obviously not true: Consider a four-feature model $m = (1, 1, 1, 0)$, where features $j = 1, 2, 3$ are strong and feature $j = 4$ is of almost no benefit. In this case, the correlation of $g(m = (1, 1, 1, 0))$ and $g(m' = (1, 1, 1, 1))$ is expected to be quite high.

To correct for such correlations, perhaps the simplest involves whitening, a method most often applied to time series. Consider the generic N_0-row data structure $(g,\, Z)$, where g holds the $N_0 \times 1$ response vector and Z holds the $N_0 \times J$ feature matrix. Further, suppose that the covariance $Cov\{g_h,\, g_i\} = \Sigma_{hi}$. Whitening transformations define a new data structure $(g',\, Z')$ such that $g' = \Sigma^{-1/2} g$ and $Z' = \Sigma^{-1/2} Z$.

(Variations to this same basic idea replace $\Sigma^{-1/2}$ with other matrices W such that $WW^T \propto \Sigma^{-1}$; the Choleski decomposition of Σ gives a lower-triangular matrix L such that $LL^T = \Sigma$; further, L is easy to invert, and one can take $W = L^{-1}$.)

For our SERP application, $n = 10^4$, so applying whitening is not straightforward. We divide the issues into three: (1) re-sampling to create replicates of $g(m)$; (2) estimating Σ; and (3) deriving the whitening matrix W_K.

5.1 $g(m)$-replicates

Our basic approach uses the bootstrap. Define an n-vector t_{ab} such that $t_{ab}[i] \in \{0, 1\}$, $\sum t_{ab}[i] = 1$, $\mathbb{E}\{t_{ab}[i]\} = 1/n$, and t_{ab} and $t_{a'b}$ are identically distributed and also independent for $a \neq a'$. From n such t_{ab} construct an n-vector of bootstrap weights $w_b = \sum_a t_{ab}$. We construct B such i.i.d. bootstraps weight vectors, indexing them by $b : \{w_b :\, b = 1, 2, ..., B\}$, and a corresponding diagonal matrix $D_b \equiv diag(w_b)$.

For a given model m, the corresponding mean square error

$$MSE(m) = SSE(m)/(n - 1 - m^T 1)$$

and we take $g(m) = -\log(MSE(m))$, where

$$SSE(m) = y^t \{I - X[,m](X^T[,m]X[,m])^{-1} X[,m]\} y.$$

The corresponding bootstrapped value $g(m, b)$ replaces $SSE(m)$ with

$$SSE(m, b) = y^t D_b \{I - X[,m](X^T[,m] D_b X[,m])^{-1} X[,m]\}\, D_b y.$$

In this way, we build out a matrix G of bootstrapped goodness-of-fit values, replicated goodness-of-fit values, with columns indexed by $m \in \mathbb{M}_0$, rows indexed by $b = 1, 2, ..., B$, and typical value $g(m)$.

5.2 Estimating Σ

Center G : $G_0 \equiv G - 1\,\bar{g}^T$, where $\bar{g}(m) = ave\, G(\bullet, m)$. For two models m and m', their covariance can be estimated by $G_0[\bullet, m]^T G_0[\bullet, m']/(B-1)$. In this sense, $S_B = G_0^T G_0/(B-1)$ is the sample variance-covariance matrix among the models, and estimates the Σ of section 5.

5.3 Whitening matrix W_K

The rank of S_B is $\min\{N_0, B-1\}$; in practice, the rank of S_B is likely to be $B-1$. Further, even if S_B were of full rank N_0, it is computationally rather unattractive to determine the inverse of a $10^4 \times 10^4$ matrix.

S_B has the eigendecomposition $\sum_{k=1}^{K} \lambda_k u_k u_k^T$, where $\{u_k\}$ are eigenvectors and $\{\lambda_k\}$ the ordered eigenvalues. S_B estimates Σ, the variance-covariance matrix of the N_0 models, and likewise $S_B(\lambda_0) \equiv S_B + \lambda_0 I$, for small λ_0, estimates Σ. The term added to S_B to create $S_B(\lambda_0)$ is sometimes called spherical, because of the uniform eigenvalues of the identity matrix, and sometimes called *whitening*, because $\lambda_0 I$ corresponds to assuming some uncorrelated or white noise component in the estimand Σ.

$S_B(\lambda_0)$ has the property of being invertible. Indeed,

$$S_B(\lambda_0) = \sum_{k=1}^{K} \lambda_k u_k u_k^T + \sum_{\ell=1}^{N_0} \lambda_0 u_\ell u_\ell^T = \sum_{k=1}^{K} (\lambda_k + \lambda_0) u_k u_k^T + \sum_{\ell=K+1}^{N_0} \lambda_0 u_\ell u_\ell^T,$$

has the associated inverse

$$S_B(\lambda_0)^{-1} = \sum_{k=1}^{K} \frac{1}{\lambda_0 + \lambda_k} u_k u_k^T + \sum_{\ell=K+1}^{N_0} \frac{1}{\lambda_0} u_\ell u_\ell^T = \frac{1}{\lambda_0}[\sum_{k=1}^{K} \frac{\lambda_0}{\lambda_0 + \lambda_k} u_k u_k^T + \sum_{\ell=K+1}^{N_0} u_\ell u_\ell^T]$$

$$= \frac{1}{\lambda_0}[\sum_{\ell=1}^{N_0} u_\ell u_\ell^T - \sum_{k=1}^{K} \frac{\lambda_k}{\lambda_0 + \lambda_k} u_k u_k^T] = \frac{1}{\lambda_0}[I_{N_0} - U D_\mu U^T], \quad (1)$$

where $D_\mu = diag(\mu)$ and $\mu_k = \lambda_k/(\lambda_0 + \lambda_k)$. In equation (1), the leading factor $1/\lambda_0$ is inessential; we seek only a matrix proportional to $\Sigma^{-1/2}$. From (1), we see the eigenvalues of $\lambda_0 S_B^{-1}$ are $1 - \mu_k$ for $\ell \le K$ and 1 for $\ell > K$, so the corresponding eigenvalues for $\lambda_0^{1/2} S_B^{-1/2}$ are $\sqrt{1-\mu_\ell}$, $\ell \le K$ and 1, otherwise. If we define the quantities $\nu_k = 1 - \sqrt{1-\mu_k}$, the foregoing considerations suggest the whitening operator of this form:

$$W_K \equiv I_{N_0} - U D_\nu U^T \quad (2)$$

This matrix has a convenient form, since the $U D_\nu U^T$ term represents merely subtracting a K-dimensional correction from the previous response-covariate matrix; W_K, which plays the role of $\Sigma^{-1/2}$ in the second paragraph of section 5, thereby filters out the K-dimensional common structure of Σ.

Figure 2 shows the QQ plot of 255 eigenvalues from the singular value decomposition of G_0. Following the scree principle from factor analysis, one can reasonably choose $K = 7$ or $K = 24$. In this paper, we present results based on $K = 24$, that is, using W_{24} as the whitening operator. Our estimate of λ_0 is 1.12×the median eigenvalue; assuming sphericity, the 1.12-factor gives an unbiased estimate of the mean eigenvalue.

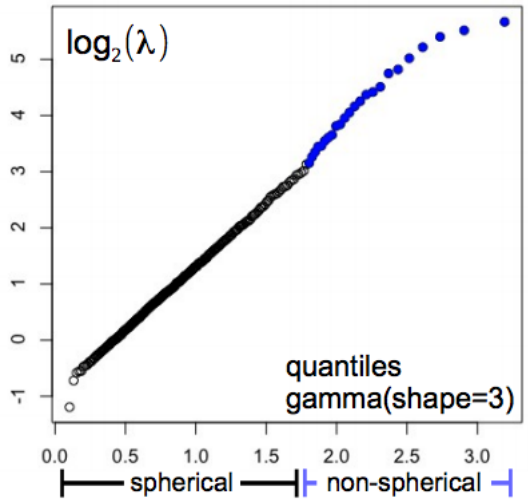

Figure 2: QQ plot of 255 log eigenvalues vs quantiles of the gamma distribution. Blue: the largest 24 eigenvalues.

5.4 Implementation with R

We arrive at this point with the $N_0 \times (J+1)$ metamodel data structure (g, Z), and intent on applying two algorithms to it, whitening and logic regression. From one point of view, the natural order is to apply the whitening operator W_K first, then logic regression to fit $\hat{g}$ second. However, $W_K Z$ is no longer a matrix of booleans; in this specific sense, there is better compatibility with existing R packages to derive the basis from logic regression prior to whitening.

To that end, we settle on this three-step approach: (1) First fit $\hat{g}_1$ by logic regression to (g, Z), recognizing this to be overfitted. (2) Having derived a linear basis Z_1 from step 1, now apply whitening, resulting in the transformed data structure $(W_K g, W_K Z_1)$. (3) Simplify the basis from the first step, Z_1 to one consisting of fewer columns, Z_2, say. This essentially involves including only those columns in $W_K Z_1$ that contribute tangibly to the fit of $W_K g$.

5.5 Analysis of $\hat{g}$-Trees

For a given meta data structure (g, Z), suppose we find a basis Z_0 with J_0 terms (i.e. Z_0 has J_0 columns) that fits g. Denote the R^2_{adj} that corresponds to this fit using basis Z_0 by $R^2(Z_0)$. Consider now the bases Z_{0j}, $j = 1, 2, ..., J_0$, that result from deleting from Z_0 only column j. The contribution of the j-th term can be described reasonably by $\Delta R^2(j) = R^2(Z_0) - R^2(Z_{0j})$. Higher values of $\Delta R^2(j)$ indicate stronger contributions from term j. As non-negative scalars, $\Delta R^2(j)$ quantifies which are the stronger terms. Denote $cols(Z)$ as the set of columns of matrix Z. $R^2(Z_0) - R^2(Z_1)$ can be used to assess the joint impact of $cols(Z_0) \backslash cols(Z_1)$, where $A \backslash B = A \cap !B$ denotes the set difference operator.

In the nomenclature of logic regression, a term with a single linear coefficient is called a *tree*. By analogy with analysis of variance, which uses changes in R^2 to quantify variable importance, we call the calculations based on $\Delta R^2(j)$ an *analysis of trees*.

6 Results

Figure 3 (a) displays the logic regression fit prior to any whitening correction. With the exception of trees 2, 3, and 4, the trees in Figure 3 (a) generally consist of logical-*or* expres-

sions. This aligns both with a logic regression practitioner's rule of thumb and our a priori expectation that certain metrics are likely to substitute for others.

The 11 terms, likely an overfit, are ordered loosely by the magnitudes of their coefficients. The coefficients have natural interpretations: For example, tree 1 has a coefficient of 0.11; this means that including variable X01 increases model precision by 11 percent. Tree 2 has a coefficient of 0.0754; this says that by including both variables X02 and X03, model precision improves by 7.54 percent.

Note that certain features appear in two terms, highlighted by "staples"; to highlight these we rearranged the presentation order to place these terms adjacent to one another. These two conventions — sorting terms by their coefficients' magnitudes, then rearranging them to staple place common terms — suggests an interesting four-block structure:

(E) The first block, the most important, point to a set of metrics that link quality to measured in part by third parties, parties outside of the Google-user dyad, and that participate because of an *economic* incentive.

(R) The third block derives from third party opinions about the SERP *relevance* (relevance in the sense of librarian judgment).

(U) The fourth block derives from *user's* behavior in response to the SERP page, calculated from the clickstream itself.

(R×U) The second block represents some interaction of (R, *relevance*) and (U, *user*) measurements. (And the contribution of term 11 is small, and disregarded.)

The four-block structure is interesting in its own right. Of $J = 29$ metrics, one can focus on just these four meta factors, a number sufficiently compact to create an intriguing narrative.

Figure 3 (b) presents the analysis of trees of the 11 terms presented in Figure 3 (a). As indicated by the darker bars, which correspond to the ΔR^2 for the whole blocks, block (E) shows the greatest importance to fitting $\hat{g}$, followed by blocks (R×U) and (U), with (R) taking fourth place. The strength of block (E) is much greater than anticipated. This finding motivated refining the analysis to the following stage, presented in Figure 4.

Figure 4 revisits the model fit and analysis of Figure 3 after whitening. The number of coefficients (trees) is reduced from 11 to 7, and most of the logic trees involve fewer metrics. In Figure 4 (a), the four-block structure of Figure 3 remains largely in place. The simpler logic trees allows us to consider metric-by-metric deletion—analysis of leaves and trees. This is presented in Figure 4 (b), which shows that factors (E) and (R×U) now have roughly equal importance, (R) continues to contribute precision, while the value derived from (U) is much diminished.

7 Conclusions

For our SERP application, the primary conclusion is the identification of three or four factors relevant for predicting SERP quality: In decreasing order, (E), (R×U), (R), and perhaps (U). Such a simplified structure allows us to speak broadly, yet with known precision, about from where the spike-and-slab ensemble draws its prediction strength. Identifying these three or four factors helps create a useful narrative about an otherwise inscrutable predictive model.

A second finding revolves around factor (E), which comes from third parties acting under economic incentives. Factor (E) is rather analogous to measures from prediction markets, and inherits some of the same elements of controversy (Hahn and Tetlock, 2006). The analysis here shows both the value of such a factor and its strength relative to the other factors. This paper offers yet another case study that empirically validates the value of such data sources.

Regarding methods, we hope that metamodels become more widely applied. Historically, applications of logic regression have involved gene presence and/or activation and similar genomic-based booleans. We believe the potential applications of logic regression to be much broader: quantifying the value of potential predictors in any proposed model, increasing the

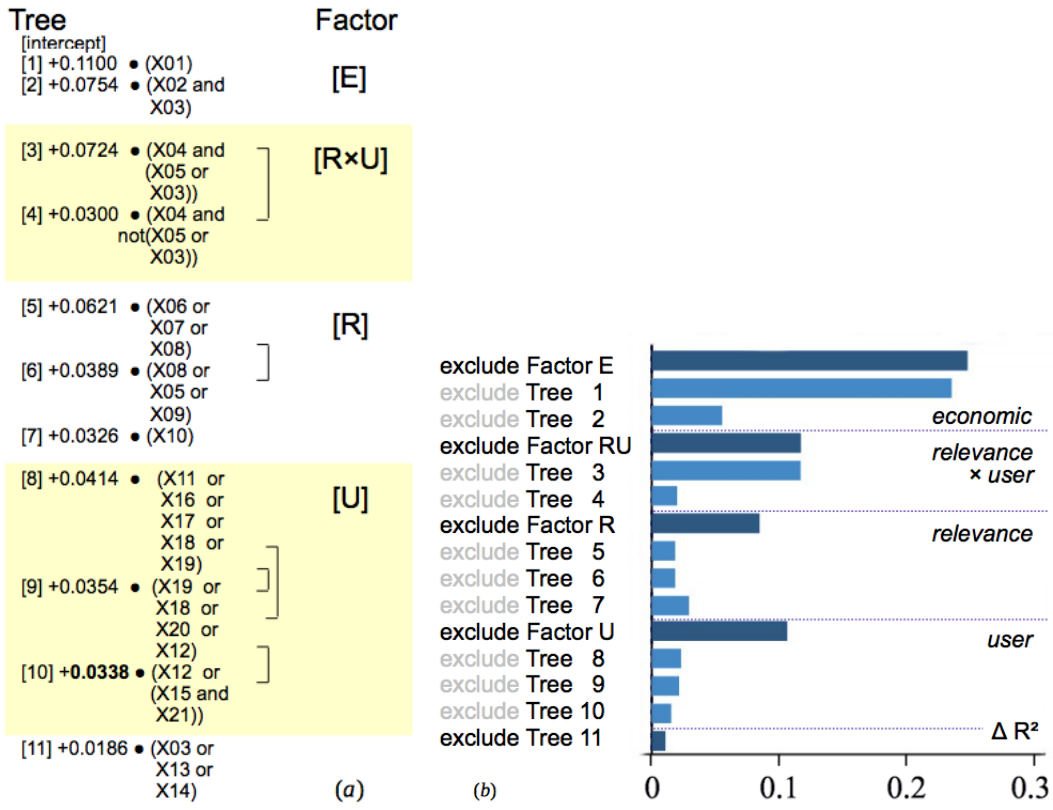

Figure 3: (a) The coefficients of the overfit logic regression model. (b) The associated analysis of trees.

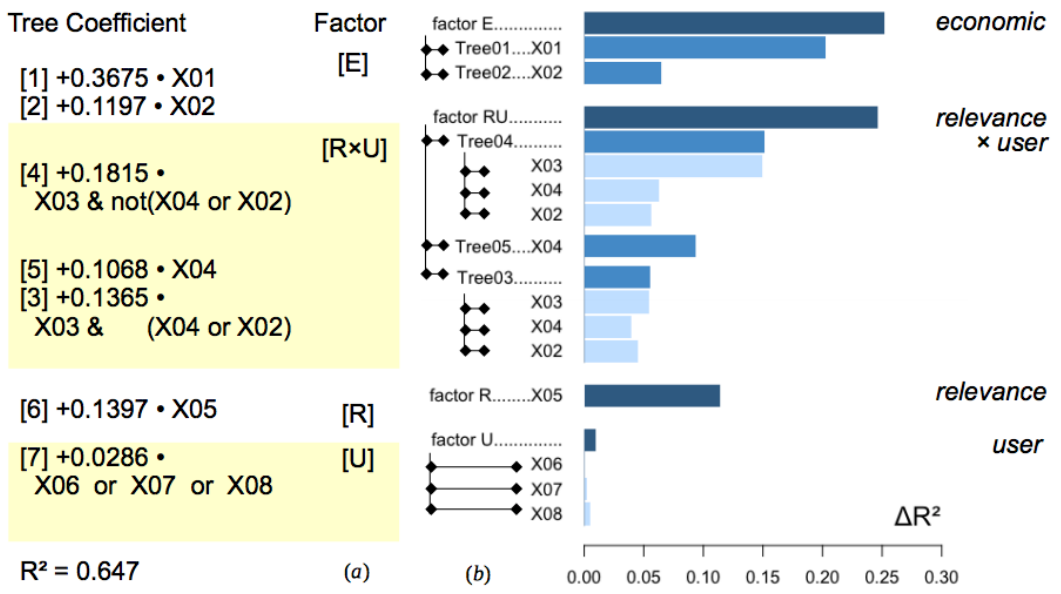

Figure 4: (a) The "whitened" coefficients of the logic regression model, now corrected for the correlations among the models. (b) The associated analysis of leaves and trees.

transparency of black box modeling methods, facilitating the due diligence and scrutiny by fiduciary authorities, and facilitating peer review and scientific publication.

We conclude by noting some open issues. In the current application, the observed ensemble $\mathbb{M}_0$ was freely available as the consequence of the MCMC estimation process of spike-and-slab modeling. On one hand, this offers a new argument for fitting Bayesian models; their algorithms naturally provide a data structure that suppports metamodels. On the other hand, it seems desirable to have an additional or complementary approach suitable for non-Bayes models. What is required is some theory for deciding which model specifications to fit, perhaps by a systematic process from experimental design theory.

A second class of issues involves choosing goodness-of-fit criteria. The current work chooses log precision, yet it seems more than plausible that greater experience may converge to a different criterion, such as AIC or BIC, that penalizes model complexity more aggressively.

References

Akaike, H (1974). "A new look at the statistical model identification," *IEEE Transactions on Automatic Control,* 19: 716-723.

Breiman, L, Friedman, JH, Olshen, RA, and Stone, CJ (1993). *Classification and Regression Trees,* Wadsworth.

Breiman, L (2001). "Random forests," *Machine Learning,* 45: 5-32.

Buja, A, Cook, D, and Swayne, DF (1995). "Interactive high-dimensional data visualization," *Journal of Computational and Graphical Statistics,* 5: 78-99.

Chipman, H, George, EI, and McCulloch, RE (2001). "The practical implementation of Bayesian model selection," *IMS Lecture Notes,* 38.

Chipman, H, George, EI, and McCulloch, RE (2010). "BART: Bayesian additive regression trees," *Annals of Applied Statistics,* 4: 266-298.

Davidian, M and Carroll, RJ (1987). "Variance function estimation," *Journal of the American Statistical Association,* 82: 1079-1091.

Dean, J, and Ghemawat, S (2004). "MapReduce: simplified data processing on large clusters," *ODSI 2004: Sixth Symposium on Operating System Design and Implementation.*

Fox, P and Hendler J (2011). "Changing the equation on scientific data visualization," *Science,* 331: 705-708.

Friedman, JH, and Tibshirani, R,(1984). "The monotone smoothing of scatterplot," *Technometrics, 26:* 243-250.

Friedman, JH (1990). "Multivariate adaptive regression splines," *Annals of Statistics,* 19: 1-67.

Friedman, JH, and Popescu, BE (2008). "Predictive learning via rule ensembles," *Annals of Applied Statistics,* 2: 916-954.

Garcia, E and Gupta, M (2009). "Lattice regression," *Advances in Neural Information Processing Systems,* 592-602.

George, EI and McCulloch, RE (1993). "Variable selection via Gibbs sampling, *Journal of the American Statistical Association,* 88: 881-889.

George, EI and McCulloch, RE (1997). "Approaches for Bayesian variable selection," *Statistica Sinica,* 7: 339-373.

Hahn, RW, and Tetlock, PC, eds. (2006). *Information Markets: a New Way of Making Decisions,* The AEI Press.

Hastie, T, and Tibshirani, R (1993). "Varying-coefficient models" (with discussion), *Journal of the Royal Statistical Society, Series B,* 55: 757-796.

Heavlin, WD (2014). "Deconstruction of effects by exposure dose," presentation at Joint Statistical Meetings, Boston.

Hohnhold, H, O'Brien, D, and Tang, D (2015). "Focusing on the long term: it's good for users and business," *Proceedings of the 21st Conference on Knowledge Discovery and Data Mining, ACM.*

Jordan, MI and Mitchell, TM (2015). "Machine learning: trends, perspectives, and prospects," *Science,* 349: 255-270.

Leamer, EE (1978). *Specification Searches: Ad Hoc Inference on Nonexperimental Data.* Wiley.

King, G, Tanner, MA, Rosen, O (2004). *Ecological Inference: New Methodological Strategies.* Cambridge University Press.

Kohavi, R, Longbotham, R, Sommerfield, D, and Henne, RM (2009). "Controlled experiments on the web: survey and practical guide," *Data Mining and Knowledge Discovery,* 18: 140-181.

Kooperberg, C and Ruczinski, I (2015). Package 'LogicReg,' Comprehensive R Archive Network.

Malewicz, G, Austern, MH, Bik, AJC, Dehnert, JC, Horn, I, Leiser, N, and Czajkowski, G (2010). *Proceedings of the 201 ACM SIGMOD International Conference of Data,* 135-146.

Mitchell, TM (2009). "Mining our reality," *Science,* 326: 1644-1645.

Ruczinski, I, Kooperberg, C, and LeBlanc, M (2003). "Logic regression," *Journal of Computational and Graphical Statistics,* 12: 475-511.

Schapire, RE, and Singer, Y (1999). "Improved boosting algorithms using confidence-rated predictions," *Machine Learning,* 80-91.

Schwarz, GE (1978). "Estimating the dimension of a model," *Annals of Statistics,* 6: 461-464.

Scott, SL (2015). Package 'BoomSpikeSlab,' Comprehensive R Archive Network.

Tang, D, Agrawal, A, O'Brien, D, and Meyer, M (2010). "Overlapping experiment infrastructure: more, better, faster experimentation," *Proceedings of the 16th Conference on Knowledge Discovery and Data Mining, ACM,* 17-26.

Waltz, D and Buchanan, BG (2009). "Automating science," *Science* 324: 43-44.