Loss-Based Estimation with Evolutionary Algorithms and Cross-Validation

David Shilane[*] Richard H. Liang[†]
Sandrine Dudoit[‡]

[*] Division of Biostatistics, School of Public Health, University of California, Berkeley, dshilane@stanford.edu

[†] Department of Statistics, University of California, Berkeley, rhliang@stat.berkeley.edu

[‡] Division of Biostatistics, School of Public Health, University of California, Berkeley, sandrine@stat.berkeley.edu

Loss-Based Estimation with Evolutionary Algorithms and Cross-Validation

David Shilane, Richard H. Liang, and Sandrine Dudoit

Abstract

Many statistical inference methods rely upon selection procedures to estimate a parameter of the joint distribution of explanatory and outcome data, such as the regression function. Within the general framework for loss-based estimation of Dudoit and van der Laan, this project proposes an evolutionary algorithm (EA) as a procedure for risk optimization. We also analyze the size of the parameter space for polynomial regression under an interaction constraints along with constraints on either the polynomial or variable degree.

Loss-Based Estimation with Evolutionary Algorithms and Cross-Validation

David Shilane, Richard H. Liang, and Sandrine Dudoit

November 11, 2007

KEY WORDS
Cross-validation, DSA algorithm, estimator selection, evolutionary algorithms, loss-based estimation, loss function, machine learning, optimization, parameter space, regression, risk, statistical estimation, variable selection.

1 Introduction

Many statistical inference methods rely on selection procedures to estimate a parameter of the joint distribution of the data structure $X = (W, Y)$ that consists of explanatory variables $W = (W_1, \dots, W_J)$, $J \in \mathbb{Z}^+$, and a scalar outcome Y. The *parameter of interest* often takes the form of a functional relationship between the outcome and explanatory variables, as in the regression setting's estimation of $E[Y|W]$, the conditional expectation of the outcome given a set of covariates. In loss-based estimation, the parameter of interest is defined as the risk minimizer for a user-supplied loss function. The quality of candidate estimators within a parameter space may be directly compared according to this or another loss function. We seek to estimate the parameter of interest by generating candidate estimators that minimize a suitably defined empirical risk function over parameter subspaces and then using cross-validation to select an optimal estimator among these candidates. Risk optimization is particularly challenging for high-dimensional estimation problems because it requires searching over large parameter spaces to accommodate general regression functions with possibly higher-order interactions among explanatory variables.

The proposed methodology is motivated by the general road map for statistical loss-based estimation using cross-validation of van der Laan and Dudoit (2003) and Dudoit and van der Laan (2005). Within this framework, Sinisi and van der Laan (2004) introduced a general Deletion/Substitution/Addition (DSA) algorithm for generating candidate estimators that seek to minimize empirical risk over subspaces demarcated by basis size (Section 3.1). However, Wolpert and MacReady (1997) have shown that no single optimization algorithm can competitively solve all problems; therefore, we are interested in generating complementary risk optimization algorithms for use in estimator selection procedures. Within the estimation road map (van der Laan and Dudoit, 2003; Dudoit and van der Laan, 2005), this project seeks to analyze the size of the parameter space for a polynomial regression function in terms of the number of explanatory variables, the maximum number of interacting variables, and either the polynomial degree or the variable degree. It also introduces an evolutionary algorithm (EA) to generate candidate estimators and minimize empirical risk within parameter subspaces. Relying upon V-fold cross-validation to select an optimal parameter subspace, the procedure effectively estimates the parameter of interest in a manner that seeks to minimize true risk.

The proposed EA for estimator selection includes a stochastic mutation mechanism that can be shown to assign all candidate estimators within a parameter subspace to a single communicating class. By doing so, the EA prevents the risk optimization from becoming trapped at local optima. An elitist selection procedure is employed to retain the best candidate estimator among those considered at every generation of the evolution process. When all candidate estimators form a single communicating class and the selection mechanism is elitist, the optimization algorithm converges asymptotically in generation to the global optimum (Fogel, 2005). The EA also includes computational parameters such as the population size and mutation probability that may be varied to produce an arbitrary number of different optimization routines; this allows the user to tailor the procedure to the problem at hand. The proposed algorithm may also be applied to general parameterizations and loss functions. Finally, the EA is modular in its design, so the user may easily substitute alternative components according to situational needs. As a result, the proposed estimator selection procedure is widely applicable in scientific settings such as regression studies in biology and public health. In addition to studying the proposed algorithm's efficacy through simulation experiments, we investigate a diabetes data set to explore the combination of factors contributing to the progression of the disease in human patients.

2 Loss-Based Estimation with Cross-Validation

2.1 The Estimation Road Map

As summarized by Dudoit and van der Laan (2005), we assume that the data X are generated according to a distribution P belonging to a statistical model $\mathcal{M}$, which is a set of possibly non-parametric distributions. Consider a parameter mapping $\Psi : \mathcal{M} \rightarrow \mathcal{F}(\mathcal{D}, \mathcal{R})$ from the model $\mathcal{M}$ into a space $\mathcal{F}(\mathcal{D}, \mathcal{R})$ (or $\mathcal{F}$ in short) of functions with domain $\mathcal{D}$ and range $\mathcal{R}$. A *parameter* is a realization $\psi \equiv \Psi(P) : \mathcal{D} \rightarrow \mathcal{R}$ of Ψ for data generating distribution P. The *parameter space* is defined as $\boldsymbol{\Psi} \equiv \{\psi = \Psi(P) : P \in \mathcal{M}\} \subseteq \mathcal{F}$. Given a random variable X and a parameter value ψ, a *loss function* $L(X, \psi) : (X, \psi) \rightarrow \mathbb{R}$ is a measure of distance between the parameter and the data. Given a data generating distribution P, we can summarize loss in terms of a corresponding *risk function*, which is defined as the expected value of the loss function:

$$\Theta(\psi, P) \equiv \int L(x, \psi) dP(x) = E[L(X, \psi)]; \qquad P \in \mathcal{M}. \tag{1}$$

It is assumed that the loss function is specified such that the parameter of interest ψ minimizes the risk function. For instance, in regression, the parameter of interest is the regression function $\psi(W) = E[Y|W]$, which minimizes risk for the L_2 loss function $L(X, \psi) = (Y - \psi(W))^2$. We can then define the optimal risk over the parameter space as:

$$\theta \equiv \Theta(\psi, P) = \min_{\psi' \in \boldsymbol{\Psi}} \Theta(\psi', P) = \min_{\psi' \in \boldsymbol{\Psi}} \int L(x, \psi') dP(x). \tag{2}$$

Given $\mathcal{X}_n$, a set of n independent, identically distributed (i.i.d.) observations of data $X_i = (W_i, Y_i)$, $i \in \{1, \ldots, n\}$, from (typically unknown) distribution P, our goal is to select an estimator $\psi_n = \hat{\Psi}(P_n)$ based upon the empirical distribution P_n of the data $\mathcal{X}_n$ in a manner that seeks to minimize true risk. The empirical risk of a parameter value ψ is defined as:

$$\Theta(\psi, P_n) \equiv \int L(x, \psi) dP_n(x) = \frac{1}{n} \sum_{i=1}^{n} L(X_i, \psi). \tag{3}$$

The general road map for loss-based estimation (Dudoit et al., 2003; van der Laan and Dudoit, 2003; Dudoit and van der Laan, 2005) contains three steps:

1. *Define the parameter of interest.* This parameter is the value that minimizes risk for a user-supplied loss function.

2. *Generate candidate estimators.* The parameter space is divided based upon a sieve of increasing dimensionality into subspaces whose union approximates the complete parameter space. Within each subspace, a candidate estimator is chosen to minimize empirical risk.

3. *Apply cross-validation.* Select the optimal estimator among the candidates produced in Step 2 using cross-validation.

2.2 Estimator Selection Procedure

It is often the case that an estimator ψ_n seeking to minimize *empirical risk* $\Theta(\psi_n, P_n)$ on the learning set $\mathcal{X}_n$ may over-fit at the expense of predictive value. The general road map for loss-based estimation (Dudoit and van der Laan, 2005) resolves the issue of over-fitting by first selecting a subspace of the parameter space using a cross-validation procedure and then selecting the optimal estimator to minimize empirical risk on the learning set over this subspace. We may apply the estimation road map in the following estimator selection procedure:

1. The user specifies a set $\mathcal{K}$ of candidate subsets of the parameter space to be searched. By default, the subspaces are indexed by $\mathcal{K} = \{1, \ldots, K\}$, with $K \in \mathbb{Z}^+$.

2. The user specifies $V \in \mathbb{Z}^+$, the number of folds to use in cross-validation. Although we employ V-fold cross-validation in this procedure, it should be noted that alternative cross-validation procedures such as Monte-Carlo cross-validation may be substituted (Dudoit and van der Laan, 2005).

3. Data points $X_i = (W_i, Y_i)$, $i \in \{1, \ldots, n\}$, from the learning set $\mathcal{X}_n$ are randomly assigned to a class in $\{1, \ldots, V\}$ such that each class contains an approximately equal number of observations. Let $Q = (q_1, \ldots, q_n)$ refer to the data's class assignments.

4. For each fold $v \in \{1, \ldots, V\}$:

   (a) Assign data points to the training set:

   $$\mathcal{T}_n(v) = \{X_i : q_i \neq v\}\,. \tag{4}$$

   (b) Assign data points to the validation set:

   $$\mathcal{V}_n(v) = \{X_i : q_i = v\}\,. \tag{5}$$

   (c) For each candidate subspace $k \in \mathcal{K}$:

      i. Search within the subspace for the candidate estimator $\psi_{k,n}$ that minimizes empirical risk on the training set $\mathcal{T}_n(v)$.

      ii. Compute the validation set risk $\Theta\left(\psi_{k,n}, P_n^{\mathcal{V}_n(v)}\right)$, where $P_n^{\mathcal{V}_n(v)}$ represents the empirical distribution on the validation set $\mathcal{V}_n(v)$.

5. Calculate the *mean cross-validated risk* for each subspace and store it in the vector

$$\Theta^{CV} \equiv \left(\Theta_1^{CV}, \ldots, \Theta_K^{CV}\right) = \left(\frac{1}{V}\sum_{v=1}^{V} \Theta\left(\psi_{1,n}, P_n^{\mathcal{V}(v)}\right), \ldots, \frac{1}{V}\sum_{v=1}^{V} \Theta\left(\psi_{K,n}, P_n^{\mathcal{V}(v)}\right)\right). \tag{6}$$

6. Select the subspace that minimizes mean cross-validated risk:

$$k_n = \underset{k \in \{1,\ldots,K\}}{\text{argmin}}\ \Theta_k^{CV}. \tag{7}$$

7. Finally, search within the parameter subspace k_n for the estimate ψ_n minimizing empirical risk $\Theta\left(\psi_n, P_n\right)$ on the learning set data $\mathcal{X}_n$.

Steps 4(c)i and 7 of the above procedure rely upon searching a parameter subspace for the estimator that minimizes empirical risk when applied to the specified (training or learning) data set. An exhaustive search of the parameter subspace may be employed when doing so is computationally tractable. However, in estimation problems over general regression functions with possibly higher-order interactions, the parameter space can grow complex and large (Section 3.3) for even a moderate number of explanatory variables. We therefore require a *search algorithm* to minimize risk within a parameter subspace in the allotted computational time. The DSA (Sinisi and van der Laan, 2004) is one candidate search algorithm; Section 4 will introduce a class of evolutionary algorithms as an alternative procedure for risk minimization.

3 The Parameter Space for Polynomial Regression

3.1 Parametrization

Given a parameter of interest ψ and a suitable loss function $L(X, \psi)$, we seek to characterize the set of candidate estimators to be searched by an estimator selection procedure. In regression, the *parameter space* can be defined by the class of *basis functions* for the explanatory variables (e.g. polynomial functions or set indicators), the choice of the *link function* (e.g. logit or probit) mapping the selected basis functions to the outcome variable, and the *constraints* that limit the way in which explanatory variables may interact. Much as in Sinisi and van der Laan (2004), the proposed estimator selection procedure may be applied to any estimation setting, including but not limited to robust and weighted regression, censored data structures, and generalized linear models for any choice of link function $h : \mathbb{R} \to \mathbb{R}$. Because of its approximation capabilities (Stoll, 2000), we will focus on the parameter space consisting of the set of polynomial combinations of the explanatory variables with real-valued coefficients. In this parametrization, the set of basis functions Φ consists of all monomial functions ϕ that can be expressed in terms of an exponent vector $\mathbf{d} = (d_1, \ldots, d_J)$ as

$$\phi = W^{\mathbf{d}} \equiv W_1^{d_1} \cdots W_J^{d_J}. \tag{8}$$

A parameter value ψ may be specified in terms of a subset of basis functions:

$$\varphi = \{\phi_{i_1}, \ldots, \phi_{i_k}\} \subseteq \Phi, \tag{9}$$

with cardinality $|\varphi| = k$ referred to as the *basis size*. Given the link function h, a size k set of basis functions φ, and a $(k+1)$-dimensional vector β of real-valued coefficients, one has a regression function of the form

$$\psi = h\left(\beta_0 + \sum_{i:\phi_i \in \varphi} \beta_i \phi_i\right). \tag{10}$$

In seeking an estimate ψ_n that minimizes true risk, we will first search for the optimal set of basis functions φ_n and then subsequently seek an optimal estimate β_n of β. Estimating β given φ_n is a standard regression problem that is solved in a closed form for linear regression and with numeric optimization methods for non-linear regression. Given φ_n and β_n, the estimate ψ_n is defined as:

$$\psi_n = h\left(\beta_{0_n} + \sum_{i:\phi_i \in \varphi_n} \beta_{i_n} \phi_i\right). \tag{11}$$

3.2 Constraints

At the user's discretion, constraints may be imposed on the set of basis functions Φ. These constraints often take the form of limits on the interaction order and the polynomial or variable degree. The interaction order constraint, which limits the number of variables that interact in a basis function, may be stated as:

$$1 \leq \sum_{j=1}^{J} \mathbf{1}\{d_j > 0\} \leq S; \qquad S \in \mathbb{Z}^+. \tag{12}$$

A polynomial degree constraint may be phrased in the form:

$$1 \leq \sum_{j=1}^{J} d_j \leq D; \qquad d_j \in \mathbb{Z}^+,\ j \in \{1, \ldots, J\}. \tag{13}$$

A variable degree constraint, which may be used as an alternative to the polynomial degree constraint, allows each component variable in a basis function to independently attain a maximum degree D_0. The variable degree constraint is:

$$0 \leq d_j \leq D_0; \qquad j \in \{1, \ldots, J\} \text{ with } \sum_{j=1}^{J} d_j \geq 1. \tag{14}$$

Although the constraints (12), (13), and (14) are not required, they allow the researcher to restrict attention to a particular subset of the class of chosen basis functions. By default, the interaction order S can be no greater than $\min(J, D)$ under constraint (13) and is limited to J under (14). The extreme cases $S = 1$ and either $S = \min(J, D)$ or $S = J$ correspond, respectively, to constraints allowing no interactions and interactions of any order.

3.3 Size of the Parameter Space

Under the above formulation, the set of basis functions Φ consists of all unique monomials ϕ corresponding to an exponent vector $\mathbf{d} = (d_1, \ldots, d_J)$ satisfying the interaction order constraint (12) and either the polynomial degree constraint (13) or the variable degree constraint (14). We can then determine the number of basis functions I using combinatorial arguments. When subject to the polynomial degree constraint (13), the value of I is given by:

$$I = \sum_{s=1}^{S} \binom{J}{s} \left[1 + \sum_{d=s+1}^{D} \sum_{k=1}^{\min(s, d-s)} \binom{s}{k} \binom{d-s-1}{k-1} \right]. \tag{15}$$

The sum from $s = 1$ to S represents all possible values for the number of variables to appear in a monomial. Once this is chosen, the $\binom{J}{s}$ term provides the number of ways to choose s variables $W_{j_1}, \ldots, W_{j_s}$ from the J total variables. Given $W_{j_1}, \ldots, W_{j_s}$, we turn our attention to the number of valid monomials using exactly all of these variables. Because the multilinear term $W_{j_1} W_{j_2} \cdots W_{j_s}$ is always included, the number of valid monomials is 1 plus the number of higher-order monomials. The sum from $d = s + 1$ to D represents the choice of higher order polynomial degree for the monomial in the set $\{s + 1, \ldots, D\}$. Once we have chosen a degree, we must distribute it over all of the selected variables. Knowing that each of these s variables must have a degree of at least one, this leaves $d - s$ powers to distribute to s variables. The sum over k chooses the number of monomial variables allocated a higher degree. A total of $\binom{s}{k}$ combinations of variables may receive higher power. Finally, by a well-known result in combinatorics, the number of ways to distribute at least one degree to each of the k variables selected to receive more power is $\binom{d-s-1}{k-1}$.

As an example, suppose we assign $S = 1$ and impose constraint (12) to preclude all variable interactions. In this case, the summations over s and k involve only a single iteration, leaving us with $\binom{J}{1} \left[1 + \sum_{d=2}^{D} \binom{1}{1} \binom{d-2}{0} \right] =$ $\binom{J}{1}[1 + (D - 1)] = JD$. That is, when no variables may interact, the set of possible basis functions consists of all choices of a single variable W_j, $j \in \{1, \ldots, J\}$, raised to a power $d_j \in \{1, \ldots, D\}$.

When the parameter space is instead restricted by the variable degree constraint (14), the number of basis functions is I_0:

$$I_0 = \sum_{s=1}^{S} \binom{J}{s} D_0^s. \tag{16}$$

In (16), all allowed basis functions may be categorized by the number of interacting variables $s \in \{1, \ldots, S\}$. Given s, a total of $\binom{J}{s}$ combinations of variables may be selected to interact. Each of these interacting variables must have a positive degree in the set $\{1, \ldots, D_0\}$ that may be assigned independently.

For a given regression setting subject to the above constraints, the number of basis functions I (15) (or I_0 (16)), provides an indication of the problem's size and may be used to guide the selection of computational parameters in the risk optimization algorithm. Because each basis function may be included in or excluded from a regression function, there are 2^I (or 2^{I_0}) possible estimators among which to choose. Within a size k parameter subspace, exactly k basis functions are included in any estimator, so the subspace contains $\binom{I}{k}$ or $\binom{I_0}{k}$ estimators, respectively.

We can analyze the size of the parameter space by providing upper and lower bounds on the number of basis functions. Using the notation of Cormen et al. (1990), the functions Ω and O may be used to specify asymptotic lower and upper bounds on the order of the number of basis functions I or I_0 in terms of the interaction order bound S and the polynomial degree bound D or variable degree bound D_0, respectively. (Please refer to the Appendix for details.) When subject to the interaction order constraint (12) and the polynomial degree constraint (13), the value of I is bounded below by a function of order $\Omega\left(2^S\right)$. When the variable degree constraint (14) is employed subject to the interaction order constraint (12), the value of I_0 is bounded below by a function that is $\Omega\left(D_0^S\right)$. When no interaction order constraint is imposed, then $S = \min(J, D)$ under the constraint (13), and so the lower bound on I is $\Omega\left(2^{\min(J,D)}\right)$. Likewise, when no interaction order constraint is imposed under the constraint (14), then $S = J$, and I_0 is bounded below by $\Omega\left(D_0^J\right)$. Because all interactions are allowed when $S = J$, the summation for I_0 in (16) may be expressed as a polynomial in D_0 of degree J. Therefore, when $S = J$, I_0 is both $\Omega\left(D_0^J\right)$ and $O\left(D_0^J\right)$, which jointly imply a tight bound on I_0. The latter bound may also be used as a loose upper bound when $S < J$. Furthermore, because any basis function allowed under the constraints (12) and (13) is also permitted under (12) and (14) when $D = D_0$, this upper bound on I_0 is also a trivial upper bound on I. Therefore, I is $O\left(D^J\right)$. Finally, because the size of the parameter space is 2^I or 2^{I_0}, these quantities are respectively bounded below by functions of order $\Omega\left(2^{2^S}\right)$ and $\Omega\left(2^{D_0^S}\right)$. Likewise, upper bounds of $O\left(2^{D^J}\right)$ and $O\left(2^{D_0^J}\right)$ may also be established, where the former is a trivial bound, and the latter is a loose bound that is only tight in the extreme case of $S = J$. These results are proved in the Appendix and summarized in Table 1.

	Polynomial Degree Constraint (13)	Variable Degree Constraint (14)
Upper Bound	$O\left(2^{D^J}\right)$	$O\left(2^{D_0^J}\right)$
Lower Bound	$\Omega\left(2^{2^S}\right)$	$\Omega\left(2^{D_0^S}\right)$

Table 1: Size of the parameter space under the interaction order constraint (12) and either the polynomial degree constraint (13) or the variable degree constraint (14).

Because the size of the parameter space is at least of a doubly exponential order of the number of variables J and the polynomial degree bound D or variable degree bound D_0 when the interaction order is not constrained, even moderate degree constraints imposed on a small number of variables may result in an intractable parameter space to search. In this setting, significant computation may be required to obtain

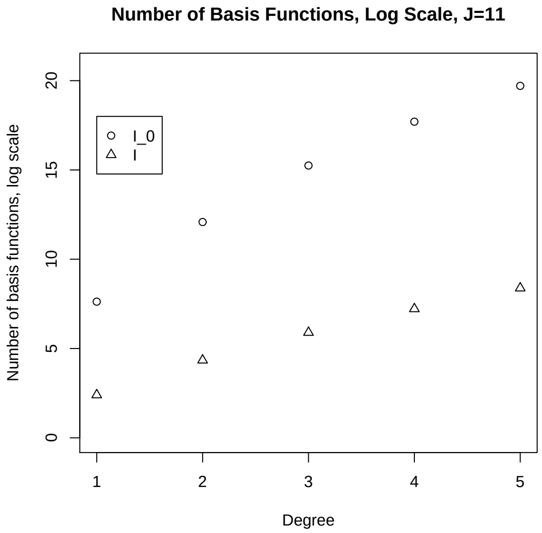

Figure 1: The natural logarithm of the numbers of basis functions I and I_0 as a function of the polynomial degree bound D and the variable degree bound D_0, respectively, for $J = 11$ variables and no interaction order constraint. For the constraint (13), we have $S = \min(J, D)$, and for the constraint (14), this value is $S = J$.

a reliable estimate of the parameter of interest. Figure 1 depicts the growth of $\log(I)$ and $\log(I_0)$ as the polynomial degree bound D and the variable degree bound D_0 increase in an estimation setting with $J = 11$ variables and no interaction order constraint; i.e. $S = \min(J, D)$ for the constraint (13), and $S = J$ for the constraint (14). The approximately linear growth on the logarithmic scale confirms that the values I and I_0 are exponential functions of their respective degree bounds. The value I is consistently smaller than I_0 because the polynomial degree constraint (13) restricts the parameter space to a subset of that specified by the variable degree constraint (14). Furthermore, the maximum value of S under the constraint (14) is J, whereas S is constrained to $\min(J, D) \leq J$ under the constraint (13). Therefore, for a fixed level of the interaction order bound S, the polynomial constraint (13) always results in a smaller parameter space than that specified by (14) when $D = D_0$.

Figure 2 plots the growth of $\log(I_0)$ as a function of the interaction order bound S for an estimation setting including $J = 11$ variables and degree bound $D_0 = 5$, which corresponds to Model 5 presented in Section 5. In practice, S is often chosen according to scientific insight for the problem at hand. However, the choice of S can also be used to effectively prune the parameter space to a manageable size.

4 Evolutionary Algorithms as Risk Optimization Procedures

Evolutionary Algorithms (EA) comprise a class of stochastic optimization algorithms that generate candidate solutions via a process similar to biological evolution (Bäck, 1996; Fogel, 2005). Although Wolpert and MacReady (1997) have shown that no single algorithm can best solve all optimization problems, EAs are sufficiently flexible to be applied to many types of problems, perform reasonably well in a variety of settings, and provide few difficulties in software development (Fogel, 2005). Furthermore, we can immediately generate many candidate algorithms for comparison by varying the proposed EA's computational parameters. In this section, we will familiarize the reader with EA methodology, elucidate the underlying stochastic nature,

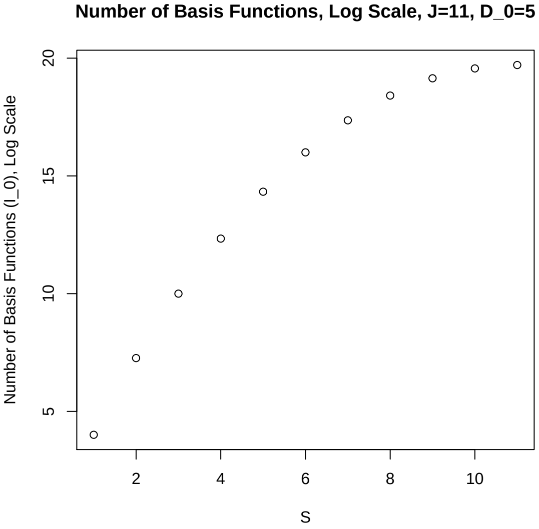

Figure 2: The natural logarithm of the number of basis functions I_0 as a function of the interaction order bound S for $J = 11$ variables with the variable degree constraint (14) specified by $D_0 = 5$.

and incorporate this class of algorithms as a search procedure in estimator selection (as in Section 2.2).

An EA seeks to optimize a real-valued *objective* (or *fitness*) function. When generating candidate estimators ψ_n, our objective is to minimize the risk $\Theta(\psi_n, P)$ over parameter subspaces, where, depending upon the context, P denotes the empirical distribution for either the full learning set or a cross-validation training set. A candidate optimum of this fitness function is given by an *individual* consisting of a *genotype* vector $\mathbf{e} = (e_1, \ldots, e_{kJ}) \in (\mathbb{R}^+)^{kJ}$ and a corresponding *phenotype* vector:

$$\mathbf{d} = \mathbf{d}(\mathbf{e}) \equiv \left(\left[d(e_1, \ldots, e_J) \right], \left[d(e_{J+1}, \ldots, de_{2J}) \right], \ldots, \left[d(e_{(k-1)J+1}, \ldots, e_{kJ}) \right] \right) \in \left(\mathbb{R}^+ \right)^{kJ}. \qquad (17)$$

Each block of J phenotypic components $\left[d(e_{(j-1)J+1}), \ldots, d(e_{jJ}) \right]$ serves as the exponent vector of a particular basis function, and the k basis functions collectively specify a subset of basis functions of the form (9) that map to a candidate optimum ψ_n. An individual's fitness is given by the risk $\Theta(\psi_n, P)$ of its associated estimate ψ_n. Although the user may proceed by directly specifying a phenotype vector, a data structure including both a genotype and a phenotype allows for a greater variety of evolutionary information to be stored in an individual. For instance, a continuous genotype may be used to break ties in the phenotype when an interaction order constraint is imposed. In this setting, the elements of the genotype vector $\mathbf{e}$ may belong to the positive real numbers $\mathbb{R}^+$, the elements of the phenotype vector $\mathbf{d}$ may be limited to the set of positive integers $\mathbb{Z}^+$, and any function with domain $\mathbb{R}^+$ and range $\mathbb{Z}^+$ may be used to map from an individual's genotype to its phenotype. In the procedure of Section 4.1, we choose this function according to the selected degree and interaction order constraints. When an interaction order constraint is imposed, a continuous genotype structure allows some genes to maintain a genotype while remaining dormant in terms of phenotype. In this scenario, if a gene whose phenotype previously interacted mutates, a gene of dormant phenotype may immediately take its place as one of the at most S interacting phenotypic components of a given basis function.

Starting from a random *initial population*, EAs typically generate *populations* of individuals in *generations* of *offspring* created from existing *parents* via iterations of *evolutionary mechanisms*. Although other mech-

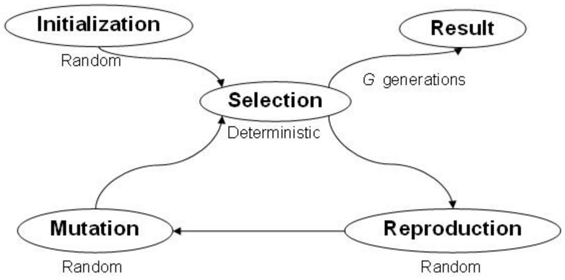

Figure 3: Schematic diagram for the proposed EA risk optimization procedure.

anisms may be used, each generation of the proposed EA consists of a reproduction, mutation, and selection phase, and these mechanisms collectively create and evaluate new individuals for quality in terms of fitness. After allowing the population to evolve for $G \in \mathbb{Z}^+$ generations, the individual with optimum observed fitness is retained as the algorithm's *result*, which specifies an estimate ψ_n with an associated risk given by $\Theta(\psi_n, P)$.

4.1 Proposed EA

The following EA is used to optimize risk within a parameter subspace of size $k \in \mathbb{Z}^+$ based on monomial basis functions of the J explanatory variables under the interaction order constraint (12) and either the polynomial degree constraint (13) or the variable degree constraint (14). A schematic diagram of this algorithm is depicted in Figure 3. Each step of the algorithm is first summarized here and then further elucidated below.

1. **Initialization**: Create a population of candidate solutions.
2. **Evolution**: Create a new population from the existing population. The three evolutionary mechanisms are performed in order at each of G generations:
   (a) **Selection**: Rank individuals according to fitness and select a proportion of the population to survive and mate.
   (b) **Reproduction**: Pair selected individuals and create offspring via a random combination of parental genotypes.
   (c) **Mutation**: Alter offspring genotypes according to a stochastic mutation process.
3. **Result**: After G generations of evolution, perform the selection mechanism on the resulting population and return the individual with optimal fitness as the EA's result. This individual's phenotype vector $\mathbf{d}$ has an associated subset of basis functions φ_n that correspond to an estimate ψ_n according to (11).

Mapping from Genes to Estimates: The following steps may be used to map from a genotype vector $\mathbf{e}$ to a candidate estimate ψ_n:

1. Each individual contains k blocks of J genes. Rank gene values within each block in decreasing order in a vector $\mathbf{r}$:
$$\mathbf{r} = \left((r_1, \ldots, r_J), (r_{J+1}, \ldots, r_{2J}), \ldots, (r_{(k-1)J+1}, \ldots, r_{kJ})\right) \tag{18}$$
so that each block $(r_{(j-1)J+1}, \ldots, r_{jJ})$, $j \in \{1, \ldots, k\}$, contains a permutation of $\{1, \ldots, J\}$. When the interaction order constraint (12) is employed, this ranking is used to select the basis function corresponding to the (at most) S genes of largest value in each block.

2. Calculate the individual's phenotype vector $\mathbf{d}$ from its genotype $\mathbf{e}$. This computation differs depending on which degree constraint is used. However, this step is the only location within the EA for which the procedure differs depending upon the constraint. This is an additional advantage of a data structure that includes both a genotype and a phenotype.

    - For the polynomial degree constraint (13): Within each block of J genes of the genotype vector $\mathbf{e}$, begin with the variable of highest rank (corresponding to the largest gene value). Assign the minimum of the floor of this gene value and the remaining polynomial degree as the variable's phenotype within the block. Repeat this procedure on each variable in order of its gene rank until the monomial is of degree D or the interaction order constraint (12) is binding.
    - For the variable degree constraint (14): Within each block of J genes, compute the phenotype by assigning the floor of the genotype for each of the at most S interacting variables. (Because all gene values are within the interval $(0, D_0+1)$, the floor function ensures that no phenotype exceeds D_0.) All non-interacting variables receive phenotype 0. This computation may be performed via the following equation:

    $$\mathbf{d} = \lfloor \mathbf{e} \rfloor \mathbf{1}\{\mathbf{r} \le S\} = (\lfloor e_1 \rfloor \mathbf{1}\{r_1 \le S\}, \ldots, \lfloor e_{kJ} \rfloor \mathbf{1}\{r_{kJ} \le S\}) \,. \tag{19}$$

    In order to ensure that the resulting monomials are all of degree at least 1 under each of the above degree constraints, the variable of highest rank within each block of J genes may receive a phenotype of 1 when all gene values within the block are less than 1.

3. Given a phenotype vector $\mathbf{d}$, an individual has an associated subset of basis functions φ_n. Calculate from φ_n the corresponding estimate ψ_n according to (11).

Initialization: The user may specify the number $Z \ge 4$, $Z \in \mathbb{Z}^+$, of individuals in the initial population. Recall that the genotype vector $\mathbf{e}$ for an individual has length kJ. Initialization consists of generating genotype vectors for each individual from a (kJ)-variate uniform distribution on $(0, D+1)^{kJ}$ or $(0, D_0+1)^{kJ}$.

Selection: Given a population of individuals, each with an associated estimate ψ_n, we will rank individuals according to fitness via the following procedure:

1. Compute empirical risk $\Theta(\psi_n, P)$, where P is the empirical distribution with respect to either a cross-validation training data set $\mathcal{T}_n$ or the learning set $\mathcal{X}_n$.

2. Rank existing individuals in order of increasing empirical risk. Select the $2\lfloor Z/4 \rfloor$ individuals with smallest empirical risk for reproduction. Refer to the ranked population as $(\mathbf{e}[1], \ldots, \mathbf{e}[Z])$, with each $\mathbf{e}[z]$ mapping to the gene vector $\mathbf{e}$ for the individual with the zth smallest risk.

An individual may be considered *cumulatively optimal* at generation g if its associated estimate ψ_n has a smaller risk than that of any other individual produced in the first g generations. The proposed selection mechanism is *elitist* in the sense that the cumulatively optimal individual is always selected at each generation. (Indeed, if the cumulatively optimal individual at generation g is not selected at generation $g+1$, then it is supplanted by some other individual with an associated estimate of smaller risk. This new individual is then cumulatively optimal at generation $g+1$.) As a result, the associated estimate of the EA's cumulatively optimal individual is monotonically non-increasing in risk as a function of the generation of evolution.

Reproduction: Create offspring from selected individuals via the following procedure:

1. Assign selected individuals to mating pairs in order of increasing empirical risk. The individuals $\mathbf{e}[1]$ and $\mathbf{e}[2]$ are assigned to a mating pair, and the process is repeated on the remaining population until $\lfloor Z/4 \rfloor$ pairs of individuals have been assigned.

2. Breed each mating pair to produce $C = 2$ offspring. For each child $c \in \{1, \ldots, C\}$, generate a Bernoulli(p) random variable γ_c. The default value of p is 0.5.

3. Given the population sorted in order of fitness, then, for $z \in \{1, \ldots, \lfloor Z/4 \rfloor\}$, construct each child's genotype vector as:

$$\mathbf{e}\left[\lfloor Z/2 \rfloor + 2(z-1) + c\right] = (\gamma_c) pmax \left(\mathbf{e}\left[2z-1\right], \mathbf{e}\left[2z\right]\right) + (1 - \gamma_c)\, pmin \left(\mathbf{e}\left[2z-1\right], \mathbf{e}\left[2z\right]\right). \qquad (20)$$

That is, a child's genotype is produced either via the pairwise maximum or pairwise minimum of the parental genes according to the flip of a weighted coin with probability p of selecting the pairwise maximum. The pairwise maximum *pmax* and pairwise minimum *pmin* are defined as the component wise maximum and minimum, respectively, of the two vector-valued arguments. Although it may seem redundant to produce two identical children if both coin flips match, dual propagation of this line may lead to a stronger evolutionary outcome over time. Furthermore, each child is subject to the mutation mechanism, so many identical twins produced at the reproduction stage may still result in genotypic differences. When the population size Z is a multiple of 4, the reproduction mechanism ensures that each offspring replaces an individual not selected for reproduction in the previous generation to maintain the population size throughout the evolution process. For other values of Z, the population size decreases to the largest multiple of 4 less than Z after the first reproduction phase.

Mutation: Each offspring is subject to mutation immediately following birth. An offspring mutates with a user-specified mutation probability η. When an offspring mutates, select the number of mutating genes by a discrete uniform random variable on $\{1, \ldots, \lfloor \lambda k J \rfloor\}$, with the mutation proportion parameter $\lambda \in [0, 1]$ supplied by the user. (A current software implementation of this procedure suggests $\eta = 0.1$ and $\lambda = 0.25$ as the default values. However, when too many genes mutate, the algorithm devolves into random search. In seeking the proper overall rate of mutation for the problem at hand, the user must weigh the number of explanatory variables J, the interaction order constraint (12), the mutation probability η, and the mutation proportion parameter λ.) Each mutating gene is selected uniformly at random and is independently assigned a new value on $(0, D+1)$ or $(0, D_0+1)$ according to a uniform random variable.

Result: After G generations of the evolutionary mechanisms, a final iteration of the selection phase is performed on the resulting population. Select the best-fit individual ($\mathbf{e}[1]$) from the final ranked population as the algorithm's result. This individual's phenotype vector $\mathbf{d}$ maps to a set of k basis functions φ_n with an associated estimate ψ_n of the parameter of interest. Because the selection mechanism is elitist, individual $\mathbf{e}[1]$ is cumulatively optimal at generation G. Within the search of size k estimators, the minimum observed risk for the specified data set is given by $\Theta\left(\psi_n, P\right)$.

In applying the EA to a particular regression setting, the user may select tuning parameters such as the population size Z, the number of generations G to run the algorithm, the mutation probability η, and the mutation proportion parameter λ specifying the maximum number of mutations that may occur in an individual. Any choice of these tuning parameters results in a new algorithm that may be compared to other estimation procedures directly in terms of risk on a test data set. Because the No Free Lunch theorem shows that no optimization algorithm solves all problems competitively (Wolpert and MacReady, 1997), the EA's adaptability allows the user to generate an arbitrary number of algorithms from which to choose. Moreover, the modular structure of the proposed search algorithm allows for alternative procedures that may remove, replace, or add to the existing evolutionary mechanisms without requiring significant changes to the software's design. For instance, the user may choose to insert an additional component of mutation that modifies the mutation probability η according to the population's phenotypic homogeneity. Crossover, which splices each parent's genes into a random number of subvectors and recombines the segments to produce two complementary children, may be used as a substitute for the proposed reproduction mechanism. Although the specific combination of evolutionary mechanisms may be chosen by the user, any evolutionary algorithm requires selection pressure to generate estimators of increasing quality and some procedure (such as mutation) to identify new candidate estimators. Additionally, the user may incorporate prior information into the EA by inserting existing estimates into the initial population or limiting evolution to a subset of genes so that particular basis functions may be forced into the estimate.

As the number of evolutionary generations G increases toward infinity, the proposed EA will converge to the size k estimate with globally minimal risk. Indeed, Fogel (2005) shows that an EA converges asymptotically in generation provided that all candidate optima form a single communicating class and an elitist selection mechanism is employed. In the proposed EA, any candidate optimum (as specified by an individual's phenotype) may be transformed into any other candidate optimum over the course of evolution through the mutation mechanism. Because all candidate optima form a single communicating class, the global optimum will eventually be reached over the course of evolution, and once this individual enters the population, elitist selection ensures that no other individual will ever supplant it. Therefore, the EA asymptotically converges as a function of generation to the estimate of globally optimal risk within the size k parameter subspace. Because of this convergence, and because cross-validation is an asymptotically optimal procedure for selecting the basis size k_n as a function of the sample size n, the proposed estimator selection procedure asymptotically converges in risk to the parameter of interest ψ as n and G tend toward infinity.

When information is known about the risk surface for an estimator selection application, it may be incorporated into the design of an appropriate optimization algorithm. However, the risk surface topology is typically unknown, so we are unable to provide any general bounds on the rate at which convergence to the global optimum is achieved. Indeed, it is possible that an EA will not improve upon full enumeration in terms of the rate of asymptotic convergence; however, because an EA evolves the population according to the risk surface, it generally outperforms random search in practical settings with limited computational resources available.

4.2 EA Example

The following example explicates the EA's evolutionary mechanisms. Suppose we are searching the parameter subspace with basis size $k = 2$ on a data set including $J = 2$ variables under either the polynomial degree constraint (13) with $D = 3$ or the variable degree constraint (14) with $D_0 = 3$ and the interaction order constraint (12) with $S = 2$ on a population of size $Z = 4$. The first step initializes the population of individuals with each consisting of a vector of $kJ = 4$ genes uniformly distributed on $(0, D_0 + 1)^4 = (0, D + 1)^4 = (0, 4)^4$:

Initialization:

Ind.	e_1	e_2	e_3	e_4
Ind_1	1.2	2.4	3.7	0.5
Ind_2	3.2	1.6	2.1	2.05
Ind_3	2.3	0.8	0.4	1.3
Ind_4	1.4	1.7	3.1	2.1

Each individual's genotype is arranged in k blocks of J genes, and the phenotype of each gene corresponds to the degree assigned to a variable within a basis function. The phenotype computation depends upon whether the polynomial degree constraint (13) or the variable degree constraint (14) is used. For constraint (13), we begin with the first block of $J = 2$ genes. The variable with the larger gene value in the block is assigned the floor of its gene value as its phenotype. Consider individual Ind_2 from the initial population. In the first block of genes, the stronger gene is e_1 with value 3.2, and its phenotype is $d_1 = \lfloor e_1 \rfloor = \lfloor 3.2 \rfloor = 3$. This assignment has exhausted the degree constraint (13) with $D = 3$, so no other variables may appear in this basis function. As a result, $d_2 = 0$. We then repeat this exercise for the second block of $J = 2$ genes (e_3 and e_4) for individual Ind_2. The stronger gene is e_3, so $d_3 = \lfloor e_3 \rfloor = \lfloor 2.1 \rfloor = 2$. Because we can still assign one more degree to the monomial and the interaction order constraint (12) allows another variable to interact, we can then assign a non-zero phenotype corresponding to gene e_4. However, this gene has the value 2.05, and its floor of 2 is larger than the remaining polynomial degree. As a result, we will assign the minimum of these quantities as its phenotype, so $d_4 = 1$. Individual Ind_2's phenotype immediately specifies the basis function subset $\varphi = \{W_1^3 W_2^0, W_1^2 W_2^1\} = \{W_1^3, W_1^2 W_2\}$, which in turn specifies a regression function for the parameter of interest. If no variable interactions are allowed ($S = 1$), then this subset is $\varphi = \{W_1^3, W_1^2\}$.

If the variable degree constraint (14) is used in place of the polynomial degree constraint (13), then gene values are first ranked within each block. Phenotypes are computed as the floor of gene values for the S genes of highest rank with all other variables receiving phenotype zero. For individual Ind_2, we consider the first block. The gene e_1 is larger than e_2, so its phenotype is assigned as $d_1 = \lfloor e_1 \rfloor = \lfloor 3.2 \rfloor = 3$. Because $S = 2$, $d_2 = 1$, and similarly, $d_3 = d_4 = 2$. This results in the basis function subset $\varphi = \{W_1^3 W_2^1, W_1^2 W_2^2\}$. If no variable interactions are allowed ($S = 1$), then this subset becomes $\varphi = \{W_1^3, W_1^2\}$. Note that individual Ind_2's genotype resulted in different phenotypes under constraints (13) and (14) for $S = 2$ but the same phenotype for $S = 1$. Once the phenotype is computed according to the selected degree and interaction order constraints, the empirical risk of the associated estimate is computed and recorded. The results for this hypothetical example are recorded below. Phenotypes and fitness values (empirical risk) corresponding to both degree constraints are shown for the individuals taken from the initial population.

Selection, Part I: Phenotype and Fitness

	Constraint (13), $D = 3$					Constraint (14), $D_0 = 3$				
Individual	d_1	d_2	d_3	d_4	Risk	d_1	d_2	d_3	d_4	Risk
Ind_1	1	2	3	0	40.3	1	2	3	0	40.3
Ind_2	3	0	2	1	45.3	3	1	2	2	52.4
Ind_3	2	0	0	1	27.9	2	0	0	1	27.9
Ind_4	1	1	3	0	44.1	1	1	3	2	39.1

Because the algorithm is otherwise identical for both degree constraints, we will proceed by only considering the genotype and phenotype structure specified by the variable degree constraint (14). The individuals within the population are sorted in order of increasing risk with the best individual placed in the first row. The best half of the population is selected for breeding, and the others die out. Individuals Ind_3 and Ind_4 are selected to breed and will form a mating pair. When larger population sizes are used, individuals are paired according to fitness rank to produce mating groups.

Selection, Part II: Fitness Ranking and Mating Pair Formation

Ind.	e_1	e_2	e_3	e_4	Risk	Selected
Ind_3	2.3	0.8	0.4	1.3	27.9	Yes
Ind_4	1.4	1.7	3.1	2.1	39.1	Yes
Ind_1	1.2	2.4	3.7	0.5	40.3	No
Ind_2	3.2	1.6	2.1	2.05	52.4	No

Evolution proceeds with the reproduction mechanism. Children are produced either via the pairwise max or the pairwise min of the parental genes according to the flip of a coin with probability p of selecting the pairwise max. For this example, suppose that child $Ind_1(1)$ receives the pairwise max and child $Ind_2(1)$ the pairwise min. As evolution proceeds, each individual is labeled with a parenthetical index storing its birth generation. The resulting population is given by:

Reproduction:

Ind.	e_1	e_2	e_3	e_4	Risk
$Ind_3(0)$	2.3	0.8	0.4	1.3	27.9
$Ind_4(0)$	1.4	1.7	3.1	2.1	39.1
$Ind_1(1)$	2.3	1.7	3.1	2.1	-
$Ind_2(1)$	1.4	0.8	0.4	1.3	-

The mutation mechanism then acts on the children $Ind_1(1)$ and $Ind_2(1)$ according to the flip of a weighted coin with success probability given by the mutation probability η. Suppose $Ind_1(1)$ mutates but $Ind_2(1)$ does not. Subsequent uniform random variables are used to select how many and which genes will mutate. For this example, the e_1 gene of individual $Ind_1(1)$ is selected to mutate, and its value is reset according to a random variable on $(0, D_0 + 1)$, just as in the initialization process. This results in the following population matrix:

Mutation:

Ind.	e_1	e_2	e_3	e_4	Risk
$Ind_3(0)$	2.3	0.8	0.4	1.3	27.9
$Ind_4(0)$	1.4	1.7	3.1	2.1	39.1
$Ind_1(1)$	**1.3**	1.7	3.1	2.1	-
$Ind_2(1)$	1.4	0.8	0.4	1.3	-

This completes one round of the selection, reproduction, and mutation mechanisms, and the evolutionary process repeats itself on the current population and continues for a total of G generations. When evolution concludes, the selection mechanism is performed on the final population, and the best individual from this population is retained. This individual has a phenotype vector $\mathbf{d}$ that maps to a set of basis functions φ_n whose corresponding estimate ψ_n represents the EA's estimate of the parameter of interest based upon the (training or learning) data set available.

5 Simulation Studies

We conducted the following simulation experiment to test the efficacy of the proposed EA-based estimator selection procedure.

5.1 Simulation Study Design

Each trial consisted of generating $n = 1000$ explanatory variables and outcomes as functions (both random and non-random) of the explanatories. The trials were designed to reproduce a subset of the results obtained in Sinisi and van der Laan (2004). With $J = 11$, the explanatory variables $W = (W_1, \ldots, W_J)$ were independently generated from the uniform distribution on (0,1). Using these data, the following outcomes were created:

$$Y_1 = W_2 + W_3^2; \qquad Y_{1e} = Y_1 + \epsilon; \qquad \epsilon \sim N(0,1); \qquad \epsilon \perp W; \tag{21}$$

$$Y_2 = W_2 W_4; \qquad Y_{2e} = Y_2 + \epsilon; \qquad \epsilon \sim N(0,1); \qquad \epsilon \perp W; \tag{22}$$

$$Y_3 = W_2 W_4 W_6^2 + W_8 W_{11}; \qquad Y_{3e} = Y_3 + \epsilon; \qquad \epsilon \sim N(0,1); \qquad \epsilon \perp W; \tag{23}$$

$$Y_5 = W_1 W_2^2 W_3^2 + W_1 W_2 W_3^2 W_4 + W_3^3 + W_5^4; \qquad Y_{5e} = Y_5 + \epsilon; \qquad \epsilon \sim N(0,1); \qquad \epsilon \perp W. \tag{24}$$

Each model a was subject to the variable degree constraint (14) with bounds of $D_1 = D_2 = 2$, $D_3 = 4$, and $D_5 = 5$. No interaction order constraint was imposed, so $S = J = 11$ by default. The total number of basis functions for each setting, which is given by the formula for I_0 in (16), is shown in Table 2. Figure 1 also shows the growth in I_0 as a function of D_0 with $J = 11$ variables and no interaction order constraint. Model 5 comprises the largest parameter space. Figure 2 shows how this parameter space can be pruned by introducing an interaction order constraint.

Model	1	1e	2	2e	3	3e	5	5e
Cross-Validation Gen.	1,000	2,000	1,000	2,000	3,000	5,000	12,000	12,000
Learning Set Gen.	5,000	5,000	5,000	5,000	10,000	12,000	10,000	15,000
D_0	2	2	2	2	4	4	5	5
I_0	177,146	177,146	177,146	177,146	48,828,124	48,828,124	362,797,055	362,797,055

Table 2: Generation values G and variable degree bounds D_0 supplied to constraint (14) for the EA estimator selection procedure in the simulation studies. With $J = 11$ variables, constraint (14), and no interaction order constraint, the number of basis functions I_0 is calculated according to (16) for each estimation setting. Generation values were increased for estimation settings with a larger parameter space.

The above experiment was repeated for a total of $B = 193$ trials. Given a subset of basis functions, the parameter vector β in (10) was estimated using Ordinary Least Squares (OLS) linear regression. Candidate basis sizes were restricted to a maximum value of $K = 5$. V-fold cross-validation was conducted with $V = 5$ and the default values of all non-specified computational parameters. Both the EA presented in Section 4 and the DSA algorithm of Sinisi and van der Laan (2004) were used to estimate the parameter of interest $\psi_a = E[Y_a|W]$ for each of the above random and non-random models $a \in \{1, 2, 3, 5\}$. On each trial, the EA and DSA algorithms each performed separate random assignments of data to their respective training sets $\mathcal{T}$ and validation sets $\mathcal{V}$. Furthermore, because the DSA algorithm includes a stopping criterion based upon relative improvement in risk, the trials do not involve the same number of model fits. In general, the DSA was allowed to run until its stopping criterion was triggered, and the EA was run for the generation limits specified in Table 2. However, for the deterministic models, the EA was allowed to halt its search if an estimate with zero risk (with a round-off error tolerance of 10^{-15}) was located. If all V searches of a parameter subspace with basis size k located estimators that attained a validation set risk of zero, then no parameter subspaces of larger basis size were searched. Likewise, the EA also halted if the learning set search located an estimate with zero empirical risk. The current software implementation of the EA also allows for an exhaustive search of a parameter subspace if doing so is more computationally efficient than running the EA for the specified number of generations. For a population of size Z, the proposed EA fits a total of T regression estimates over G generations of evolution, where T is given by:

$$T = Z + 2G\lfloor Z/4 \rfloor. \tag{25}$$

The EA first fits regression estimates for each of the Z individuals in the initial population. At each generation, a total of $2\lfloor Z/4 \rfloor$ offspring are created. Because regression estimates are computationally costly, the value of T may be reduced if individuals within the population specify the same candidate solution. Similarly, when $\binom{I}{k} \leq T$ or $\binom{I_0}{k} \leq T$ for constraints (13) or (14), respectively, an exhaustive search of the size k parameter subspace is more computationally efficient than evolution. As an additional computational parameter, the user may specify a *leeway* value l such that an exhaustive search is used provided that $\binom{I}{k} \leq T + l$ or $\binom{I_0}{k} \leq T + l$, which may be preferred when the computational limits are close to those required for an exhaustive search.

5.2 Simulation Study Results

For each simulation model studied, the EA and DSA produce estimates of the parameter of interest. We can assess these results in terms of the selected basis size, *sensitivity*, *specificity*, *cross-validated risk*, *empirical learning set risk*, and *empirical test set risk*. With respect to the true parameter ψ, an estimate ψ_n's sensitivity reflects the proportion of true basis functions included in the estimate, and its specificity denotes the proportion of the selected basis functions that are contained in the set generating the true parameter. If both quantities are one, then the estimator includes the same set of basis functions as that specifying the parameter of interest. In terms of the set of basis functions φ_n that generate an estimate ψ_n of ψ, we can define the sensitivity and specificity as follows:

$$sensitivity(\psi, \psi_n) = \frac{|\varphi \cap \varphi_n|}{|\varphi|}; \tag{26}$$

$$specificity(\psi, \psi_n) = \frac{|\varphi \cap \varphi_n|}{|\varphi_n|}. \tag{27}$$

The cross-validated risk is the minimum component of the vector (6) of mean validation set risks. The empirical learning set risk is the risk of the selected estimator on the learning set $\mathcal{X}_n$. The empirical test set risk is the risk of the selected estimator on a test set of 1 million observations. Because we seek to minimize risk, smaller quantities are preferred for the cross-validated, empirical learning set, and empirical test set risks. When independent and identical trials are conducted for a given algorithm on separate data sets, we can combine the results in terms of a performance metric. We are primarily concerned with the distribution of each type of risk in general and the median value in particular. The results of the simulation study are contained in Tables 3–5 and Figures 4–9. These figures contain *notched* boxplots, and evidence of a significant performance difference between the DSA and EA is noted when the notches of the respective boxplots fail to overlap (Chambers et al., 1983).

The simulation's sensitivity results for the random and non-random models are summarized in Table 3. For non-random models, the EA consistently produces a sensitivity of 1 for estimating $E[Y_a|W]$ on Models 1, 2, and 3. Although results are variable for Model 5, the median sensitivity is 1. Meanwhile, the DSA produces strong results for some models but not for others. In the random models, the results are more varied. The EA successfully locates the proper basis functions for Model 1e and at least one true basis function for Models 3e and 5e but does not locate the proper term for Model 2e. The DSA performs similarly to the EA on Models 1e, 2e, and 5e, but it does not locate any true basis functions for Model 3e.

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
Model 1 EA	1.00	1.00	1.00	1.00	1.00	1.00
Model 1 DSA	1.00	1.00	1.00	1.00	1.00	1.00
Model 1e EA	0.00	0.50	1.00	0.81	1.00	1.00
Model 1e DSA	0.00	1.00	1.00	0.88	1.00	1.00
Model 2 EA	1.00	1.00	1.00	1.00	1.00	1.00
Model 2 DSA	0.00	0.00	0.00	0.00	0.00	0.00
Model 2e EA	0.00	0.00	1.00	0.58	1.00	1.00
Model 2e DSA	0.00	0.00	0.00	0.00	0.00	0.00
Model Y_3 EA	1.00	1.00	1.00	1.00	1.00	1.00
Model Y_3 DSA	0.00	0.00	0.00	0.00	0.00	0.00
Model 3e EA	0.00	0.00	0.00	0.23	0.50	1.00
Model 3e DSA	0.00	0.00	0.00	0.00	0.00	0.00
Model Y_5 EA	0.50	1.00	1.00	0.89	1.00	1.00
Model Y_5 DSA	0.50	0.50	0.50	0.50	0.50	0.50
Model 5e EA	0.00	0.00	0.25	0.18	0.25	0.75
Model 5e DSA	0.00	0.00	0.25	0.23	0.25	0.50

Table 3: Six number summaries for sensitivity measurements in the simulation study.

The specificity results are displayed in Table 4. The EA consistently selects only proper basis functions for Models 1, 2, and 3, with 1 improper term and 4 correct terms typically selected for Model 5. The DSA includes both proper and improper terms for Models 1 and 5 but trails the EA in specificity on all non-random models. However, for random models, the DSA appears to perform better than the EA on Models 1e and

5e, equally on Model 2e, and worse on Model 3e.

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
Model 1 EA	1.00	1.00	1.00	1.00	1.00	1.00
Model 1 DSA	0.40	0.40	0.40	0.44	0.40	1.00
Model 1e EA	0.00	0.25	0.40	0.36	0.40	1.00
Model 1e DSA	0.00	0.50	1.00	0.79	1.00	1.00
Model 2 EA	1.00	1.00	1.00	1.00	1.00	1.00
Model 2 DSA	0.00	0.00	0.00	0.00	0.00	0.00
Model 2e EA	0.00	0.00	0.20	0.15	0.20	1.00
Model 2e DSA	0.00	0.00	0.00	0.00	0.00	0.00
Model 3 EA	0.67	1.00	1.00	1.00	1.00	1.00
Model 3 DSA	0.00	0.00	0.00	0.00	0.00	0.00
Model 3e EA	0.00	0.00	0.00	0.11	0.20	1.00
Model 3e DSA	0.00	0.00	0.00	0.00	0.00	0.00
Model 5 EA	0.40	0.80	0.80	0.78	1.00	1.00
Model 5 DSA	0.40	0.40	0.40	0.40	0.40	0.50
Model 5e EA	0.00	0.00	0.20	0.16	0.20	0.75
Model 5e DSA	0.00	0.00	0.33	0.34	0.50	1.00

Table 4: Six number summaries for specificity measurements in the simulation study.

Table 5 shows the basis size error. The error is standardized across models by subtracting the true basis size from the selected size on each trial, so an error of zero is desirable. In terms of median performance, the EA consistently selects the appropriate basis size on all non-random models but occasionally overestimates on Model 5. The DSA consistently overestimates the basis size for all non-random models. However, for random models, the EA appears to overestimate the true basis size while the DSA either produces a smaller overestimate (Models 2e and 3e), selects the appropriate size (Model 1e), or underestimates the basis size (Model 5e).

The performance difference between the EA and DSA becomes clear when we compare the two procedures in terms of risk. Figure 4 (non-random models) and Figure 5 (random models) summarize the cross-validated risk for the estimates produced in the simulation study. Both procedures consistently locate the appropriate set of basis functions in the cross-validation stage of estimator selection on Model 1, but the EA produces a smaller median cross-validated risk for the other seven models studied. Furthermore, the EA consistently locates an estimate resulting in essentially zero cross-validated risk (within a tolerance of 10^{-15} for rounding error) for the deterministic models.

Empirical learning set risk in the simulation study is displayed in the boxplots of Figures 6 and 7. Just as in the cross-validated risk measures, the EA consistently performs as well or better than the DSA in terms of median empirical learning set risk. Finally, Figures 8 and 9 convey an estimate of the EA and DSA's true risk obtained from a separate test data set consisting of one million observations. Again, the EA performs as well or better than the DSA on the non-random models of Figure 8. The DSA appears to outperform the EA on Model 1e, but the EA results in a superior median empirical test set risk on each of the other random models of Figure 9. For the random models, the true risk (the risk of the true regression function) is given by the variance of the residual vector ϵ, which is 1 in this case because the residuals were generated from standard Normal random variables. For the simulation study, any increase above 1 in the test set risk can be attributed to a bias introduced by the selection of improper basis functions by the EA or DSA. Across all simulations, it appears that the EA's estimates produce a test set risk that exhibits greater variance than that of the DSA.

The cross-validated risk, empirical learning set risk, and empirical test set risk are all estimates of true risk

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
Model 1 EA	0.00	0.00	0.00	0.00	0.00	0.00
Model 1 DSA	0.00	3.00	3.00	2.77	3.00	3.00
Model 1e EA	0.00	2.00	3.00	2.60	3.00	3.00
Model 1e DSA	0.00	0.00	0.00	0.37	1.00	3.00
Model 2 EA	0.00	0.00	0.00	0.00	0.00	0.00
Model 2 DSA	1.00	1.00	1.00	1.27	1.00	4.00
Model 2e EA	0.00	3.00	4.00	3.47	4.00	4.00
Model 2e DSA	0.00	1.00	1.00	1.26	1.00	4.00
Model 3 EA	0.00	0.00	0.00	0.01	0.00	1.00
Model 3 DSA	3.00	3.00	3.00	3.00	3.00	3.00
Model 3e EA	0.00	2.00	3.00	2.47	3.00	3.00
Model 3e DSA	−1.00	0.00	1.00	1.33	2.00	3.00
Model 5 EA	0.00	0.00	1.00	0.61	1.00	1.00
Model 5 DSA	0.00	1.00	1.00	0.99	1.00	1.00
Model 5e EA	−1.00	0.00	1.00	0.69	1.00	1.00
Model 5e DSA	−2.00	−2.00	−1.00	−0.97	0.00	1.00

Table 5: Six number summaries for basis size error measurements in the simulation experiments.

for a given estimate of the regression function ψ. However, it is well known that the empirical learning set risk tends to underestimate true risk (Dudoit and van der Laan, 2005). In the figures mentioned above, the median empirical learning set risk for the simulation results is smaller than the corresponding median cross-validated risk or empirical test set risk in each of the random models studied for both the EA and DSA. (In many of the non-random models, each median is zero.) In general, we prefer the empirical test set risk to the cross-validated risk in assessing an estimate's quality; because the test set data are not used in the estimator selection process, the resulting estimate cannot over-fit to the test set data. Although the median cross-validated and empirical test set risks were both close to the true risk of 1, the cross-validated risk exhibits significantly greater variability across trials than the corresponding empirical test set risk on each model. Therefore, the empirical test set risk appears to estimate true risk more reliably than the cross-validated risk in the random simulation models.

6 Data Analysis for a Diabetes Study

The proposed EA for estimator selection may be applied in a wide variety of estimation settings to investigate which explanatory variables contribute to an outcome of interest and examine the ways in which these variables interact. As an example of our procedure, we will study a diabetes data set used in Efron et al. (2004) to predict a quantitative measure of disease progression taken one year after the onset of the disease. Explanatory variables include age, sex, body mass index (BMI), blood pressure (BP), and quantitative measures of six blood serum levels $S1, \ldots, S6$. Data are available for a total of $n = 442$ patients. We seek to estimate the expected value of disease progression given a particular 10-dimensional covariate profile.

In order to estimate true risk, we divided the diabetes data at random into a learning set of 392 observations and a test set of 50 data points. Using an R language implementation of the EA estimator selection procedure, we supplied the parameter values in Table 6 (with all other computational parameters set to the default values) to obtain an estimate of the expected disease progression given the covariate values on the learning set. Additionally, we allowed the DSA to run with the same candidate basis sizes, cross-validation folds V, and interaction order bound for constraint (12) as those supplied to the EA. For the DSA, the polynomial degree constraint (13) with $D = 3$ was used in place of the variable degree constraint (14) employed by

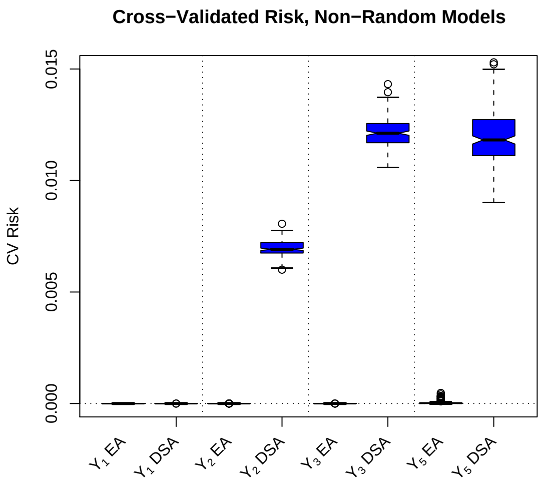

Figure 4: Cross-validated risk of estimates produced by the EA and DSA algorithms for the non-random simulation models.

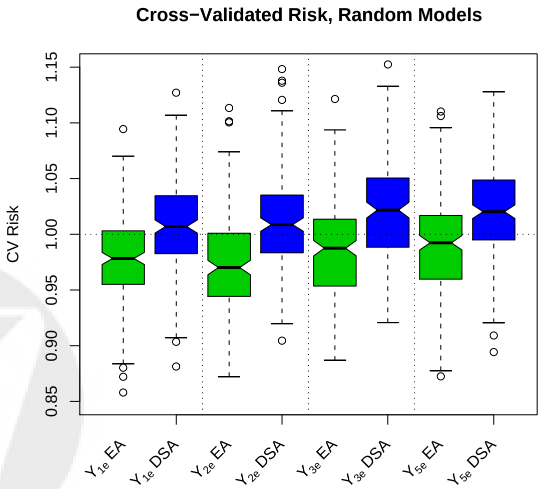

Figure 5: Cross-validated risk of estimates produced by the EA and DSA algorithms for the random simulation models.

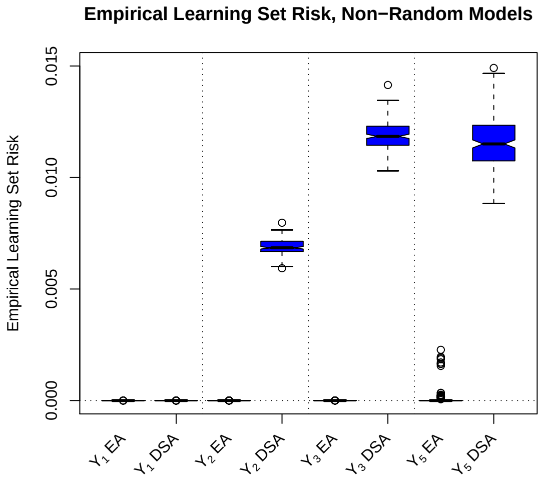

Figure 6: Empirical learning set risk of estimates produced by the EA and DSA algorithms for the non-random simulation models.

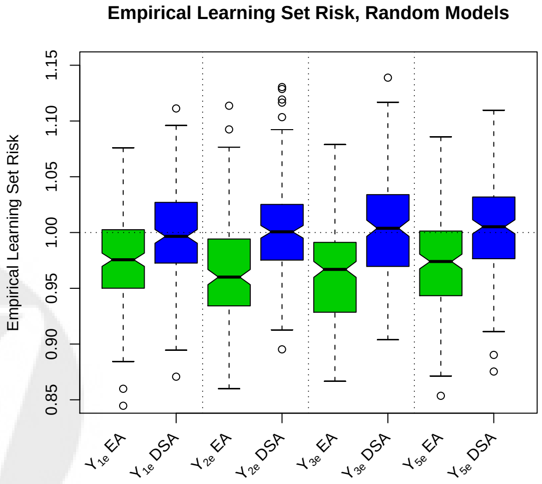

Figure 7: Empirical learning set risk of estimates produced by the EA and DSA algorithms for the random simulation models.

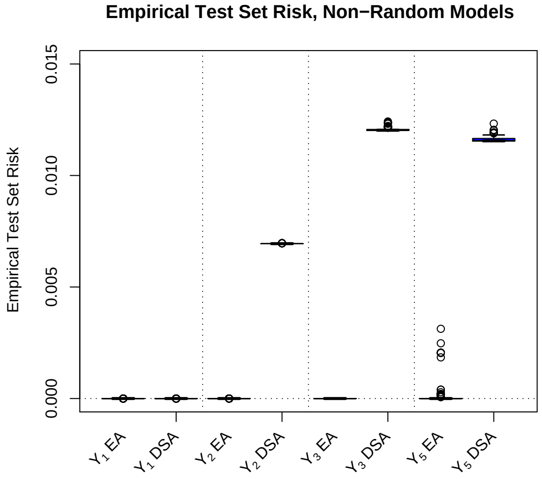

Figure 8: Empirical test set risk of estimates produced by the EA and DSA algorithms for the non-random simulation models.

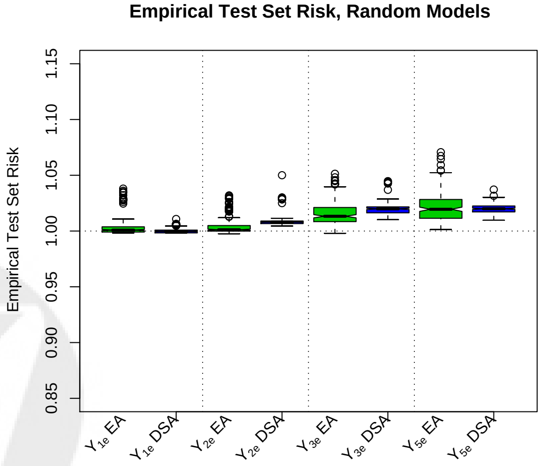

Figure 9: Empirical test set risk of estimates produced by the EA and DSA algorithms for the random simulation models.

Basis Sizes	V	D_0/D	S	Population Size, Z	CV Generations, G	Learning Set Generations, G
$\{0, 1, \ldots, 8\}$	5	3	3	20	$5,000$	$10,000$

Table 6: Tuning parameter values for the EA estimator selection algorithm applied to the diabetes data set of Efron et al. (2004).

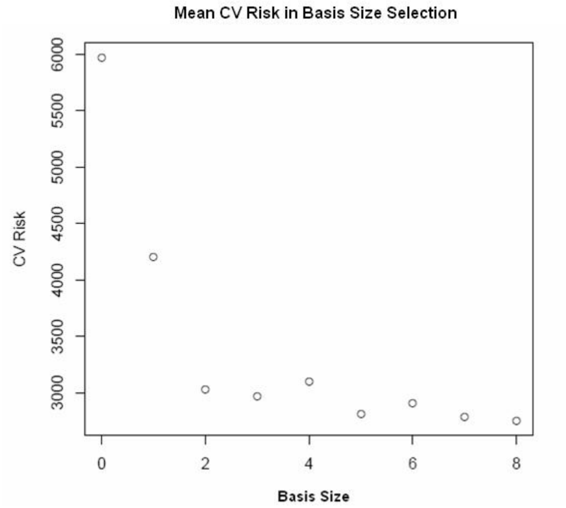

Figure 10: Mean cross-validated risk of EA by candidate basis size.

the EA with $D_0 = 3$. Because the two procedures were not subject to the same constraints, the estimated regression functions are not directly comparable. It should be noted that an updated software release of the DSA (version 2.2.1) was used for this analysis compared to version 2.0.2 employed in the simulation study of Section 5. Computations were performed on a Unix workstation with approximately 11 gigabytes of RAM and a 2 megaHertz processor in the University of California, Berkeley's Statistical Computing Facility. The generation limits of Table 6 were chosen so that the computation could be performed overnight, which was considered the maximum acceptable search time for the study. In total, this required 5.7 hours of computation.

Figure 10 displays the cross-validated risk for each candidate basis size considered by the EA. During the cross-validation phase, the estimator selection algorithm selected a basis size of 8, the maximum considered. Figure 11 plots empirical learning set risk as a function of generation in the learning set risk optimization within the size 8 parameter subspace. Because the cumulatively optimal individual is retained at each generation, risk decreases monotonically as a function of generation. Somewhat after the $8,000$th generation, the EA located an estimate that was not improved upon in the subsequent generations. The estimator selection procedure results in the OLS coefficient estimates contained in Table 7. Ordinarily, these coefficients are accompanied by estimated standard errors, t-statistics, and p-values for testing the null hypothesis of a zero coefficient. However, such inferences can only be drawn through a model of the underlying distribution of the estimator, which is currently an open problem for estimator selection procedures such as those considered in this paper. Similarly, Table 8 shows the regression coefficient estimates obtained by the DSA. The basis function including the $S5$ serum measurement was selected by both the EA and DSA, but otherwise the selected basis functions differed in terms of degree, order of interaction, and coefficient estimates. Most of the basis functions selected by the EA contain higher powers, a maximal order of interaction, and generally large coefficient estimates. In contrast, the DSA produced an estimate with no higher powers assigned to

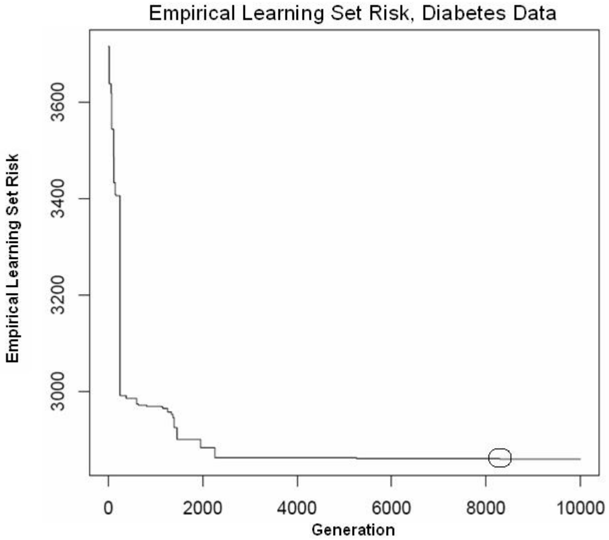

Figure 11: Empirical learning set risk as a function of generation in risk optimization on estimates of size 8. The circled region contains the generation at which the final estimate was located by the EA.

Int.	$S5$	$Sex^3{:}BMI^3{:}S1^3$	$Sex^2{:}BMI^3{:}S4^2$	$Age^3{:}S3^3{:}S6^3$	$Age{:}S6$	$Sex^2{:}BMI$	$BP^2{:}S4^3{:}S6^2$	$BMI^2{:}S4^3{:}S6^2$
-3.67	5.95e02	-1.48e11	3.04e08	2.03e11	3.72e03	3.04e05	7.28e08	-3.76e08

Table 7: EA regression coefficient estimates for disease progression in the diabetes study.

any variable, relatively few variable interactions, and smaller coefficient estimates that produce a simpler interpretation for the effect of each variable. It is possible that the EA would also produce a more meaningful estimate if the polynomial degree constraint (13) were used in place of the variable degree constraint (14), which would limit the parameter space to a subspace of that considered here. However, at the time of this analysis, the software implementation of the EA for the polynomial degree constraint (13) was not yet available.

Int.	BMI	$S5$	$S3$	BP	SEX	$BMI : BP$	$AGE : SEX$
-5.84	525.98	549.76	315.14	295.34	-255.71	3910.98	3913.89

Table 8: DSA regression coefficient estimates of disease progression in the diabetes study.

We then compared the EA and DSA estimates to those obtained by a variety of other estimator selection procedures considered by Durbin et al. (2005). These estimates' test set risks were calculated on a total of $B = 100$ bootstrap samples produced from the test set data. Although the learning and test sets were identical to those used by Durbin et al. (2005), the specific bootstrap test set samples previously used were not available. However, the bootstrap test sets generated in this analysis are i.i.d. observations produced from the sampling technique of Durbin et al. (2005). The results are displayed in Table 9. In terms of mean test set risk, both the EA and the DSA improved upon the performance of all estimators considered by Durbin et al. (2005). In particular, the EA's estimate resulted in a mean test set risk that improved upon all previous results by approximately 7.9%. Moreover, the DSA's estimate improved upon that of the EA by approximately 10.2%. Figure 12 displays a notched boxplot of bootstrap test set risk for each estimator

Estimator	Bootstrap Test Set Risk	Ratio	Covariates
lm	3182.0 (1966.2, 4397.7)	1.079	(all)
LARS (CV)	3270.9 (2118.6, 4423.2)	1.109	Sex, BMI, BP, $S1-S3$, $S5$, $S6$
polymars	3301.8 (2123.8, 4479.9)	1.119	Sex, BMI, BP, $S3$, $S5$, $S6$
LARS (Cp)	3336.7 (2206.2, 4467.3)	1.131	Sex, BMI, BP, $S2$, $S3$, $S5$, $S6$
full nnet	3552.4 (2297.2, 4807.6)	1.204	(all)
nnet-DSA	3565.2 (2368.4, 4175.8)	1.208	(all)
rpart	3692.0 (2498.9, 4885.0)	1.251	BMI, BP, $S2$, $S3$, $S5$, $S6$
DSA	**2649.3 (1337.1, 3961.6)**	0.898	Sex, BMI, BP, $S3$, $S5$
EA	2950.3 (1670.6, 4230.0)	1	Age, Sex, BMI, BP, $S1$, $S3-S6$

Table 9: Empirical test set risk of several estimator selection procedures on the diabetes data of Efron et al. (2004) based upon $B = 100$ bootstrap samples of the diabetes data test set of 50 observations. The EA and DSA results are compared in terms of risk to a number of estimators tested in Durbin et al. (2005) on the diabetes data . The table shows the mean bootstrap test set risk and 95% risk confidence interval for each estimator based on 100 bootstrap samples from the test set. Confidence intervals were produced from normal theory according to estimates of the mean and standard deviation for each estimator's risk. The third column compares each procedure's mean risk ratio to that of the EA, and the final column shows which covariates were included in each algorithm's selected estimator. It appears that all results obtained by Durbin et al. (2005) were at least 7.9% larger in test set risk than that obtained from the EA, and the DSA subsequently improved on the EA by approximately 10.2%.

selection algorithm. These results may be directly compared to those contained in Durbin et al. (2005), which are reproduced in Figure 13 with the permission of the authors. Because its notches do not overlap with those of any other estimator, it appears that the DSA significantly outperforms all other estimators considered for this particular problem. The EA and DSA's 95% confidence intervals for test set risk appear to be wider than those of the other estimators studied. It is possible that the proposed EA produces a greater variability in its estimates on account of its stochastic mechanisms in the reproduction and mutation stages.

7 Conclusion

In light of the size of parameter spaces for the constraint profiles characterized in Section 3, estimator selection procedures operating according to the general road map for loss based estimation must be able to search quickly and effectively for candidate estimators minimizing empirical risk within parameter subspaces. EAs and similar stochastic optimization algorithms provide an aggressive approach to risk optimization and are sufficiently flexible to offer high-quality estimates in a wide variety of settings. The results of the simulation study and diabetes analysis establish the proposed EA as a competitive alternative to other procedures. Because the No Free Lunch Theorem (Wolpert and MacReady, 1997) shows that no single algorithm can always outperform all others, the proposed EA may be used as a complement to the DSA as a general tool for estimator selection in regression settings. The EA is an attractive alternative because its computational parameters can be adapted to the problem at hand, and its modular design allows for variations of its evolutionary mechanisms without requiring significant changes in the overall software implementation. Furthermore, the EA converges asymptotically in generation to the global optimum within the size k parameter subspace to be searched. While the DSA search algorithm shifts between parameter subspaces of different basis size, the EA independently searches each subspace. This separation allows for parallel computing techniques to simultaneously search different parameter subspaces on additional processors and also allows the user to tune computational parameters like the population size, mutation probability, and number of generations according to the size of the subspace. Although the EA described is designed to search a parameter space consisting of polynomial regression functions, the proposed methodology applies to general

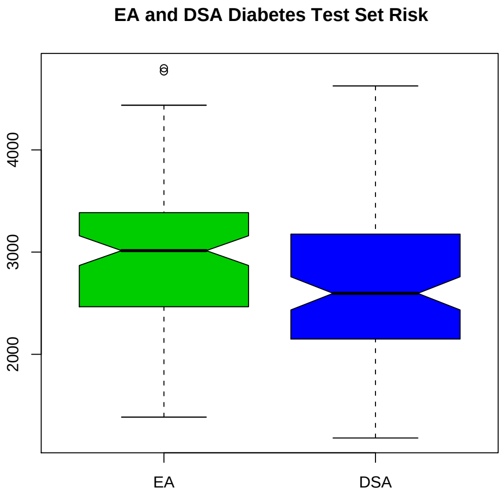

Figure 12: Boxplots of bootstrap test set risk of the EA and DSA estimates obtained from the diabetes data based upon $B = 100$ bootstrap samples of the test set. These results may be directly compared to those obtained by Durbin et al. (2005) in Figure 13.

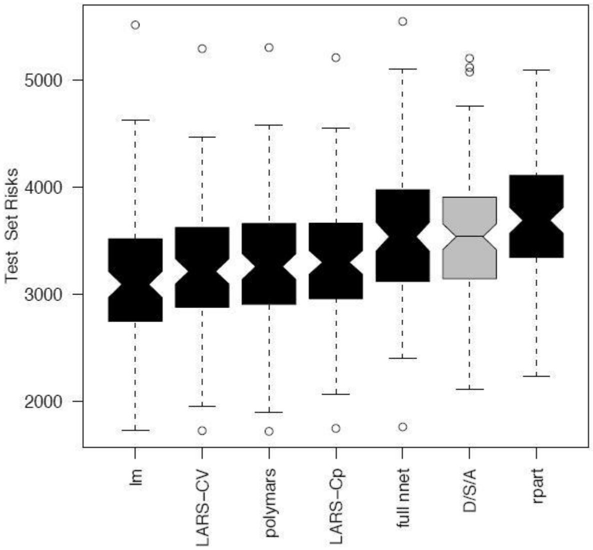

Figure 13: Empirical test set risk of several estimator selection procedures on the diabetes data based upon $B = 100$ bootstrap samples of the diabetes data test set of 50 observations. This figure was originally produced by Durbin et al. (2005) and is reproduced here with the permission of the authors. These results may be directly compared to those of the EA and DSA, which are displayed in Figure 12.

parametrizations (e.g. histogram regression and neural networks), which is an appealing feature of both the EA and the DSA.

The results obtained in this study come with a few caveats: first, in general the EA required significantly more time to produce its estimates than the DSA. This time difference may be attributed to the DSA's implementation in the C programming language, which is significantly faster than R. Although this project illustrates the EA's utility, it also demonstrates the need to improve the algorithm's speed in subsequent software packages. Future implementations of the EA may also apply parallel computing techniques to simultaneously search distinct subspaces or training sets in the cross-validation phase.

Because the DSA was treated as a black box in the simulations, a comparison to the EA in terms of the number of model fits required to obtain an estimate of a given quality is currently unavailable. However, the simulation results suggest that the DSA is vulnerable to local optima: when its set of deletion, substitution, and addition moves do not reduce risk, the algorithm halts at the current estimate. It is currently unknown whether all candidate estimators within a parameter subspace necessarily form a single communicating class for the DSA. If not, asymptotic convergence in time to the global optimum within a parameter subspace cannot be ensured. Otherwise, the DSA's vulnerability to local optima observed in this study may be a consequence of the software's stopping criteria. Unlike EAs, the DSA risk-optimizing search procedure is deterministic for a given split of the data into training and validation sets. Future versions of the DSA may consider introducing a stochastic component akin to the EA's mutation mechanism to work in concert with its existing elitist selection procedure. If the proposed augmentation ensures that all estimators within a parameter subspace form a single communicating class, then this modified DSA would asymptotically converge in time to the global optimum.

Additionally, estimator selection software packages may provide the researcher with the opportunity to include particular basis functions in all candidate estimates so that known causal relationships remain fixed while searching for additional factors that contribute to a quantity of interest. When the researcher wishes to compare results from a large number of distinct algorithms, an arbitrary number of alternative search procedures may be generated by varying the EA's tuning parameters such as the mutation probability. For a particular problem, an additional cross-validation procedure may be used to select among candidate mutation probabilities or other tuning parameters. Finally, the variability of the EA's results may be investigated as a function of generation to guide the choice of these computational parameters.

Appendix

Section 3.3 analyzed the size of the parameter space for polynomial regression under the interaction order constraint (12) and the polynomial degree constraint (13) or the variable degree constraint (14). We wish to substantiate the conclusions summarized in Table 1.

Under constraint (13), the number of basis functions is given by the value of I (15), which can be bounded below as follows:

$$I = \sum_{s=1}^{S} \binom{J}{s} \left[1 + \sum_{d=s+1}^{D} \sum_{k=1}^{\min(s,d-s)} \binom{s}{k} \binom{d-s-1}{k-1} \right] \geq \sum_{s=1}^{S} \binom{J}{s} \geq \sum_{s=1}^{S} \binom{S}{s} = 2^S - 1 \Rightarrow \Omega\left(2^S\right). \qquad (28)$$

The first equality restates (15), and the first inequality follows because all terms in the nested summations are positive. Under constraint (13), then $S \leq \min(J, D) \leq J$, and $\binom{S}{s} \leq \binom{J}{s}$ for all $s \in \{1, \ldots, S\}$, so the second inequality holds. The final equality is a direct consequence of the Binomial Theorem. Therefore, I is bounded below by a function of order $\Omega\left(2^S\right)$. For the extreme case of $S = \min(J, D)$, then $I = \Omega\left(2^S\right) = \Omega\left(2^{\min(J,D)}\right)$, which is an exponential function of the number of variables J and the polynomial degree bound D.

We then turn our attention to the case of I_0 under the variable degree constraint (14). We can bound I_0 from below as follows:

$$I_0 = \sum_{s=1}^{S} \binom{J}{s} D_0^s \geq \sum_{s=1}^{S} \binom{S}{s} D_0^s = (D_0 + 1)^S - 1 > D_0^S \Rightarrow \Omega\left(D_0^S\right). \tag{29}$$

The first equality restates (16), and the first inequality follows because $J \geq S$ when constraint (14) is imposed. The next equality follows from the Binomial Theorem, and the remaining polynomial is of degree S. In the extreme case of $S = J$, the number of basis functions is then $\Omega\left(D_0^J\right)$. Because the summation in (29) is solved in a closed form and results in a polynomial when $S = J$, this asymptotic lower bound is also an asymptotic upper bound, and both are tight (Cormen et al., 1990). Therefore the number of basis functions is both $\Omega\left(D_0^J\right)$ and $O\left(D_0^J\right)$ when $S = J$. Because S is maximized, this upper bound is an overall upper bound on the number of basis functions under the variable degree constraint (14). Furthermore, because the set of basis functions under the polynomial degree constraint (13) is a subset of those under the variable degree constraint (14) when $D = D_0$, then the number of basis functions I is trivially bounded above by a function of order $O\left(D^J\right)$. Likewise, the value I_0 for constraint (14) is loosely bounded above by a function of order $O\left(D_0^J\right)$ that becomes tight if $S = J$.

The size of the parameter space is 2^I or 2^{I_0} in the constraint profiles of Section 3. By applying the previous bounds for I and I_0 to the parameter space analysis, we arrive at the conclusions summarized in Table 1. It should be noted that the order functions Ω and O imply that the bounds can be stated as a constant times the given function. In expressing the size of the parameter space in terms of the number of basis functions under different constraint profiles, the constant for the order of the size of the parameter space differs from that for the order of the number of basis functions.

Acknowledgments

The authors wish to thank Ron Peled, Cathy Tuglus, Burke Bundy, and Mark van der Laan for their suggestions and computing advice. Blythe Durbin provided information about the design of her previous diabetes analysis in order to ensure a fair comparison of the proposed method to other predictors. Richard Liang gratefully acknowledges the support of the Natural Sciences and Engineering Research Council (NSERC) of Canada.

References

T. Bäck. *Evolutionary Algorithms in Theory and Practice.* Oxford University Press, 1996.

J. M. Chambers, W. S. Cleveland, and P. A. Tukey. *Graphical Methods for Data Analysis.* Duxbury Press, 1983.

T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms.* MIT Press, 1990.

S. Dudoit and M. J. van der Laan. Asymptotics of cross-validated risk estimation in estimator selection and performance assessment. *Statistical Methodology*, 2(2):131–154, 2005.

S. Dudoit, M. J. van der Laan, S. Keleş, A. M. Molinaro, S. E. Sinisi, and S. L. Teng. Loss-based estimation with cross-validation: Applications to microarray data analysis. In G. Piatetsky-Shapiro and P. Tamayo, editors, *Microarray Data Mining*, volume 5 of *SIGKDD Explorations*, pages 56–68. ACM, 2003. URL `http://www.acm.org/sigs/sigkdd/explorations/issue5-2.htm`.

B. Durbin, S. Dudoit, and M. J. van der Laan. Optimization of the architecture of neural networks using a Deletion/Substitution/Addition algorithm. Technical Report 170, Division of Biostatistics, University of California, Berkeley, 2005. URL `www.bepress.com/ucbbiostat/paper170`.

B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *Annals of Statistics*, 32(4), 2004.

D. B. Fogel. *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence*. IEEE Press, 2005.

S. E. Sinisi and M. J. van der Laan. Deletion/substitution/addition algorithm in learning with applications in genomics. *Statistical Applications in Genetics and Molecular Biology*, 3(1):Article 18, 2004. URL `www.bepress.com/sagmb/vol3/iss1/art18`.

M. Stoll. *Introduction to Real Analysis*. Addison Wesley, 2000.

M. J. van der Laan and S. Dudoit. Unified cross-validation methodology for selection among estimators and a general cross-validated adaptive ϵ-net estimator: Finite sample oracle inequalities and examples. Technical Report 130, Division of Biostatistics, University of California, Berkeley, 2003. URL `http://www.bepress.com/ucbbiostat/paper130`.

D. H. Wolpert and W. G. MacReady. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1):67–82, 1997.