УДК 519.226:04.855

O. Mytnyk

DOI: 10.18523/2617-70807202444-50

ROBUST BAYESIAN REGRESSION MODEL IN BERNSTEIN FORM

In this paper, we present an inductive method for constructing robust Bayesian Polynomial Regression (BPR) models in Bernstein form, referred to as PRIAM (Polynomial Regression Inductive AlgorithM). PRIAM is an algorithm designed to determine stochastic dependence between variables. The triple nature of PRIAM combines the advantages of Bayesian inference, the interpretability of neurofuzzy models in Bernstein form, and the robustness of the support vector approach. This combination facilitates the integration of state-of-the-art machine learning techniques in decision support systems. We conduct experiments using well-known datasets and real-world economic, ecological, and meteorological models. Furthermore, we compare the forecast errors of PRIAM against several competitive algorithms.

Keywords: PRIAM, Bayesian inference, BPR, neurofuzzy model, polynomials in Bernstein form.

Introduction

Data mining competitions are an effective tool for evaluating the performance of specific methods among the growing variety of approaches. Recent contests, such as those hosted on Kaggle (https://www.kaggle.com/competitions) and the Data Mining Cup (http://www.data-mining-cup.com) have demonstrated the advantages of Bayesian and support vector (SV) methods. However, despite their high performance, these methods often face challenges in seamless integration into decision support systems. In contrast, neurofuzzy modeling offers an appealing framework for knowledge representation. This work seeks to combine the strengths of Bayesian reasoning, the robustness of the SV approach, and the interpretability of neurofuzzy modeling.

Brief historical outlook. Bayesianism began with Savage's personalistic school of thought and gained strength through the objective selection of prior probabilities based on the maximum entropy principle [1]. Since then, Bayesianism has inspired a series of significant contributions. For instance, Bayesian Occam's razor was demonstrated by Gull [2] as a method to estimate the parameters of prior probabilities in regression analysis. This concept was later applied by MacKay for the regularization of artificial neural networks in the so-called Bayesian evidence framework [3]. Additionally, the theory of Gaussian Processes (GP) incorporates evidence, also referred to as marginal likelihood, as a fundamental component of Bayesian inference [4].

Brown and Harris [5] established a correspondence between associative memory networks and fuzzy logic in neurofuzzy adaptive models. These models combine the transparent knowledge representation of fuzzy systems with the analytical ability to learn from observations. The ability to describe the behavior of neurofuzzy models as a series of human-readable linguistic rules makes them particularly well-suited for expert systems. However, conventional neurofuzzy models often suffer from the curse of dimensionality. To address this, Hong and Harris [6] proposed a polynomial complexity neurofuzzy approach.

Another efficient approach to process analysis is the Statistical Learning Theory (SLT) developed by Vapnik [7]. SLT is founded on the structural risk minimization principle, which is implemented in support vector machines (SVM) for classification problems [8]. SVM has since been extended to regression problems, leading to the development of support vector regression (SVR)[**?**], and further refined into Bayesian SVR[10]. In this work, we leverage the support vector (SV) approach to enhance the robustness of our models.

General problem statement. Suppose we observe the data:

$$\mathcal{D} = \{(y_j, \mathbf{x}_j)\}_{j=1}^{N}, \quad y \in \mathbb{R}, \quad \mathbf{x} \in \mathcal{X} = \mathbb{R}^n.$$

We hypothesize the existence of a stochastic dependence that maps each $\mathbf{x}$ to some value y obtained from a random trial governed by the law $p(y|\mathbf{x})$. To determine this stochastic dependence, we aim to identify the probability density function $p(y|\mathbf{x})$. However, this inverse problem is inherently ill-posed. Using the finite training set $\mathcal{D}$, we can only estimate posterior predictive distribution $p(y|\mathbf{x}, \mathcal{D})$. This estimation depends on the confidence in the observed data and the regularization methods applied to make the problem well-posed. In this paper, we focus on finding the mean of the posterior predictive distribution along with its variance.

The paper is organized as follows. We begin with an overview of the Bayesian framework. Next, we explore neurofuzzy models in Bernstein form and introduce a procedure for searching suboptimal models. Robustness is incorporated into the model afterward. As a result, we propose PRIAM – an inductive algorithm for constructing robust BPR models in Bernstein form, capable of encoding prior knowledge and generalizing effectively. Finally, we conduct experiments with PRIAM on both synthetic and real-world datasets, comparing its performance to that of other competitive algorithms.

Bayesian Framework

Let the systematic component of the stochastic dependence be described by a latent function f of a model $\mathcal{M}$ from the model space $\mathcal{H}$. This raises the following questions: how should the model space $\mathcal{H}$ be chosen, how should the model $\mathcal{M} \in \mathcal{H}$ be selected, and how can the function $f \in \mathcal{M}$ be determined. Bayesian reasoning provides answers to the last two questions.

Selection of a model $\mathcal{M}$. According to the Bayesian approach, the model with the maximum posterior probability $P(\mathcal{M}|\mathcal{D})$ is selected. Assuming a flat prior distribution of models over the space $\mathcal{H}$ (i.e. complete ignorance), the models are ranked by their marginal likelihood $p(\mathcal{D}|\mathcal{M})$, also known as evidence. Evidence reflects the ability of the model $\mathcal{M}$ to generate the data $\mathcal{D}$ and is defined as the following Lebesgue integral:

$$p(\mathcal{D}|\mathcal{M}) = \int_{\mathcal{M}} p(\mathcal{D}|f, \mathcal{M}) d\mu(f), \qquad (1)$$

where $\mu(f)$ represents the prior probability measure on the function space $\mathcal{M}$. The likelihood $p(\mathcal{D}|f, \mathcal{M})$ reflects the ability of the function $f \in \mathcal{M}$ to generate the data $\mathcal{D}$. Assume that the random component of the stochastic dependence is represented by additive noise, so that $y_j = f(\mathbf{x}_j) + \delta_j$, where δ_j are independent and identically distributed random variables. Under this assumption, the likelihood takes the following form:

$$p(\mathcal{D}|f, \mathcal{M}) = \prod_j^N p(\delta_j|f, \mathcal{M}),$$

where $p(\delta|f, \mathcal{M})$ is a noise model. Both the noise model and the prior measure $\mu(f)$ will be selected later.

The evidence (1) can be approximated using various techniques, including expectation propagation (EP), Laplace's method, Markov chain Monte Carlo (MCMC). An overview and comparison of these methods are provided by Kuss [11].

Selection of a function f. The posterior probability measure can be derived using Bayes' rule:

$$d\mu(f|\mathcal{D}) = \frac{p(\mathcal{D}|f, \mathcal{M}) d\mu(f)}{p(\mathcal{D}|\mathcal{M})}.$$

Our goal is to determine the posterior predictive distribution, which represents the posterior beliefs about the output value y. This distribution is obtained by integrating over the posterior uncertainty of the function:

$$p(y|\mathbf{x}, \mathcal{D}, \mathcal{M}) = \int_{\mathcal{M}} p(y|\mathbf{x}, f, \mathcal{M}) d\mu(f|\mathcal{D}).$$

According to Bayesian decision theory, to obtain a single function estimate $g \in \mathcal{M}$ for regression $y(\mathbf{x})$, we minimize the Bayesian risk, defined as the expectation of a loss functional L:

$$R(g) = \mathbb{E}_f \left[L(f, g) \right] = \int_{\mathcal{M}} L(f, g) d\mu(f|\mathcal{D}).$$

The loss functional L reflects the researcher's subjective attitude toward risk. Typically, the choice of L is a point of debate among researchers. We will establish our choice of L later.

In the next section, we address the question of how to select the model space and organize the model search process.

Polynomial Regression in Bernstein Form

Hong and Harris [6] introduced neurofuzzy models based on the following truncated ANOVA decomposition for input variable $\mathbf{x} = \left\{x^i\right\}_{i=1}^n$:

$$f(\mathbf{x}) = b + \sum_{k=1}^{n} B_k^d(x^k) + \sum_{q>p}^{n} B_{pq}^d\left(x^p, x^q\right). \qquad (2)$$

B_k^d, B_{pq}^d are univariate and bivariate polynomials in Bernstein form, defined as linear combinations of Bernstein basis polynomials of degree d:

$$B_k^d(x^k) = \textstyle\sum_{j=0}^d w_j^k \phi_j^d \left[s(x^k)\right],$$
$$B_{pq}^d(x^p, x^q) = \textstyle\sum_{i+r+t=d} w_{irt}^{pq} \phi_{irt}^d \left[\mathbf{u}(x^p, x^q)\right].$$

We refer to models (2) as neurofuzzy models in Bernstein form. The Bernstein basis polynomials are defined as:

$$\phi_j^d(s) = \binom{d}{j} \cdot s^j (1-s)^{d-j},$$
$$\phi_{irt}^d(\mathbf{u}) = \binom{d}{i, r, t} u^i v^r (1-u-v)^t.$$

To determine the barycentric coordinates s and $\mathbf{u} = \{u, v\}$ we follow the approach proposed in [12], which introduces a fast inverse de Casteljau mapping based on a uniform knot layout.

After training such neurofuzzy models, dependencies can be interpreted using fuzzy logic and generate a set of fuzzy rules [5]. It is well-known that Bernstein basis polynomials are non-negative and satisfy the unity of support property: $\sum_j \phi_j = 1$. Therefore, Bernstein basis polynomials are valid fuzzy membership functions. The advantages of this approach include: transparency of the model structure, interpretation of dependencies in terms of fuzzy logic, and polynomial complexity of the resulting set of fuzzy rules.

Selection of a model space $\mathcal{H}$. Let us leverage the advantages of neurofuzzy modeling. To achieve this, we define the configuration of the model space as an upper triangular $[n \times n]$ matrix $\mathbf{C}$. Each diagonal element c_k represents the degree of a univariate polynomial in Bernstein form for the factor x^k. Each element above the diagonal, $c_{q>p}$, represents the degree of a bivariate polynomial in Bernstein form for the pair x^p and x^q. The corresponding model space is expressed as:

$$\mathcal{M}(\mathbf{w}, b, \mathbf{x}) = \mathcal{H}(\mathbf{C}, \mathbf{w}, b, \mathbf{x}) = \\ = b + \sum_{k=1}^{n} B_k^{c_k}(x^k) + \sum_{q>p}^{n} B_{pq}^{c_{pq}}(x^p, x^q), \quad (3)$$

where parameters $\mathbf{w} = \{\ldots, w_j^k, \ldots, w_{irt}^{pq}, \ldots\}$. The models in the form (3) generalize those in the form (2). Furthermore, the models (3) can also be considered linear in parameters $\mathbf{w}$ within a high-dimensional Euclidian space $\mathcal{W}$ with the canonical scalar product $\langle \cdot, \cdot \rangle$ and norm $\| \cdot \|$:

$$f(\mathbf{x}) = \mathcal{M}(\mathbf{w}, b, \mathbf{x}) = \langle \mathbf{w}, \Phi(\mathbf{x}) \rangle + b,$$

where $\Phi(\mathbf{C}) : \mathcal{X} \to \mathcal{W}$ represents the mapping:

$$\Phi : \mathbf{x} \mapsto \{\ldots, \phi_j^{c_k}\left(x^k\right), \ldots, \phi_{irt}^{c_{pq}}\left(x^p, x^q\right), \ldots\}.$$

Model search in neurofuzzy model space. Algorithm 1 demonstrates how an initial model guess can be refined using evidence-based calculations.

Algorithm 1 Model search

Input: observations $\mathcal{D}$, convergence level $\nu > 0$, initial model $\mathcal{M}^{(0)} = \mathcal{H}\left(\mathbf{C}^{(0)}\right)$
Result: suboptimal model $\mathcal{M}_{\text{opt}}$
Iterator $t \longleftarrow 0$
repeat
  $\mathcal{M}_{\text{opt}} \longleftarrow \mathcal{M}^{(t)}$
  Generate a set of candidate models: $\left\{\mathcal{M}_{ij}^{(t+1)} = \mathcal{H}\left(\mathbf{C}_{ij}^{(t+1)}\right)\right\}_{ij}$, where $\mathbf{C}_{ij}^{(t+1)} = \mathbf{C}^{(t)} \pm \mathbf{1}_{ij}, \quad 1 \leqslant i \leqslant j \leqslant n.$
  Choose the model with the maximum evidence: $\mathcal{M}^{(t+1)} = \arg\left[p^{(t+1)} = \max p\left(\mathcal{D}|\mathcal{M}_{ij}^{(t+1)}\right)\right]$
  $t \longleftarrow t + 1$
until $p^{(t)} < p^{(t-1)} + \nu$

First, we define the space $\mathcal{H}$ of models $\mathcal{M}$ in the form (3). Based on prior assumptions about the model structure, the initial configuration $\mathbf{C}^{(0)}$ is constructed. At each step, a set of candidate models is generated, each differing in the degree of one polynomial in Bernstein form. For each candidate model, the evidence is calculated. The model with the maximum evidence is selected. The new model is accepted if its evidence exceeds that of the previous model by a threshold ν. This threshold determines the linear convergence speed and reflects the degree of confidence in the prior model structure.

Robust BPR in Bernstein Form

To leverage the robustness of Support Vector Regression (SVR), we define the noise model as:

$$p(\delta_j | f, \mathcal{M}) = \frac{\beta}{2(1 + \epsilon\beta)} \exp(-\beta |\delta_j|_\epsilon), \quad (4)$$

where $|\cdot|_\epsilon$ is the ϵ-insensitive loss function (ϵ-ILF), which provides sparseness and robustness to the BPR models. The parameters ϵ and β are referred to as hyperparameters. In this work we assume flat priors for these hyperparameters.

Let the prior probability measure $\mu(f)$ have a density $p(\mathbf{w}|\mathcal{M})$ with respect to the Lebesgue measure on the parameter space $\mathcal{W}$:

$$d\mu(f) = p(\mathbf{w}|\mathcal{M}) d\mathbf{w}.$$

Let this density be a multivariate Gaussian with $\mathbf{0}$ mean and identity covariance matrix $\mathbf{I}$:

$$p(\mathbf{w}|\mathcal{M}) = \mathcal{N}(\mathbf{0}, \mathbf{I}).$$

We define the loss functional L as $L(f, g) = \{0 \text{ if } f = g; 1 \text{ if } f \neq g\}$. In this case, the

Bayesian risk is minimized at the mode of the posterior function distribution, yielding the so-called Maximum a Posteriori (MAP) estimate. The MAP estimate for BPR with the noise model (4) corresponds to the canonical SVR problem:

$$R(f) = \beta \sum_{j=1}^{N} |\delta_j|_\epsilon + \frac{1}{2}\|\mathbf{w}\|^2 \longrightarrow \min_f, \qquad (5)$$

with solution in the form:

$$f_{\text{map}}(\mathbf{x}) = \textstyle\sum_j (\alpha_j - \alpha_j^*)\langle \Phi(\mathbf{x}_j), \Phi(\mathbf{x})\rangle + b_{\text{map}},$$
$$\mathbf{w}_{\text{map}} = \textstyle\sum_j (\alpha_j - \alpha_j^*)\Phi(\mathbf{x}_j),\ \alpha_j, \alpha_j^* \in (0, \beta),$$
$$b_{\text{map}} = \text{mean}_j \left\{ \begin{array}{l} y_j - \langle \mathbf{w}, \Phi(\mathbf{x}_j)\rangle - \epsilon \\ y_j - \langle \mathbf{w}, \Phi(\mathbf{x}_j)\rangle + \epsilon \end{array} \right\},$$

where α_j, α_j^* are the Lagrange multipliers of the corresponding quadratic programming (QP) problem.

The algorithm 1, under the assumptions described above, is referred to as PRIAM. In the following subsections, we demonstrate how to estimate the evidence and error bars in PRIAM.

Evidence Estimation

For fast evidence estimation, we adopt the approach described in [13], where a locally smoothed loss function is used to approximate ϵ-ILF. This approach yields the following approximation, referred to as Bayesian Evidence Criterion (BEC), for negative logarithm of the Bayesian evidence:

$$-\ln p(\mathcal{D}|\mathcal{M}) \approx \text{BEC}(\mathcal{M}, \epsilon, \beta) = \\ = R(\mathbf{w}_{\text{map}}) - N \ln \frac{\beta}{2(1+\epsilon\beta)}. \qquad (6)$$

Although the BEC approximation is not entirely accurate, it preserves sparseness and is recognized as the fastest method for model comparison.

While evidence should be maximized, BEC should be minimized. Additionally, since BEC depends on the hyperparameters ϵ and β, it can also be minimized with respect to these hyperparameters:

$$\text{BEC}(\mathcal{M}) = \min_{\epsilon,\beta} \text{BEC}(\mathcal{M}, \epsilon, \beta). \qquad (7)$$

This is a nonlinear minimization problem. The gradient of BEC is expressed as:

$$\nabla^{\text{BEC}}_{\epsilon,\beta} = \left[\frac{N\beta}{1+\epsilon\beta} - \beta N_{\text{SV}}, N R_{\text{emp}} - \frac{N}{\beta(1+\epsilon\beta)} \right]$$

where N_{SV} is the number of support vectors, and the empiric risk is defined as $R_{\text{emp}} = \sum_{j=1}^{N} |\delta_j|_\epsilon$. To minimize (7) with respect to the hyperparameters, we employ the Interior Reflective Newton (IRN) method. IRN is known for its global and quadratic convergence properties [14].

Estimation of error bars. The variance of the noise model (4) can be easily computed as:

$$\sigma_N^2 = \frac{2}{\beta^2} + \frac{\epsilon^2(\epsilon\beta + 3)}{3(\epsilon\beta + 1)}.$$

It is well-known that, for Gaussian noise, the posterior predictive distribution is also Gaussian $p(y_*|\mathbf{x}_*, \mathcal{D}) = \mathcal{N}(y_*|f_{\text{mean}}(\mathbf{x}_*), \sigma^2)$, with variance

$$\sigma^2 = k_{**} - \mathbf{k}_*^\top (\mathbf{K} + \sigma_N^2 \mathbf{I})^{-1} \mathbf{k}_* + \sigma_N^2, \qquad (8)$$

where matrix $\mathbf{K} \sim k(\mathbf{x}_i, \mathbf{x}_j) = \langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j)\rangle$, $\mathbf{k}_* = [k(\mathbf{x}_1, \mathbf{x}_*), \ldots, k(\mathbf{x}_N, \mathbf{x}_*)]^\top$ and $k_{**} = k(\mathbf{x}_*, \mathbf{x}_*)$. Although our noise model differs from the normal distribution, it can be shown that the distribution (4) is sufficiently close to a normal distribution with the same variance to justify using (8) as an acceptable approximation for the posterior variance. Furthermore, as demonstrated by Gao [15], the computation of $\mathbf{k}_*$ and $\mathbf{K}$ can be reduced to the marginal support vectors $\mathbf{X}_M = \{\mathbf{x}_i : |y_i - f(\mathbf{x}_i)| = \epsilon\}$. Finally, we use $\pm 2\sigma$ to represent the 95% confidence interval.

Pros and cons of the SVR with BEC approach. Advantages:

1. SV expansion is independent of the input space dimension, mitigating the curse of dimensionality in reconstruction problems.
2. A unique solution is obtained after training, as it is derived from solving a QP problem.
3. The SVR model exhibits robustness and sparseness.
4. The BEC provides an exceptionally fast model search method.

Disadvantages:

1. The BEC computation lacks precision, necessitating the selection of models with significantly smaller BEC values during model comparison.
2. Relying on the mode of the posterior function distribution for a given model (MAP estimation) deviates from pure Bayesian inference principles.

Experiments

For the experiments, we selected the Longley and Filip datasets from the Statistical Reference Datasets project [16]. Additionally, we included the well-known synthetic Friedman dataset. The AutoMPG dataset, which represents city cycle fuel consumption, was obtained from the UCI Machine Learning Repository [17]. The CPI and RCON datasets correspond to economic models of the Consumer Price Index and Real Consumption, respectively, as studied in [18]. The WIW dataset represents a meteorological wind-induced

wave model, while IBSS corresponds to an ecological model of macrozoobenthic biomass.

We compared PRIAM with GMDH (Group Method of Data Handling), NF-GMDH (Neurofuzzy Group Method of Data Handling, implemented in "GMDH Modeler 0.9.37"), RNN (Recurrent Neural Networks), and ANFIS (Adaptive Neuro Fuzzy Inference System), both implemented in "NeuroSolutions 5". Additionally, we evaluated GPR (Gaussian Process Regression) and XGBoost (Extreme Gradient Boosting), both available as Python packages. A brief excerpt of our experimental results is shown in Table 1.

The table also includes information about the size of the full dataset (N), the size of the learning dataset (N_{learn}), and the number of input factors (n) for each problem. A significant discrepancy between the MSEs of two different methods can be detected using Fisher statistics $F^{(80\%)}_{N-n,N-n}$.

Let us create a rating table for different algorithms. The algorithm with the smallest MSE result receives 10 points, the second 8 points, and so on. The two algorithms with the worst results receive no points. If multiple algorithms show insignificant differences in results, they share the same number of points. This way 30 points are distributed among all algorithms for each dataset.

As shown in Table 2, PRIAM achieves the highest rating among all the algorithms. Its performance is stable and never ranks among the worst. It is worth noting, however, that this rating is not absolute but instead reflects the strength of a specific algorithm in a particular implementation.

In the next subsection, we provide a detailed description of the result for the RCON dataset.

Dynamics of real consumption. The model for real consumption ($RCON$) is defined as a function of two factors: the interest rate (R) and real domestic income (RDI). The dataset consists of 24 observations, corresponding to monthly samples over a two years period. The first 14 points are used for training, while the last 10 points are reserved for forecasting and calculating the generalization error. The initial configuration of the model $\mathbf{C}^{(0)} = \mathrm{diag}\{1, 1\}$ reflects our prior belief in linear dependencies. The optimal PRIAM model is shown in Fig. 1.

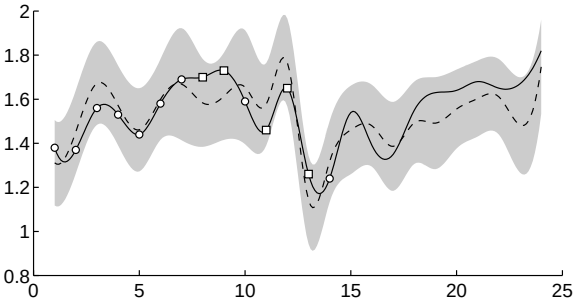
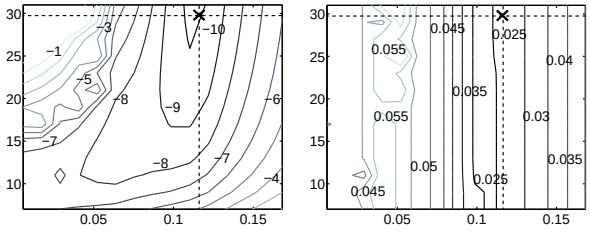


Figure 1. RCON dataset and optimal PRIAM model with 95% confidence interval. Squares stand for SVs, circles are vectors inside ϵ-tube.



(a) BEC contours (b) NMSE contours

Figure 2. The contour plots illustrate the dependencies of BEC and generalization error on the width of the ϵ-tube (abscissa axis) for different values of the hyperparameter β (ordinate axis). Crosses indicate the optimal hyperparameter values.

The corresponding optimal model configuration is given by $\mathbf{C}_{\mathrm{opt}} = \mathrm{diag}\{1, 2\}$, highlighting the reinforcing effect of RDI. The normalized model representation using the SV expansion is as follows:

$$f_{\mathrm{opt}}(\mathbf{x}) = 0.43 + 6.6k(\mathbf{x}_8, \mathbf{x}) + 25.5k(\mathbf{x}_9, \mathbf{x}) - \\ - 29.8k(\mathbf{x}_{11}, \mathbf{x}) - 2.6k(\mathbf{x}_{12}, \mathbf{x}) + 0.3k(\mathbf{x}_{13}, \mathbf{x}).$$

Dual model representation in neurofuzzy space:

$$f_{\mathrm{opt}}(\mathbf{x}) = 0.43 + 0.06\phi_0^1(x^1) - 0.06\phi_1^1(x^1) - \\ - 0.58\phi_0^2(x^2) + 0.13\phi_1^2(x^2) + 0.45\phi_2^2(x^2).$$

where $x^1 \equiv R$, $x^2 \equiv RDI$. Here, we observe a weak dependence of $RCON$ on R.

To evaluate the efficiency of BEC, we conduct a more detailed analysis of the relationship between BEC and generalization error with respect to the hyperparameters. According to the BEC contours (Fig. 2a), the optimal hyperparameter region is characterized by ϵ near 0.12, and high values of β. PRIAM successfully identifies the optimal hyperparameters, as $\beta = 29.8$ and $\epsilon = 0.12$. MSE contours (Fig. 2b) further confirm the efficiency of $\epsilon \approx 0.12$. However, they also indicate that forecasting is largely indifferent to the value of β for

Table 1. Comparison of algorithms on different datasets by normalized MSE

Algorithm	Longley	Filip	Friedman	AMPG	CPI	RCON	WIW	IBSS
$N(N_{learn})$	16(11)	82(33)	1000(500)	392(300)	24(14)	24(14)	166(100)	50(25)
n	6	1	10	4	4	2	3	2
$F^{(80\%)}_{N-n,N-n}$	2.2	1.3	1.0	1.0	1.8	1.7	1.1	1.4
PRIAM	0.017	0.004	0.009	0.022	0.003	0.027	0.033	0.076
GMDH	0.051	0.016	0.029	0.024	0.130	0.042	0.025	0.124
NF-GMDH	0.001	0.039	0.037	0.026	0.008	0.098	0.043	0.053
RNN	0.018	0.012	0.008	0.028	0.090	0.101	0.033	0.083
ANFIS	0.002	0.001	0.009	0.027	0.003	0.051	0.031	0.110
GPR	0.054	0.010	0.008	0.025	0.096	0.043	0.032	0.086
XGBoost	0.022	0.001	0.004	0.019	0.073	0.063	0.041	0.140

Table 2. Algorithm rating

Algorithm	Longley	Filip	Friedman	AMPG	CPI	RCON	WIW	IBSS	Rating
PRIAM	4	6	3	8	9	10	5	6	**51**
GMDH	0	0	0	6	0	6	10	1	**23**
NF-GMDH	9	0	0	2	6	0	0	10	**27**
RNN	4	3	7	0	2	0	5	6	**27**
ANFIS	9	9	3	0	9	6	5	1	**42**
GPR	0	3	7	4	2	6	5	6	**33**
XGBoost	4	9	10	10	2	2	0	0	**37**

wide ϵ-tubes. The significance of β becomes notable only for narrower tubes.

Generation of fuzzy rules. We demonstrate how fuzzy rules can be generated based on the model in neurofuzzy space [19]. A balanced neurofuzzy model, like the one described above, can be decomposed into two neurofuzzy submodels in the canonical form due to the unity of support property:

$$f_1(x^1) = 0.49\mu_{A_0^1}(x^1) + 0.37\mu_{A_1^1}(x^1),$$
$$f_2(x^2) = -0.15\mu_{A_0^2}(\cdot) + 0.56\mu_{A_1^2}(\cdot) + 0.88\mu_{A_2^2}(\cdot)$$

where the Bernstein basis polynomials are used as fuzzy membership functions, $\phi_j^d \equiv \mu_{A_j^d}$, with the corresponding fuzzy labels on input space: A_0^1 — *low R*, A_1^1 — *high R*, A_0^2 — *low RDI*, A_1^2 — *average RDI*, A_2^2 — *high RDI*. Each submodel generates simplified rules independently and contributes to a fuzzy knowledge base of reduced complexity.

The rules and their confidences can be easily determined if the output fuzzy membership functions are represented as B-splines. In this case, at most two adjacent coefficients are nonzero. Let us define fuzzy membership functions for the *RCON* output variable, normalized on [0; 1], using three second-order B-splines μ_{B_k} with a triangular shape. These B-splines are defined over the knots {-0.5; 0; 0.5; 1; 1.5} with peaks at {0; 0.5; 1}. They correspond to the following fuzzy labels: B_0 — *low RCON*, B_1 — *average RCON*, B_2 — *high RCON*.

The rule R_j^i produced by submodel f_i is expressed as: "if $x^i \in A_j^i$, then $y \in B_k$ with confidence c_{kj}^i", where the rule confidences c_{kj}^i are determined by converting the weights of the model in the neurofuzzy space as follows:

$$c_{kj}^i = \mu_{B_k}\left(f_i\left(\arg\max_{x^i} \mu_{A_j^i}(x^i)\right)\right).$$

Thus, we derive five rules:

R_0^1: *if R is low, then RCON is low* (0.02) *or average* (0.98)
R_1^1: *if R is high, then RCON is low* (0.26) *or average* (0.74)
R_0^2: *if RDI is low, then RCON is low* (1.0)
R_1^2: *if RDI is average, then RCON is low* (0.08) *or average* (0.92)
R_2^2: *if RDI is high, then RCON is average* (0.24) *or high* (0.76)

Although these rules are not suitable for making exact forecasts. They enable experts in the application domain to understand relationships between variables, verify the trained model, and collaborate with machine learning engineers in model fusion.

Conclusion

We have presented an inductive method for constructing robust Bayesian Polynomial Regression models in Bernstein form. This method integrates the strengths of Bayesian inference, the support vector approach, and neurofuzzy modeling. The dual model conception—combining support vector expansion with Bernstein form – enables PRIAM to remain competitive with modern machine learning algorithms while also being suitable for knowledge representation in expert systems. The use of fuzzy rules with reduced complexity allows domain experts to contribute at various stages of the modeling process, including such tasks as prior setup, model validation, and knowledge extraction.

Our experiments on real-world economic datasets demonstrate that PRIAM outperforms many modern algorithms while adhering to a parsimonious model construction logic. Notably, bivariate dependencies appear only when the true underlying function (as observed in synthetic datasets) explicitly includes a product of endogenous factors.

References

1. E. T. Jaynes, *Probability Theory: The Logic of Science* (Cambridge : Cambridge University Press, 2003).
2. S. Gull, in: Maximum Entropy and Bayesian Methods, ed. by Erickson G. J., Smith C. R. (Dordrecht: Kluwer Academic, 1988), pp. 53–74.
3. D. J. C. Mackay, Neural Computations. **4**, 448–472 (1992).
4. C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning* (Cambridge, MA: MIT Press, 2006).
5. M Brown and C. J. Harris, *Neurofuzzy adaptive modelling and control* (Hemel Hempstead: Prentice Hall, 1994).
6. X. Hong and C. J. Harris, IEEE Trans. Neural Networks. **11** (4), 889–902 (2000).
7. V. N. Vapnik, *Statistical learning theory* (New York: John Wiley and Sons Inc., 1998).
8. C. Cortes and V. Vapnik, Machine Learning. **20**, 273–297 (1995).
9. V. Vapnik, S. Golowich, and A. Smola, Advances in Neural Information Processing Systems. **9**, 281–287 (1997).
10. W. Chu, S. Keerthi, and C. J. Ong, IEEE Trans. Neural Networks. **15** (1), 29–44 (2004).
11. M. Kuss and C. E. Rasmussen, Journal of Machine Learning Research. **6**, 1679–1704 (2005).
12. O. Y. Mytnyk and P. I. Bidyuk, System Research and Information Technologies. **2**, 24–34 (2004).
13. O. Y. Mytnik, Cybernetics and Sys. Anal. **43** (4), 613–620 (2007).
14. T. F. Coleman and Y. Li, Mathematical Programming. **67** (2), 189–224 (1994).
15. J. B. Gao, S. R. Gunn, C. J. Harris, and M. Brown, Machine Learning. **46** (1–3), 71–89 (2002).
16. NIST Standard Reference Database 140, https://www.itl.nist.gov/div898/strd.
17. UCI Repository of machine learning databases, https://archive.ics.uci.edu/datasets.
18. Y. Gorodnichenko, *Effects of intergovernmental aid on fiscal behavior of local governments: the case of Ukraine : EERC MA thesis* (NaUKMA, 2001).
19. O. Y. Mytnyk, in: Proceedings of 2nd International Conference on Inductive Modelling, 15–19 Sept. 2008 (Kyiv, 2008), pp. 148–152.

Митник О. Ю.

РОБАСТНА МОДЕЛЬ БАЄСІВСЬКОЇ РЕГРЕСІЇ У ФОРМІ БЕРНШТЕЙНА

Тут представлений індуктивний метод побудови робастних моделей баєсівської поліноміальної регресії (БПР) у формі Бернштейна, що отримав назву ПРІАМ. ПРІАМ – це алгоритм, призначений для визначення стохастичної залежності між змінними. Трикомпонентна природа ПРІАМ поєднує переваги баєсівського висновку, прозорість та лінгвістичну інтерпретовність нейронечітких моделей у формі Бернштейна, робастність методу опорних векторів.

Алгоритм апробовано на відомих штучних наборах даних, а також на реальних моделях різного розміру та рівня зашумленості. Складено рейтинг, який демонструє переваги запропонованого алгоритму за більшістю метрик.

Ключові слова: ПРІАМ, баєсівський висновок, БПР, нейронечітка модель, поліноми в формі Бернштейна.

Матеріал надійшов 27.12.2024