A handbook of

Tree-Classifier for Gaussian process regression

Outlier Detection and Feature Delection

Bin Cao

Materials Genome Institute, Shanghai University, Shanghai 200444, China

Zhejiang Laboratory, Hangzhou 311100, China

We propose a novel machine learning model, named Tree-Classifier for Gaussian process regression (TCGPR), for detecting abnormal recorded data and highlighting essential features in the field of materials informatics. The proposed model utilizes the global Gaussian messy factor (GGMF) to profile the coherence of data distributions, which allows for the evaluation of consistency and stability of an datum in an infinite mapping space. Empirical evaluations of the model demonstrate its outstanding performance in outlier detection and feature selection tasks on small material datasets compared to other popular machine learning strategies. This approach is particularly useful when dealing with small material datasets that have large level of noise. The proposed model provides a valuable tool for data-driven science and has the potential to accelerate the development of material informatics and is open-source.

Keywords: Materials informatics, TCGPR, GGMF, outlier detection, feature selection

GGMF

The TCGPR algorithm has been developed as a pre-processing technique for detecting and separating heterogeneous distributions caused by the presence of outliers and/or redundant features within datasets. Messy order, which are defined as data points or features that are inconsistent or redundant, are particularly susceptible to a factor termed as the global Gaussian messy factor (GGMF). TCGPR is based on the concept of GGMF, which emphasizes the coherence of the Gaussian distribution among data, in order to detect outliers and redundant features. The main objective of TCGPR is to identify anomalies and/or partition datasets by minimizing GGMF and clarifying the main distribution within the data. To achieve this, TCGPR utilizes the Gaussian radial kernel function within the Gaussian Process regression (GPr) model, along with Leave One Out Cross-Validation (LOOCV)(Arlot and Celisse, 2010; Pedregosa et al., 2011) techniques to fit the datasets and evaluate the coherence of data. GGMF assesses the coherence of the data distribution across the entire dataset by examining the response surface of the GPr model, and takes into account the length scale parameter in the Gaussian radial kernel function,

$$K(\boldsymbol{X}_i, \boldsymbol{X}_j) = exp\left(-\frac{\left\|\boldsymbol{x}_i - \boldsymbol{x}_j\right\|^2}{2\vartheta^2}\right), \tag{1}$$

where $\|\cdot\|$ represents the Euclidean distance of argument vector, $\boldsymbol{X}_i$ and $\boldsymbol{X}_j$ are datum argument vectors. The length scale parameter, ϑ, is associated with the data dimension. A smaller ϑ implies a higher 'activity', i.e., lower correlation among the data (Huang et al., 2006), which corresponds to an unreasonably sharp fitting surface.

Consider a dataset containing n data follows a definite distribution $D = \{\boldsymbol{X}_i = (x_{i1}, \cdots, x_{im}), \boldsymbol{Y}_i = (y_{i1}, \cdots, y_{ip})\}\ (i = 1, \cdots, n)$, $\boldsymbol{X}_i \in \mathbb{R}^m$ and $\boldsymbol{Y}_i \in \mathbb{R}^p$. The inclusion of an exterior datum (x^*, y^*) into an existing dataset may be warranted if the new observation follows the same distribution as the existing data and is unlikely to disrupt the data consistency. In such case, a GPr model with Gaussian radial kernel is trained on the expended dataset $D^* = D \cup (x^*, y^*)$, where the optimized length scale parameter (ϑ) models the response surface and evaluates the offset of new datum from the distribution in a mapped infinite feature space. The LOOCV is performed to predict the optimal response surface for each validation datum, generating a prediction response vector, $\widehat{\boldsymbol{Y}} = \begin{pmatrix} \widehat{\boldsymbol{Y}}_i \\ \cdots \\ \widehat{\boldsymbol{Y}}_n \end{pmatrix}$.

A good data distribution's consistency evaluation factor should be stable and robust. Thus, we apply a logarithmic transformation to smooth the variation of indicators and avoid the fluctuation induced by singular values. The GGMF is defined as follows,

$$GGMF = \frac{(1-R(\boldsymbol{Y},\widehat{\boldsymbol{Y}}))}{\ln(\vartheta+c_2)}, \tag{2}$$

Where $R(\boldsymbol{Y},\widehat{\boldsymbol{Y}})$ is a fitting goodness evaluator, e.g., Pearson correlation coefficient (R, default setting), Coefficient of determination (R^2), etc.. The $\boldsymbol{Y} = \begin{pmatrix} \boldsymbol{Y}_1 \\ \cdots \\ \boldsymbol{Y}_n \end{pmatrix}$ and $\widehat{\boldsymbol{Y}} = \begin{pmatrix} \widehat{\boldsymbol{Y}}_i \\ \cdots \\ \widehat{\boldsymbol{Y}}_n \end{pmatrix}$ are real and prediction response matrixes. c is a smooth constant and default as 1. The increasing the value of ϑ is desirable because it denotes the decreases of model's sensitivity (Huang et al., 2006).

TCGPR takes into account the evaluation uncertainty by treating GGMF as a normal distribution. The modeling approach uses LOOCV to estimate prediction standard deviations (symbolized s_i) associated with each validation datum. Specifically, LOOCV leaves out one datum at a time and then fits a Gpr model to the remaining data to predict the standard deviation of the left-out point's distribution. This process is repeated for each data point in the set, resulting in $n+1$ prediction standard deviations (where $n+1$ is the number of data points). The mean of prediction standard deviation, σ, is expressed as $\sigma = \frac{1}{n}\sum_{i=1}^{n+1} s_i$ for gauging the robustness of this GGMF evaluation. Henceforth, the probability distribution of the Gaussian Graphical Model Feature (GGMF) for a given dataset, which incorporates an identified data point

(x^j, y^j), is represented by a normal distribution $N(GGMF_j, \sigma_j^2)$ as specified in equation (3).

$$GGMF \sim N(GGMF_j, \sigma_j^2) \tag{3}$$

Here, $GGMF_j$ is the factor computed using the equation (2), and the notation $N(\cdot,\cdot)$ denotes the normal distribution. Consequently, the TCGPR employs this approach to carry out the tasks of detecting outliers and selecting relevant features.

Sequential Identification and Expected Decrease

In order to detect anomalies, TCGPR employs the Sequential Forward/Backward Identification (SFI/SBI) algorithm, which is depicted in Figure 1. The SFI approach begins with a small set of cohesive data points and then adds a batch of $p \geq 1$ data points/features sequentially from the dataset to identify outliers/redundant features. The algorithm maximizes the Expected Decrease (ED) of the GGMF by adding these batches of data/features. On the other hand, the SBI method starts with the full dataset and sequentially drops out data points/features. The SFI approach leads to a dynamic expansion based on an initial dataset/features set without any outliers/redundant features, viz., which follows a definite distribution.

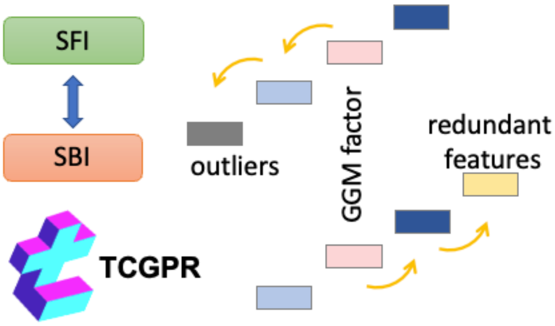

Fig.1 function modules of TCGPR package.

To illustrate the forward data screening module of TCGPR, we consider a specific initial dataset that is firstly initialized and continuously expanded by adding new data. The data infilling process follows a chain along the decrease of the GGMF. At any given link t, there exists a sub-dataset, S^t (follows same distribution), along with several candidates, S_j (contains newly added data), having larger data capacity. TCGPR selects one of the candidates by considering the expected decrease of GGMF. In order to achieve this, an effective strategy is to maximize the expected distance between the current dataset's GGMF value, $GGMF^t$, and the expanded dataset's GGMF value, $GGMF_j$, which results in the following expression:

$$ED = E\left(max\left[GGMF^t - N\_GGMF_j,\ 0\right]\right), \tag{4}$$

Where $E(\cdot)$ represents the exception. The $GGMF^t$ is a real value for a given dataset, S^t, while $N\_GGMF_j \sim N\left(GGMF_j, \sigma_j^2\right)$ is a normal distribution as introduced in Eq.(3).

The use of truncate function $max\left[GGMF^t - N\_GGMF_j,\ 0\right]$ ensures that the expected value of $GGMF_j$ is bounded below by the current value $GGMF^t$. Let term $Z \triangleq \left(GGMF^t - N\_GGMF_j\right)$ obeys the distribution of $N\left(GGMF^t - GGMF_j, \sigma_j^2\right)$. The expected value of Eq.(4) is yielded in,

$$\int_0^{+\infty} Zf(Z)dZ = \left(GGMF^t - GGMF_j\right)\phi\left(\frac{\left(GGMF^t-GGMF_j\right)}{\sigma_j}\right) + \sigma_j\varphi\left(\frac{\left(GGMF^t-GGMF_j\right)}{\sigma_j}\right) \tag{5}$$

where $f(Z) = \frac{1}{\sqrt{2\pi\sigma_j^2}} exp\left(-\frac{\left(Z-\left(GGMF^t-GGMF_j\right)\right)^2}{2\sigma_j^2}\right)$ is the probability distribution function of Z, $\varphi(\cdot)$ and $\phi(\cdot)$ are the standard normal density function and standard normal cumulative distribution function.

Assuming a dataset with N discrete data, the initial cohesive dataset in TCGPR is comprised of p $(p \ll N)$ data, and the addition path is chosen as q, viz., add q data into the last dataset at each sequential along the expansion chain. This results in C_{N-p}^q possible candidates at current link. If the inequality $C_{N-p}^q \leq C$ holds true, then the brute force search method is employed for selecting the optimal solution among the C_{N-p}^q possible situations with the highest expected decreases in GGMF, as defined in equation (4). However, if the inequality does not hold true, C situations are randomly sampled from the entire set of C_{N-p}^q cases and evaluated to recommend the best candidate. Here, C denotes an exceedingly large constant. After the initial expansion, the dataset is augmented with $(p + q)$ discrete data points. Subsequently, additional q

data points are incorporated by exploring $C_{N-(p+q)}^{q}$ possible situations, and the process is continued in the direction of expected decrease of GGMF until either the remaining data is insufficient or the stopping criteria of TCGPR are met, i.e., the fitting-goodness saturates. Conversely, the Sequential Bayesian Inference (SBI) follows a reverse process to detect anomalies .

Initial dataset and Stop criteria

The selection of an appropriate initial dataset is of utmost importance in the context of SFI-TCGPR. The initial dataset serves as the basis for the dynamic expression chain and has a direct impact on the outcome of data screening process. SFI-TCGPR offers two distinct interfaces for selecting the initial dataset: (1) Interface one, where the user can directly assign an initial dataset/feature set, is particularly useful for researchers who possess extensive domain knowledge pertaining to their data. (2) Interface two involves a brute-force search of all C_N^p potential initial datasets, with the best candidate being selected based on the highest expected decrease of the GGMF as compared to an empty dataset. Here, N represents the number of studied data, p refers to the data capacity of the initial dataset, and the GGMF of an empty dataset is defined as $+\infty$. In cases where C_N^p exceeds a predetermined threshold C, C random samples are compared and the best candidate is chosen to conserve computing power.

TCGPR constructs a long chain of sub-datasets or sub-feature sets, where each link represents a specific subset. Along the chain, the compacity of subsets is sequentially increased (SFI) or decreased (SBI). The Pearson correlation coefficient, R, is recorded along the expansion path with the calculated GGMF, which is represented by $R_0, R_1, \cdots, R_k \cdots, R_E \in [0,1]$, where R_0 represents the starting terminal and R_E represents the ending terminal. To identify outliers in the SFI-TCGPR, the E^{th} link is converted to an end terminal if $R_E < (1-\eta) \times max\{R_0, R_1, \cdots, R_{E-1}\}$ (criteria 1), where η is a tolerance coefficient, default $\eta = 0.1$. On the other hand, in the outlier identification of SBI-TCGPR and feature selection modules, only neighboring links are considered, and the stop criteria is set as $R_E < (1+\tau) \times R_{E-1}$ (criteria 2), where τ is a tolerance coefficient, default $\tau = 0.005$. It is worth noting that criteria 1 is a relative slack constraint. The SFI-TCGPR and SBI-TCGPR are illustrated in Algorithm 1 and Algorithm 2 respectively.

Algorithm 1 : SFI-TCGPR for outlier identify (feature selection follows the same algorithm)

```
Input a studied dataset $S$ contains $N$ descried data, the capacity of initial dataset $p$, the chain path of $q$, stopping tolerance $\eta$ ($\tau$, for feature selection), and up-search threshold $C$.
Initialize select an initial dataset $S_0$ with $p$ data (interface 1 or 2) and calculate $GGMF$ of dataset $S_0$ as $GGMF_0$.
While True (infinite iteration):
>>> Set residual dataset as $S_r$, where $S_r = S - S_0$
>>> produce $C_{|S_r|}^{q}$ candidate datasets, denotes as $S_j$ ($j = 1, \ldots, C_{|S_r|}^{q}$), $S_j$ contains $|S_0| + q$ data.
>>> if $C_{|S_r|}^{q} \leq C$:
        >>>for $j$ in range($C_{|S_r|}^{q}$):
            >>> calculate the expected decrease of $GGMF$ of dataset of the $S_j$ as $ED_j$
        >>>$ED_* = \max_{j=1,..,C_{|S_r|}^{q}} (ED_j)$, $S_*$ is the associated dataset and $GGMF_*$ is the correspond GGMF
>>> else $C_{|S_r|}^{q} > C$:
        >>> selected $C$ candidates randomly from $C_{|S_r|}^{q}$ situations
        >>>for $j$ in range($C$):
            >>> calculate the expected decrease of $GGMF$ of dataset of the $S_j$ as $ED_j$
        >>>$ED_* = \max_{j=1,..,\mathrm{C}} (ED_j)$, $S_*$ is the associated dataset and $GGMF_*$ is the correspond GGMF
>>>calculate R value of $S_*$ as $R_t$ at $t^{th}$ iteration (t starts from 0)
>>>if $R_t < (1-\eta) \times max\{R_0, R_1, \cdots, R_{t-1}\}$ ($R_t < (1+\tau) \times R_{t-1}$ for feature selection) :
    >>>break iteration and print out the result
>>>else:
    >>>$S_0 := S_*$, $GGMF_0 := GGMF_*$.
```

Algorithm 2 : SBI-TCGPR for outlier identify (recommend, has higher computation efficiency)

```
Input a studied dataset $S$ contains $N$ descried data, the chain path of $q$, stopping tolerance $\tau$, and up-search threshold $C$.
Initialize calculate $GGMF$ of input dataset $S$ as $GGMF_0$.
While True (infinite iteration):
>>> Set residual dataset as $S_r$, where $S_r = \emptyset$
>>> produce $C_{|S_r|}^{q}$ candidate datasets, denotes as $S_j$ $(j = 1, \dots, C_{|S_r|}^{q})$, $S_j$ contains $|S_0| - q$ data.
>>> if $C_{|S_r|}^{q} \leq C$:
        >>>for $j$ in range($C_{|S_r|}^{q}$):
            >>> calculate the expected decrease of $GGMF$ of dataset of the $S_j$ as $ED_j$
        >>>$ED_* = \max_{j=1,..,C_{|S_r|}^{q}} (ED_j)$, $S_*$ is the associated dataset and $GGMF_*$ is the correspond GGMF
>>> else $C_{|S_r|}^{q} > C$:
        >>> selected $C$ candidates randomly from $C_{|S_r|}^{q}$ situations
        >>>for $j$ in range($C$):
            >>> calculate the expected decrease of $GGMF$ of dataset of the $S_j$ as $ED_j$
        >>>$ED_* = \max_{j=1,..,\mathrm{C}} (ED_j)$, $S_*$ is the associated dataset and $GGMF_*$ is the correspond GGMF
>>>calculate R value of $S_*$ as $R_t$ at $t^{th}$ iteration (t starts from 0)
>>>if $R_t < (1 + \tau) \times R_{t-1}$ :
    >>>break iteration and print out the result
>>>else:
    >>>$S_0 := S_*$, $GGMF_0 := GGMF_*$.
```

Execution Template

Our team has developed an open-source Python package, which is available on GitHub at github.com/Bin-Cao/TCGPR. This package is designed to be compatible with multiple operating systems, including Windows, Linux, and MAC OS, and can be easily installed and used.

Installation and updating

The TCGPR package can be easily installed on personal laptops using the pip command. Simply execute "pip install PyTcgpr" and the latest version of the TCGPR model will be automatically downloaded to your computer. If a new version of TCGPR is developed, users can check for the latest version by using the package version checking function in Python, which will alert them to any updates published by our team. The latest version can then be installed using the command "pip install --upgrade PyTcgpr".

Running

These commands are designed to be user-friendly and easy to understand, enabling researchers and practitioners to easily incorporate the TCGPR algorithm into their own projects and analyses. To facilitate the application of the package, we provide the following operating commands:

Code 1 : outlier identify (SFI, python)	**Code 2** : outlier identify (SBI, python)	**Code 3** : feature selection (SFI, python)
# coding=utf-8	# coding=utf-8	# coding=utf-8
from PyTcgpr import TCGPR	**from PyTcgpr import TCGPR**	**from PyTcgpr import TCGPR**
# name of studied dataset	# name of studied dataset	# name of studied dataset
dataSet = "data.csv"	**dataSet** = "data.csv"	**dataSet** = "data.csv"
# the capacity of initial dataset p	# the chain path of q	# assign mission type
initial_set_cap = 3	**sampling_cap** =2	**Mission** = ‘FEATURE’
# the chain path of q	# stopping tolerance τ, sufficient small	# the capacity of initial dataset p
sampling_cap =2	**ratio** = 0.001	**initial_set_cap** = 3
# stopping tolerance η	# up-search threshold C	# the chain path of q
ratio = 0.1	**up_search** = 500	**sampling_cap** =2
# up-search threshold C	# the task number of regression	# stopping tolerance τ, sufficient small
up_search = 500	**target** = 1	**ratio** = 0.001

# the task number of regression **target** = 1 # execute TCGPR model **TCGPR.fit** (**filePath** = dataSet, **initial_set_cap** = initial_set_cap, **sampling_cap** = sampling_cap, **ratio** = ratio, **up_search** = up_search **target**= target,)	# execute TCGPR model **TCGPR.fit** (**filePath** = dataSet, **sampling_cap** = sampling_cap, **ratio** = ratio, **up_search** = up_search **target**= target,)	# up-search threshold C **up_search** = 500 # the task number of regression **target** = 1 # execute TCGPR model **TCGPR.fit** (**filePath** = dataSet, **Mission**= Mission **initial_set_cap** = initial_set_cap, **sampling_cap** = sampling_cap, **ratio** = ratio, **up_search** = up_search
Note : symbol **#** is the interpretation of the parameter at the following line, and more information see, **github.com/Bin-Cao/TCGPR**		

Reference

Arlot, S., Celisse, A., 2010. A survey of cross-validation procedures for model selection.

Huang, D., Allen, T.T., Notz, W.I., Zeng, N., 2006. Global optimization of stochastic black-box systems via sequential kriging meta-models. Journal of global optimization 34, 441-466.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., 2011. Scikit-learn: Machine learning in Python. the Journal of machine Learning research 12, 2825-2830.