Exploiting Expertise Rules for Statistical Data-Driven Modeling

Ling Jian, Member, IEEE, Jundong Li, and Shihua Luo

Abstract—A variety of real-world applications such as complex industry process usually are lack of abundant training samples since the data acquiring process is time and labor consuming. Hence, it is important to utilize the limited training samples to build a sophisticated data-driven model which may improve industry productivity. Recently, nonlinear learning models such as artificial neural networks and support vector machines have shown to be effective in modeling small-scale data by their strong modeling ability. However, these nonlinear learning models work as a black box and are often not human understandable and are difficult to be interpreted. In addition, in many applications, domain experts could provide us valuable expertise knowledge which may help further improve the modeling process. In this paper, we propose to integrate expertise knowledge to the nonlinear learning model to advance the data driven modeling process in real-world applications. Experimental results on six benchmark datasets and a real-world industry application validate the effectiveness of the proposed model.

Index Terms—Nonlinear model, rule based model, multiple kernel learning, blast furnace ironmaking process

I. INTRODUCTION

Accurate data-driven modeling focuses on leveraging machine learning techniques to tackle real-world problems [1] and enjoys its popularity in solving a variety of industry applications such as blast furnace ironmaking process [2] and laser welding process [3], [4]. Typically, given some training data, a machine learning algorithm works by modeling the statistical relationship between the training input and the training output.

Recent years have witnessed an increasing amount of data-driven modeling approaches, ranging from simple linear regression to sophisticated nonlinear statistical models. By virtue of their strong approximation ability, nonlinear statistical models such as artificial neural network (ANN), support vector machines (SVMs), kernel principal component analysis (KPCA) and Kernel Fisher discriminant analysis (KFA) have more powerful modeling ability than simple linear models in practical applications such as monitoring of complicated chemical plants, nonlinear bioprocesses and fault diagnosis, etc. [5]–[7]. However, one major limitation of such kinds of nonlinear models is the lack of transparency and comprehensibility [8]. In other words, these nonlinear models are not human understandable and are difficult to be interpreted.

On the other hand, instead of directly building a sophisticated statistical model, rule based model first mines the rules (e.g., if···then··· rules) from the data [9]. Typically, since rules are normally in the format of rule_antecedent → rule_consequent, they are human readable and interpretable. Due to this reason, rule based model has been extensively studied [10], [11] and are used for building decision support systems such as medical diagnosis and financial fraud detection [12]. Another benefit of rule based models is that they can easily be edited since decision makers can sift through all these rules to edit and update according to their own needs, which is important for many real-world applications. However, since rule based models heavily rely on some data discretization techniques to preprocess the data, usually, they do not have a strong modeling power as nonlinear statistical models.

Considering all of the above factors, it is natural to question whether it is possible to seek an effective way to inherit the merits of nonlinear statistical models and rule based models for real-world industry applications. To answer this question, in this paper, we propose a novel modeling method to incorporate the prior expertise rules into the nonlinear statistical models to form an intelligent model that can be guided by experts. The merits of the proposed model are two folds: improve the prediction ability and enhance transparency of statistical learning models. In particular, if the expertise rules are not available as a prior, we propose to automatically mine the expertise rules from a portion of sampled data first. The major contributions of this paper are summarized as follows:

- Present a principled transformation formula to transform the original data into rule based data according to the mined expertise rules;
- Employ $\ell_2$-norm multiple kernel learning (MKL) algorithm to fuse different rule based data;
- Input the fused rule based model and the nonlinear statistical model into a sigmoid function to generate posterior probabilities for a final prediction model;
- Conduct experiments on six benchmark datasets and a real-world industry application to demonstrate the effectiveness of the proposed framework.
The remainder of this paper is organized as follows. In Section 2, we discuss several related works on improving nonlinear statistical modeling by incorporating prior knowledge. In Section 3, we introduce the proposed framework of rule aided nonlinear statistical modeling. In Section 4, empirical evaluations on real-world datasets are presented to show the performance of the proposed model. In Section 5, we present the conclusion of this paper.

II. RELATED WORK

In this section, we briefly review some related work on improving nonlinear statistical modeling by incorporating prior knowledge. Recently, incorporating prior knowledge into the nonlinear statistical models [13]–[20] has received an increasing attention. Wang et al. [13] and Gabrys [14] studied the hybrid neuro-fuzzy models to combine the learning ability of neural networks and the interpreting abilities of fuzzy systems. Under the framework of maximum entropy, Palmieri et al. [15], [17] and Caticha et al. [16] proposed to model expert knowledge as entropic prior which shows great potentials in the signal processing. Maclin et al. [18] proposed a mechanism for incorporating prior knowledge in the form of simple rules into the support vector methods for both classification and regression tasks. Specifically, prior knowledge is incorporated by adding inequality constraints on the associated data points that match the priori. Zhou et al. [19] proposed a consistent Bayesian framework for introducing strong prior knowledge into a system to extract human gait. To incorporate nonlinear knowledge without its kernelization, Mangasarian and Wild [20] used a theorem to convert nonlinear prior knowledge into linear inequalities. The aforementioned approaches are designed for some specified nonlinear models and are difficult to be extended for other nonlinear models. In this paper, we propose a more general framework to incorporate rule knowledge into common nonlinear statistical models such as ANN, SVMs and naive Bayes, etc.

III. FRAMEWORK OF RULE AIDED NONLINEAR STATISTICAL MODEL

In this section, we fist summarize some symbols used throughout this paper in Table I. We use uppercase character to denote matrices (e.g., A), transpose of A as $A^T$, bold lowercase character to denote vectors (e.g., $a$, $(i,j)$-th entry of $A$ as $A_{ij}$). $I$ is an identity matrix and $1$ is a vector whose elements are all $1$. In the following context, we focus on binary classification problem for a dataset with $l$ samples $\{\mathbf{x}_k, y_k\}_{k=1}^l$, where $\mathbf{x}_k = (x_{k1}, \ldots, x_{kn}) \in \mathbb{R}^n$ and $y_k \in \{-1, 1\}$.

In the following of this section, we introduce details about the proposed framework which exploits expertise rules for nonlinear statistical data-driven modeling in four steps: (i) format expert rules or extract rules from the sampled dataset if the expertise rules are not available as a priori; (ii) transform the original data into rule based data according to the mined rules; (iii) employ multipurpose kernel learning (MKL) algorithm to derive a fused rule based model from different rule based data; (iv) apply sigmoid function to cast the numerical prediction results of data based and rule based models into posterior probabilities for model ensemble. The illustration of the proposed approach is shown in Fig. 1. We will introduce the detailed four steps from Section III-A to Section III-D.

A. Rule Extraction through Decision Tree

If the expertise rules are known a priori, we can format these expertise rules into standard if· · · then· · · logical rules. Otherwise, one can use decision tree to obtain expertise rules from sampling data. Classification and regression trees (CART) [21] is one of the most popular rule extraction methods. Therefore, in the current work, we choose CART to extract rules directly from data. We briefly review CART as follows. The decision tree can be linearized into decision rules, where the outcome is the contents of the leaf node, and the conditions along the path form a conjunction in the if clause [21]. It has been shown that finding a minimal decision tree consistent with the training set is NP-hard [22]. Consequently, CART, one kind of heuristic method, is proposed to solve the problem. CART is greedy by nature and constructs the decision tree in a top-down, recursive manner (also known as “divide and conquer”). In particular, CART has two main steps: one step is to produce the maximal binary tree by recursive partitioning; the other step is to prune this maximal binary tree to reduce the tree complexity. During the construction of the maximal binary tree, the Gini diversity index is used to recursively find the most suitable input variable and its corresponding threshold for splitting tree nodes. Decision tree is easy to interpret and implement. In addition, it performs well if a few highly relevant attributes exist. However, in the case of many complex interactions exist in the attributes, e.g., blast furnace ironmaking process, its accuracy may seriously deteriorate. Therefore, in the current work, CART is used as a preprocessing step to extract if· · · then· · · rules.

B. Rule based Data Expression

For some expert, assume that there are $m$ items if· · · then· · · logical rules composed of the following two kinds

- if $x_{i_1} \in V_{i_1}^d$ and · · · and $x_{i_t} \in V_{i_t}^d$, then $y = y^d$. 

<table>
<thead>
<tr>
<th>TABLE I</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Symbols description</strong></td>
</tr>
<tr>
<td>Notations</td>
</tr>
<tr>
<td>---</td>
</tr>
<tr>
<td>$b$</td>
</tr>
<tr>
<td>$e_i$</td>
</tr>
<tr>
<td>$n$</td>
</tr>
<tr>
<td>$l$</td>
</tr>
<tr>
<td>$y_i$</td>
</tr>
<tr>
<td>$\hat{y}_i$</td>
</tr>
<tr>
<td>$\nu$</td>
</tr>
<tr>
<td>$\tau$</td>
</tr>
<tr>
<td>$\mu_i$</td>
</tr>
<tr>
<td>$\alpha \in \mathcal{R}^l$</td>
</tr>
<tr>
<td>$w$</td>
</tr>
<tr>
<td>$x_i \in \mathcal{R}^n$</td>
</tr>
<tr>
<td>$y \in \mathcal{R}^d$</td>
</tr>
<tr>
<td>$K \in \mathcal{R}^{l \times l}$</td>
</tr>
</tbody>
</table>
Fig. 1. Exploiting expertise rules for nonlinear learning modeling framework includes four steps: ① experts’ knowledge is extracted as if \( \cdots \) then rules; ② original data are transformed into rule based data according to the rules; ③ multiple kernel learning is used to integrate different rule based data and deduce the rule based predictor; ④ sigmoid function casts the numerical prediction results of data based and rule based models into posterior probabilities for model ensemble.

- if \( x_{i1} \in V_{i1}^j \) or \( \cdots \) or \( x_{i1} \in V_{i1}^j \), then \( y = y^j \).

Here \( \{V_{i1}^j\}_{q=1}^{\cdots m} \) is called basic rule interval including three types, i.e., \( (-\infty, a) \), \( [b, +\infty) \) and \( [c, d) \). \( \{i_1, \cdots, i_1\} \subset \{1, \cdots, n\} \) is the index of features used in the \( j \)th rule. Next we define the \( i_q \)th feature of the \( k \)th sample’s membership, denoted as \( \delta_{V_{i1}^j}(x_{kiq}) \in [0, 1] \), on the rule interval \( V_{i1}^j \) as

\[
\delta_{V_{i1}^j}(x_{kiq}) = \begin{cases} 
\frac{1}{1 + e^{s_{i1}^q - x_{kiq}}}, & \text{if } V_{i1}^j \triangleq [b, +\infty) \\
\frac{e^{s_{i1}^q - \sigma_{i1}^q}}{e^{s_{i1}^q} + e^{\sigma_{i1}^q}}, & \text{if } V_{i1}^j \triangleq [c, d) \\
\frac{1}{1 + e^{x_{kiq} - b_{i1}^q}}, & \text{if } V_{i1}^j \triangleq (-\infty, a),
\end{cases}
\]

where \( s_{i1}^q = \min_k \{x_{ki1} | x_{ki1} \in V_{i1}^j, k = 1, \cdots, l\} \) and \( b_{i1}^q = \max_k \{x_{ki1} | x_{ki1} \in V_{i1}^j, k = 1, \cdots, l\} \). \( m_{i1}^j \) and \( \sigma_{i1}^q \) denote the mean and standard deviation of the \( i_q \)th feature of samples in the set \( \{x_{ki1} | x_{ki1} \in V_{i1}^j, k = 1, \cdots, l\} \). Then, operators \( \lor \) and \( \land \) are used to define the or-type and and-type rule’s density support of input sample \( x_k \) as

\[
r_j(x_k) = \lor_{q=1}^l \delta_{V_{i1}^q}(x_{kiq}) = \max_{1 \leq q \leq l} \delta_{V_{i1}^q}(x_{kiq}),
\]

and

\[
r_j(x_k) = \land_{q=1}^l \delta_{V_{i1}^q}(x_{kiq}) = \min_{1 \leq q \leq l} \delta_{V_{i1}^q}(x_{kiq}),
\]

respectively. \( r_j(x_k) \) depicts the support degree of the \( k \)th sample \( x_k \) to the \( j \)th rule. In this way, we generate the rule based data for the specified expert as

\[
R = \{r_k, y_k\}_{k=1}^L
\]

where \( r_k = (r_1(x_k), \cdots, r_m(x_k)) \in [0, 1]^m \). As expert knowledge is embedded into the rule based data, the rule based datasets are different from one another. In the next subsection, we will discuss how to employ MKL techniques to integrate different rule based datasets for the purpose of data fusion.

C. Rule based Data Fusion by \( \ell_2 \) Multiple Kernel Learning

MKL has shown its potential to integrate heterogeneous information in a variety of applications [23]. Yu et al. [24] and Kloft et al. [25] studied \( \ell_2 \) MKL which can lead to non-sparse optimal kernel coefficients. Empirically, \( \ell_2 \) MKL is superior to sparse integration method since it thoroughly combines complementary information in the heterogeneous data sources. In the current work, we employ \( \ell_2 \) MKL to fuse rule based datasets generated in Section III-B. It is worth noting that the proposed framework can be applied to any kernel based classification model such as SVMs and least square SVMs (LS-SVMs), etc. Suppose that the model’s optimal value is \( \omega(K) \), where \( K \) is a specified kernel matrix. \( \ell_2 \) MKL works by minimizing \( \omega(\sum_{i=1}^p \mu_i K_i) \) under the constraint of \( \|\mu\|_2 = 1 \) to fuse rule based datasets \( R_i \) \((i = 1, \cdots, p)\). Here, \( K_i \) is the corresponding kernel matrix generated by \( R_i \). In the following, we select LS-SVMs as a special case to show the procedure of rule based data fusion in detail. We briefly review the LS-SVMs and \( \ell_2 \) MKL as follows.

LS-SVMs proposed by Suykens and Vandewal [26] has been successfully applied in many real-world applications. LS-SVMs employs equality constraints and square loss in its formulation, its analytical solutions can be obtained by solving a saddle point system. Given training set \( \{x_i, y_i\}_{i=1}^L \), the LS-SVMs model can be formulated as follows.

\[
\begin{align*}
\min_{w, b, \varepsilon} & \quad \frac{1}{2} w^T w + \frac{1}{2\nu} \sum_{k=1}^L \epsilon_k^2 \\
\text{s.t.} & \quad y_k = w^T \Phi(x_k) + b + \epsilon_k, \quad k = 1, \cdots, l.
\end{align*}
\]

where \( \Phi(x) \) is the mapping from input space to the feature space, \( \epsilon_k \in R \) is the error, and \( \nu \) is a regularization parameter that balances the margin and the error. Lagrangian function of Eqs.(5,6) is

\[
L(w, b, \varepsilon; \alpha) = \frac{1}{2} w^T w + \frac{1}{2\nu} \sum_{k=1}^L \epsilon_k^2 - \sum_{k=1}^L \alpha_k (w^T \Phi(x_k) + b + \epsilon_k - y_k)
\]

where \( \alpha_k \) is the \( k \)th Lagrange multiplier. As for convex programming Eqs.(5,6), the Slater constraint qualification holds [27]. Therefore, \( \min_{w, b, \varepsilon} L(w, b, \varepsilon; \alpha) = \min_{\alpha} \max_{w, b, \varepsilon} L(w, b, \varepsilon; \alpha) \).
latter is well known as the dual programming of the former problem.

\[
\max_{\alpha} \quad \alpha^T y - \frac{1}{2} \alpha^T K \alpha - \frac{\nu}{2} \alpha^T \alpha \tag{7}
\]

s.t. \[
\sum_{i=1}^{l} \alpha_i = 0, \tag{8}
\]

where \(K_{ij} = k(x_i, x_j)\) and \(k(\cdot, \cdot)\) is a user specified kernel function. By solving the dual programming in Eqs.(7,8), we obtain the decision function

\[
f(x) = \sum_{i=1}^{l} \alpha_i y_i k(x_i, x) + b. \tag{9}
\]

Obviously, the optimal value of quadratic programming Eqs.(7,8) depends on the kernel \(K\). Denote the aforementioned dual programming’s optimal value as \(\omega(K)\)

\[
\omega(K) = \max \{ \alpha^T y - \frac{1}{2} \alpha^T K \alpha - \frac{\nu}{2} \alpha^T \alpha | \alpha^T 1 = 0 \} = - \min \{ \frac{1}{2} \alpha^T K \alpha + \frac{\nu}{2} \alpha^T \alpha - \alpha^T y | \alpha^T 1 = 0 \}.
\]

In the context of MKL, one would try to find the optimal combinations of a series of pre-specified kernels \(K_1, \cdots, K_p\) under some constraints, e.g., kernel coefficients satisfy convex constraints. Generally, MKL is formulated as \(\min_{\mu} 2 \omega(\sum_{i=1}^{p} \mu_i K_i)\), and for the convenience of presentation, we transform it into an equivalent format \(\max_{\mu} -2 \omega(\sum_{i=1}^{p} \mu_i K_i)\).

\[
\max_{\mu \geq 0, ||\mu||=1} -2 \omega(\sum_{i=1}^{p} \mu_i K_i) \tag{10}
\]

\[
= \max_{\mu \geq 0, ||\mu||=1} \min_{1 \leq \alpha^T 1 \leq 0} \{ \sum_{i=1}^{p+1} \mu_i \alpha^T K_i \alpha - 2 \alpha^T y \} \tag{11}
\]

\[
= \max_{\mu \geq 0, ||\mu||\leq 1} \min_{1 \leq \alpha^T 1 \leq 0} \{ \sum_{i=1}^{p+1} \mu_i \alpha^T K_i \alpha - 2 \alpha^T y \}. \tag{12}
\]

Under the assumption that \(K_i (i = 1, \cdots, p + 1)\) are positive definite, the optimal solution of Eq.(11) satisfies \(||\mu^*|| = 1\), and the non-convex constraint \(||\mu|| = 1\) can be relaxed to convex constraint \(||\mu|| \leq 1\). So the equality sign \(\max\) holds. To tackle the above reformulated problem, Sonnenburg et al. reformulated this max-min problem as the following semi-infinite programming (SIP) [29]

\[
\max_{\mu, \theta} \quad \theta \tag{13}
\]

s.t. \[
||\mu|| \leq 1, \tag{14}
\]

\[
\mu_i \geq 0, i = 1, \cdots, p + 1, \tag{15}
\]

\[
\sum_{i=1}^{p+1} \mu_i f_i(\alpha) - 2 \sum_{k=1}^{l} \alpha_k y_k \geq \theta, \tag{16}
\]

\[
\sum_{k=1}^{l} \alpha_k = 0, \tag{17}
\]

where \(f_i(\alpha) = \alpha^T K_i \alpha, i = 1, \cdots, p + 1\). In the formulation of SIP, the LS-SVMs \(\ell_2\) MKL can be decomposed into two subproblems and these two subproblems can be solved iteratively. Among these two subproblems, one is a single kernel LS-SVMs learning problem which can be reduced to a saddle point system (a special linear equations problem) and the other one is a coefficients optimization problem which can be reduced to a quadratically constrained linear program problem of small-scale. In this way, the LS-SVMs \(\ell_2\) MKL can be solved efficiently.

**D. Ensemble Data and Rule based Model by Sigmoid Fitting**

Given a classifier \(f(x)\), its decision value indicates the confidence of the corresponding prediction \(\tilde{y} = sgn(f(x))\). Typically, the higher the absolute value of the decision value, the higher we are confident about the prediction. In our study, we have two models, one is data based model \(f_{data}\) and the other is rule based model \(f_{rule}\). Although two different models may compensate each other for a final prediction, the output of these two classifiers may also contradict with each other. Therefore, we should be cautious when combining the prediction results of these two different models. In the current study, we employ Platt’s method [30] to normalize the decision value of the classifier into an interval between \([0, 1]\) through fitting a Sigmoid function. Details are shown as follows. Putting the learning model’s decision value into a Sigmoid function

\[
P(y = 1|x) \approx \frac{1}{1 + \exp(\varepsilon f(x) + \gamma)}, \tag{18}
\]

and fixing the parameters \(\varepsilon\) and \(\gamma\) by maximizing the likelihood function of the training data, i.e., \(\max_{\varepsilon, \gamma} \Pi_{k=1}^{l} P(y = y_k|f(x_k))\), we have:

\[
\max_{\varepsilon, \gamma} \Pi_{k=1}^{l} P_k^{\varepsilon}(1 - P_k)^{1-t_k}, \tag{19}
\]

where

\[
\begin{align*}
    & t_k = \begin{cases}
        1, & \text{if } y_k = 1 \\
        0, & \text{if } y_k = -1
    \end{cases} \\
    & P_k = \frac{1}{1 + \exp(\varepsilon f(x_k) + \gamma)}.
\end{align*}
\]

In practice, the optimization problem in Eq.(19) is solved by minimizing the negative log transformation of the objective function:

\[
\min_{\varepsilon, \gamma} - \sum_{k=1}^{l} t_k \log(P_k) + (1 - t_k) \log(1 - P_k). \tag{20}
\]

The optimization problem in Eq.(20) can be effectively solved by Levenberg-Marquardt algorithm [31]. When the optimal parameter pair \((\varepsilon, \gamma)\) are derived for \(f_{data}\) and \(f_{rule}\), we get the probability output for the data based and the rule based models, i.e., \(P_{data}\) and \(P_{rule}\), respectively. At last, rule aided statistical data-driven model can be obtained by the ensemble of \(P_{data}\) and \(P_{rule}\) in a straightforward way as follows:

\[
\tilde{y} = \begin{cases}
    1, & \text{if } P_{data} + P_{rule} \in [0.5, 1] \\
    -1, & \text{if } P_{data} + P_{rule} \in [0, 0.5].
\end{cases}
\]
It is worth noting that in the current work the weights of two models are equal. Some adaptive strategy to dynamically adjust the weights of different models according to their predicting accuracies may further improve the prediction accuracy and it deserves deep investigation in the future.

The pseudocode for the rule aided nonlinear modeling algorithm is summarized as follows.

**Algorithm 1 Rule Aided Statistical Data-driven Modeling**

**Input**: data set $\mathcal{D} = \{x_i, y_i\}_{k=1}^1$, rule sets $\mathcal{E}_1, \mathcal{E}_2, \cdots, \mathcal{E}_p$

**Output**: decision value $\hat{y}$

1. With $\mathcal{E}_1$, transform $\mathcal{D} = \{x_k, y_k\}_{k=1}^1$ into rule based data $\mathcal{R}_1 = \{r_1^k, y_k\}_{k=1}^1$ through Eqs.(1-4);

2. Generate kernel matrix $K_1$ according to $\mathcal{R}_1 (i = 1, \cdots, p)$;

3. Employ $\ell_2$ MKL algorithm to learn $\mu_i (i = 1, \cdots, p)$ through Eqs.(13-17) and derive the rule based classifier $f_{rule}$;

4. Train LS-SVMs classifier on $\mathcal{D} = \{x_k, y_k\}_{k=1}^1$, and get the data based classifier $f_{data}$;

5. Optimize $\varepsilon$ and $\gamma$ in Eq.(18), and transform the decision values $f_{rule}$ and $f_{data}$ into posterior probabilities;

6. Ensemble rule based model and data based model, and output the decision value $\hat{y}$ through Eq.(21).

**IV. RESULTS AND DISCUSSIONS**

In this section, we conduct experiments to evaluate the performance of the proposed rule aided statistical data-driven modeling framework. First, we present an illustrative example to show the basic idea of the proposed rule aided data-driven modeling framework. Next, we use six benchmark datasets for experimental evaluation. Then, a real-world application problem (tendency prediction of the thermal state of blast furnace) is presented. We evaluate the performance of rule aided statistical data-driven modeling method on two medium-sized blast furnaces with volume of 2500 m$^3$ and 2000 m$^3$, respectively.

All experiments are performed in MATLAB 7.14 environment on a PC with 3.4 GHz Intel Core i5 processors and 8G RODRAM running under the Windows 7 operating system.

**A. An Illustrative Example**

In this experiment, we select a subset of the 20 newsgroup dataset i.e., “Christianity” and “Mac Computer” to form the samples. Each document is represented by a vector of 19487 dimension using the “bag-of-words” model, including 997 “Christianity” documents and 958 “Mac Computer” samples. Suppose that experts use the keywords in the document to perform classification task. There are two experts denoted as expert1 and expert2 whose expertise rules are shown in Fig. 2. Expert1 thinks that if keywords “christ” or “god” or “love” appears in one document, this documents may be classified as “Christianity”, on the other hand if keywords “storage” or “computer” or “display” appears in one document, it should be classified as “Mac Computer”.

Eqs.(1-4) generate the rule based data for expert1 and expert2 denoted as $\mathcal{R}_1$ and $\mathcal{R}_2$, respectively. The rule based data are further transformed into kernel matrices $K_1$ and $K_2$. Here, we adopt the Laplacian kernel:

$$k(x_i, x_j) = \exp(-\|x_i - x_j\|/\tau)$$

due to its robustness to noise and its well performance in modeling complex systems. The parameter $\tau$ is specified as the dimension of inputs. LS-SVMs based $\ell_2$ MKL, i.e., Eqs.(13-17) is used to learn the kernel coefficients. In addition, regularization parameter $\nu$ is also jointly optimized. In this way, it enables us to integrate different expert knowledge and to derive the rule based classifier $f_{rule}$. In addition, we can also build a single kernel LS-SVMs model through Eqs.(5-9) and derive the data based classifier $f_{data}$ based on the original data. Here, linear kernel is used in this illustrative example for the inputs are of high-dimensional and the regularization parameter $\nu$ is set as 1. By feeding the decision values of the model into a sigmoid function through Eqs.(18-20), we can transform the decision value of $f_{rule}$ and $f_{data}$ into two posterior probabilities. Finally, we get the rule aided statistical data-driven model through aggregating of the rule based model and data based model through Eq.(21).

In this illustrative example, the dataset is divided into two separate part randomly: select $d$% samples to form test set and the rest form the training set, here $d \in \{10, 30, 50, 70, 90\}$. The random division of the data into test set and training set is repeated 10 times and the average accuracy is reported in Fig. 3. It can be seen from Fig. 3 that data based model (say, Data) outperforms the rule based model (say, Rule) significantly for the expert knowledge is extremely limited (only 5-6 keywords used by every expert). Even though, the integration of expert expertise can further improve the data based model consistently.

Next, we perform validation on benchmark datasets and real-world application datasets. We execute the following procedures for each dataset.

1) To avoid the problem of overfitting, the dataset is divided into two separate part: the first 200 samples, denoted as $\mathcal{D}_R$, is used for rule extraction; the rest dataset, denoted as $\mathcal{D}_V$, is used for model validation.

2) A bootstrap sample $\mathcal{D}_B$ is selected from $\mathcal{D}_R$ and then we generate the decision tree from $\mathcal{D}_B$. This procedure is repeated four times and gives four different decision trees $T_1, T_2, T_3$, and $T_4$.

3) The validation set is randomly splitted into a test set and a training set: randomly select $d$% samples from
B. Evaluating the Performance on Benchmark Datasets

In this subsection, we evaluate the empirical performance of the proposed rule aided statistical data-driven modeling algorithm on a number of publicly available benchmark datasets. Details of the datasets are listed in TABLE II. All the datasets can be obtained from the UCI machine learning repository [32] and the LIBSVM website [33]. Take the German dataset for an example, we generate four decision trees as shown in Fig. 4. Afterwards, we obtain the rules from the leaf nodes of the decision trees. For each decision tree (expert), we transform the original data in the validation set into rule based data according to Eqs.(1-4). In this way, we get four rule based datasets $R_1$, $R_2$, $R_3$, and $R_4$. The rule based data are further transformed into kernel matrices $K_1$, $K_2$, $K_3$, and $K_4$ through the Laplacian kernel function with the kernel width $\tau$ specified as the dimension of inputs.

Fig. 5 summarizes the results of the experiments. It can be seen from Fig. 5 that in almost all cases, the rule aided statistical data-driven model (say, Ensemble) consistently outperforms the data based model (say, Data) and the rule based model (say, Rule).

![Figure 3: A toy experiment about two-class document classification.](image)

![Figure 4: Four decision trees for rule extraction on German. Red circle stands for positive point and green square stands for negative point.](image)

![Figure 5: Experimental results on six benchmark datasets.](image)

### TABLE II

**Details of Four Benchmark Datasets**

<table>
<thead>
<tr>
<th>Dataset</th>
<th># Positives</th>
<th># Negatives</th>
<th># Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cancer</td>
<td>239</td>
<td>444</td>
<td>9</td>
</tr>
<tr>
<td>Codma</td>
<td>180</td>
<td>320</td>
<td>8</td>
</tr>
<tr>
<td>German</td>
<td>300</td>
<td>700</td>
<td>24</td>
</tr>
<tr>
<td>Heart</td>
<td>137</td>
<td>160</td>
<td>13</td>
</tr>
<tr>
<td>Ijcnn1</td>
<td>111</td>
<td>889</td>
<td>22</td>
</tr>
<tr>
<td>Ionosphere</td>
<td>126</td>
<td>225</td>
<td>33</td>
</tr>
</tbody>
</table>

For a through comparison, we conduct 10-fold cross-validation experiments to compare the performance of the proposed rule aided statistical data-driven model with two basic methods (i.e., rule based classifier $f_{Rule}$ and LS-SVMs model $f_{Data}$) and two ensemble methods (i.e., rule ensemble...
method bagging decision forest model $f_{Bagging}$ and weak classifiers ensemble model $f_{AdaBoost}$ on the validation sets. As decision trees use a “divide and conquer” strategy in training the model, they do not perform well if many complex interactions exist among the attributes [34]. Decision forest aims to improve the predictive performance of a single decision tree by training multiple trees and combining multiple prediction results. Bagging (Bootstrap aggregating) is a simple yet effective method for generating an ensemble of models in general and decision forest in particular. Each decision tree in the forest is trained on a sample of instances taken as a replacement from the original training set. All trees are trained using the CART algorithm. In the experiments, 50 bootstrap replicates are used for the bagging decision forest model. AdaBoost (short for “Adaptive Boosting”) consists of two parts: a weak classifier and a boosting part. The weak classifier tries to find the optimal threshold in one of selected features to separate the data into two classes. The boosting part calls the base classifier iteratively, and after each classification step it changes the weights of misclassified samples. This creates a cascade of “weak classifiers” which are combined into a weighted sum representing the final output of the boosted classifier. In this experiments, the number of training iterations for the AdaBoost model is set as 50. Compared with the basic classifiers on six benchmark datasets, rule aided statistical data-driven model, i.e., $f_{Ensemble}$ ranks 2nd in accuracy on Codrna dataset, and 1st on the rest five datasets. Following the Statlog method of ordering classifiers by their average ranks [35], $f_{Ensemble}$ is considered the best classifier on the six test datasets with an average rank of 1.17. The average rank of $f_{AdaBoost}$, $f_{Bagging}$, $f_{Data}$, and $f_{Rule}$ is 2.67, 3.00, 3.17 and 5, respectively.

C. Tendency Prediction of Thermal State of Blast Furnace

Blast furnace (BF), a metallurgical reactor, has been extensively used to produce pig iron, often called as hot metal. The spatial and temporal distribution of variables in blast furnace process, together with high temperatures, pressure, and erosive environment make it very hard to assess the internal state of BF by direct measuring. The silicon content of the hot metal (often denoted by Si) always acts as a chief indicator to reflect the molten iron quality and also the thermal state of BF: A high silicon content generally indicates a surplus of coke (a large “energy reserve”) in the hearth of the furnace, while a low value indicates a depleted energy reserve. Due to the above facts, the hot metal silicon content needs to be controlled with caution and its prediction has attracted much attention [37]. Gao et al. pioneered the tendency prediction of thermal state of blast furnace hearth through a classification model [36]. In this classification model, the tendency change of Si is selected as an output variable. Typically, an increasing Si is denoted as 1 whereas a decreasing Si is denoted as -1 because the evolution of the Si is strongly affected by the variables related to the charged solid raw materials, the compressed gaseous materials and other hot metal components. Following [36], we select eight variables (see Table IV) in modeling the blast furnace (a) and seven variables (see Table V) in modeling the blast furnace (b).

There are, in total, 800 consecutive data points sampled from the selected blast furnaces. The sampled data are divided into two separate parts: the first 200 data samples are used for rule extraction and the residual 600 data samples form the validation set. Bootstrap strategy is used to generate multiple decision trees based on the first 200 samples: repeatedly sample data points with replacement and fit decision trees to these samples. We generate four decision trees for blast furnace (a) and seven variables (see Table V) in modeling the blast furnace (b).

Previous studies show that the differentiated data of some available variables exhibit stronger correlation to the tendency change of Si compared with the original data [36]. In the case study of blast furnace (a), we select the differentiated data of blast volume, feed speed, oxygen enrichment percentage, CO percentage in top gas, basicity of ingredients, smelt intensity denoted as $\Delta BV$, $\Delta FS$, $\Delta O_2$, $\Delta CO$, $\Delta BI$, $\Delta SI$, respectively, as model input variables. In addition, the latest silicon content and sulfur content, denoted as $Si_{n-1}$ and $S_{n-1}$ are also selected as input variables since they are highly correlated to the tendency change of the Si. For blast furnace (b), differentiated data of BV, EFe, TFe, BN, DT and the original data of $Si_{n-1}$ and GP are selected as model input. Details of these variables can be found in TABLE V.

To avoid attributes with larger magnitudes dominating the attributes with smaller magnitudes in the data based model, we normalize each of the original data through the equation $x_{nor} = x_{raw} - (\text{mean of } x_{raw}) / (\text{std of } x_{raw})$, where $x_{nor}$ denotes the normalized data and $x_{raw}$ denotes the original data.

Fig. 6 summarizes the results of Si tendency prediction. It can be seen from Fig. 6 that the rule aided statistical data-driven model performs best for 12 times out of 17 tries on blast furnace (a) and it consistently outperforms the
we propose a general rule aided nonlinear statistical model to incorporate expertise knowledge into the data-driven model. In particular, in the first step, we propose to mine the expertise rules from the data if prior knowledge is not available. Then we investigate how to transform the original data into rule based data according to the expertise rules and how to integrate multiple rule based datasets through multiple kernel learning. Later, we propose an effective method to ensemble rule based model and data based model. Experimental results on benchmark datasets and a real-world application validate the effectiveness of the proposed model.

APPENDIX

In the Appendix, we compare the $\ell_2$ MKL algorithm Eq.(10) with $\ell_1$ MKL which can be formulated as:

$$\max_{\mu \geq 0, ||\mu||_1=1} -2\omega \sum_{i=1}^{p+1} \mu_i K_i.$$

Eq.(23) can further be transformed into SIP and can be solved efficiently. In addition, we also compare the performance of Laplacian kernel with Gaussian kernel

$$k(x_i, x_j) = \exp(-||x_i - x_j||^2/\tau)$$

on three benchmark datasets. In the experiment, we set the kernel width $\tau$ of Laplacian and Gaussian kernel as the input dimension. The average accuracy and the running time of 10-fold cross-validation are shown in TABLE VII. For running time, the $\ell_2$ MKL and $\ell_1$ MKL are both transformed into SIP and can be solved efficiently, thus the cost time is comparable. The column of average accuracy shows that Laplacian kernel performs pretty close to Gaussian kernel. On the whole, $\ell_2$ MKL outperforms $\ell_1$ MKL since it can thoroughly combines complementary information.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>MKL</th>
<th>Kernel</th>
<th>Accuracy</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>German</td>
<td>$\ell_1$-norm</td>
<td>Gaussian</td>
<td>69.95 ± 3.37</td>
<td>1.03</td>
</tr>
<tr>
<td></td>
<td>$\ell_2$-norm</td>
<td>Gaussian</td>
<td>72.30 ± 2.91</td>
<td>1.07</td>
</tr>
<tr>
<td></td>
<td>$\ell_2$-norm</td>
<td>Laplacian</td>
<td>73.75 ± 2.86</td>
<td>1.01</td>
</tr>
<tr>
<td>Cancer</td>
<td>$\ell_1$-norm</td>
<td>Gaussian</td>
<td>95.89 ± 2.74</td>
<td>0.37</td>
</tr>
<tr>
<td></td>
<td>$\ell_2$-norm</td>
<td>Laplacian</td>
<td>95.84 ± 2.87</td>
<td>0.38</td>
</tr>
<tr>
<td></td>
<td>$\ell_2$-norm</td>
<td>Gaussian</td>
<td>96.69 ± 2.65</td>
<td>0.39</td>
</tr>
<tr>
<td></td>
<td>$\ell_2$-norm</td>
<td>Laplacian</td>
<td>96.45 ± 2.35</td>
<td>0.36</td>
</tr>
<tr>
<td>Ijcnn1</td>
<td>$\ell_1$-norm</td>
<td>Gaussian</td>
<td>88.23 ± 2.26</td>
<td>0.94</td>
</tr>
<tr>
<td></td>
<td>$\ell_2$-norm</td>
<td>Laplacian</td>
<td>88.42 ± 2.61</td>
<td>0.93</td>
</tr>
<tr>
<td></td>
<td>$\ell_2$-norm</td>
<td>Gaussian</td>
<td>88.57 ± 2.56</td>
<td>0.98</td>
</tr>
</tbody>
</table>

ACKNOWLEDGMENTS

The authors would like to thank Dr. Huan Liu for helpful discussions.
REFERENCES


Ling Jian (M’15) received the B.Sc. degree in applied mathematics from Qufu Normal University, Qufu, China, in 2003, the M.Sc. degree in operational research and cybernetics from Zhejiang University, Hangzhou, China, in 2006, and the Ph.D. degree in operational research and cybernetics from the Dalian University of Technology, Dalian, China, in 2012. Since March 2006, he has been with the China University of Petroleum, Qingdao, China, where he is currently an Associate Professor. His scientific interests include machine learning, data mining, kernel methods, and optimization.

Jundong Li is a Ph.D student of computer science and engineering at Arizona State University. He received his B.Eng degree from College of Computer Science and Technology, Zhejiang University in 2012 and M.Sc. degree from Department of Computing Science, University of Alberta in 2014. His research interests include data mining, machine learning. Specifically, he is interested in feature selection, dynamic networks analysis, anomaly detection, associative classification, spatial and spatio-temporal data mining.
Shihua Luo received the B.S. degree and M.S. degree in mathematics from Jiangxi Normal University, China, in 1998 and 2001, and the Ph.D. degree in operational research and cybernetics from Zhejiang University, China, in 2007. He is currently a Professor of Jiangxi University of Finance and Economics. His research interests are in the field of modeling and optimization of complex systems, especially on the extraction of nonlinear characteristics of chemical process and financial process.