SVM-Based Automatic Classification Algorithm for Eclipsing Binary Light Curves: A Postprint

Authors: Yuan Huiyu, Zhao Juan, Dai Haifeng, Yang Yuan-Gui

Date: 2018-09-10T00:00:00+00:00

Abstract

We propose a machine learning-based automatic classification algorithm for eclipsing binary light curves. The algorithm first preprocesses the data by normalizing the eclipsing binary light curve data and reducing noise through filtering/interpolation. It then extracts frequency signals as feature vectors using Fast Fourier Transform, and trains a Support Vector Machine with these vectors to obtain an automatic classification model. The algorithm is implemented in Python and validated using data scraped from CALEB and GCVS, analyzing the effects of feature vectors, SVM kernel functions, and penalty coefficients on classification accuracy. After optimization, the classification model achieves accuracies of 92.8% on the training set and 89.0% on the testing set. Finally, when applied to third-party data classification, the model achieves an accuracy of 88.8%, thereby demonstrating the effectiveness of the proposed algorithm.

Full Text

An Automatic Classification Algorithm for Light Curves of Eclipsing Binaries Based on SVM

Huiyu Yuan[1], **Juan Zhao**[2], **Haifeng Dai**[2], **Yuangui Yang**[2]* [1]Information College, Huaibei Normal University, Huaibei 235000, China
[2]Huaibei Normal University, Huaibei 235000, China

Abstract: This paper proposes an automatic classification algorithm for light curves of eclipsing binary stars based on machine learning. The algorithm first preprocesses the data by normalizing the light curves and reducing noise through filtering and interpolation. It then employs Fast Fourier Transform to extract frequency signals as feature vectors, which are used to train a Support Vector Machine and obtain an automatic classification model. Using data from CALEB and GCVS, we analyze the effects of feature vectors, SVM kernel functions, and

penalty coefficients on classification accuracy. Implemented in Python, the algorithm achieves a classification accuracy of 88.8% on third-party data, demonstrating the effectiveness of the proposed classification method. The optimized model reaches correct rates of 92.8% (training set) and 89.0% (test set).

Keywords: automatic classification of light curves; support vector machine; eclipsing binary stars

Introduction

Driven by emerging technologies such as information and computational science, astronomical research has transitioned from traditional single-target observation and manual data processing to multi-target observation and automated data processing. Large-scale sky surveys including ROTSE, ASAS, SuperWASP, MACHO, OGLE, SDSS, LAMOST, and Kepler have provided massive datasets for astronomical research. Computers now automatically perform tasks such as target cross-identification, observation, real-time data processing, and analysis to obtain spectral, photometric, periodic, and classification data. As data volumes continue to grow, single servers can no longer process data in real time, prompting the application of distributed computing to improve processing efficiency. Faced with these massive astronomical datasets, artificial intelligence algorithms such as support vector machines, neural networks, and genetic algorithms have become essential for data analysis and information mining. Examples include classifying celestial objects in SDSS and XMM data using random forest methods, detecting radio pulse signals through machine learning, and classifying close binary systems based on Roche potentials. These developments mark the entry of astronomy into the era of big data.

Light curves obtained from observations of eclipsing binaries enable rapid type determination and identification of binary systems with special evolutionary significance, providing important windows for studying unique celestial objects and phenomena. This is crucial for enriching and advancing binary star research and for understanding the formation and evolution of star clusters and galaxies through eclipsing binaries. Previous work has used polynomial fitting of light curves to determine types based on the width and depth of primary and secondary minima. Other studies have employed Fourier transforms to extract frequency features from light curve data for classification. However, these approaches often used ideal light curve data from software calculations for parameter setting, utilized limited features, and did not account for data fluctuations caused by instrumental errors and weather conditions. Consequently, they could only perform preliminary classification and could not identify light curves containing special astronomical phenomena.

This paper proposes an automatic classification algorithm for eclipsing binary light curves based on Support Vector Machines, using frequency signals obtained from Fast Fourier Transform as features to train the SVM model.

1. Automatic Classification Algorithm

Eclipsing binary light curves can be classified into three types: EA, EB, and EW. The proposed classification method is illustrated in Figure 1 [Figure 1: see original paper]. The first step preprocesses the raw data through normalization and noise reduction. The second step extracts frequency signals via Fast Fourier Transform to serve as feature data. The third step trains a classification model using the SVM algorithm. Finally, the process is optimized to obtain the best classification model.

1.1 Data Preprocessing

While ASAS uses theoretical light curves without noise influence, this study employs real observational data from CALEB[2] (Catalog and AtLas of Eclipsing Binaries), including phase and differential magnitude values. Due to weather factors and instrumental errors, real data inevitably contains noise. To reduce noise effects, preprocessing is performed as follows:

First, normalization is applied. Phase values already fall within [0,1] and require no processing. Differential magnitudes can be normalized to [0,1] using the formula where m' is the normalized differential magnitude, m is the original value, and m_{max} and m_{min} are the maximum and minimum differential magnitudes, respectively.

Second, mean filtering and linear interpolation algorithms reduce noise. Let m_i represent the final preprocessed differential magnitude value. The phase is uniformly divided into n segments. If segment k contains exactly one data point (b=1) within its phase range, that value is used as m_i. If b>1, mean filtering is applied to obtain a new m_i value. If b=0, linear interpolation is used to generate a new m_i value. This produces equally-spaced, normalized data.

1.2 Light Curve Feature Extraction

Original light curves are time-series data that require feature extraction for analysis. Common features include differences between primary and secondary minima and the full width at half maximum of primary minima. This study adopts frequency characteristics as feature values. In practice, fast discrete Fourier transform converts phase/differential magnitude data into frequency domain signals. The frequency signals and corresponding light curve types form the feature dataset {f, T}, where f represents frequency components and T represents the light curve type.

1.3 Support Vector Machine Classification Algorithm

Support Vector Machine is a supervised machine learning algorithm based on VC dimension theory and structural risk minimization principles from statistical learning theory. The fundamental concept involves mapping feature values into a high-dimensional vector space to obtain a hyperplane that separates different

classes. This algorithm is commonly used for automatic classification. In practice, data is divided into training and test sets. The training set is used to train the SVM model to obtain the mapping function and separating hyperplane (i.e., the classification model), while the test set validates the resulting model.

2. Experiments and Results Analysis

The algorithm was implemented in Python, an object-oriented interpreted programming language that has become one of the most popular languages due to its ease of use, simplicity, and extensibility. Python offers numerous scientific computing libraries that were utilized in this implementation.

2.1 Classification Experiment Implementation

First, raw data was downloaded and collected. Using urllib3 and BeautifulSoup libraries, we automatically analyzed CALEB webpage data and stored coordinates, star names, types, and 747 light curves for 300 variable stars. However, the website does not provide light curve types, which were obtained through cross-comparison with GCVS[3] (General Catalogue of Variable Stars new version) data using the variable star coordinates.

Next, light curve data preprocessing was implemented. Using V-band data for three variable stars—BE Vul (EA), YY Cet (EB), and TW Cet (EW)—as examples, the original data is shown in Figure 2 Figure 2: see original paper. Due to observational equipment limitations, the data quality is poor, characterized by inconsistent numbers of data points, large fluctuations, and discrete values. After dividing the phase into new points with 0.005 intervals and applying normalization, mean filtering, and linear interpolation, the resulting data is shown in Figure 2(b). The preprocessing preserves the original data trends while producing smoother results.

The third step uses numpy and scipy libraries to perform Fast Fourier Transform on the preprocessed data for frequency domain transformation. Using the three stars mentioned above as examples, the resulting frequency values are shown in Figure 3 [Figure 3: see original paper], where the horizontal axis represents signal harmonic frequencies.

The fourth step involves SVM model training. After processing 747 light curves using the above method, we obtained dataset {f, T}, where f represents the set of independent frequency components and T represents the light curve type. The SVM model used a linear kernel function. We first tested the impact of frequency component selection on model training, using [f_i, f_j] to represent continuous frequency component sets from f_i to f_j. The training set contained 373 data points, the test set contained 374 data points, and the penalty factor was set to 1.0. The kernel function is the algorithm that maps input space to high-dimensional space, while the penalty factor represents tolerance for misclassification. Lower tolerance yields better training results but may cause overfitting.

The final results are shown in Figure 4 [Figure 4: see original paper]. Using even harmonic components as feature values yields high classification accuracy (data points a, b, and c). Even using only f0 achieves 78.6% accuracy (point a). Using odd harmonic components as feature values achieves a maximum accuracy of only 57.8% (points d, e), indicating that odd harmonics are unsuitable as features. Comparing results f through i, accuracy increases with the number of frequency components selected, demonstrating that more frequency components help optimize classification. The difference between training and test set accuracies is less than 2%, proving effective training without overfitting. Overall, even harmonic components are suitable as feature values.

2.2 Support Vector Machine Optimization

Next, SVM parameter settings were optimized for better results, focusing on kernel function selection and penalty factor configuration. Using dataset {f0, f2, f4, f6, f8} as feature values with different kernels and penalty factors, the results are shown in Figure 5 [Figure 5: see original paper]. The four kernel functions rank in order of performance as: linear, rbf, sigmoid, and poly. Increasing the penalty factor initially significantly improves accuracy for linear, rbf, and sigmoid kernels, but accuracy stabilizes beyond a certain threshold. The penalty factor has no effect on poly kernels. Using a linear kernel with penalty factor set to 2.0 yields the optimal classification model with accuracies of 89.8% (training set) and 84.8% (test set). The trained model can be saved for classifying new light curve data.

2.3 Experimental Results Analysis and Data Correction

While the trained model shows high accuracy and meets classification requirements, some misclassifications remain. We analyzed these errors to identify their causes.

Misclassifications originate from two main sources. First, inconsistent light curve and classification information between the two websites. For example, original and preprocessed data for AU Pup and AW Lac are shown in Figure 6 Figure 6: see original paper. The light curves should be EW type, but GCVS classifies them as EB type. This error can be eliminated by correcting the original light curve type data.

Second, the lack of clear classification standards for light curve types. As shown in Figure 6(b), GCVS classifies XZ Cmi and SW Lyn as EB and EA types respectively, but their CALEB light curve data are very similar. Clear classification standards must be established, and original data must be manually reviewed and verified. Due to the substantial workload, this has not yet been completed.

After correcting 14 targets with classification errors in the original data, SVM model training and testing were repeated, with results shown in Figure 7 [Figure 7: see original paper]. Since sigmoid and poly kernels performed poorly in previous training, only linear and rbf kernels were tested. The results show the

linear kernel performs better, achieving 92.8% accuracy (training set) and 89.0% (test set) when the penalty factor is set to 5.8. Using the rbf kernel with penalty factor 5.6 achieves 90.9% (training set) and 86.4% (test set).

Subsequently, 160 light curve data points were prepared and tested using both trained models, yielding 88.8% accuracy for both. Error analysis revealed that misclassifications primarily occur between EA and EB type light curves.

3. Summary and Outlook

This paper proposes an automatic classification algorithm for light curves based on machine learning, using Fast Fourier Transform to extract frequency signals from target data. Even-order frequency components are selected as light curve features to train a Support Vector Machine model. The algorithm was implemented and optimized in Python using real observational data from CALEB and classification data from GCVS. Results show that using [f0, f2, f4, f6, f8] as feature values with a linear kernel and penalty factor of 2.0 yields optimal classification results of 89.8% (training set) and 84.8% (test set), basically meeting classification requirements.

Analysis of misclassified data reveals two error sources. First, inconsistencies between CALEB light curve data and GCVS classification information, which can be eliminated by correcting classification data. Second, the lack of clear classification standards, where very similar light curves are assigned different types, interfering with test results. Establishing clear standards and reclassifying original data would prevent this error. After correcting the first type of error, accuracy improved to 92.8% (training set) and 89.0% (test set). The second error source remains unaddressed due to the lack of established classification standards.

Automation technology is increasingly applied in astronomical observations, generating ever-larger datasets. Routine observational data often contains special data indicating unusual astronomical phenomena such as binary mergers. Identifying and prioritizing observation of these special targets yields more valuable results. The challenge lies in rapidly screening special data from massive datasets. Future research will compile special light curve data as sample data to train SVM algorithms, enabling rapid identification of anomalous light curves and quick response to transient phenomena.

References

[1] Cui Chenzhou, Yu Ce, Xiao Jian, et al. Astronomy research in big-data era. Chin Sci Bull, 2015, 60(Z1):445-449
[2] Zhang Hailong, Nie Jun, Zhao Qing, et al. Xinjiang Astronomical Observatory Data Center Custom Uploading Crossmatcher. Astronomical Research & Technology, 2017, 14(03):347-
[3] Lewis H, Raffi G. Advanced Software, Control, and Communication Systems

for Astronomy. Proc Spie, 2004,5496(1):65-68.
[4] Wei Shoulin, Liu Pengxiang, Wang Feng. Real-Time Data Processing in Mingantu Ultrawide Spectral Radio Heliograph Based on Spark Streaming. Astronomical Research & Technology, 2017, 14(03):421-428.
[5] Chen Shuxin, Luo Ali, Sun Weiming. Application of R language in LAMOST Spectral Analysis. Astronomical Research & Technology, 2017, 14(03):363-368.
[6] Zhang Y X, Zhou X L, Zhao Y H, et al. Statistical Study of 2XMMi-DR3/SDSS-DR8 Cross-correlation Sample. Astronomical Journal, 2013,145(2):531-544.
[7] Devine T R, Goseva-Popstojanova K, McLaughlin M. Detection of dispersed radio pulses: a machine learning approach to candidate identification and classification. Monthly Notices of the Royal Astronomical Society, 2016,459(2):w655.
[8] Yang Yuangui, Lai Chunfu. Calculating the Roche Potential of Close Binaries. Journal of Huaibei Normal University (Natural Science). 2011,32(01):29-32.
[9] Kirk B, Conroy K, Prša A, et al. Kepler Eclipsing Binary Stars. VII. The Catalog of Eclipsing Binaries Found in the Entire Kepler Data-Set. Astronomical Journal, 2016,151(3):68.
[10] Pojmański G. The All Sky Automated Survey. Astronomische Nachrichten, 1997,325(6-8):467-481.
[11] Akerlof C, Amrose S, Balsano R, et al. ROTSE All-Sky Surveys for Variable Stars. I. Test Fields. Astronomical Journal, 2000,119(4):1901.
[12] Pojmanski G. The All Sky Automated Survey. Variable Stars in the 0h - 6h Quarter of the Southern Hemisphere. Physics, 2002.
[13] https://www.researchgate.net/profile/Y-G_{Yang}.

Note: Figure translations are in progress. See original paper for figures.