Estimation of Stellar Atmospheric Parameters with Light Gradient Boosting Machine Algorithm and Principal Component Analysis

Junchao Liang1, Yude Bu1, Kefeng Tan2, Jingchang Pan3, Zhenping Yi3, Xiaoming Kong3, and Zhou Fan2

1 School of Mathematics and Statistics, Shandong University, Weihai, 264209, Shandong, People’s Republic of China; buyude@sdu.edu.cn
2 CAS Key Laboratory of Optical Astronomy, National Astronomical Observatories, Chinese Academy of Sciences, Beijing, 100101, People’s Republic of China
3 School of Mechanical, Electrical & Information Engineering, Shandong University, Weihai, 264209, Shandong, People’s Republic of China

Received 2021 August 21; revised 2022 January 7; accepted 2022 January 18; published 2022 March 8

Abstract

In this paper, we propose a new method to estimate stellar atmospheric parameters with photometric data, which is based on principal component analysis (PCA) and light gradient boosting machine (LightGBM) algorithms. We first use PCA to transform all band photometric data ($u$, $v$, $g$, $r$, $i$, and $z$) and then utilize LightGBM to estimate stellar atmospheric parameters. The experimental results show that the root mean square errors of the method for estimating the effective temperature, surface gravity, and metallicity are 90 K, 0.40 dex, and 0.20 dex, respectively. We then compare PCA + LightGBM with the original photometry data (OPD) + LightGBM and the color index data (CID) + LightGBM. The experimental results show that the performance of PCA + LightGBM is better than that of CID + LightGBM and OPD + LightGBM, and PCA + LightGBM can solve the problems of model instability and inaccurate estimation results caused by direct use of OPD or CID as input for LightGBM. We believe the new features obtained by PCA can be used on photometric data collected by the Chinese Space Station Telescope.

Unified Astronomy Thesaurus concepts: Astronomy data analysis (1858); Astronomical methods (1043); Interstellar abundances (832)

1. Introduction

The main task of modern astrophysics is to find out when and how galaxies were formed and evolved. Our own galaxy, the Milky Way, offers a unique opportunity to study galaxies in considerable detail by measuring and analyzing the properties of stars (Juric et al. 2008). The principal properties of stellar analysis are determined by the accurate measurement of stellar atmospheric parameters, such as effective temperature ($T_{\text{eff}}$), surface gravity ($\log g$), and metallicity ([Fe/H]). The photometric method is widely used to estimate the atmospheric parameters of stars.

Effective temperature is relatively easy to estimate compared with other stellar atmospheric parameters, and has been widely considered in previous studies. Strömgren (1956) and Crawford (1958) proposed a $uvby\beta$ photometric system, which yielded four commonly used color indexes—namely $b - y$, $m_1$, $c_1$, and $\beta$—and accurately calculated the atmospheric parameters ($T_{\text{eff}}$, $\log g$, and [Fe/H]) for stars of different spectral types. Magain (1987) used the InfraRed Flux Method (IRFM) to establish the relation between $B - V$, $b - y$, $V - K$, and $T_{\text{eff}}$, which can be used to estimate the $T_{\text{eff}}$ of dwarf and subgiant stars. Alonso et al. (1996) studied 475 dwarf and subdwarf stars sampled by using the IFRM and obtained their $T_{\text{eff}}$ with an average accuracy of approximately 1.5%. Alonso et al. (1999) applied the IRFM to a sample of approximately 500 giant stars to derive their $T_{\text{eff}}$ with an internal mean accuracy of approximately 1.5% and a maximum uncertainty in the zero point of the order of 0.9%. They also established a relation among $T_{\text{eff}}$, color, and [Fe/H]. Holmberg et al. (2007) consolidated the calibrations of $uvby\beta$ photometry into $T_{\text{eff}}$, [Fe/H], distance, and age for $F$ and $G$ stars. For different $b - y$ ranges, the relation between $b - y$ and $T_{\text{eff}}$ was established, and the accuracy was approximately 60 K. Casagrande et al. (2010) presented an investigation based on the IRFM using a large set of solar twins to set the absolute zero point of the $T_{\text{eff}}$ scale within a few degrees. They also established a color–$T_{\text{eff}}$ relation with [Fe/H] ranging from $-5$ to 0.4. For luminosity classes IV and V dwarfs and for classes II and III giants, Huang et al. (2015) devised an empirical calibration of $T_{\text{eff}}$ and color based on metallicity. The method was based on direct $T_{\text{eff}}$ measurements of approximately 200 nearby stars, which were better than 2.5%. The $T_{\text{eff}}$ range of calibration was $3100$–$10,000$ K for dwarf stars with $M_5$ to A0 spectral types and $3100$–$5700$ K for giant stars with $K5$ to $G5$ spectral types.

Estimating [Fe/H] with photometric information has also been widely studied. Olsen (1984) discussed four-color $uvby$ photometry for several hundred late-type dwarf stars of types $G$, $K$, and $M$. The relation between $uvby$ photometry and [Fe/H] was established, and the accuracy of [Fe/H] was approximately 0.17 dex. Nissen (1981) performed photoelectric measurements on $F0$–$G2$ giant stars and determined the [Fe/H] of 179 stars. Holmberg et al. (2007) revised the relation between [Fe/H] and $uvby$ photometry established in previous studies and obtained a more complicated relation among [Fe/H] and $b - y$, $m_1$, and $c_1$, with an accuracy of 0.12 dex. Juric et al. (2008) used more than 6000 $FG$ stars from the Sloan Digital Sky Survey (SDSS) to perform polynomial fitting of [Fe/H] with $u - g$ and $g - r$, and the experimental accuracy was almost similar to the spectral prediction results.

Most of the above studies use the color index data to determine the stellar parameters. However, there are two obstacles for extending the use of color index data (CID) to large-scale prediction of stellar atmospheric parameters. First, these features have strong correlation with each other, which
The $m$th regression decision tree of GBDT fitting $T_{dd}$ with photometric data. The box represents feature selection according to the mean square error, and the ellipse represents the prediction result. The $m$th regression decision tree fits the residuals (the difference between the real value and the predicted value) of the $(m-1)$ th regression decision tree. With an increase in $m$, the fitting result of the $m$th regression decision tree tends to 0. The final fitting result of GBDT is the sum of the fitting results of samples in each regression decision tree.

Figure 1.

will cause the instability of the model. Therefore, the application of previous models to different types of stars is difficult. Second, not all bands have been fully considered. For example, it is believed that only $u-g$ and $g-r$ are sensitive features of [Fe/H]. However, the experiments showed that the weighted combination of all bands could impact the [Fe/H] of stars.

Considering the aforementioned problems, we first use principal component analysis (PCA) to construct features and eliminate the correlation among variables, and then employ the light gradient boosting machine (LightGBM) algorithm to estimate stellar atmospheric parameters. The experimental results show that PCA+LightGBM method can estimate the stellar parameters accurately, and is superior to LR, support vector regression (SVR), artificial neural networks (ANNs), and other tree model algorithms (such as XGBoost and random forest) in terms of the performance.

The paper is organized as follows. We introduce the LightGBM algorithm and PCA in Sections 2 and 3, respectively. Then, we introduce the data in Section 4. We compare various experiments in Section 5 and detail the extraction of the feature importance of stellar atmospheric parameters. Finally, we summarize the study in Section 6.

2. Method: LightGBM

LightGBM is a gradient boosting decision tree algorithm (GBDT) with gradient-based one-side sampling (GOSS) and exclusive feature bundling (EFB). It uses a histogram-based algorithm to find the best split points. The GOSS and EFB techniques are introduced to process a large number of data instances and features, respectively. The histogram-based algorithm can reduce the number of split points and accelerate training. Hence, LightGBM is mainly used to solve the problem that traditional GBDT is slow during massive data training. For our experiment, as the feature dimension is very low and the sample size is relatively large, we only use GOSS and the histogram-based algorithm. We briefly introduce GBDT in Section 2.1 and then explain GOSS, histogram-based algorithm in Sections 2.2 and 2.3, respectively. The detailed introduction of EFB can be found in Ke et al. (2017).

2.1. GBDT

In the field of machine learning, a learner whose generalization performance is slightly better than random guessing is called a weak learner. GBDT is an ensemble algorithm that takes regression decision trees as weak learners. Each regression decision tree of GBDT is established in order, and the resulting structure is shown in Figure 1. In this section, we briefly introduce the calculation process for GBDT. For more detail, we refer the reader to Friedman (2001).

Suppose that $(x_i, y_i)$, $i = 1, 2, …, N$ represents the data set, where $x_i = (x^1_i, x^2_i, …, x^n_i)^{T}$ is the magnitude data vector corresponding to n bands, and $y_i$ contains the stellar atmospheric parameters corresponding to $x_i$. We refer to $y_i$ as the target value of $x_i$. Given an input $x_i$, the output of GBDT is

$$F_M(x_i) = \sum_{m=1}^{M} h_m(x_i),$$  \hspace{1cm} (1)

where $h_m$ is the $m$th regression decision tree. $h_m(x_i)$ denotes the predicted value of the $m$th regression decision tree for $x_i$. Constant $M$ is the number of regression decision trees. GBDT takes the following steps to obtain Equation (1).

1. Initialize $F_0(x_i) = 0$.
2. Calculate residuals $g_{mi} = y_i - F_{m-1}(x_i)$ and obtain the sample set:

$$O = \{(x_i, g_{mi}), …, (x_N, g_{mN})\}.$$  \hspace{1cm} (2)

3. Fit sample $O$ to obtain the regression decision tree, $h_m(x_i)$.
4. Update $F_m(x_i) = F_{m-1}(x_i) + h_m(x_i)$.

In the four aforementioned steps, $m$ ranges from 1 to $M$ in turn. Finally, a total of $M$ regression decision trees are created, and the prediction value is obtained through the following formula:

$$F_M(x_i) = F_{M-1}(x_i) + h_m(x_i) = \sum_{m=1}^{M} h_m(x_i).$$

Many ensemble algorithms based on decision trees are different in the third step. GBDT adopts a simpler method to build $h_m$ in the third step. That is, each region (parent node) is recursively divided into two regions (left and right child node) in sample
set \( O \), and the output value of each region is determined by performing the following three steps to construct a binary decision tree.

1. Starting from the root node, select the optimal splitting variable \( j \), and the optimal splitting point \( d \) (the value of variable \( j \)), and minimize the square error between the predicted value and the real value of the two regions (left and right child nodes):

\[
\min_{j,d} \left[ \min_{c_1} \sum_{x_i \in L} (g_{mi} - c_1)^2 + \min_{c_2} \sum_{x_i \in R} (g_{mi} - c_2)^2 \right],
\]

where \( L = \{x_i \in O: x_i^j \leq d\} \), \( R = \{x_i \in O: x_i^j > d\} \), and \( c_1 \) and \( c_2 \) are the predicted values of the left and right child nodes, respectively.

2. After the first step, the left and right child nodes are used as new parent nodes of the next split. According to the first step, split the new parent node.

3. Repeat the first and second steps and establish \( h_m \).

After the three aforementioned steps are performed, \( h_m \) is established, as shown in Figure 1. The residuals in set \( O \) are different. The samples with small residuals converge to the real value, and fitting them again leads to overfitting of the GBDT and increases the calculation cost of Equation (2). To improve performance, GOSS and the histogram-based algorithm are proposed and described in the following subsection.

2.2. GOSS

GOSS is a statistical method to treat different residual values \((g_{n}(x_i))\) differently. It achieves a balance between reducing the amount of data and improving the accuracy of GBDT by changing the second step of GBDT. The principle of GOSS is as follows:

1. Sort absolute values of residuals in descending order as follows:

   \[ |g_{m1}| \geq |g_{m2}| \geq \ldots \geq |g_{mN}|. \]

2. The data corresponding to the first \( a\% \) residuals are recorded as subset \( A \). The remaining data are marked as subset \( B \), including the \( 1 - a\% \) samples with the smallest residuals.

   \[
   A = \{(x_1, g_{m1}), \ldots, (x_p, g_{mp})\} \\
   B = \{(x_{p+1}, g_{m(p+1)}), \ldots, (x_N, g_{mN})\}
   \]

   where \( \frac{p}{N} = a\% \).

3. Randomly sample \( b\% \) of data from subset \( B \), and record it as subset \( C \). The sample weight is assigned as \((1-a)/b\) for subset \( C \), then

   \[
   C = \left\{ \left( x_{p+1}, \frac{1-a}{b} g_{m(p+1)} \right), \ldots, \left( x_{p+q}, \frac{1-a}{b} g_{m(p+q)} \right) \right\}
   \]

   where \( \frac{q}{N} = b\% \).

The above three steps correspond to the calculation of the residual error of the second step of GBDT. Ke et al. (2017) proved that GOSS can increase the generalization ability of the model compared to GBDT by paying attention to samples with large residuals without losing accuracy.

2.3. Histogram-based Algorithm

The main cost in GBDT is to find the best split points. In order to alleviate the problem, Ke et al. (2017) used histogram-based algorithm to bucket continuous variable values into discrete bins and used these bins to construct feature histograms during training. That is, the \( i \)th variable is divided into \( s \) bins in ascending order of \( x_i^j \) \((i = 1, 2, \ldots, N)\), and each bin (called feature histogram) includes \( \frac{1}{s} \) samples. In this way, the splitting point \( d \), is changed from \( x_1^j, x_2^j, \ldots, x_N^j \) to \( x_{1b}^j, x_{2b}^j, \ldots, x_{Nb}^j \). Finally, in the same way as Equation (2), we find the optimal split variable \( j \) and the optimal splitting point \( d \) in \( x_{1b}^j, x_{2b}^j, \ldots, x_{Nb}^j \) to obtain the optimal \( h_m \) by minimizing the following formula on \( A \cup C \):

\[
\min_{j,d} \left[ \min_{c_1} \left( \sum_{x_i \in A_k} (g_{mi} - c_1)^2 + \sum_{x_i \in C_k} \left( \frac{1-a}{b} g_{mi} - c_1 \right)^2 \right) \right] \]

where

\[
A_k = \{x_i \in A: x_i^j \leq d\}, \quad A_k = \{x_i \in A: x_i^j > d\} \\
C_k = \{x_i \in C: x_i^j \leq d\}, \quad C_k = \{x_i \in C: x_i^j > d\},
\]

and \( c_1 \) and \( c_2 \) are the predicted values for each node.

It changes the third step of GBDT by reducing the number of split points (from \( N \) to \( s \)) to achieve the effect of speedup. Because \( s \) is much less than \( N \), this model’s fitting speed is faster than that of GBDT.

3. PCA

In this section, we briefly introduce PCA. We refer to Abdi & Williams (2010) and Jolliffe & Cadima (2016) for more information about PCA. PCA is widely used in spectral classification and dimensionality reduction (Storrö-Lombardi et al. 1994; Singh et al. 1998; Bu et al. 2014), which is also used in the data processing of regression analysis to reduce data collinearity and improve the stability of regression model.

For consistency and simplicity, we suppose that the original data contains \( n \) variables, and \( X = (x_1, x_2, \ldots, x^n) \), in which \( x^i \) \((i = 1, 2, \ldots, n)\) is the \( i \)th band variable. PCA combines the original variables into new variables through the following linear combination

\[
\begin{align*}
F_1 &= a_1^T X \equiv a_{11}x^1 + a_{12}x^2 + \ldots + a_{1n}x^n \\
F_2 &= a_2^T X \equiv a_{21}x^1 + a_{22}x^2 + \ldots + a_{2n}x^n \\
&\vdots \\
F_n &= a_n^T X \equiv a_{n1}x^1 + a_{n2}x^2 + \ldots + a_{nn}x^n
\end{align*}
\]  

(3)

where \( F_i \) \((i = 1, 2, \ldots, n)\) is the \( i \)th new variable (called the \( i \)th principal component). \( a_{ij} \) \((i = 1, 2, \ldots, n)\) is the coefficient to be calculated, which satisfies \( a_1^Ta_1 = 1 \).

To make each principal component represent the potential structure of the original data, PCA requires that the variance of each principal component be reduced in turn and the covariance of different principal components be 0, which can
be expressed mathematically as follows.

\[
\begin{align*}
\text{var}(F_i) &\geq \text{var}(F_j) \geq \ldots \geq \text{var}(F_n) \\
\text{cov}(F_i, F_j) & = 0, \quad i \neq j.
\end{align*}
\]

Based on Equation (3), we obtain the following.

\[
\text{var}(F_i) = \text{cov}(a_i^T X, a_i^T X) = a_i^T \text{var}(X) a_i.
\]

We calculate each principal component in turn according to the order of principal component variance in Equation (4). Therefore, \( F_1 \) is first calculated by using \( a_1 \), and \( a_i \) can be obtained according to Equation (5) in the following way:

\[
a_i = \arg \max a_i \text{var}(F_1) = \arg \max a_i^T \Sigma a_i,
\]

where \( a_i^T a_i = 1 \). According to Equation (6), we can calculate \( a_1 \) through the Lagrange multiplier method by optimizing

\[
\lambda(a_1) = a_1^T \text{var}(X) a_1 - \lambda_1(a_1^T a_1 - 1),
\]

where \( \lambda_1 \) is the Lagrange factor. As

\[
\begin{align*}
\frac{\partial \lambda}{\partial a_i} & = 2(\text{var}(X) - \lambda)a_i = 0 \\
\frac{\partial \lambda}{\partial \lambda} & = a_1^T a_1 - 1 = 0,
\end{align*}
\]

\([\text{var}(X) - \lambda_1] = 0 \), and \( \lambda_1 \) is the eigenvalue of \( \text{var}(X) \). According to Equation (7), \( a_1 \) is the eigenvector of \( \text{var}(X) \) belonging to eigenvalue \( \lambda_1 \). Thus,

\[
\text{var}(F_1) = a_1^T \text{var}(X) a_1 = a_1^T \lambda_1 a_1 = \lambda_1 a_1^T a_1 = \lambda_1.
\]

According to Equations (6)–(8), \( \text{var}(F_1) \) is the largest eigenvalue of \( \text{var}(X) \), and \( a_1 \) is the eigenvector corresponding to the largest eigenvalue of \( \text{var}(X) \). Similarly, \( a_i \) is the eigenvector corresponding to the \( i \)th largest eigenvalue of \( \text{var}(X) \). Substituting \( a_i \) into Equation (3), we obtain each principal component \( F_i \). These new variables, \( (F_1, F_2, \ldots, F_n) \), represent the potential structure of the original data, this is convenient to explore the data and improve the generalization ability of the model.

### 4. Data Introduction

In this section, we introduce data sources and the data preprocessing process. To simulate the future Chinese Space Station Telescope (CSST) and obtain more similar data, we select common stars of the Apache Point Observatory Galactic Evolution Experiment Data Release 16 (APOGEE DR16), Stellar Abundance and Galactic Evolution (SAGE), and Panoramic Survey Telescope and Rapid Response System Data Release 1 (Pan-STARRS DR1) to obtain \( u, v, g, r, i, z \)-band photometric data and stellar atmospheric parameters (Teff, log g, and [Fe/H]) of 37,979 stars.

APOGEE is one of the programs in the Sloan Digital Sky Survey III (Majewski et al. 2017) and Sloan Digital Sky Survey IV (Blanton et al. 2017), and it is a large-scale, stellar spectroscopic survey conducted in the near-infrared (\( H \)-band) portion of the electromagnetic spectrum. APOGEE is a large spectroscopic survey that is expected to exceed 700,000 stars, mainly consisting of galactic red giants from all stellar populations, but also including the Magellanic clouds and other nearby dwarf galaxy red giants, as well as numerous cool (FGKM) dwarf stars (Zasowski et al. 2013, 2017). The stellar atmospheric parameters in our experiment are taken from APOGEE DR16, which contains high-resolution (\( R \sim 22,500 \)), near-infrared (15140–16940 Å) spectra for approximately 430,000 stars covering both the northern and southern sky, from which radial velocities, stellar parameters, and chemical abundances of up to 26 species are determined (Jónsson et al. 2020).

The photometric data of the \( u \) and \( v \) bands are taken from the SAGE survey (Fan et al. 2018), which used the 2.3 m Bok telescope of the Steward Observatory of the University of Arizona to complete the observation of the \( u \) and \( v \) bands, obtaining high-precision photometric data for approximately 500 million stars. The actual 100\( u \) depths of the single-epoch images are \( u \sim 17.3 \) and \( v \sim 16.8 \) (Zheng et al. 2019). The SAGE system is designed to be more sensitive to stellar atmospheric parameters than \( uvby/\beta \), so the accuracy of the SAGE system’s stellar parameter measurement could actually be better than that of the broadband photometry.

The photometry data for the \( g, r, i, z \) bands are taken from Pan-STARRS DR1. Located on Haleakala, and dedicated to sky survey observations, Pan-STARRS system is a 1.8 m aperture, \( f/4.4 \) telescope illuminating a 1.4 Gpixel detector spanning a 3\( \times \)3 field of view (Tonry et al. 2012). Its mean wavelengths of \( g, r, i, z \) are 486.6 nm, 621.5 nm, 754.5 nm, and 867.9 nm, respectively. Although the Pan-STARRS filter system has a lot in common with the filter system used in previous investigations such as the SDSS (York et al. 2000), the \( g \) filter extends 20 nm in the direction of \( g \) of SDSS, which costs the price of 5577 Å sky emission for obtaining higher sensitivity and a lower photometric redshift system. The \( z \) filter is cut off at 920 nm, making its response different from that of the detector defined by the \( z \) filter of SDSS. Therefore, these bands are more sensitive to stellar atmospheric parameters. Chambers & Pan-STARRS Team (2018) introduced Pan-STARRS data in detail.

We first correct the measured photometric magnitudes of each band for interstellar reddening with \( E(B-V) \) (Schlegel et al. 1998) and adopted reddening coefficients deduced from Schlafly & Finkbeiner (2011). After reddening corrections, we use following criteria to select data:

1. Delete the samples with missing values of each band.
2. According to the distribution of magnitude error, samples with magnitude error of \( u, v \) bands less than 0.1 mag and magnitude error of bands (\( g, r, i, z \)) less than 0.05 mag are selected.

We obtain a data set containing 37,979 stars in total. The distribution of stellar atmospheric parameters of these stars is shown in Figure 2.

It is clear that our data sets have a large number of stars with high temperatures and high [Fe/H], and include a large number of dwarf stars and giant stars. Therefore, our data sets cover a wide range of atmospheric stellar parameters, and the performance based on these data sets can fully demonstrate the generalization ability of our method.

### 5. Data Experiment

In this section, we first use original photometric data (OPD) + LightGBM and CID + LightGBM experiments to explain in detail the two problems faced by previous studies, namely, the strong correlation between variables and the lack of...
consideration of the weighted combination of the whole band in the color index method. Finally, we use a PCA + LightGBM experiment to analyze the performance improvement after solving the above problems.

We randomly divide the data into a training set (80%) and test set (20%), and use root mean square error (RMSE; see Equation (9)) as the metric of performance of the model

\[ RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2}. \]

We use maximal information coefficient (MIC) measure dependence between variables, such as linear and nonlinear relations, even for nonfunctional dependence (Reshef et al. 2011). The MIC values range from 0 to 1. The closer to 1, the stronger the relation between variables, and thus can we understand the relation among the data before the experiment. We use variance gain ratio as the metric of performance of the model

\[ \text{Feature importance} = \frac{\sum_{j=1}^{n} \text{Gain}(x^j)}{\sum_{j=1}^{n} \sum_{i=1}^{m} \text{Gain}(x^i)} \]

where

\[ \text{Gain}(x^j) = \frac{1}{n} \left( \frac{\left( \sum_{i:x^j \in A_L} g_{mi} + \frac{1-a}{b} \sum_{i:x^j \in C_L} g_{mi} \right)^2}{\sum_{i:x^j \in A_L} + \sum_{i:x^j \in C_L}} \right) \]

\[ + \frac{\left( \sum_{i:x^j \in A_R} g_{mi} + \frac{1-a}{b} \sum_{i:x^j \in C_R} g_{mi} \right)^2}{\sum_{i:x^j \in A_R} + \sum_{i:x^j \in C_R}} \]

\[ A_L = \{ x^j \in A: x^j \leq d \}, \quad A_R = \{ x^j \in A: x^j > d \} \]

\[ C_L = \{ x^j \in C: x^j \leq d \}, \quad C_R = \{ x^j \in C: x^j > d \}. \]

\( x^j \) represents the \( j \)th variable of the \( i \)th sample, \( \sum_{j} \text{Gain}(x^j) \) represents the sum of variance gain generated by the \( j \)th variable \( x^j \) at all nodes (Ke et al. 2017). The larger the variance gain of the variable, the more important it is to estimate the stellar atmospheric parameters.

The experiments are carried out in a PyCharm 2019.3.1 environment running on a personal computer with an Intel core i7 2.60 GHz CPU and 16 GB of memory.

5.1. OPD + LightGBM

We first use the original magnitude data of six bands as input of LightGBM to derive stellar atmospheric parameters. The results on the test set are RMSE(\( T_{\text{eff}} \)) = 106 K, RMSE (log g) = 0.51 dex and RMSE([Fe/H]) = 0.25 dex, which are shown in Figure 3. The experimental results show that LightGBM cannot estimate stellar atmospheric parameters well. The main reasons are as follows:

1. The MIC values between variables and stellar atmospheric parameters are low. As shown in subgraph A of Figure 4, the MIC value of \( u \) band and \( T_{\text{eff}} \) is only 0.25. The MIC values between other stellar atmospheric parameters and bands are lower than 0.21. Hence the model cannot build a good functional relationship between the bands and the stellar parameters. In this sense, LightGBM cannot estimate stellar atmospheric parameters well.

2. The strong MIC between variables makes the model unstable, that is, the important features identified by the model do not have high MIC values. For example, as shown in subgraph A of Figure 4, the MIC between variables is between 0.37 and 0.95, which is very strong. The feature importance of \( T_{\text{eff}} \) and log g are more related to the \( u \) band and \( z \) band than other bands in Figure 5. However, the MIC between \( T_{\text{eff}} \) and the \( v \) band is 0.21, which is greater than that with \( z \) band (0.1). This makes it difficult to estimate \( T_{\text{eff}} \) accurately. And the same is true for log g.

Therefore, we believe that the photometric data cannot be directly input to estimate the stellar atmospheric parameters. It is necessary to transform the variables appropriately to reduce the MIC of each band, and increase the MIC between each band and the stellar atmospheric parameters to make the model more stable.
The CID is the difference of magnitude of any two wave bands of the same star. There is a strong relationship between the CID and stellar parameters, such as $T_{\text{eff}}$ and $V-K$. Therefore, we also input CID to estimate the stellar atmospheric parameters.

The CID we used in the experiment contains 15 variables ($C_6^2$):

\[ u-v, u-r, u-i, u-z, u-g, v-i, v-r, v-z, 
  v-g, g-i, g-r, g-z, r-z, r-i, i-z. \]

The result are shown in Figures 6 and 7. We find the MIC between the new variables and the parameters ($T_{\text{eff}}, \log g$) in subgraph B of Figure 4 is improved, which in turn reduces the RMSE of $T_{\text{eff}}$ and $\log g$ in the test set by 9% and 12% (compared with the OPD method in Section 5.1), respectively. However, this method do not reduce the MIC among the new variables in subgraph B of Figure 4, with the result that LightGBM is unable to fully recognize important features. For example, as shown in subgraph B of Figure 4, $u-g$ and $T_{\text{eff}}$ have high MIC (0.65), which means that it plays an important role.

5.2. CID + LightGBM

The CID is the difference of magnitude of any two wave bands of the same star. There is a strong relationship between the CID and stellar parameters, such as $T_{\text{eff}}$ and $V-K$. Therefore, we also input CID to estimate the stellar atmospheric parameters.

The CID we used in the experiment contains 15 variables ($C_6^2$):

\[ u-v, u-r, u-i, u-z, u-g, v-i, v-r, v-z, 
  v-g, g-i, g-r, g-z, r-z, r-i, i-z. \]
role in $T_{\text{eff}}$ prediction. However, as shown in Figure 7, the model considers $u-g$ is an unimportant feature of $T_{\text{eff}}$, and its importance to $T_{\text{eff}}$ accounts for only 0.2%. This is the consequence of the model instability caused by the high MIC of variable $u-g$ and other variables; that is, the model does not fully establish the functional mapping from variables to $T_{\text{eff}}$. This may affect the prediction ability of the model. The same is true for log $g$ and $[\text{Fe}/\text{H}]$.

Although the CID method enhances the MIC between new variables and parameters, it does not reduce the strong MIC among new variables. Therefore, the model cannot fully recognize the important features of the data itself, so it cannot be directly applied to other data sets.

5.3. PCA + LightGBM

To investigate whether PCA + LightGBM can improve the performance of estimating stellar atmospheric parameters, we first apply PCA to process the data such that the correlation between the new variables (PCs) is 0 (see Equation (4)). Then, we applied LightGBM to the PCs, and the experimental results are shown in Figure 8. Compared with the CID method in Section 5.2, the RMSE decreased by 6%, 11%, and 13%, respectively. Furthermore, the model can accurately identify the important features of the data.

As shown in subgraph C of Figure 4, the main reasons are that (1) the MIC between the new variables PCs is less than 0.26 (the correlation between the new variables PCs is 0), and
the MIC between new variable $F_2$ and parameters ($T_{\text{eff}}$, log $g$) is enhanced. Therefore, LightGBM can identify the important variables of the parameters ($T_{\text{eff}}$, log $g$) and improve the model’s generalization performance. For example, the maximal MIC (as shown in subgraph C of Figure 4) of $T_{\text{eff}}$ (or log $g$) is $F_2$, and LightGBM shows the important features (as shown in Figure 9) of $T_{\text{eff}}$ (or log $g$) are also $F_2$. The MIC values of $T_{\text{eff}}$ and other PCs ($F_1$, $F_3$, $F_5$, $F_4$ and $F_6$) are all lower than 0.14. Consistent with the MIC value, LightGBM model shows that the total importance of these PCs ($F_1$, $F_3$, $F_5$, $F_4$, and $F_6$) to $T_{\text{eff}}$ is only 7.6%. In addition, there is no significant difference in the proportion of importance between PCs, which are 3.1%, 2.3%, 0.9%, 0.8%, and 0.5%, respectively, indicating that LightGBM reasonably establish the functional mapping from PCs to $T_{\text{eff}}$. The same is true for log $g$.

For the estimation of [Fe/H], neither CID nor PCs improve the MIC of variables and [Fe/H], so the model do not significantly distinguish the feature importance of [Fe/H]. However, PCs reduce the MIC between variables and thus reduce the predicted RMSE of [Fe/H].

PCA makes the new variables more independent and enhances the MIC between the new variables and stellar parameters, and thus improves the final model prediction accuracy. PCA itself can be considered as a generalization of the CID method, and the CID method only considers the difference of two variables, so does not comprehensively consider the weighted sum of each variable. Therefore, the new variable obtained by PCA takes more data information into account, and the final model is more stable. The specific combination formula of each new variable ($F_i$, $i = 1, 2, 3, 4, 5,$
We take Equation 6 is shown in Equation (11)

\[
\begin{align*}
F_1 &= -0.36u - 0.38v - 0.41r - 0.45i - 0.43z - 0.39g \\
F_2 &= 0.62u + 0.51v - 0.19r - 0.35i - 0.44z + 0.04g \\
F_3 &= -0.15u + 0.54v - 0.32r + 0.12i + 0.4z - 0.64g \\
F_4 &= -0.68u + 0.55v + 0.18r - 0.11i - 0.27z + 0.34g \\
F_5 &= -0.05u - 0.01v - 0.63r - 0.31i + 0.47z + 0.54g \\
F_6 &= 0.03u - 0.0002v + 0.52r - 0.74i + 0.39z - 0.16g.
\end{align*}
\]

(11)

We take Equation (11) as a generalization of the CID. Here we consider not only the difference of magnitude of any two wave bands, but also the weighted sum of the data for each band. In this way, problems caused by the CID method are solved, and the performance of the model is improved.

5.4. Comparison of Experimental Results of Different Algorithms

To further assess the performance of LightGBM, we perform similar experiments with LR, SVR, Random Forest, GBDT, XGBoost, and ANN on PCs. These algorithms have their own advantages. For example, LR has strong interpretability, SVR has powerful kernel functions, GBDT, Random Forest, and XGBoost are the classic ensemble learning algorithm based on a decision tree, and ANN is a widely used machine learning algorithm.

The algorithms used in our test were created by the Python library. LR, SVR, GBDT, and random forest were built with the sklearn library, XGBoost and LightGBM were respectively created XGBoost library and LightGBM library. Bayesian optimization is used to select hyperparameters to make the model achieve the best result, and it was built with Bayesian-optimization library (Nogueira 2014). Bayesian optimization uses Gaussian process to optimize objective functions which are very costly or slow to evaluate. And it selects the best parameters in continuous space and can be effective in practice even if the underlying optimized function is stochastic, nonconvex, or even noncontinuous. Therefore, using Bayesian optimization to select model hyperparameters is more efficient than grid search and random search (Dewancker et al. 2015).

- Table 1: The Model Configurations of SVR and LR using Bayesian Optimization

| Model Configurations | SVR | LR |
|----------------------|-----|----|
|                      | Set$^a$ | Best$^b$ | Set$^a$ | Best$^b$ |
| Kernel               | (rbf, poly, linear) | [rbf, rbf] | × | × |
| Degree               | (1, 3) | [1, 1, 1] | (1, 8) | [3, 3, 4] |
| $C$                  | (0, 5) | [5, 4.9, 5] | × | × |

Notes.

$^a$ Set indicates the value range of the hyperparameters, × means that the model does not have this hyperparameter.

$^b$ Best represents the best hyperparameters of the model for $T_{\text{eff}}$, log $g$, and [Fe/H], respectively, × means that the model does not have this hyperparameter.

The experimental results are listed in Table 4. We find that the training speed of LightGBM is more than three times faster than that of XGBoost and more than ten times faster than that of GBDT, and the RMSE of LightGBM is almost lower than that of other algorithms. The high efficiency of the LightGBM algorithm provides a new reason for using photometric data to estimate stellar atmospheric parameters. For future spectral data, we will use the EFB algorithm with LightGBM for feature binding, which will further improve the efficiency of the LightGBM algorithm.

5.5. Optimize the LightGBM

In order to further study the LightGBM algorithm, we quantitatively analyze the impact of hyperparameters on estimating $T_{\text{eff}}$ from the following two aspects, and the impact on other parameters is similar.

1. We explore the impact of $a$ and $b$ on the results, and limit the ranges of $a$ and $b$ as 0.001 to 0.5 (other hyperparameters take the best value in Table 2). The experimental results are shown in Figure 10. We find that when $a = 0.05$ and $b = 0.45$, the RMSE of $T_{\text{eff}}$ reaches the minimum. This shows that in order to improve the generalization ability of the model, we need to pay...
attention to the samples with large residuals in the first 5%, rather than treat all samples equally. The setting of $a$ and $b$ is an important reason why the generalization ability of LightGBM is better than that of other algorithms.

2. We explore the impact of the number of bins (max_bin) on the results, and limit the number of bins varying from 250 to 10,000 in steps of 513 (20 steps); that is, the range of max_bin is 250 to 10,000 (other hyperparameters take the best value in Table 2). The experimental results are shown in Figure 11. We find that the training time of the model increases linearly with the increase of the number of bins. The good setting of the number of bins is the main reason why LightGBM algorithm is faster than other algorithms.

The advantages of LightGBM are mainly reflected in the above two aspects. The use of GOSS makes the model focus on the samples with high residuals, which improve its generalization ability. The setting of the number of bins reduces the number of splits. The good setting of the number of bins is the main reason why LightGBM algorithm is faster than other algorithms.
Figure 10. The influence of GOSS on $T_{\text{eff}}$ prediction RMSE. Axis $a$ and axis $b$ represent the value range of $a$ and $b$ of GOSS algorithm, and the vertical axis represents the RMSE of $T_{\text{eff}}$ in the test set. The red dot in the figure indicates the position when the RMSE is the lowest, and different colors indicate the change of RMSE.

Figure 11. The influence of max_bin on $T_{\text{eff}}$ training time. The horizontal axis represents the number of bins. The vertical axis represents the time consuming of LightGBM training.
number of the selection of the splitting node $d$ and speeds up model training. This advantage of LightGBM provides a selection for regression analysis of a large number of data in the future.

6. Conclusions

In this study, we propose a new method to determine the stellar parameters with color data. We first analyze why the previous methods cannot be applied to our data by introducing the MIC. We use OPD + LightGBM and CID + LightGBM to analyze why MIC between the input variables affect the accuracy of stellar atmospheric parameters. To overcome these problems, we use PCA to transform the data and then apply LightGBM to estimate the stellar atmospheric parameters. The main conclusions are as follows:

1. Estimating stellar atmospheric parameters directly by the OPD or CID is difficult, because the strong MIC between variables leads to the instability of the model and inability to correctly identify the important features of stellar atmospheric parameters.

2. PCA can eliminate the MIC between new variables through weighted sum of variables and enhance the MIC between new variables and stellar atmospheric parameters, making the final model more stable and the estimation more accurate.

3. The LightGBM algorithm is more efficient than other algorithms in terms of RMSE and time consuming. By setting $a$, $b$, and max_bin and cooperating with other hyperparameters, we use a small amount of data to train the model and accurately estimate stellar atmospheric parameters. Therefore, the LightGBM algorithm can be used to predict the stellar atmospheric parameters of the CSST.

4. PCA + LightGBM can be used to determine the sensitive variables of stellar atmospheric parameters. The sensitive variable of $T_{\text{eff}}$, $\log g$ is $F_2$. Other variables have little effect on the stellar atmospheric parameters. However, the lower MIC between PCs is helpful to the prediction of stellar atmospheric parameters (including $\text{[Fe/H]}$).

We thank the anonymous referee for useful comments that helped us improve the manuscript substantially. We would also like to thank Yuqin Chen for providing us with useful comments. This work is supported by the National Natural Science Foundation of China (NSFC) under grant No. 11873037, the science research grants from the China Manned Space Project with No. CMS-CSST-2021-B05 and CMS-CSST-2021-A08, and partially supported by the Young Scholars Program of Shandong University, Weihai (2016WHWLJH09). J.P. is supported by the National Natural Science Foundation of China under grant No. U19311209. K.T. is supported by National Natural Science Foundation of China under grant No. 11673030, 12090044. X.K. is supported by National Natural Science Foundation of China under grant No. 11803016.

ORCID iDs

Yude Bu https://orcid.org/0000-0002-9474-4734
Kefeng Tan https://orcid.org/0000-0003-0173-6397

References

Abdi, H., & Williams, L. J. 2010, Computational Statistics, Wiley Interdisciplinary Reviews Vol 2 (New York: Wiley), 433
Alonso, A., Arribas, S., & Martínez-Roger, C. 1996, A&AS, 117, 227
Alonso, A., Arribas, S., & Martínez-Roger, C. 1999, A&AS, 140, 261
Blanton, M. R., Bershady, M. A., Abolfathi, B., et al. 2017, AJ, 154, 28
Bu, Y., Chen, F., & Pan, J. 2014, NewA, 28, 35
Casagrande, L., Ramírez, I., Meléndez, J., Bessell, M., & Asplund, M. 2010, A&A, 512, A54
Chambers, K. C. & Pan-STARRS Team 2018, AAS Meeting, 231, 102.01
Crawford, D. L. 1958, ApJ, 128, 185
Dewancker, I., McCourt, M., & Clark, S. 2015, arXiv:1612.04858
Fan, Z., Zhao, G., Wang, W., et al. 2018, PrA, 36, 101, https://kns.cnki.net/kcms/detail/detail.aspx?FileName=TWJZ201802002&DbName=CTJQ2018
Friedman, J. H. 2001, AnSta, 29, 1189
Holmberg, J., Nordström, B., & Andersen, J. 2007, A&A, 475, 519
Huang, Y., Liu, X. W., Yuan, H. B., et al. 2015, MNRAS, 454, 2863
Jolliffe, I. T., & Cadima, J. 2016, RSPTA, 374, 20150202
Jónsson, H., Holtzman, J. A., Allende Prieto, C. A., et al. 2020, AJ, 160, 120
Jurić, M., Ivezic, Z., Brooks, A., et al. 2008, ApJ, 673, 864
Ke, G., Meng, Q., Finley, T., et al. 2017, in Advances in Neural Information Processing Systems, ed. I. Guyon et al., Vol. 30 (Red Hook, NY: Curran Associates, Inc.), 3146, https://proceedings.neurips.cc/paper/2017/file/6449f44a102fde848669bdd9eb676fa-Paper.pdf
Magain, P. 1987, A&A, 181, 323
Majewski, S. R., Schiavon, R. P., Frinchaboy, P. M., et al. 2017, AJ, 154, 94
Nissen, P. E. 1981, A&A, 97, 145
Nogueira, F. 2014, Bayesian Optimization: Open Source Constrained Global Optimization Tool for Python, https://github.com/fmfn/BayesianOptimization
Olsen, E. H. 1984, A&AS, 57, 443
Reshef, D. N., Reshef, Y. A., Finucane, H. K., et al. 2011, Sci, 334, 1518
Schlafly, E. F., & Finkbeiner, D. P. 2011, ApJ, 737, 103
Schlegel, D. J., Finkbeiner, D. P., & Davis, M. 1998, ApJ, 500, 525
Singh, H. P., Gulati, R. K., & Gupta, R. 1998, MNRAS, 295, 312
Storrie-Lombardi, M. C., Irwin, M. J., von Hippel, T., & Storrie-Lombardi, L. J. 1994, VA, 38, 331
Strömgren, B. 1956, VA, 2, 1336
Tonry, J. L., Stubbs, C. W., Lykke, K. R., et al. 2012, ApJ, 750, 99
York, D. G., Adelman, J., Anderson Jr, J. E., et al. 2000, AJ, 120, 1579
Zasowski, G., Cohen, R. E., Choi, S. R., et al. 2017, AJ, 154, 198
Zasowski, G., Johnson, J. A., Frinchaboy, P. M., et al. 2013, AJ, 146, 81
Zheng, J., Zhao, G., Wang, W., et al. 2019, AR&T, 16, 93