entropy

MDPI

Article

Product Design Time Forecasting by Kernel-Based Regression with Gaussian Distribution Weights

Zhi-Gen Shang [1,2] and Hong-Sen Yan [1,*]

[1] MOE Key Laboratory of Measurement and Control of Complex Systems of Engineering, School of Automation, Southeast University, Nanjing 210096, China; zgshang@ycit.edu.cn
[2] Department of Automation, Yancheng Institute of Technology, Yancheng 224051, China
* Correspondence: hsyan@seu.edu.cn; Tel.: +86-25-8379-2418

Academic Editors: Badong Chen and Jose C. Principe
Received: 20 April 2016; Accepted: 16 June 2016; Published: 21 June 2016

Abstract: There exist problems of small samples and heteroscedastic noise in design time forecasts. To solve them, a kernel-based regression with Gaussian distribution weights (GDW-KR) is proposed here. GDW-KR maintains a Gaussian distribution over weight vectors for the regression. It is applied to seek the least informative distribution from those that keep the target value within the confidence interval of the forecast value. GDW-KR inherits the benefits of Gaussian margin machines. By assuming a Gaussian distribution over weight vectors, it could simultaneously offer a point forecast and its confidence interval, thus providing more information about product design time. Our experiments with real examples verify the effectiveness and flexibility of GDW-KR.

Keywords: design time forecast; kernel-based regression; Kullback-Leibler divergence; heteroscedasticity

1. Introduction

Product design is a complex and dynamic process, and its duration is affected by a number of factors, most of which are of fuzzy, random and uncertain characteristics. As product design tasks occur in different companies, uncertain characteristics may vary from product to product. The heteroscedasticity thus constitutes another important feature of product design. The mapping from the factors to design time is highly nonlinear, and it is impossible to describe this mapping relationship by definite mathematical models. The degree of reasonability of the supposed distribution of product design time is a key factor in product development control and decisions [1–3].

The triangular probability distribution was chosen by Cho and Eppinger [1] to represent design task durations, and a process modeling and analysis technique for managing complex design projects was proposed by using advanced simulation. However, if the assumed distribution of design activity durations does not reflect the true state, the proposed algorithm may fail to obtain ideal results. Yan and Wang [2] proposed a time-computing model with its corresponding design activities in concurrent product development process. Yang and Zhang [3] presented an evolution and sensitivity design-structure matrix to reflect overlapping and their impact on the degree of activity sensitivity and evolution in the process model, and the model can be used for better project planning and control by identifying overlapping and risk for process improvements, but with the two algorithms mentioned above, normal duration of each design activity should be determined before the algorithm is executed, and if activity durations are incompatible with the actual ones, the proposed algorithm may fail to function well. Apparently, the accuracy of predetermined design time is crucial to the planning and controlling of product development processes.

Traditionally, approximate design time is analyzed by means of qualitative approaches. With the rapid development of computer and regression techniques, new forecast methods keep emerging. Bashir and Thomson [4] came up with a modified Norden model to estimate project duration in

conjunction with the effort-estimation model. Griffin [5] related the length of the product development cycle to project, process and team structure factors by a statistical method, and quantified the impact of project newness and complexity on the increasing length of development cycle, but with no proposal for design time forecasts. Jacome and Lapinskii [6] developed a model to forecast electronic product design efforts based on a structure and process decomposition approach. Only a small portion of the time factors, however, are taken into account by the model. Xu and Yan [7] proposed a design-time forecast model based on a fuzzy neural network, which exhibits good performance when the sample data are sufficient. However, only a small number of design cases are available to a company, which weakens the validity of the fuzzy neural network. Therefore, a novel approach should be adopted.

Recently, kernel methods have been identified as one of the leading means for pattern classification and function approximation, and successfully applied in various fields [8–14]. Support vector machine (SVM), initially developed by Vapnik for pattern classification, is one of the most used models. With the introduction of the ε-insensitive loss function, SVM has been extended in use to solve nonlinear regression problems, and thus is also called support vector regression (SVR). ε-insensitive loss functions contribute to the sparseness property of SVR, but the value of ε, chosen a priori, is hard to determine. A new parameter v was then introduced and v-SVR proposed, whereby v controls the number of support vectors and training errors [11]. v-SVR has overcome the difficulty of ε determination. In recent years, much research has been done on kernel methods. Kivinen *et al.* considered online learning in a reproducing kernel Hilbert space in [15]. Liu *et al.* [16] proved that the kernel least-mean-square algorithm can be well posed in reproducing kernel Hilbert spaces without adding an extra regularization term to penalize solution norms as was suggested by [15]. Chen *et al.* developed a quantized kernel least mean square algorithm based on a simple online vector quantization method in [17], and proposed the quantized kernel least squares regression in [18]. Wu *et al.* [19] derived the kernel recursive maximum correntropy in kernel space and under the maximum correntropy. Furthermore, by combining fuzzy theory with v-SVR, Yan and Xu [20] proposed Fv-SVM to forecast the design time, which could be used to solve regression problems with uncertain input variables. However, both Fv-SVM and v-SVR assume that the noise level is uniform throughout the domain, or at least, its functional dependency is known beforehand [21]. It is thus clear that the time forecast of product design based on Fv-SVM is deficient simply due to the heteroscedasticity of product design. For better planning and controlling of product development process, any good forecast method is expected to yield not only highly precise forecast values, but also valid forecast intervals.

In terms of Gaussian margin machines [22], the weight vector of binary classifier maintains a Gaussian distribution, and what should be struck for is the least information distribution that classifieds training samples with a high probability. Gaussian margin machines provide the probability that a sample belongs to a certain class. The idea given by Gaussian margin machines is extend to the regression for the forecast of product design time. Shang and Yan [23] proposed Gaussian margin regression (GMR) on the basis of combining Gaussian margin machines and kernel-based regression. However, GMR assumes that the forecast variances are same, which is inconsistent with the heteroscedasticity that exits in design time forecast. Like Fv-SVM, GMR also fails to provide valid forecast intervals. By combining Gaussian margin machine and extreme learning machine [24,25], a confidence-weighted extreme learning machine was proposed for regression problems of large samples [26].

The present study adopts the kernel-based regression with Gaussian distribution weights (GDW-KR) by combining Gaussian margin machines with the kernel-based regression, aiming to solve problems of small samples and heteroscedastic noise in design time forecasting, providing both forecast values and intervals. Inheriting the merits of Gaussian margin machines, GDW-KR maintains a Gaussian distribution over weight vectors, seeking the least information distribution that will make each target be included in its corresponding confidence interval. The optimization problem of GDW-KR is simplified, and an approximate solution of the simplified problem is obtained by using

the results of regularized kernel-based regression. On the basis of this model, a forecast method for product design time and its relevant parameter-determining algorithm are then put forward.

The rest of this paper is organized as follows: Gaussian margin machines are introduced in Section 2. GDW-KR and the method for solving the optimization problem are described in Section 3. In Section 4, the application in injection mold design is presented, and GDW-KR is then compared with other models. An extended application of GDW-KR is also given. Section 5 draws the final conclusions.

2. Gaussian Margin Machines

Suppose the samples $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{l}$, where $\boldsymbol{x}_i \in \mathbf{R}^m$ is a column vector and $y_i \in \{-1, 1\}$ is a scalar output. The weight vector $\boldsymbol{w}$ of a linear classifier is supposed to follow a multivariable normal distribution $N_m(\boldsymbol{\mu}_1, \Sigma_1)$ with mean $\boldsymbol{\mu}_1 \in \mathbf{R}^m$ and covariance matrix $\Sigma_1 \in \mathbf{R}^{m\times m}$. For the sample $\boldsymbol{x}_i$, we get the normal distribution:

$$\boldsymbol{x}_i{}^{\mathrm{T}}\boldsymbol{w} \sim N(\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{\mu}_1, \boldsymbol{x}_i^{\mathrm{T}}\Sigma_1\boldsymbol{x}_i). \tag{1}$$

The linear classifier is designed to properly classify each sample with a high probability, that is:

$$\Pr(y_i\boldsymbol{x}_i{}^{\mathrm{T}}\boldsymbol{w} \geqslant 0) \geqslant \rho, \tag{2}$$

where $\rho \in (0.5,\ 1]$ is the confidence value.

By combining Equations (1) and (2), we get:

$$\Pr\left(\frac{y_i\boldsymbol{x}_i{}^{\mathrm{T}}\boldsymbol{w} - y_i\boldsymbol{x}_i{}^{\mathrm{T}}\boldsymbol{\mu}_1}{\sqrt{\boldsymbol{x}_i^{\mathrm{T}}\Sigma_1\boldsymbol{x}_i}} \leqslant \frac{-y_i\boldsymbol{x}_i{}^{\mathrm{T}}\boldsymbol{\mu}_1}{\sqrt{\boldsymbol{x}_i^{\mathrm{T}}\Sigma_1\boldsymbol{x}_i}}\right) \leqslant 1-\rho. \tag{3}$$

GMM aims to seek the least informative distribution that classifies the training set with high probability, which is achieved by seeking a multivariable normal distribution $N_m(\boldsymbol{\mu}_1, \Sigma_1)$ with minimum Kullback-Leibler divergence with respect to an isotropic distribution $N_m(0, a\boldsymbol{I}_m)$. The Kullback-Leibler divergence between $N_m(\boldsymbol{\mu}_1, \Sigma_1)$ and $N_m(0, a\boldsymbol{I}_m)$ is denoted by $D_{\mathrm{KL}}\left(N_m(\boldsymbol{\mu}_1, \Sigma_1)\,||\,N_m(0, a\boldsymbol{I}_m)\right)$ (the subscript KL is the abbreviation of Kullback-Leibler and D is the abbreviation of divergence), and is obtained by calculating:

$$\frac{1}{2}\mathrm{lndet}(a\boldsymbol{I}_m\Sigma_1^{-1}) + \frac{1}{2}\mathrm{tr}\left((a\boldsymbol{I}_m)^{-1}(\boldsymbol{\mu}_1\boldsymbol{\mu}_1^{\mathrm{T}} + \Sigma_1 - a\boldsymbol{I}_m)\right). \tag{4}$$

The optimization problem of GMM is described as:

$$\begin{aligned} &\min_{\boldsymbol{\mu}_1,\Sigma_1} D_{\mathrm{KL}}\left(N_m(\boldsymbol{\mu}_1, \Sigma_1)\,||\,N_m(0, a\boldsymbol{I}_m)\right) \\ &\text{s.t.}\quad \Pr\left(\frac{y_i\boldsymbol{x}_i{}^{\mathrm{T}}\boldsymbol{w} - y_i\boldsymbol{x}_i{}^{\mathrm{T}}\boldsymbol{\mu}_1}{\sqrt{\boldsymbol{x}_i^{\mathrm{T}}\Sigma_1\boldsymbol{x}_i}} \leqslant \frac{-y_i\boldsymbol{x}_i{}^{\mathrm{T}}\boldsymbol{\mu}_1}{\sqrt{\boldsymbol{x}_i^{\mathrm{T}}\Sigma_1\boldsymbol{x}_i}}\right) \leqslant 1-\rho \\ &\qquad \Sigma_1 > 0, \quad i = 1, \cdots, l. \end{aligned} \tag{5}$$

After omitting the constant terms in the objective function and transforming the constraints of Equation (5), we get:

$$\begin{aligned} &\min_{\boldsymbol{\mu}_1,\Sigma_1} \tfrac{1}{2}\left(-\mathrm{lndet}\Sigma_1 + \tfrac{1}{a}\mathrm{tr}(\Sigma_1) + \tfrac{1}{a}\boldsymbol{\mu}_1\boldsymbol{\mu}_1^{\mathrm{T}}\right) \\ &\qquad \text{s.t.}\ \ y_i\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{\mu}_1 \geqslant \Phi^{-1}(\rho)\sqrt{\boldsymbol{x}_i^{\mathrm{T}}\Sigma_1\boldsymbol{x}_i} \\ &\qquad\qquad \Sigma_1 > 0, \quad i = 1, \cdots, l, \end{aligned} \tag{6}$$

where $\Phi^{-1}(\rho)$ is the inverse cumulative distribution function of a standard normal distribution. $\Phi^{-1}(\rho)$ is further equal to $\sqrt{2}erf^{-1}(2\rho - 1)$, where erf^{-1} denotes the inverse Gauss error function.

Theorem 1. *The training samples* $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{l}$ *are given, and a prior distribution over the weight vector* $N_m(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$ *is set. Then, for any* $\delta \in [0,1]$ *and any posterior distribution* $N_m(\boldsymbol{\mu}_1, \Sigma_1)$*, the following holds with the probability of at least* $1-\delta$ *:*

$$\varphi\left(N_m(\boldsymbol{\mu}_1, \Sigma_1), D\right) \leqslant C_1 \frac{1}{l} \sum_{i=1}^{l} \Phi\left(-\frac{y_i \boldsymbol{x}_i^{\mathrm{T}} \boldsymbol{\mu}_1}{\sqrt{\boldsymbol{x}_i^{\mathrm{T}} \Sigma_1 \boldsymbol{x}_i}}\right) + C_2 \frac{D_{\mathrm{KL}}\left(N_m(\boldsymbol{\mu}_1, \Sigma_1) \,||\, N_m(\boldsymbol{\mu}_0, \Sigma_0)\right) + \ln \frac{2l}{\delta}}{l-1}, \tag{7}$$

where $\varphi(N_m(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1), D) = \mathrm{E}[\varphi(\boldsymbol{w}, (\boldsymbol{x}, y)) \,|\, (\boldsymbol{x}, y) \sim D, \boldsymbol{w} \sim N_m(\boldsymbol{\mu}_1, \Sigma_1)]$*,* $\varphi(\boldsymbol{w}, (\boldsymbol{x}, y))$ *is* $0-1$ *loss function,* $C_1 = 1 + \sqrt{2}/2$*,* $C_2 = 2 + \sqrt{2}/2$*, and* D *is the distribution of* $(\boldsymbol{x}, y)$ [22,27].

Proof of Theorem 1. See Appendix A. □

3. Kernel-Based Regression with Gaussian Distribution Weights

3.1. Optimization Problem of GDW-KR

A finite number of independent non-duplicate observations $\{(\boldsymbol{x}_i, t_i)\}_{i=1}^{l}$ with $\boldsymbol{x}_i \in \mathbf{R}^m$ and $t_i \in \mathbf{R}$ are considered. A kernel-based regression model approximates the unknown regression function $f(\boldsymbol{x})$ as follows:

$$\hat{f}(\boldsymbol{x}) = \sum_{j=1}^{l} w_j k(\boldsymbol{x}, \boldsymbol{x}_j), \tag{8}$$

where $k(\boldsymbol{x}, \boldsymbol{x}_j)$ is a predefined kernel function, and $\boldsymbol{w} = (w_1, \ldots, w_l)^{\mathrm{T}}$.

Definition 1. *(kernel function) A kernel is a function k that for all* $\boldsymbol{x}$*,* $\boldsymbol{z}$ *from a space* χ *(which needs not be a vector space) satisfies:*

$$k(\boldsymbol{x}, \boldsymbol{z}) = < \phi(\boldsymbol{x}), \phi(\boldsymbol{z}) >, \tag{9}$$

where ϕ *is a mapping from the space* χ *to a Hilbert space F that is usually called the feature space* $\phi : \boldsymbol{x} \in \chi \mapsto \phi(\boldsymbol{x}) \in F$ [28].

By assuming $\boldsymbol{w} \sim N_l(\boldsymbol{\mu}, \Sigma)$ with $\boldsymbol{\mu} \in \mathbf{R}^l$ and the positive definite covariance matrix $\Sigma \in \mathbf{R}^{l \times l}$, we maintain a distribution over alternative weight vectors rather than committing to a single specific vector. Let y_i denote the forecasted value by the model for a given observation $\boldsymbol{x}_i$, and we obtain:

$$y_i \sim N_l(\boldsymbol{K}_i \boldsymbol{\mu}, \boldsymbol{K}_i \Sigma \boldsymbol{K}_i^{\mathrm{T}}), \tag{10}$$

where $\boldsymbol{K}_i$ is the ith row of the symmetric kernel matrix $\boldsymbol{K}$, and $\boldsymbol{K}_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$, $i = 1, \ldots, l$, $j = 1, \ldots, l$. Weight vectors are required to make the target value be included in the confidence interval of the forecast value. Thus, we have the following constraint conditions:

$$\begin{aligned} &\boldsymbol{K}_i \boldsymbol{\mu} - \eta \sqrt{\boldsymbol{K}_i \Sigma \boldsymbol{K}_i^{\mathrm{T}}} \leqslant t_i, \\ &t_i \leqslant \boldsymbol{K}_i \boldsymbol{\mu} + \eta \sqrt{\boldsymbol{K}_i \Sigma \boldsymbol{K}_i^{\mathrm{T}}}, \quad i = 1, \ldots, l. \end{aligned} \tag{11}$$

The confidence interval needs to be large enough to impose a high confidence level. To make the level higher than 95%, η should be greater than 1.96 computed by $\Phi^{-1}(1-(1-0.95)/2)$. Considering the independence of noise between samples, $\boldsymbol{K}_i \Sigma \boldsymbol{K}_{j \neq i}^{\mathrm{T}}$ is set to be 0. Since the row vector $\boldsymbol{K}_i$ cannot be a zero vector, we have $\boldsymbol{K}_i \Sigma \boldsymbol{K}_i^{\mathrm{T}} > 0$, where Σ is a positive definite matrix. Hence, the covariance matrix of $\boldsymbol{K} \Sigma \boldsymbol{K}^{\mathrm{T}}$ should be a positive definite diagonal matrix:

$$\begin{aligned} &\boldsymbol{K}_i \Sigma \boldsymbol{K}_{j \neq i}^{\mathrm{T}} = 0, \\ &\Sigma > 0, i = 1, \cdots, l, j = 1, \cdots, l, \end{aligned} \tag{12}$$

which indicates that kernel matrix $\boldsymbol{K}$ must be invertible because $\text{rank}(\boldsymbol{K}\Sigma\boldsymbol{K}^{\mathrm{T}}) = l$ and $\text{rank}(\boldsymbol{K}\Sigma\boldsymbol{K}^{\mathrm{T}}) \leqslant \text{rank}(\boldsymbol{K}) \leqslant l$.

Under the constraint conditions (11) and (12), GDW-KR aims at the least informative distribution that has the smallest Kullback-Leibler divergence with respect to an isotropic Gaussian distribution $N_l(0, a\boldsymbol{I}_l)$ for some constant scalar $a > 0$. Thus, the optimization problem of GDW-KR is expressed as:

$$\begin{aligned} &\min_{\boldsymbol{\mu},\Sigma} -\tfrac{1}{2}\ln\det\Sigma + \tfrac{1}{2a}\text{tr}(\Sigma) + \tfrac{1}{2a}\boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{\mu} \\ &\text{s.t.} \quad \boldsymbol{K}_i\boldsymbol{\mu} - \eta\sqrt{\boldsymbol{K}_i\Sigma\boldsymbol{K}_i^{\mathrm{T}}} \leqslant t_i, \\ &\qquad t_i \leqslant \boldsymbol{K}_i\boldsymbol{\mu} + \eta\sqrt{\boldsymbol{K}_i\Sigma\boldsymbol{K}_i^{\mathrm{T}}}, \\ &\qquad \boldsymbol{K}_i\Sigma\boldsymbol{K}_{j\neq i}^{\mathrm{T}} = 0, \\ &\qquad \Sigma > 0, i = 1, \cdots, l, j = 1, \cdots, l. \end{aligned} \tag{13}$$

3.2. Simplification of Optimization Problem

In the problem (13), the number of unknown parameters is $l + l(l+1)/2$, which can be lowered by handling properly its constraints. First of all, let us suppose:

$$\boldsymbol{K}\Sigma\boldsymbol{K}^{\mathrm{T}} = \boldsymbol{\Lambda}, \tag{14}$$

where $\boldsymbol{\Lambda} = \text{diag}(\lambda_1^2, \ldots, \lambda_l^2)$, and $\lambda_i > 0$, $i = 1, \cdots, l$. If the diagonal elements of $\boldsymbol{\Lambda}$ are treated as unknown parameters taking the place of Σ, the number of unknown parameters in the problem (13) is reduced to $2l$. Then, the objective function of Equation (13) is rewritten as:

$$\min_{\boldsymbol{\mu},\boldsymbol{\Lambda}} -\frac{1}{2}\ln\det(\boldsymbol{K}^{-1}\boldsymbol{\Lambda}\boldsymbol{K}^{-1}) + \frac{1}{2a}\boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{\mu} + \frac{1}{2a}\text{tr}(\boldsymbol{K}^{-1}\boldsymbol{\Lambda}\boldsymbol{K}^{-1}). \tag{15}$$

As $\ln(\det(\boldsymbol{K}^{-1}\boldsymbol{\Lambda}\boldsymbol{K}^{-1})) = \ln(\det(\boldsymbol{K}^{-1}\boldsymbol{K}^{-1}\boldsymbol{\Lambda}))$, we have:

$$-\frac{1}{2}\ln\det(\boldsymbol{K}^{-1}\boldsymbol{\Lambda}\boldsymbol{K}^{-1}) = -\sum_{i=1}^{l}\ln\lambda_i - \frac{1}{2}\ln\det\boldsymbol{P}, \tag{16}$$

where $\boldsymbol{P} = \boldsymbol{K}^{-1}\boldsymbol{K}^{-1}$. Since $\text{tr}(\boldsymbol{K}^{-1}\boldsymbol{\Lambda}\boldsymbol{K}^{-1}) = \text{tr}(\boldsymbol{K}^{-1}\boldsymbol{K}^{-1}\boldsymbol{\Lambda})$ and both $\boldsymbol{K}^{-1}$ and $\boldsymbol{K}$ are symmetric and invertible matrices, we obtain:

$$\text{tr}(\boldsymbol{K}^{-1}\boldsymbol{K}^{-1}\boldsymbol{\Lambda}) = \frac{1}{2a}\sum_{i=1}^{l}(\boldsymbol{P})_{ii}\lambda_i^2, \tag{17}$$

where $(\boldsymbol{P})_{ii} > 0$. Disregarding the term $-\frac{1}{2}\ln\det\boldsymbol{P}$ in the objective function, problem (13) is rewritten as:

$$\begin{aligned} &\min_{\boldsymbol{\mu},\lambda} -\sum_{i=1}^{l}\ln\lambda_i + \tfrac{1}{2a}\sum_{i=1}^{l}(\boldsymbol{P})_{ii}\lambda_i^2 + \tfrac{1}{2a}\boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{\mu} \\ &\text{s.t.} \quad \boldsymbol{K}_i\boldsymbol{\mu} - \eta\lambda_i \leqslant t_i, \\ &\qquad t_i \leqslant \boldsymbol{K}_i\boldsymbol{\mu} + \eta\lambda_i, \\ &\qquad \lambda_i > 0, i = 1, \cdots, l. \end{aligned} \tag{18}$$

Assuming $\lambda_i = \lambda$ where in $i = 1, \cdots, l$, the problem of GMR is obtained as:

$$\begin{aligned} &\min_{\boldsymbol{\mu},\lambda} -l\ln\lambda + \tfrac{1}{2a}\lambda^2\sum_{i=1}^{l}\boldsymbol{P}_{ii} + \tfrac{1}{2a}\boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{\mu} \\ &\text{s.t.} \quad \boldsymbol{K}_i\boldsymbol{\mu} - \eta\lambda \leqslant t_i, \\ &\qquad t_i \leqslant \boldsymbol{K}_i\boldsymbol{\mu} + \eta\lambda, \\ &\qquad \lambda > 0, i = 1, \cdots, l. \end{aligned} \tag{19}$$

Comparing the problems (18) and (19) reveals that GMR is a special case of GDW-KR.

3.3. Analysis of Optimization Problem

Proper generalization of GDW-KR can be guaranteed by Theorem 1 based on the two-sided PAC-Bayesian theorem. However, that of GDW-KR is realized here by analyzing Equation (18) based on the empirical Rademacher complexity [29].

Definition 2. *(empirical Rademacher complexity) Let G be a family of functions mapping from X to $[a, b]$ and $(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_l)$ a fixed sample of size l with elements in $\boldsymbol{x}$. Then, the empirical Rademacher complexity of G with respect to $(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_l)$ is defined as:*

$$\hat{S}(G) = \mathop{\mathrm{E}}_{\sigma}\left[\sup_{g \in G}\left|\frac{2}{l}\sum_{i=1}^{l}\sigma_i g(\boldsymbol{x}_i)\right|\right],$$

where $\sigma = (\sigma_1, \ldots, \sigma_l)^{\mathrm{T}}$ with σ_i s independent uniform random variables taking values in $\{-1, +1\}$ [30].

Theorem 2. *GDW-KR can be properly generalized, which is guaranteed by keeping the balance between the empirical Rademacher complexity and the fitting error.*

Proof. The objective function of the problem (18) is rewritten as:

$$-a\sum_{i=1}^{l}\ln\lambda_i + \frac{1}{2}\sum_{i=1}^{l}(\boldsymbol{P})_{ii}\lambda_i^2 + \frac{1}{2}\boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{\mu}. \tag{20}$$

Suppose the function set is as follows:

$$Q_c = \{\sum_{j=1}^{l}\mu_j k(\boldsymbol{x}, \boldsymbol{x}_j)\,\Big|\,\boldsymbol{x} \in \mathbf{R}^m, \boldsymbol{\mu} \in \mathbf{R}^l, \boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{K}\boldsymbol{\mu} \leqslant c^2\}, \tag{21}$$

where c is a positive real number. Let $\hat{S}(Q_c)$ denote the empirical Rademacher complexity of Q_c.

Suppose another function set is defined as:

$$H_c = \{< \boldsymbol{\beta}, \phi(\boldsymbol{x}) > \,|\, ||\boldsymbol{\beta}|| \leqslant c\}, \tag{22}$$

where ϕ is the feature mapping corresponding to the kernel k.

For any $h(\boldsymbol{x})$ in H_c, letting $\boldsymbol{\beta} = \sum_{i=1}^{l}\mu_i\phi(\boldsymbol{x}_i)$ gives:

$$h(\boldsymbol{x}) =< \boldsymbol{\beta}, \phi(\boldsymbol{x}) >=< \sum_{i=1}^{l}\mu_i\phi(\boldsymbol{x}_i), \phi(\boldsymbol{x}) >= \sum_{i=1}^{l}\mu_i k(\boldsymbol{x}, \boldsymbol{x}_i), \tag{23}$$

and:

$$||\boldsymbol{\beta}||^2 =< \sum_{i=1}^{l}\mu_i\phi(\boldsymbol{x}_i), \sum_{j=1}^{l}\mu_j\phi(\boldsymbol{x}_j) >= \sum_{i,j=1}^{l}\mu_i\mu_j < \phi(\boldsymbol{x}_i), \phi(\boldsymbol{x}_j) > = \sum_{i,j=1}^{l}\mu_i\mu_j k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{K}\boldsymbol{\mu}. \tag{24}$$

Then, H_c is a superset of Q_c. Based on the derivation in [30], we obtain $\hat{S}(Q_c) \leqslant \hat{S}(H_c)$ and the following:

$$\begin{aligned}
\hat{S}(H_c) &= \mathop{\mathrm{E}}_{\sigma}\left[\sup_{h\in H_c}\left|\frac{2}{l}\sum_{i=1}^{l}\sigma_i h(\boldsymbol{x}_i)\right|\right] \\
&= \mathop{\mathrm{E}}_{\sigma}\left[\sup_{||\boldsymbol{\beta}||\leqslant c}\left|\left\langle \boldsymbol{\beta}, \frac{2}{l}\sum_{i=1}^{l}\sigma_i\boldsymbol{\phi}(\boldsymbol{x}_i)\right\rangle\right|\right] \\
&\leqslant \frac{2c}{l}\mathop{\mathrm{E}}_{\sigma}\left[||\sum_{i=1}^{l}\sigma_i\boldsymbol{\phi}(\boldsymbol{x}_i)||\right] \\
&= \frac{2c}{l}\mathop{\mathrm{E}}_{\sigma}\left[\left(\left\langle\sum_{i=1}^{l}\sigma_i\boldsymbol{\phi}(\boldsymbol{x}_i), \sum_{j=1}^{l}\sigma_j\boldsymbol{\phi}(\boldsymbol{x}_j)\right\rangle\right)^{1/2}\right] \\
&\leqslant \frac{2c}{l}\left(\mathop{\mathrm{E}}_{\sigma}\left[\sum_{i,j=1}^{l}\sigma_i\sigma_j k(\boldsymbol{x}_i,\boldsymbol{x}_j)\right]\right)^{1/2} \\
&= \frac{2c}{l}\left(\sum_{i=1}^{l}k(\boldsymbol{x}_i,\boldsymbol{x}_i)\right)^{1/2}.
\end{aligned}$$

Then, we have:

$$\hat{S}(F_c) \leqslant 2c\sqrt{\mathrm{tr}(\boldsymbol{K})}/l. \tag{25}$$

In view of Equation (21), c can be minimized by minimizing $\boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{K}\boldsymbol{\mu}$. Calculating by Cauchy–Schwarz inequality yields:

$$\boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{K}\boldsymbol{\mu} = \langle\boldsymbol{\mu}, \boldsymbol{K}\boldsymbol{\mu}\rangle \leqslant ||\boldsymbol{\mu}|| \cdot ||\boldsymbol{K}\boldsymbol{\mu}|| \leqslant ||\boldsymbol{K}|| \cdot ||\boldsymbol{\mu}||^2. \tag{26}$$

Since the kernel function is predefined, $\frac{1}{2}\boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{\mu}$ in Equation (20) can reduce the empirical Rademacher complexity of Q_c.

Under the constraints of the problem (18), the smaller λ_i, the less the fitting error. The term:

$$-a\sum_{i=1}^{l}\ln\lambda_i + \frac{1}{2}\sum_{i=1}^{l}(\boldsymbol{P})_{ii}\lambda_i^2.$$

prevents λ_i from getting too small or too large, and thus the model $\sum_{j=1}^{l}\mu_j k(\boldsymbol{x}^*, \boldsymbol{x}_j)$ is free from overfitting and underfitting the training data. So the term can be taken as a special loss function. Thereby, it can be concluded that proper values of a and η guarantee the balance between the empirical Rademacher complexity and the fitting error. Thus, GDW-KR promises a desirable generalization performance. Then, we have Theorem 2. □

Theorem 2 shows that balancing the empirical Rademacher complexity and the fitting loss is consistent with the two-sided PAC-Bayesian theorem for GDW-KR.

3.4. Solution of Optimization Problem

The results of regularized kernel-based regression are used to obtain the approximate solution of the problem (18). Regularized kernel-based regression is described as:

$$\begin{aligned}
&\min_{\boldsymbol{\mu},\varepsilon}\tfrac{1}{2}(\boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{\mu} + C\sum_{i=1}^{l}\varepsilon_i^2) \\
&\text{s.t. } \boldsymbol{K}_i\boldsymbol{\mu} - t_i = \varepsilon_i, \\
&\qquad i = 1, \cdots, l,
\end{aligned} \tag{27}$$

where C is the regularization parameter.

Let $\overline{\boldsymbol{\mu}}$ be the solution to Equation (27). Using the KKT conditions, $\overline{\boldsymbol{\mu}}$ is analytically computed as:

$$\overline{\boldsymbol{\mu}} = (\frac{\boldsymbol{I}}{C} + \boldsymbol{K}^{\mathrm{T}}\boldsymbol{K})^{-1}\boldsymbol{K}^{\mathrm{T}}\boldsymbol{t}. \tag{28}$$

Then, assuming that $\boldsymbol{\mu}$ is known as $\overline{\boldsymbol{\mu}}$ and ignoring the term $\frac{1}{2a}\boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{\mu}$ in the objective function, then we rewrite Equation (18) as:

$$\begin{array}{l} \min\limits_{\lambda_i} -\ln\lambda_i + \frac{(\boldsymbol{P})_{ii}}{2a}\lambda_i^2 \\ \text{s.t.} \quad \lambda_i \geqslant t_i^*, \\ \qquad \lambda_i > 0, \end{array} \tag{29}$$

where $t_i^* = |\boldsymbol{K}_i\overline{\boldsymbol{\mu}} - t_i|/\eta, i = 1, \cdots, l$. The second derivative of the objective function of the problem (29) is ${\lambda_i}^{-2} + (\boldsymbol{P})_{ii}/a$ that must be larger than 0 when $\lambda_i > 0$. Let $\overline{\lambda}_i$ be the solution to Equation (29), which is determined by:

$$\overline{\lambda}_i = \begin{cases} \sqrt{a/(\boldsymbol{P})_{ii}}, & t_i^* \leqslant \sqrt{a/(\boldsymbol{P})_{ii}}; \\ t_i^*, & t_i^* > \sqrt{a/(\boldsymbol{P})_{ii}}. \end{cases} \tag{30}$$

Thus, the algorithm consists of the following steps:

***Step 1*:** Make independent non-duplicated observations $\{(\boldsymbol{x}_i, t_i)\}_{i=1}^{l}$.
***Step 2*:** Select the kernel function, and choose the proper relevant parameter (s).
***Step 3*:** Compute $\boldsymbol{K}^{-1}$ and $\boldsymbol{P}$.
***Step 4*:** Solve the problem (27), and let $\overline{\boldsymbol{\mu}}$ be its solution.
***Step 5*:** Substitute $\boldsymbol{K}$ and $\overline{\boldsymbol{\mu}}$ into Equation (18), and obtain $\overline{\lambda}$ from Equation (30).

For the observation $\boldsymbol{x}^*$, the forecast value is:

$$\boldsymbol{s}^{\mathrm{T}}\overline{\boldsymbol{\mu}} = \sum_{j=1}^{l} \overline{\mu}_j k(\boldsymbol{x}^*, \boldsymbol{x}_j), \tag{31}$$

where $\overline{\boldsymbol{\mu}} = (\overline{\mu}_1, \ldots, \overline{\mu}_l)^{\mathrm{T}}$, and $\boldsymbol{s} = (k(\boldsymbol{x}^*, \boldsymbol{x}_1), \ldots, k(\boldsymbol{x}^*, \boldsymbol{x}_l))^{\mathrm{T}}$. And, the forecast interval is calculated as:

$$\left[\boldsymbol{s}^{\mathrm{T}}\overline{\boldsymbol{\mu}} - \eta^*\sqrt{\boldsymbol{s}^{\mathrm{T}}\overline{\Sigma}\boldsymbol{s}}, \boldsymbol{s}^{\mathrm{T}}\overline{\boldsymbol{\mu}} + \eta^*\sqrt{\boldsymbol{s}^{\mathrm{T}}\overline{\Sigma}\boldsymbol{s}}\right], \tag{32}$$

where $\overline{\Sigma} = \boldsymbol{K}^{-1}\overline{\boldsymbol{\Lambda}}\boldsymbol{K}^{-1}$ and $\eta^* > 0$.

3.5. Kernel Function and Model Selection

The kernel function plays an important role in kernel function methods. There are three common types of kernel functions: linear function, polynomial function and radial basis function (RBF). Many actual applications demonstrate that RBF tends to display its desirable performance under general smoothness assumptions. With no additional knowledge of the data set available, that makes the very reason for our adoption of the kernel function [31]:

$$k(\boldsymbol{x}, \boldsymbol{x}_j) = \exp\left\{-||\boldsymbol{x} - \boldsymbol{x}_j||^2/2\sigma^2\right\}. \tag{33}$$

Hyper-parameters also bear heavily on the generalization performance of kernel function methods. Model selection is to seek proper values of hyper-parameters commonly by means of cross-validation and grid search [32]. The k-fold cross-validation [12,13] partitions the training data into k disjoint subsets of approximately equal size. A series of k models are then trained, each using a different combination of $k-1$ subsets. The model selection criterion, such as the mean squared error, is then evaluated for each model in each case, utilizing the subset of the data not used in training that model. Recently, evolutional algorithms, such as genetic algorithm and particle swarm optimization, have

been adopted to guide the parameters selection process [33–36]. Regularized kernel-based regression uses genetic algorithm to seek the proper values of σ and C. An individual in genetic algorithm represents a possible parameter combination. The fitness of each individual is calculated by the k-fold cross-validation.

4. Experiments

Experiments were performed to verify the effectiveness of the proposed GDW-KR. The models were built using MATLAB 7.7. The quadratic problems involved were solved through the optimization toolbox QP in MATLAB. The experiments were made on a computer with a Win7 32 bit OS running on 3.1-GHz Intel Core i5-3450 with 4 GB RAM.

4.1. Formulation of Product-Design Time Forecast

To validate the proposed method, the design of plastic injection molds is studied. An injection mold is a kind of single-piece-designed product and the design process is usually driven by customer orders. The design process of injection mold is involved in many product development projects. The design time forecast is meaningful for the optimization of the whole product development process.

Factor values of product-design time are obtained by fuzzy measurable house of quality (FM-HOQ) [7]. Suppose that a design order for a kind of injection mold and the specification of the molding product are given to us. Then the customer demands should be analyzed and some useful mold characteristics should be extracted. The technical customer demands are taken into account. Some demands are originally described as quantitative information (e.g., the mold life is 3000 h), while others are expressed as qualitative information (e.g., the molding product precision is high). A unified fuzzy measurement scheme for all these demands is established, five linguistic levels are used [7]. The importance degrees of these demands are also represented by fuzzy weight sets.

For the specific mold design, the designer should specify the grades of membership of demand weights and demand measures, whose assignments can be made based on the customer demands given on the design order, and on the designer's objective evaluation of the degrees of importance and scope of the demands. A survey-based methodology is applied for identifying engineering characteristics and time factors, which is performed through self-administered questionnaires from several mold companies in Nanjing. Then, nine kinds of engineering characteristics are selected: mold structure, cavity number, wainscot gauge variation, injection pressure, injection capacity, ejector type, runner shape, manufacturing precision and form feature number. Then we can construct a planning FM-HOQ to map and measure characteristics for technical demands. Among the time characteristics with large influencing weights are structure complexity (SC), model difficulty (MD), wainscot gauge variation (WGV), cavity number (CN), mold size (MS) and form feature number (FFN), the first three of which are expressed as linguistic variables and the last three as numerical ones. Here, the influencing weights that indicate the influence degree on product-design time are different from the indexes of importance in FM-HOQs. Figure 1 presents the application procedure of our model.

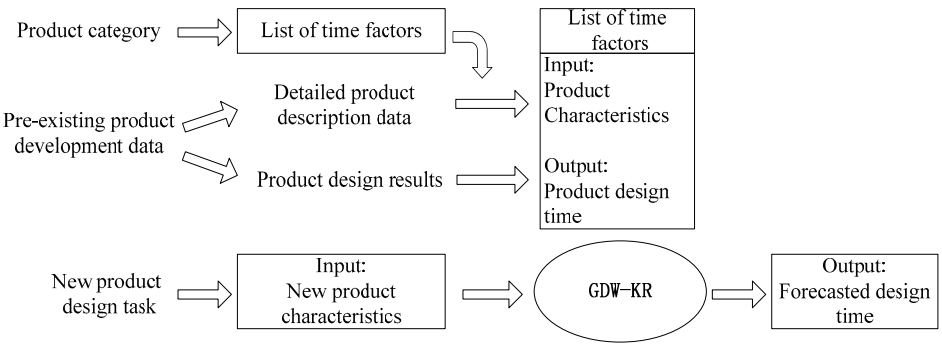

Figure 1. The application procedure of the GDW-KR model.

4.2. Product-Design Time Forecast Based on GDW-KR

In our experiments, 72 sets of molds with corresponding design time were obtained from a typical company. The detailed characteristic data and design time of these molds compose the corresponding patterns, as shown in Table 1. Numerical variables were normalized to be within [0, 1] by:

$$\overline{x}_i^d = \frac{x_i^d - \min(x_i^d|_{i=1}^l)}{\max(x_i^d|_{i=1}^l) - \min(x_i^d|_{i=1}^l)}, \tag{34}$$

where l denotes the number of samples, d the number of numerical variables, x_i^d the origin value of the dth number variable, and $\overline{x}_i^d$ the normalized value of the dth number variable. The linguistic variables, VL, L, M, H and VH, were transformed into the crisp values in terms of expertise: 0.1, 0.25, 0.5, 0.75 and 0.95.

Table 1. Training and testing data of injection model design.

Molds		Input Data						Desired Outputs (h)
No.	**Name**	**SC**	**MD**	**WGV**	**CN**	**MS**	**FFN**	
1	Global handle	L	L	L	4	3.1	3	23
2	Water bottle lid	H	L	H	4	0.56	7	45.5
3	Medicine lid	H	M	VL	4	1.5	6	37
4	Footbath basin	VL	VL	VL	1	0.5	3	10
5	Litter basket	L	M	H	1	2.1	12	42.5
6	Plastic silk flower	L	M	M	1	7.1	4	29.5
...	...	...	...	...	...	...	...	...
71	Paper-lead pulley	L	M	H	8	6.1	6	55
72	Winding tray	M	M	VH	1	2.18	7	41.5

First of all, η should be determined, mainly based on the confidence level at which the forecast interval includes the target. To make the confidence level higher than 95%, η should be greater than 1.96 computed by $\Phi^{-1}(1-(1-0.95)/2)$. The value of η is then set to 1.96, and the same is true of η^*. The target outputs were normalized to be within [0, 1].

The root mean square error (RMSE), the mean absolute percentage error (MAPE) and the mean absolute error (MAE) are three criteria used to optimize model parameters:

$$\begin{aligned} \text{RMSE} &= \sqrt{\frac{1}{l}\sum_{i=1}^{l}(t_i - \hat{t}_i)^2}, \\ \text{MAPE} &= \frac{1}{l}\sum_{i=1}^{l}\left|\frac{t_i - \hat{t}_i}{t_i}\right|, \\ \text{MAE} &= \frac{1}{l}\sum_{i=1}^{l}|t_i - \hat{t}_i|, \end{aligned}$$

where $\hat{t}_i$ is the forecast value for $\boldsymbol{x}_i$. The underlying assumption for using the RMSE is that the errors are not biased and follow a normal distribution [37]. The MAPE cannot be used if there is a zero value in $\{t_1, ..., t_l\}$, and puts a heavier penalty on negative errors ($t_i < \hat{t}_i$) than on positive errors. The MAE is suitable to be used for uniformly distributed errors. Because model errors are likely to follow a normal distribution rather than a uniform distribution, the RMSE is a better criterion than the MAE [37]. Thus, we apply the RMSE as a criterion for optimizing model parameters.

The whole data set is divided into several subsets. We choose one subset as the testing set and other ones as the training set. The combination of the genetic algorithm and 5-fold cross-validation is implemented to seek its optimal parameters to minimize the RMSE for the training set. In the genetic algorithm, each individual is evaluated by performing 5-fold cross-validation on the training set. After the optimal parameters are obtained, the model is estimated by using the training set. Then,

we calculate the forecast values and three criteria for the testing set. This procedure is repeated until each subset has been used once as the testing set. The testing results of the experiments are averaged over disjoint testing sets which cover the entire dataset. The selection ranges of σ and C are [0.01, 5] and [0.01, 10^6] respectively. The value of a was selected from [10^{-6}, 10^6].

The whole data set is first divided into six disjoint subsets. When subset 6 is used as the testing set, the optimal combinational parameters of regularized kernel-based regression are selected as $\sigma = 2.119$ and $C = 998746.999$, and the optimal parameter of GDW-KR turns out to be $a = 910.190$. As illustrated by Figure 2, our GDW-KR gives the valid forecast intervals, excluding T1 and T10. In T10, the forecast interval fails to cover its corresponding target value. In T1, the interval range is too large to provide useful information.

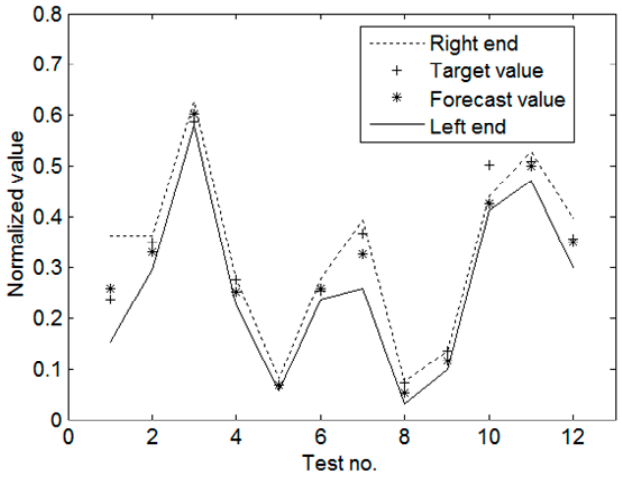

Figure 2. Testing results of GDW-KR when using subset 6 as the testing set.

Actual forecast values are listed in Table 2 for comparison of the models. The RMSE, the MAPE, the MAE and the average testing time are introduced to compare the forecast performance of different models. Here, the testing time means the time that is spent on solving the optimization problem and on obtaining the testing results when the hyper-parameters are given. Table 3 shows the results from four forecast models, which indicate that GDW-KR promises as high precision as other models do, and that GDW-KR can generate the forecast intervals simultaneously, thus facilitating product development to a certain extent.

Table 2. Forecast results from four different models when using subset 6 as the testing set.

No.	Designed Outputs	Forecast Results			
		Fv-SVM	v-SVR	GMR	GDW-KR
T1	31	35.315	31.236	30.134	32.928
T2	41	40.672	39.167	38.186	39.155
T3	62	62.029	63.521	64.075	63.291
T4	34.5	33.313	32.754	30.900	32.232
T5	16	16.424	16.761	16.877	16.156
T6	32.5	32.965	32.418	32.243	32.801
T7	42.5	40.243	38.516	38.566	38.811
T8	16.5	15.394	15.280	15.346	14.708
T9	22	21.521	21.066	19.963	20.417
T10	54.5	46.391	47.324	46.821	47.789
T11	55	54.149	52.771	53.509	54.304
T12	41.5	41.752	39.893	39.883	40.894

Table 3. Error statistics of four forecast models.

Model	Testing Results			Average Testing Time (s)
	RMSE	MAPE	MAE	
Fv-SVM	2.374	0.042	1.905	0.781
v-SVR	2.387	0.041	1.814	0.764
GMR	2.549	0.055	2.137	0.572
GDW-KR	2.366	0.041	1.848	0.583

The whole data set is then divided into 4 disjoint subsets. Figure 3 illustrates the results of GDW-KR from the first 54 training samples, and demonstrates that GDW-KR still performs well. Table 4 shows error statistics of four forecast models. GDW-KR does provide a satisfactory performance with small samples, and has thus been proved to be of better performance, appropriate to cases with small samples.

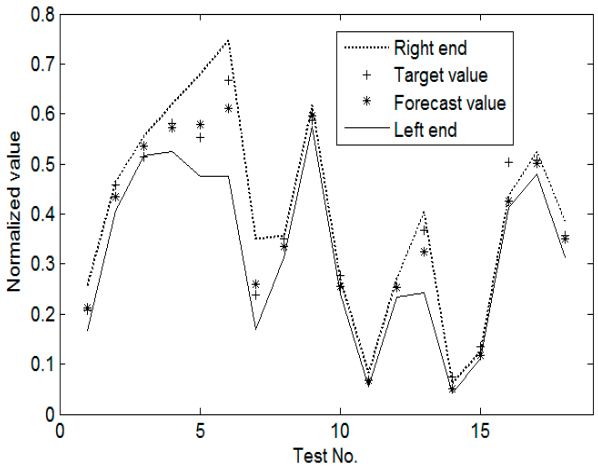

Figure 3. Testing results of GDW-KR from the first 54 training samples.

Table 4. Error statistics of four forecast models from 54 training samples.

Model	Testing Results			Average Testing Time (s)
	RMSE	MAPE	MAE	
Fv-SVM	2.156	0.038	1.615	0.770
v-SVR	2.141	0.037	1.617	0.728
GMR	2.205	0.038	1.624	0.568
GDW-KR	2.133	0.037	1.599	0.579

4.3. Extended Application of GDW-KR

Besides design time forecast, GDW-KR can also be extended to other regression problems with small samples. The Slump Test dataset, the Machine CPU dataset and the Yacht Hydrodynamics dataset, which are all from the UCI repository [38], are used to evaluate the extended application of GDW-KR. In these datasets, Fv-SVM behaves the same as v-SVR, as there is no fuzzy variable. Thus, the results of Fv-SVM are not presented here. Each dataset is divided into 6 disjoint subsets. In our experiments, both the target output and numerical attributes were normalized to be within [0, 1].

The Concrete Slump Test covers seven input and three output variables as well as 103 data points. The 28-day Compressive Strength is taken as the desired output variable. For the case of the Concrete Slump Test, the results of GDW-KR are compared with those of other two models. Concrete Slump Test results are shown in Figure 4. The three error indices of different three models are given in Table 5.

On the Slump Test, GDW-KR offers forecast values with high accuracy and forecast intervals with good validity.

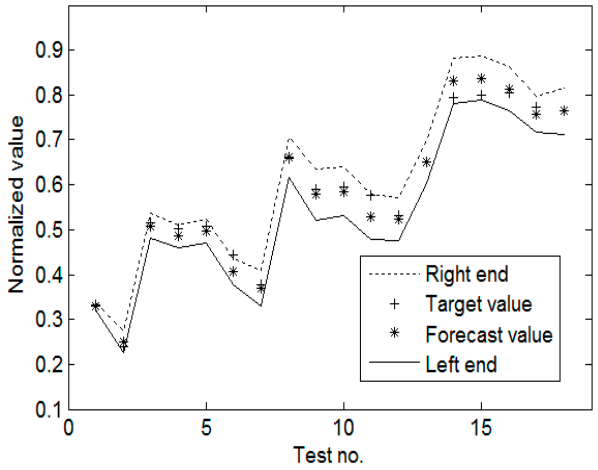

Figure 4. Concrete slump test results of GDW-KR when using subset 6 as the testing set.

Table 5. Error statistics of three forecast models on the Slump Test dataset.

Model	Testing Results			Average Testing Time (s)
	RMSE	MAPE	MAE	
v-SVR	0.021	0.044	0.014	0.795
GMR	0.023	0.047	0.015	0.583
GDW-KR	0.019	0.055	0.014	0.607

For the Machine CPU dataset and the Yacht Hydrodynamics, the error statistics of three forecast models are presented in Tables 6 and 7, respectively. Figures 5 and 6 indicate the forecast results when using subset 6 as the testing set.

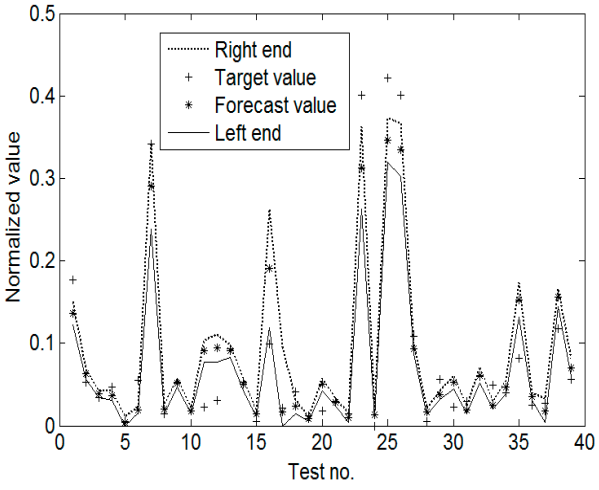

Figure 5. Machine CPU results of GDW-KR when using subset 6 as the testing set.

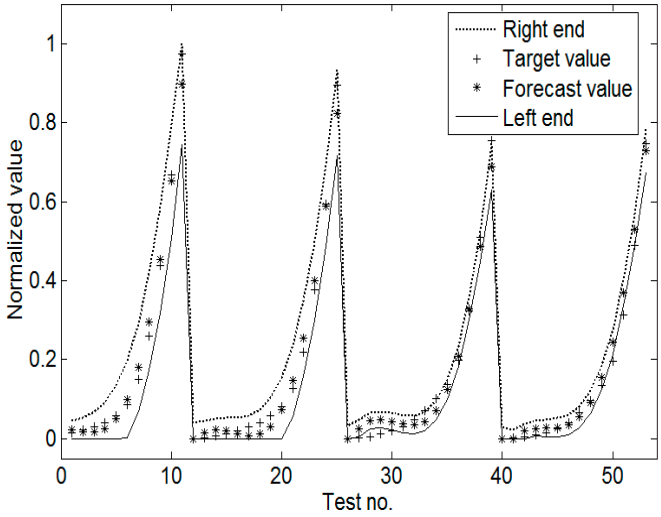

Figure 6. Yacht Hydrodynamics results of GDW-KR when using subset 6 as the testing set.

Table 6. Error statistics of three forecast models on the Machine CPU.

Model	Testing Results			Average Testing Time (s)
	RMSE	MAPE	MAE	
v-SVR	0.038	0.660	0.024	0.941
GMR	0.039	0.638	0.023	0.695
GDW-KR	0.038	0.555	0.023	0.783

Table 7. Error statistics of three forecast models on the Yacht Hydrodynamics.

Model	Testing Results			Average Testing Time (s)
	RMSE	MAPE	MAE	
v-SVR	0.034	3.585	0.025	1.196
GMR	0.036	3.772	0.027	0.710
GDW-KR	0.034	3.471	0.026	0.894

5. Conclusions

The control and decision of product development are based on the reasonable degree of the distribution of product design time. In design time forecasting, the problems of small samples and heteroscedastic noise ought to be considered.

This paper has presented a new model of kernel-based regression with Gaussian distribution weights for product-design time forecasts, which combines Gaussian margin machines with kernel-based regression. The kernel method performs well for the problem of small samples. Unlike GMR, which assumes that the covariance matrix of the forecast values in the training set is an identity matrix multiplied by a positive scalar, GDW-KR assumes that this matrix is a positive definite diagonal matrix. GDW-KR is more suitable for addressing the problem of heteroscedastic noise than GMR, and has the advantage of providing both point forecasts and confidence intervals simultaneously.

The plastic injection mold was studied before modeling. For convincing evaluation, experiments with 72 real samples were conducted. Results from them have verified that GDW-KR promises not only as high forecast accuracy as F*v*-SVM and *v*-SVR but forecast intervals crucial to the control and decision of product development. Undoubtedly, GDW-KR benefits from the merits of Gaussian margin machines.

Acknowledgments: This work was jointly funded by the National Natural Science Foundation of China under Grants 50875046 and 60934008 the Fundamental Research Funds for the key Universities of China under Grant 2242014K10031, and the Priority Academic Program Development of Jiangsu Higher Education Institutions (PAPD). We thank the three reviewers and Li Lu for their valuable comments and suggestions.

Author Contributions: Zhi-Gen Shang wrote the first draft. Hong-Sen Yan corrected and improved it. Both authors have read and approved the final manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

Proof of Theorem 1. Suppose $p, q \in [0,1]$, and let $\mathrm{D_{KL}}(p \,||\, q)$ denote the Kullback-Leibler divergence between a Bernoulli variable with bias p to a Bernoulli variable with bias q. Then, we have:

$$\mathrm{D_{KL}}(p \,||\, q) = p\ln(p/q) + (1-p)\ln((1-p)/(1-q)).$$

If $q > p$, we have $\mathrm{D_{KL}}(p \,||\, q) \geqslant (q-p)^2/(2q)$, which implies that if $\mathrm{D_{KL}}(p \,||\, q) \leqslant x$, then:

$$q \leqslant p + \sqrt{2px} + 2x.$$

Using $\sqrt{px} \leqslant (p+x)/2$, we have:

$$q \leqslant (1+\sqrt{2}/2)p + (2+\sqrt{2}/2)x = C_1 p + C_2 x. \tag{A1}$$

Let S be $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{l}$. We obtain:

$$\varphi\left(N_m(\boldsymbol{\mu}_1, \Sigma_1), S\right) = \frac{1}{l}\sum_{i=1}^{l} \varphi\left(N_m(\boldsymbol{\mu}_1, \Sigma_1), (\boldsymbol{x}_i, y_i)\right) = \frac{1}{l}\sum_{i=1}^{l} \Pr(y_i \boldsymbol{x}_i{}^{\mathrm{T}} \boldsymbol{w} \leqslant 0) = \frac{1}{l}\sum_{i=1}^{l} \Phi^{-1}(-\frac{y_i \boldsymbol{x}_i^{\mathrm{T}} \boldsymbol{\mu}_1}{\sqrt{\boldsymbol{x}_i^{\mathrm{T}} \Sigma_1 \boldsymbol{x}_i}}).$$

Based on the two-sided PAC-Bayesian theorem (or a Gaussian version of a theorem of McAllester) [27], we have for any $\delta \in [0,1]$, with probability at least $1-\delta$ over S, for all posterior distributions $N_m(\boldsymbol{\mu}_1, \Sigma_1)$, the following holds:

$$\mathrm{D_{KL}}\left(\varphi\left(N_m(\boldsymbol{\mu}_1, \Sigma_1), S\right) \,||\, \varphi\left(N_m(\boldsymbol{\mu}_1, \Sigma_1), D\right)\right) \leqslant \frac{\mathrm{D_{KL}}\left(N_m(\boldsymbol{\mu}_1, \Sigma_1) \,||\, N_m(\boldsymbol{\mu}_0, \Sigma_0)\right) + \ln\frac{2l}{\delta}}{l-1}. \tag{A2}$$

Equation (A2) demonstrates that the average generalization error diverges from the average training error by no more than a quantity which depends on the Kullback-Leibler divergence between the posterior and prior distributions over weight vectors.

Combining Equations (A1) and (A2) yields for any $\delta \in [0,1]$, with probability at least $1-\delta$ over S, for $N_m(\boldsymbol{\mu}_1, \Sigma_1)$, the following holds:

$$\varphi\left(N_m(\boldsymbol{\mu}_1, \Sigma_1), D\right) \leqslant C_1 \frac{1}{l}\sum_{i=1}^{l} \Phi(-\frac{y_i \boldsymbol{x}_i^{\mathrm{T}} \boldsymbol{\mu}_1}{\sqrt{\boldsymbol{x}_i^{\mathrm{T}} \Sigma_1 \boldsymbol{x}_i}}) + C_2 \frac{\mathrm{D_{KL}}\left(N_m(\boldsymbol{\mu}_1, \Sigma_1) \,||\, N_m(\boldsymbol{\mu}_0, \Sigma_0)\right) + \ln\frac{2l}{\delta}}{l-1}. \;\square$$

References

1. Cho, S.H.; Eppinger, S.D. A simulation-based process model for managing complex design projects. *IEEE Trans. Eng. Manag.* **2005**, *52*, 316–328. [CrossRef]
2. Yan, H.S.; Wang, B.; Xu, D.; Wang, Z. Computing completion time and optimal scheduling of design activities in concurrent product development process. *IEEE Trans. Syst. Man Cybern. Part. A Syst. Hum.* **2010**, *40*, 76–89. [CrossRef]

3. Yang, Q.; Zhang, X.F.; Yao, T. An overlapping-based process model for managing schedule and cost risk in product development. *Concurr. Eng. Res. Appl.* **2012**, *20*, 3–7. [CrossRef]
4. Basher, H.A.; Thomson, V. Models for estimating design effort and time. *Des. Stud.* **2001**, *22*, 141–155. [CrossRef]
5. Griffin, A. Modeling and measuring product development cycle time across industries. *J. Eng. Technol.* **1997**, *14*, 1–24. [CrossRef]
6. Jacome, M.F.; Lapinskii, V. NREC: Risk assessment and planning for complex designs. *IEEE Des. Test Comput.* **1997**, *14*, 42–49. [CrossRef]
7. Xu, D.; Yan, H.S. An intelligent estimation method for product design time. *Int. J. Adv. Manuf. Technol.* **2006**, *30*, 601–613. [CrossRef]
8. Burges, C.J.C. A tutorial on support vector machines for pattern recognition. *Data Min. Knowl. Discov.* **1998**, *2*, 110–115. [CrossRef]
9. Chen, S.T. Mining informative hydrologic data by using support vector machines and elucidating mined data according to information entropy. *Entropy* **2015**, *17*, 1023–1041. [CrossRef]
10. Vapnik, V.N. *The Nature of Statistical Learning Theory*, 2nd ed.; Springer-Verlag New York, Inc.: New York, NY, USA, 1999.
11. Schölkopf, B.; Smola, A.J.; Williamson, R.; Bartlett, P.L. New support vector algorithms. *Neural Comput.* **2000**, *12*, 1207–1245. [CrossRef] [PubMed]
12. Santiago-Paz, J.; Torres-Roman, D.; Figueroa-Ypiña, A.; Argaez-Xool, J. Using generalized entropies and OC-SVM with Mahalanobis kernel for detection and classification of anomalies in network traffic. *Entropy* **2015**, *17*, 6239–6257. [CrossRef]
13. Ibrahim, R.W.; Moghaddasi, Z.; Jalab, H.A.; Noor, R.M. Fractional differential texture descriptors based on the Machado entropy for image splicing detection. *Entropy* **2015**, *17*, 4775–4785. [CrossRef]
14. Benkedjouh, T.; Medjaher, K.; Zerhouni, N.; Rechak, S. Remaining useful life estimation based on nonlinear feature reduction and support vector regression. *Eng. Appl. Artif. Intel.* **2013**, *26*, 1751–1760. [CrossRef]
15. Kivinen, J.; Smola, A.J.; Williamson, R.C. Online learning with kernels. *IEEE Trans. Signal Process.* **2004**, *52*, 2165–2176. [CrossRef]
16. Liu, W.F.; Pokharel, P.P.; Principe, J.C. The kernel least mean square algorithm. *IEEE Trans. Signal Process.* **2008**, *56*, 543–554. [CrossRef]
17. Chen, B.D.; Zhao, S.L.; Zhu, P.P.; Principe, J.C. Quantized kernel least mean square algorithm. *IEEE Trans. Neural Netw. Learn. Syst.* **2012**, *23*, 22–32. [CrossRef] [PubMed]
18. Chen, B.D.; Zhao, S.L.; Zhu, P.P.; Principe, J.C. Quantized kernel recursive least squares algorithm. *IEEE Trans. Neural Netw. Learn. Syst.* **2013**, *24*, 1484–1491. [CrossRef] [PubMed]
19. Wu, Z.Z.; Shi, J.H.; Zhang, X.; Ma, W.T.; Chen, B.D. Kernel recursive maximum correntropy. *Signal Process.* **2015**, *117*, 11–16. [CrossRef]
20. Yan, H.S.; Xu, D. An approach to estimating product design time based on fuzzy v-support vector machine. *IEEE Trans. Neural Netw.* **2007**, *18*, 721–731. [PubMed]
21. Hao, P.Y. New support vector algorithms with parametric insensitive/margin model. *Neural Netw.* **2010**, *23*, 60–73. [CrossRef] [PubMed]
22. Crammer, K.; Mohri, M.; Pereira, F. Gaussian margin machines. In Proceedings of the 12th International Conference on Artificial Intelligence Statistics, Clearwater, FL, USA, 16–18 April 2009; pp. 105–112.
23. Shang, Z.G.; Yan, H.S. Forecasting product design time based on Gaussian margin regression. In Proceedings of the 10th International Conference on Electronic Measurement & Instruments, Chengdu, China, 16–18 August 2011; pp. 86–89.
24. Huang, G.B.; Zhu, Q.Y.; Siew, C.K. Extreme learning machine: theory and applications. *Neurocomputing* **2006**, *70*, 489–501. [CrossRef]
25. Feng, G.; Huang, G.B.; Lin, Q.; Gay, R. Error minimized extreme learning machine with growth of hidden nodes and incremental learning. *IEEE Trans. Neura Netw.* **2009**, *20*, 1352–1357. [CrossRef] [PubMed]
26. Shang, Z.G.; He, J.Q. Confidence-weighted extreme learning machine for regression problems. *Neurocomputing* **2015**, *148*, 544–550. [CrossRef]
27. McAllester, D. Simplified PAC-Bayesian margin bounds. In Proceedings of the 16th conference on Learning Theory and 7th Kernel Workshop, Washington DC, WA, USA, 24–27 August 2003; pp. 203–215.

28. Shawe-Taylor, J.; Sun, S.L. A review of optimization methodologies in support vector machines. *Neurocomputing* **2011**, *74*, 3609–3618. [CrossRef]
29. Robin, C.G.; Theodore, B.T. Quadratic programming formulations for classification and regression. *Optim. Meth. Softw.* **2009**, *24*, 175–185.
30. Shawe-Taylor, J.; Cristianini, N. *Kernel Methods for Pattern Analysis*; Cambridge University Press: Cambridge, UK, 2004.
31. Smola, A.J.; Schölkopf, B.; Müller, K. The connection between regularization operators and support vector kernels. *Neural Netw.* **1998**, *11*, 637–649. [CrossRef]
32. Li, B.; Song, S.J.; Li, K. A fast iterative single data approach to training unconstrained least squares support vector machines. *Neurocomputing* **2013**, *115*, 31–38. [CrossRef]
33. Hong, W.C. Chaotic particle swarm optimization algorithm in a support vector regression electric load forecasting model. *Energy Convers. Manag.* **2009**, *50*, 105–117. [CrossRef]
34. Yuan, S.F.; Chu, F.L. Fault diagnosis based on support vector machines with parameter optimization by artificial immunization algorithm. *Mech. Syst. Signal Process.* **2007**, *21*, 1318–1330. [CrossRef]
35. Pai, P.F.; Hong, W.C. Forecasting regional electricity load based on recurrent support vector machines with genetic algorithms. *Electr. Power Syst. Res.* **2005**, *74*, 417–425. [CrossRef]
36. Lin, S.W.; Lee, Z.J.; Chen, S.C.; Tseng, T.Y. Parameter determination of support vector machine and feature selection using simulated annealing approach. *Appl. Soft Comput.* **2008**, *8*, 1505–1512. [CrossRef]
37. Chai, T.; Draxler, R.R. Root mean square error (RMSE) or mean absolute error (MAE)?—Arguments against avoiding RMSE in the literature. *Geosci. Model Dev.* **2014**, *7*, 1247–1250. [CrossRef]
38. UC Irvine Machine Learning Repository. Available online: http://archive.ics.uci.edu/ml (accessed on 17 June 2016).