2014 International Joint Conference on Neural Networks (IJCNN)
July 6-11, 2014, Beijing, China

A Practical SIM Learning Formulation with Margin Capacity Control

Thomas Vacek

***Abstract*—Given a finite i.i.d. dataset of the form (y_i, x_i), the Single Index Model (SIM) learning problem is to estimate a regression of the form $u \circ f(x_i)$ where u is some Lipschitz-continuous nondecreasing function and f is a linear function. This paper applies Vapnik's Structural Risk Minimization principle to SIM learning. I show that a risk structure for the space of model functions f gives a risk structure for the space of functions $u \circ f$. Second, I provide a practical learning formulation for SIM using a risk structure defined by margin-based capacity control. The new learning formulation is compared with support vector regression.**

SINGLE-INDEX MODEL (SIM) is a regression technique that extends the Generalized Linear Model in that the link function—instead of being predefined—is learned jointly with the linear model.[1] The problem is to estimate a regression of the form $\hat{y}_i = u \circ f(x_i)$ to minimize some loss function $\frac{1}{n}\sum_{i=1}^{n} l(\hat{y}_i, y_i)$, where $\{(y_i, x_i)\}$ is an i.i.d. sample, u belongs to the class of Lipschitz-continuous nondecreasing functions $\mathcal{M}$, and f belongs to the model space $\mathcal{F}$ of linear functions. For the remainder of this paper, I call u the *link function* and f the *model function*.

SIM has a number of useful features, in principle. It allows for some nonlinearity in the hypothesis class while retaining the interpretablity of linear models. In GLMs, a link function must be chosen based on assumptions about the data distribution. SIM is a non-parametric method, so such assumptions are not required. Moreover, I will show that, with the right capacity control structure, SIM's nonlinear hypothesis class properly contains a linear hypothesis class, yet has the same statistical complexity bound. The principle drawback for SIM is that model inference is practically intractable. In addition to proposing a capacity control structure for SIM, this paper presents an efficient approach to model inference.

According to Vapnik's principle of Structural Risk Minimization [1], a machine learning formulation should choose a hypothesis space that jointly minimizes the empirical risk of the best element of the space and the confidence interval associated with the chosen hypothesis space. I will focus on the following empirical risk functional:

$$R(f) = \frac{1}{n}\sum_{i=1}^{n} l\left(y_i, f(x_i)\right)$$

where $l(y, f) = |y - f|^q$ for $q = 1$ or 2 and y_i is a real-valued regression target.[2]

There are two approaches to a SIM learning formulation:

1) Choose parametric representations of $\mathcal{M}$ and $\mathcal{F}$, and jointly optimize to find the parameters for $u^* \in \mathcal{M}$ and $f^* \in \mathcal{F}$. This is the approach taken in previous papers.
2) Use a learning formulation that (approximately) minimizes $\hat{R}(f)$, where

$$\hat{R}(f) = \inf_{u \in \mathcal{M}} \frac{1}{n}\sum_{i=1}^{n} l(y_i, u \circ f(x_i))$$

without explicitly finding u^*. After f^* is known, find u^* via a 1-dimensional regression.

I focus on the latter approach, which I call *implicit learning*. I call $\hat{R}$ *isotonic-invariant risk* because it is invariant for any isotonic transformation of f. ("Isotonic" is a synonym for "nondecreasing.")

To practically use this approach, the following questions must be addressed:

1) Can a risk structure be defined for SIM?
2) Can a learning formulation be given with desirable properties, such as convexity and computational efficiency?

This paper gives a constructive, affirmative answer to these questions. To answer the first question, I show that the statistical complexity of any hypothesis space of the form $\mathcal{MH} = \{u \circ f : u \in \mathcal{M}, f \in \mathcal{H}\}$ (for any hypothesis space $\mathcal{H}$) can be characterized by a measure of statistical complexity of $\mathcal{H}$. Applying the result to SIM, $\mathcal{MF}$ inherits any risk structure that can be placed on $\mathcal{F}$. To answer the second question, I propose a risk structure for the set of linear functionals that is computationally tractable and amenable to SIM learning, which is based on the margin hyperplane principle.

I. Previous work

The Single-Index Model learning problem has a small body of literature in the statistics field [2], [3], [4], [5]. There are two contemporary papers in the machine learning field that extend the Perceptron to the SIM framework. The first of these papers proposed a formulation called the *Isotron* [6]. It was only of theoretical interest and not practical because it required a large

Thomas Vacek is a PhD candidate at the University of Minnesota (Minneapolis, MN 55455) and a student of Vladimir Cherkassky. He is an employee of Thomson Reuters, which had no part in this work. (email: vacek@cs.umn.edu)

[1]The term 'single index model' is also commonly used in economics to refer an asset pricing model.

[2]It is also possible to consider the ϵ-insensitive loss $|y-f|_\epsilon^q = \min(0, |y-f| - \epsilon)^q$ with a simple modification to my approach.

978-1-4799-1484-5/14/$31.00 ©2014 IEEE

number of samples to prevent overfitting. The second paper set out to address that problem by placing capacity control on the link function [7]. Both of these papers parametrize the class of isotonic functions by piecewise linear functions, and then apply a form of alternating minimization to jointly optimize the representations. This is a nonconvex problem, although the papers still prove learning guarantees.

By contrast, we can interpret this paper's Theorem 1 (*infra*) to argue that capacity control should be applied at the level of the model function. Therefore, just as SVM extended the Perceptron via margin-based capacity control, this paper extends the single-index formulations with margin-based capacity control. In addition, the formulation proposed here is convex.

Finally, the authors of [8] propose learning isotonic transformations of the input features for the classification task. In other words, whereas the single-index formulation finds an isotonic link function following a linear model, this formulation has a linear model following an isotonic transformation of the features. While the idea is practically useful, I am not aware of a way to motivate the approach by learning theory or to characterize the statistical complexity of the hypothesis space.

Viewing ridge regression (RR) and support vector regression (SVR) as members of a family of methods based on trading off model complexity (measured by L^2 regularization of model parameters) with empirical risk, then my proposed method should be included in that family as well. Moreover, I establish a risk structure for my SIM formulation using the same bound (VC dimension of margin hyperplanes) that has been used to analyze SVR and RR (although there are many alternative approaches [9, cf. 12.4]). Therefore, I take support vector regression as the most directly comparable method for evaluation purposes.

II. Risk structure

In this section, I take up the first of the questions required to produce a useful SIM learning formulation: *Can a risk structure be defined for SIM?* A risk structure for regression requires controlling the capacity of the loss class: $\{[l(y_1, f(x_1)), \ldots, l(y_n, h(x_n))] : h \in \mathcal{H}, y \in \mathbb{R}^n\}$ where $\mathcal{H}$ is an arbitrary hypothesis space. Without a prespecified link function, it would seem little could be said about the loss class. However, Vapnik proved that the VC dimension of the loss class for squared (L^2) and quantile (L^1) loss is bounded above and below by a constant of the *level VC dimension* of $\mathcal{H}$ [10, p. 108]. Therefore, I use the bound to ignore the loss class and focus directly on the level VC-dimension of the hypothesis class $\mathcal{H}$. It turns out that this is easy to characterize for the proposed SIM risk structure where $\mathcal{H} \coloneqq \mathcal{MF}$.

The level VC dimension of some real-valued hypothesis class $\mathcal{H}$ is simply the VC dimension of the class under composition with the set of all characteristic functions. More specifically, define characteristic function $\mathcal{X}(z) = 0$ if $z < 0$ and 1 otherwise. Then the level VC dimension of $\mathcal{H}$ is the VC dimension of the set of characteristic functions $\{\mathcal{X}(f - \beta) : f \in \mathcal{H}, \beta \in \mathbb{R}\}$[1, p. 191].

The following theorem shows that the isotonic link function transformation effectively adds no statistical complexity to the underlying hypothesis space, according to the level VC bound:

Theorem 1: Let $\mathcal{H}$ be any family of functions $\mathcal{Z} \to \mathbb{R}$, and let $\mathcal{M}$ be the class of isotonic (nondecreasing) functions $\mathbb{R} \to \mathbb{R}$. Let $\mathcal{MH} = \{u \circ f : u \in \mathcal{M}, f \in \mathcal{H}\}$. The level VC-dimension of $\mathcal{MH}$ is the same as the level VC-dimension of $\mathcal{H}$.

Proof: First, $\mathcal{MH} \supseteq \mathcal{H}$ (since $\mathcal{M}$ contains the identity), so the VC dimension of $\mathcal{MH}$ cannot be less than for $\mathcal{H}$. The opposite inclusion is trivial from the definition of level VC-dimension. Suppose a set can be labeled in a particular way by $\mathcal{X}(u \circ f(x) - \beta)$, and that $u^{pre}(\beta)$ is defined. Then the same labeling can be attained by $\mathcal{X}(f(x) - u^{pre}(\beta))$. If $u^{pre}(\beta)$ is not defined, then take any $\hat{u}$ which is a nondecreasing function that agrees with u on set to be labeled and for which $\hat{u}^{pre}(\beta)$ is defined. (Such a $\hat{u}$ always exists since the set to be labeled is finite.) Then $\mathcal{X}(f(x) - \hat{u}^{pre}(\beta)$ gives the desired labeling.

Therefore, any set that can be shattered by $\mathcal{MH}$ can also be shattered by $\mathcal{H}$, so the level VC-dimension of the former is not greater than the latter. ■

Letting $\mathcal{H}$ be an arbitrary element of a risk structure on $\mathcal{F}$, we see that a risk structure on $\mathcal{F}$ will be inherited by $\mathcal{MF}$. Therefore, if we can find a risk structure for $\mathcal{F}$, we shall have our answer to the question asked at the outset.

A learning formulation implements a risk structure by computing the following for any element of the risk structure: 1) the empirical risk of the best hypothesis in that element and 2) the statistical complexity of that element. I shall shortly propose a tractable approximation of isotonic-invariant empirical risk based on pairwise interactions of examples. It would be convenient to find a measure of the statistical complexity of a set of linear functionals based on the same criterion. Just such a measure was proposed by Vapnik alongside his well-known proof of the VC-dimension of margin hyperplanes [1, pp. 359-361]. The proof idea is similar to Herbrich's bound on statistical complexity for support vector ordinal regression [11], which considers the VC dimension of a hypothesis class on the pairwise differences of a set.

To summarize the previous section, we have the following chain of reasoning:

1) The level VC dimension of the loss class is bounded (up to a constant) by the level VC-dimension of the hypothesis class.
2) The level VC dimension of the hypothesis class $\mathcal{MH}$ (for an arbitrary $\mathcal{H}$) is the same as the level VC dimension of the class $\mathcal{H}$.
3) The level VC dimension of a class $\mathcal{H}$ containing linear functionals can be defined based on pairwise interactions of points, according to Vapnik's proof.

The relationship with support vector regression is the following: The risk structure for SVR has elements which are sets of linear functionals with bounded VC dimension. The risk structure on $\mathcal{F}$ in SIM also has elements which are sets of linear functions with bounded VC dimension. Since $\mathcal{MH} \supsetneq \mathcal{H}$ for any $\mathcal{H} \subset \mathcal{F}$, each risk structure element in SIM in principle properly contains a risk structure element of SVR, yet the level VC bound is the same for both. (Of course, the way in which the best functional is chosen from a given risk structure element differs greatly between the two methods.)

III. Capacity-controlled SIM formulation

Isotonic-invariant empirical risk is difficult from a computational perspective. It is not convex or continuous in f. Assuming some hypothesis space $\mathcal{H}$ contains an open subset of the the set of linear functionals $X \to \mathbb{R}$ (where X is the feature space), it is easy to see that isotonic-invariant risk is constant almost everywhere in Lebesgue measure. It only changes when the sort order of $[f(x_i)]$ changes. To make this tractable, the isotonic-invariant empirical risk will be replaced by a tractable lower bound.

Let us consider the following isotonic-invariant empirical risk functional for $q = 1$ or 2:

$$\min_{u \in \mathcal{M}} \frac{1}{n} \sum_{1}^{n} |y_i - u \circ f(x_i)|^q$$

The bound I propose is based on the minimum loss for each pair of points, considered independently. For just two points, the isotonic-invariant empirical risk is trivial: If the two points are ranked in correct order ($y_j > y_i$ and $f_j > f_i$), then there is an isotonic function that attains zero loss. If the order is inverted ($f_i > f_j$), then the minimum-loss isotonic function is constant, and the loss can be computed elementarily. See Figure 1.

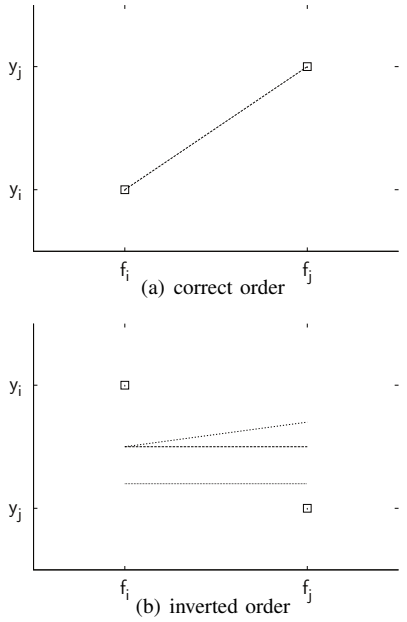


Fig. 1. Minimum-loss isotonic functions for a pair of points in correct order and incorrect order. In the latter case, it apparent that the minimum loss is attained by a constant regressor; the up-sloped line cannot be the minimizer. For L^2 loss, the regressor is unique (middle line), while for L^1 loss any constant function in the range $[y_j, y_i]$ attains the minimum, such as the lowest line in addition to the middle line.

I define the loss as c^1_{ij} and c^2_{ij} for L^1 and L^2 loss, respectively:

$$L^1 : \min_u |y_i - u| + |y_j - u| \quad = |y_i - y_j| \quad := c^1_{ij}$$
$$L^2 : \min_u (y_i - u)^2 + (y_j - u)^2 \quad = \frac{1}{2}(y_i - y_j)^2 \quad := c^2_{ij}$$

The global loss ξ is found by assigning loss to each point in a way that minimizes the overall loss but is consistent with each pairwise loss. That is, if x_i and x_j are inverted, then the global loss should satisfy $\xi_i + \xi_j \geq c_{ij}$. This can be written as a linear program:

$$\begin{aligned} \hat{R}_P(f) = \min_{\xi \geq 0} \frac{1}{n} \sum_{k=1}^{n} \xi_k \\ \text{s.t. } \forall (i,j) \in \mathcal{P} : \\ \xi_i + \xi_j \geq c^q_{ij}\, l(f(x_i), f(x_j), \mathcal{P}) \end{aligned}$$

where $\mathcal{P}$ is the preference set induced from the regression targets by $(i,j) \in \mathcal{P} \Leftrightarrow y_j > y_i$, and l is the pairwise rank inversion indicator defined as

$$l = \begin{cases} 1 & (i,j) \in \mathcal{P} \text{ and } w \cdot x_i < w \cdot x_j \\ 1 & (j,i) \in \mathcal{P} \text{ and } w \cdot x_i > w \cdot x_j \\ 0 & \text{otherwise} \end{cases}$$

This is an underestimator of isotonic-invariant empirical risk, since c^q_{ij} is the *minimum* loss over each pair.[3] The bound is only attained if there exists a regression that bisects each loss pair as in Figure 1(b).

This loss function can also be interpreted in terms of the task assignment problem [12] applied to the pairwise loss matrix $L_{ij} = c^q_{ij}\, l(f(x_i), f(x_j), \mathcal{P})$. The task assignment problem is to select a subset of elements from a matrix so that each row and column has exactly one element selected and the sum of the selected elements is minimized (or maximized, as in this case). The Hungarian (Munkres) algorithm [13], [14] can be interpreted as a constructive proof of the existence of a potential function ξ such that $\sum \xi$ is the optimal assignment cost and for every i, j, $\xi_i + \xi_j \geq c_{ij}$ for some pairwise cost c_{ij}. The isotonic-invariant risk approximation $\hat{R}_P$ is exactly the assignment cost from the potential function with costs L_{ij}; therefore it has an equivalent interpretation as an instance of the task assignment problem.[4]

Adding the penalty for model complexity, the learning formulation for the model function is the following:

$$\begin{aligned} \min \frac{1}{2} w^T w + C \sum_{i=1}^{n} \zeta \\ \text{s.t.} \forall (i,j) \in \mathcal{P} : \\ w^T (x_i - x_j) \geq 1 - \xi_{ij} \\ \zeta_i + \zeta_j \geq c_{ij} \xi_{ij} \\ \zeta \geq 0 \end{aligned}$$

Here, ξ_{ij} is the hinge loss approximation of the pairwise rank inversion indicator. This is a convex QP with $d + n$ variables and $O(n^2)$ constraints. An interior-point algorithm can be formulated (using numerous symbolic eliminations in the KKT system) with step complexity $O\left(n^3 + nds\right)$, where n is the number of examples, d is the dimension of the examples, and s is their average sparsity.[5]

[3] However, the rank-inversion indicator is not convex and this has to be relaxed to a hinge function to make a tractable learning formulation; thus, the underestimation property is lost in the practical algorithm.

[4] A loss function that requires solving the linear assignment problem as also proposed for the ranking task by [15].

[5] To the best of my knowledge, a dual simplex algorithm can be formulated with a step complexity of $O(n^2 + nds)$.

IV. Link function regression

One final piece of the puzzle remains: To find the link function. One-dimensional curve fitting has been studied a great deal. Strictly speaking, this application requires isotonic curve fitting, and choices here are more constrained. Moreover, there is opportunity for expert intervention since 1-d curves are easily visualized. Practically, the link function is found with the following steps:

1) Project each example with the model function.
2) Fit a link function on the graph of the regression targets versus their projections.

The graph is not generally perfectly isotonic, as the model function makes errors. Naive curve-fitting could give a non-isotonic curve, which would certainly be overfit.

I wanted an automated end-to-end system for evaluation on account of the large number of models to be fit and the need to present an unbiased analysis. Thus, I invested a great deal of effort into the problem of finding good link functions without supervision, even though in practice this is less important. I examined isotonic regression [16], linear spline regression, and cubic smoothing splines. Isotonic regression naturally produces jagged models, with level regions punctuated by sharp increases. While the jagged parts tend to be within the noise variance of the data, the lack of smoothness hurts their performance. Moreover, there is not a way to extrapolate outside of the support of the training set, except as a constant function. Linear splines allow for extrapolation and isotonicity and had competitive performance. However, knot number and placement was difficult to automate.

I recommend the cubic smoothing spline [17]. It solves problems of knot placement and number, replacing these with a single smoothness parameter. The drawback is that one cannot enforce isotonicity. One paper [18] proposed a novel isotonic cubic smoothing spline that can be found by solving a second-order cone programming problem. I implemented this with CVX/Sedumi, but the solver often ran into numerical problems, and I chose not to use this method. The classical smoothing spline can be adapted to enforce *knot isotonicity*; that is, the spline is nondecreasing on the set of knots (but possibly decreasing anywhere else). The matter is perhaps trivial because in all the experiments I examined, I only found a few examples of a smoothing spline failing to be isotonic, and I never found a knot-isotonic spline that was not fully isotonic. Still, the isotonic constraint is part of the theoretical method and a formulation should attempt to satisfy it.

I also experimented with replacing the classical squared loss in the smoothing spline with Huber's robust loss (some properties proved in [19]). For the interested reader, the evaluation gives results both for the classical cubic smoothing spline and the enhanced version with knot isotonicity and Huber loss.

V. Evaluation

I seek to validate the following hypotheses in an empirical evaluation:

1) A full analysis of risk bounds is beyond the scope of this paper, but it is axiomatic that they depend positively on both the empirical risk and the statistical complexity of the hypothesis class. That is, reducing the empirical risk will reduce the bound on true risk if the hypothesis class complexity is not changed. I have argued that each element of the SIM risk structure properly contains one for linear SVR, but has the same complexity bound. Therefore, SIM formulations should have better risk bounds and should attain better empirical performance than linear methods.
2) The hypothesis class in SIM is restricted compared to a general nonlinear method. For high-dimensional low sample-size problems, the restricted hypothesis class in SIM (with corresponding lower complexity) should outperform more general nonlinear methods (such as kernel SVR).

A. Experiments

SIM, linear SVR, and RBF kernel SVR are compared on the following tasks:

1) Regression with high-dimensional synthetic data and nonlinear label transformation: Synthetic datasets are created in the following way: For each example, features are drawn from a high-dimensional Gaussian $x \sim \mathcal{N}(0, I)$. Labels are created by composing a link function u with a projection operator w, plus white noise: $y_i = u(w^T x_i) + \epsilon$. The noise level is chosen to preserve a desired signal-to-noise ratio. The true underlying generation process (u and w) is hidden from the learner. Although this process creates datasets which correspond to SIM's hypothesis space, many natural phenomena follow a power law or logistic pattern based on several factors, so this is a reasonable model for real-life datasets.
2) Regression problems using well-known UCI datasets.

All experiments are repeated 10 times on random partitions of the data into training, validation, and test sets. The random partition in one experiment is the same for all methods. The reported measure is normalized root mean squared error (NRMSE): $\sqrt{\frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}{n\sigma^2(y)}}$. In all cases, the training, validation, and test sets are disjoint, with the training and validation sets having the same number of examples and the test set being the remainder of the dataset. For the synthetic experiments, the test set contained 10,000 examples.

Data were prepared by standardizing each feature in the training set, followed by a global scaling so that the largest-magnitude training example had unit norm. This transformation was then applied to the validation and test sets. Regression targets were also standardized.

I chose small-scale evaluations in order to keep computation time reasonable because codes are not optimized and because the automated nature of the model selection (for fairness in evaluation) requires a great deal of computation. Nevertheless, the performance of the method is relative to the sample size and dimensionality, so a small-scale test is conclusive.

B. Model selection

For linear and kernel SVR, the cost parameter C and loss insensitivity parameter ϵ were found using the analytic "rule-

of-thumb" method of Cherkassky [10, p. 448]. The method gives cost and loss-insensitivity parameters based on an analysis of the training set, including estimating its noise variance. In my experiments, each analytic parameter was expanded to a set by a range of scalings $x \to [.25x, .5x, x, 2x, 4x]$, and grid search with validation was used to select the best combination. For RBF kernel SVR, the kernel parameter was selected by considering a range of scalings $(2^{-3}, 2^{-2}, \ldots, 2^3)$ around the median pairwise distance of the training examples.

The SIM formulation has a cost parameter for the model function which is analogous to the C parameter in SVR. This parameter was chosen from the analytic set used in the SVR formulations. The SIM formulation can make use of an ϵ-insensitive loss like SVR. Given that the SIM formulation is based on an optimistic approximation of isotonic-invariant risk, it seems likely that the insensitive loss would be helpful in high-noise settings. However, insensitive loss did not significantly change the results of the experiments I am reporting. It is likely that none of the experiments had high enough noise for this phenomenon to arise. Overall, the insensitive loss function does not seem promising for SIM, as experiments suggest that SIM has little advantage over linear methods in high-noise settings.

In addition, the 1-d link function fit can require parameters. A smoothing spline requires a single parameter. The grid search procedure for selecting the model function and link function is given in pseudocode in Algorithm 1. (Note that the calls to the model function learner can be reused during the link function fitting, and that spline fitting is computationally inexpensive compared to the model function learning.) I found that using validation to choose from a small, fixed set of smoothing parameters $(.96, .97, .98, .99, .995)$ was sufficient to produce good link functions. For the enhanced spline minimizing Huber's robust loss, the loss parameter was chosen to be the noise standard deviation, which was estimated as part of the analytic model selection process.

Algorithm 1 Grid search model selection procedure

Input: training data (y, X), validation data $(\check{y}, \check{X})$, ranking model parameter space $\mathcal{C}$, isotonic model parameter space $\mathcal{U}$.
Output: optimal model and link functions
for $c \in \mathcal{C}, v \in \mathcal{U}$ **do**
  $w_c \leftarrow \text{SIMLearn}(y, X, c)$
  $u_{c,v} \leftarrow \text{smoothingSplineFit}(y, Xw, v)$
  $r_{c,v} \leftarrow \sum l(\check{y}_i, u(\check{X}_i \cdot w))$
end for
$(c, v) \leftarrow \arg\min r_{c,v}$
$(w^*, u^*) \leftarrow (w_c, u_{c,v})$

The evaluation also included *prediction clipping*. For each dataset, a histogram of regression targets was analyzed. If the distribution was uniform on its support, then predictions were restricted to the range of targets seen in the training set (clipped). Clipping prevents the loss from being dominated by a few extreme points that receive highly unlikely predictions.

I also wished to simulate the effect of expert intervention in fitting the link function. For this experiment, the model function is learned with (only) the training data, but once it is fixed, the validation set is combined with the training set and the link function is fit using cross validation over the combined set. The argument for this being a reliable proxy for expert intervention is that it would be reasonable practice for an expert to combine the sets in this way, as 1-d curve fitting is not prone to overfitting. These results have "CV" in their identifier. One might object that this gives the SIM method an unfair advantage over the SVR methods which do not have the benefit of the additional examples. The response is that the major part of the SIM procedure was completed without the additional examples, with curve fitting being a less important part. Nevertheless, a set of results is given ("SIM IS") with the training and validation sets strictly separated.

Finally, this experiment considers two curve fitting techniques: classical cubic smoothing splines and an enhanced version enhanced using Huber loss and enforcing isotonicity at the knots. Results from this experiment have a suffix "SS" or "IS," which is mnemonic for smoothing spline and isotonic spline, respectively.

C. Implementation details

The following is a list of codes used in this paper and their descriptions.

1) SIM model solver: The formulation is a convex quadratic program, and it was implemented using an interior-point solver following a description in [20]. The solver is a naive implementation, resulting in $O(n+d)^3$ step cost, which is suboptimal for high-dimensional or sparse data compared to an optimized formulation. Finding the link function required one of the cubic spline routines, documented below.
2) Classical cubic spline solver: I used Matlab's *csaps* routine, which was written by Carl de Boor following his description in [17, ch. 14].
3) Huber-loss cubic smoothing spline with knot isotonicity: I wrote this code in Matlab, reimplementing de Boor's code with appropriate changes. This requires a quadratic program, which was solved with *quadprog*.
4) Support Vector Regression solver: This is the standard formulation found in LIBSVM and SVM^{light} codes:
$$\min \frac{1}{2} w^T w + C \sum l_\epsilon^1 (y_i - w \cdot x_i - b).$$
It was implemented (in the Langrangian dual) with Matlab's *quadprog* interior-point-convex solver.

D. Synthetic experiments

The data generation process has already been described. I evaluate the results for SNR values of 20, 40, and 80, and logistic $\frac{1}{1+\exp -x}$ and exponential link functions on data realizations with 50 examples and 20 dimensions. Since the data are generated by a function that is in the hypothesis space of the SIM method, it is expected to perform considerably better than linear SVR, which does not contain the function. The poor performance of kernel SVR suggests that the SIM formulation does have an advantage in a high-dimensional low sample-size setting. The results suggest that SIM can recover the true model even in the high-dimensional low-sample setting. It is important to note that noise level, data dimension, and sample size in this experiment were chosen

TABLE I. SYNTHETIC DATA RESULTS

SNR	SIM CV IS	SIM CV SS	SIM IS	L SVR	K SVR
logistic 50 examples × 20 dimensions					
20	.407 (.09)	.407 (.09)	.411 (.09)	.520 (.06)	.557 (.05)
40	.350 (.07)	.353 (.07)	.352 (.07)	.498 (.06)	.536 (.03)
80	.277 (.06)	.279 (.06)	.282 (.06)	.460 (.05)	.523 (.03)
exponential 50 examples × 20 dimensions					
20	.362 (.04)	.364 (.04)	.385 (.04)	.474 (.04)	.494 (.03)
40	.295 (.05)	.283 (.04)	.312 (.05)	.472 (.05)	.499 (.05)
80	.276 (.06)	.273 (.07)	.290 (.07)	.463 (.06)	.481 (.05)

for the effect. With an SNR of 80 and a sample size of 30 (smaller than the evaluation in Table I), SIM performs well in comparison to linear SVR (.44 v. .59). If the SNR is lowered to 20, both the methods perform about the same (.61 v. .63). This example suggests that SIM's nonlinearity isn't useful in high noise problems, but that it fails gracefully, becoming similar to linear SVR when nonlinearity is of no help.

The results of the experiments (mean and std over 10 partitions of the data) are shown in Table I. Recall that three variations of SIM are evaluated in order to compare techniques for fitting a link function. "SIM CV IS" and "SIM CV SS" both use additional points from the validation set to fit the link function (but not the model function), while "SIM IS" uses only the training set. Additionally, "SIM CV SS" uses the classical smoothing spline instead of an enhanced isotonic spline.

As expected, SIM is able to find much better models than the SVR formulations. One of the experiments for the logistic link function with SNR=20 is shown in Figure 2 as an illustration of the method. The upper plot shows the projections of the synthetic data on the x-axis, and y-axis giving the result of transformation by the link function and addition of white noise. The lower plot shows the same information, but with the projection learned by SIM and the link function regression. (The x-axis scale is not significant in the learned projection.)

E. UCI datasets

The datasets considered were Boston Housing [21], Concrete Strength [22], Body Fat,[6] Auto MPG,[7] CPU Small,[8] and Yacht Hydrodynamics[9] [23]. Since I am evaluating a general-purpose tool, I avoided any domain-specific attributes of the problems. The number of predictors for each dataset is given in the results table.

Table II gives the test set loss mean and standard deviations for 10 random realizations of each experiment. The size of the training (and validation) set is reported in the first column. Again, the reported results include three SIM formulations which differ only in terms of the link function procedure. Since they tend to be fairly similar, I will refer to them in common.

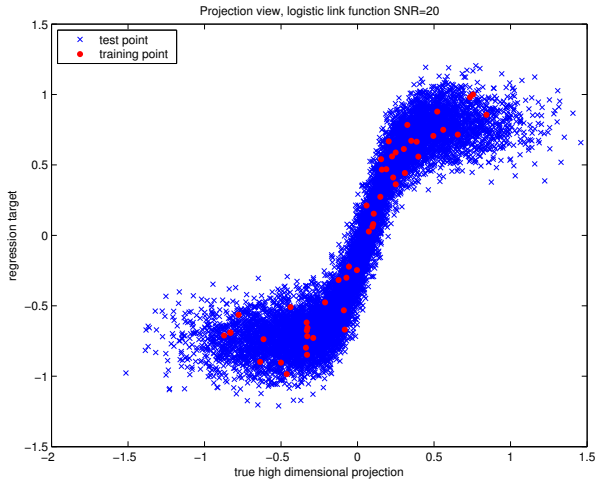
SIM outperformed linear SVR (perhaps modestly) in all experiments but Body Fat and Cpu Small. Visualizing the projections learned by SIM, it is apparent that the learned link

[6]http://lib.stat.cmu.edu/datasets, retrieved from http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/

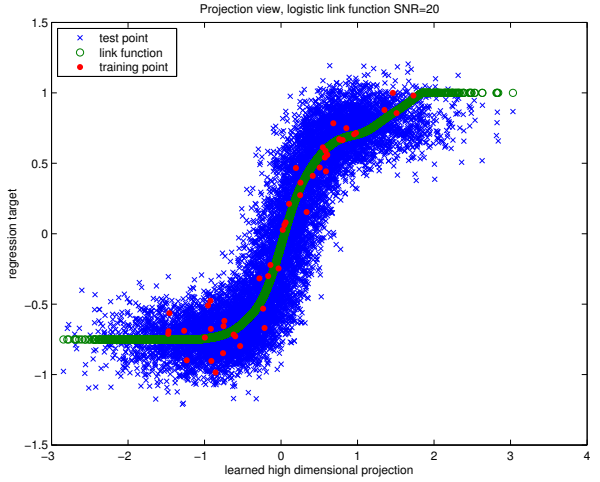
[7]http://lib.stat.cmu.edu/datasets, retrieved from http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/

[8]retrieved from http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/

[9]retrieved from http://archive.ics.uci.edu/ml/datasets/Yacht+Hydrodynamics#



(a) true



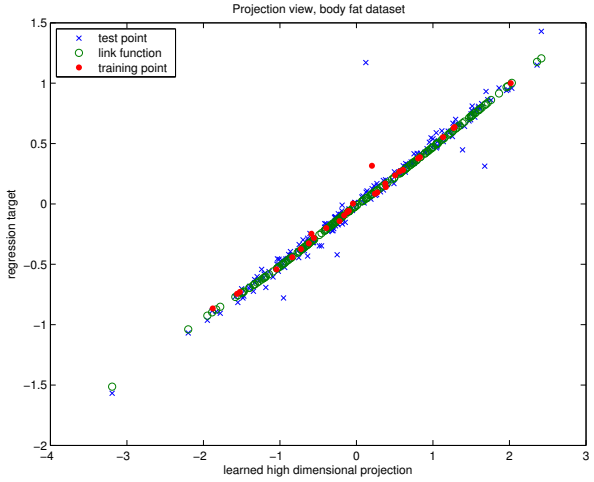
(b) learned

Fig. 2. Illustration of a synthetic dataset. The upper frame gives the regression values versus the ground truth projection of the data. The lower frame shows the projection returned by the SIM learner and the spline that was fit through the training examples. The x-axis scale of the lower plot is not significant, as it depends on the regularization parameter C.

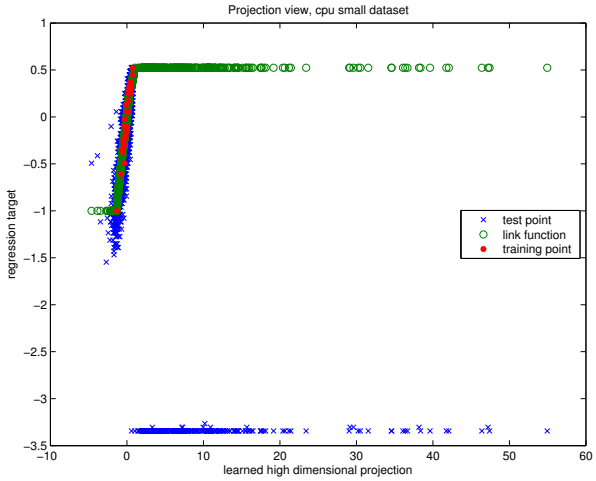
TABLE II. UCI RESULTS

size	SIM CV IS	SIM CV SS	SIM IS	L SVR	K SVR
Boston Housing (13 predictors)					
50	.530 (.03)	.523 (.03)	.533 (.03)	.592 (.04)	.527 (.03)
40	.538 (.03)	.536 (.02)	.551 (.03)	.592 (.03)	.547 (.04)
30	.539 (.04)	.537 (.04)	.543 (.04)	.602 (.04)	.552 (.05)
Body Fat (14 predictors)					
50	.184 (.02)	.185 (.02)	.190 (.03)	.180 (.03)	.207 (.03)
40	.154 (.04)	.158 (.03)	.157 (.05)	.137 (.05)	.193 (.03)
30	.183 (.03)	.189 (.03)	.181 (.03)	.165 (.03)	.229 (.03)
Concrete Strength (8 predictors)					
50	.670 (.04)	.669 (.03)	.677 (.03)	.690 (.03)	.576 (.02)
40	.680 (.03)	.693 (.05)	.691 (.02)	.695 (.03)	.586 (.03)
30	.676 (.06)	.671 (.05)	.709 (.04)	.723 (.04)	.629 (.05)
MPG (7 predictors)					
50	.673 (.04)	.675 (.04)	.685 (.05)	.713 (.03)	.624 (.06)
40	.693 (.07)	.689 (.06)	.698 (.06)	.709 (.05)	.627 (.07)
30	.669 (.07)	.699 (.07)	.702 (.06)	.704 (.07)	.639 (.05)
Cpu small (12 predictors)					
50	.520 (.20)	.516 (.23)	.555 (.21)	.503 (.16)	.591 (.16)
40	.412 (.09)	.423 (.09)	.539 (.27)	.475 (.17)	.578 (.15)
30	.610 (.22)	.597 (.27)	.653 (.17)	.629 (.15)	.704 (.16)
Yacht Hydrodynamics (6 predictors)					
50	.148 (.05)	.148 (.05)	.164 (.05)	.606 (.04)	.502 (.06)
40	.202 (.11)	.207 (.12)	.213 (.11)	.606 (.06)	.515 (.06)
30	.278 (.06)	.274 (.06)	.314 (.06)	.613 (.03)	.589 (.08)

functions are very nearly linear, so the class of nonlinear link functions fails to add to the explaining power of the model. See Figure 3. It is worth noting that the non-optimality for SIM on these datasets is fairly small, though for the cpu dataset we have to bring in validation points to fit a good link function.



(a) body fat



(b) cpu

Fig. 3. Visualizations of the data projected by the optimal SIM model functions for body fact and CPU datasets. For the body fat dataset, the optimal link function is already linear, so it is unsurprising that linear SVR is (slightly) superior to SIM. The cpu dataset is challenging because the test distribution appears to contain two distributions: the points forming the steep upsloped line on the left and the level points trailing low to the right. For the points forming the steep line, a linear link function is optimal. Clipping was critical to avoiding outsized errors on the points to the right, since none of them were part of the training sample.

We also see that SIM generally found better models than kernel SVR, with the exception of the Concrete Strength and MPG datasets. Without investigating deeper, it suggests that these datasets have locality properties that the RBF kernel is able to harness.

Finally, we see that the SIM formulation outperformed linear and kernel SVR significantly on the yacht dataset. Figure 4 is a visualization of the optimal SIM model function projection. It appears to be a case where SIM would work well. I am not aware of an explanation why kernel SVR failed to model the problem.

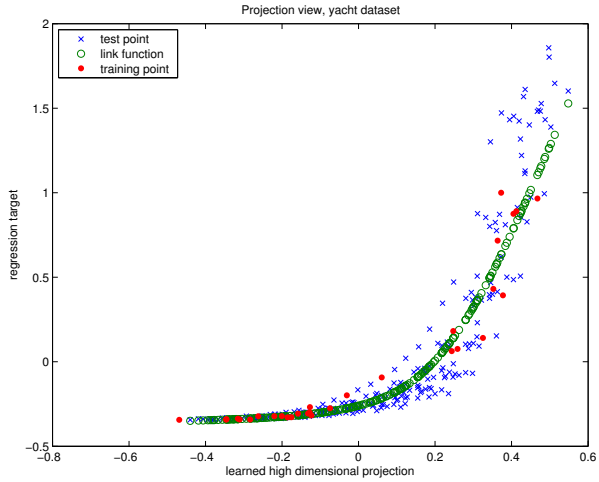


Fig. 4. Visualization of the optimal SIM model function projection of the yacht dataset.

F. Conclusions

As expected, the method is reliably better than linear SVR as expected, but the improvement is somewhat modest—in the range of a 10% relative reduction. We see that SIM can find better models than kernel SVR, but only when the problem is data-constrained. Even when it is not an improvement, SIM is never significantly worse than linear SVR. It attains modest improvements in the test set loss in a number of instances, and significant gains for the synthetic data and one UCI dataset.

The evaluation confirms the intuition from theory, that SIM method retains the benefits of linear regression (interpretation and visualization) and low statistical capacity while providing improved flexibility. Moreover, I have shown that the method can be superior to more general nonlinear methods under certain circumstances, such as in the high-dimensional low sample-size setting. Finally, it appears there exists at least one natural problem (yacht dataset) where SIM hypothesis explains the observations much better than linear or kernel SVR.

Moreover, there are also model selection considerations to keep in mind:

1) The model search space for SIM was much smaller than the space for RBF kernel SVR because SIM has no kernel.
2) SIM did not require a loss insensitivity parameter (as in SVR formulations) for these datasets. The extent to which this holds generally is not known.

Given these merits, I believe the SIM method has the potential to earn a respectable place among high-dimensional regression tools.

G. Scalability

The goal of this paper was to propose a learning formulation based on a theoretical foundation while taking into account computational issues in a high-level way. Nevertheless, to give the reader some idea of the computational requirement, I evaluate SIM on a larger problem. I applied the previous

evaluation procedure to the Ames Housing Dataset [24]. After processing, the data had 318 predictors. A single realization of 500 training examples was used. The naive SIM implementation required 74 seconds to optimize the model function, whereas SVR required about 5 seconds.[10] The naive SIM implementation is known to have poor scaling in d. Much of the solution time came from fill-in when solving the KKT system, since it is represented as a sparse linear equation. The fill-in can be prevented using clever block eliminations. For the comparatively smaller problems in the evaluation, solution times were under one second.

H. Future work

We saw that for most datasets, SIM offered modest improvement, while for some it offered great improvement. At present, the only way to know is to try the method and evaluate using withheld data or cross-validation. It would be useful to have a way to characterize datasets which have a potential gain in performance.

The chief drawback of the method, at present, is the high cost of model inference. This method will require a faster inference algorithm to be highly useful. Much of this is simply a matter of efficient implementation, as the $O\left(n^3 + nds\right)$ scalability of an optimized interior-point implementation is acceptable for many applications. For larger problems, I intend to investigate a dual simplex solver, which would have a step complexity of $O(n^2)$, similar to dual simplex SVM solvers [25] which compare favorably to well-known SVM solvers, such as LIBSVM and SVM^{light}.

REFERENCES

[1] V. Vapnik, *Statistical learning theory*, ser. Adaptive and learning systems for signal processing, communications, and control. Wiley, 1998. [Online]. Available: http://books.google.com/books?id=GowoAQAAMAAJ

[2] H. Ichimura, "Semiparametric least squares (sls) and weighted sls estimation of single-index models," Minnesota - Center for Economic Research, Working Papers, 1991. [Online]. Available: http://EconPapers.repec.org/RePEc:fth:minner:264

[3] J. L. Horowitz and W. H ardle, "Direct semiparametric estimation of single-index models with discrete covariates," 1994.

[4] W. Zhao, R. Zhang, Z. Huang, and J. Feng, "Partially linear single-index beta regression model and score test," *Journal of Multivariate Analysis*, vol. 103, no. 1, pp. 116–123, 2012. [Online]. Available: http://EconPapers.repec.org/RePEc:eee:jmvana:v:103:y:2012:i:1:p:116-123

[5] Z. Huang, Z. Pang, and T. Hu, "Testing structural change in partially linear single-index models with error-prone linear covariates," *Comput. Stat. Data Anal.*, vol. 59, pp. 121–133, Mar. 2013. [Online]. Available: http://dx.doi.org/10.1016/j.csda.2012.10.002

[6] A. T. Kalai and R. Sastry, "The isotron algorithm: High-dimensional isotonic regression." in *COLT*, 2009. [Online]. Available: http://dblp.uni-trier.de/db/conf/colt/colt2009.html#KalaiS09

[7] S. M. Kakade, A. Kalai, V. Kanade, and O. Shamir, "Efficient learning of generalized linear and single index models with isotonic regression," in *Advances in Neural Information Processing Systems 24*, J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Weinberger, Eds., 2011, pp. 927–935.

[8] A. Howard and T. Jebara, "Learning monotonic transformations for classification," in *Advances in Neural Information Processing Systems 20*, J. Platt, D. Koller, Y. Singer, and S. Roweis, Eds. Cambridge, MA: MIT Press, 2008, pp. 681–688.

[10] Kernel SVR had .346 NRMSE and SIM had .352 NRMSE, while linear SVR had .384 NRMSE. The best known model [24] has .28 NRMSE (converted from R^2 score).

[9] B. Scholkopf and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Cambridge, MA, USA: MIT Press, 2001.

[10] V. S. Cherkassky and F. Mulier, *Learning from Data: Concepts, Theory, and Methods*, 2nd ed. New York, NY, USA: John Wiley & Sons, Inc., 2007.

[11] R. Herbrich, T. Graepel, and K. Obermayer, "Support vector learning for ordinal regression," in *International Conference on Artificial Neural Networks*, 1999, pp. 97–102.

[12] R. Burkard, M. Dell'Amico, and S. Martello, *Assignment Problems*, revised reprint ed., ser. SIAM e-books. Society for Industrial and Applied Mathematics (SIAM, 3600 Market Street, Floor 6, Philadelphia, PA 19104), 2012. [Online]. Available: http://books.google.com/books?id=nHIzbApLOr0C

[13] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval Research Logistics Quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955. [Online]. Available: http://dx.doi.org/10.1002/nav.3800020109

[14] J. Munkres, "Algorithms for the assignment and transportation problems," *Journal of the Society for Industrial and Applied Mathematics*, vol. 5, no. 1, pp. pp. 32–38, 1957. [Online]. Available: http://www.jstor.org/stable/2098689

[15] Q. V. Le, A. Smola, O. Chapelle, and C. H. Teo, "Direct optimization of ranking measures." [Online]. Available: http://arxiv.org/pdf/0704.3359

[16] T. Robertson, F. Wright, and R. Dykstra, *Order restricted statistical Inference*, ser. Probability and Statistics Series. John Wiley & Sons Canada, Limited, 1988. [Online]. Available: http://books.google.com/books?id=sqZfQgAACAAJ

[17] C. De Boor, *A practical guide to splines*, ser. Appl. Math. Sci. New York, NY: Springer, 1978.

[18] X. Wang and F. Li, "Isotonic smoothing spline regression," *Journal of Computational and Graphical Statistics*, vol. 17, no. 1, pp. 21–37, 2008. [Online]. Available: http://www.tandfonline.com/doi/abs/10.1198/106186008X285627

[19] Z. Shang, "Convergence rate and bahadur type representation of general smoothing spline m-estimates," *Electronic Journal of Statistics*, vol. 4, pp. 1411–1442, 2010. [Online]. Available: http://dx.doi.org/10.1214/10-EJS588

[20] J. Nocedal and S. J. Wright, *Numerical optimization*. Springer Science+ Business Media, 2006.

[21] D. Harrison and D. Rubinfeld, "Hedonic prices and the demand for clean air," *J. Environ. Economics & Management*, vol. 5, pp. 81–102, 1978.

[22] I.-C. Yeh, "Modeling of strength of high performance concrete using artificial neural networks," *Cement and Concrete Research*, vol. 28, no. 12, pp. 1797–1808, 1998.

[23] I. Ortigosa, R. Lopez, and J. Garcia, "A neural networks approach to residuary resistance of sailing yachts prediction," in *Proc. International Conference on Marine Engineering*, 2007.

[24] D. DeCock, "Ames, iowa: Alternative to the boston housing data as an end of semester regression project," *Journal of Statistics Education*, vol. 19, no. 3, 2011. [Online]. Available: www.amstat.org/publications/jse/v19n3/decock.pdf

[25] C. Sentelle, G. Anagnostopoulos, and M. Georgiopoulos, "Efficient revised simplex method for svm training," *Neural Networks*, vol. 22, no. 10, pp. 1650 –1661, oct. 2011.