Mining Quasar Candidates from Large Sky Surveys

Zhang, Yanxia
Key Laboratory of Optical Astronomy
National Astronomical Observatories
Chinese Academy of Sciences
Datun Road 20A, Chaoyang District
Beijing 100012, China
E-mail: zyx@bao.ac.cn

Luo, Ali
Key Laboratory of Optical Astronomy
National Astronomical Observatories
Chinese Academy of Sciences
Datun Road 20A, Chaoyang District
Beijing 100012, China
E-mail: lal@bao.ac.cn

Zhao, Yongheng
Key Laboratory of Optical Astronomy
National Astronomical Observatories
Chinese Academy of Sciences
Datun Road 20A, Chaoyang District
Beijing 100012, China
E-mail: yzhao@lamost.org

With the development and operation of various large sky survey projects, how to improve and optimize the efficiency and scientific output of telescopes is a hot issue. Thus the careful preparation of survey programs and input catalogs are of great value. The development of robust data mining techniques for ground-based instruments (such as Chinese LAMOST telescope) is a key element in preselecting quasar candidates from other photometric surveys. Taking quantity and complexity of astronomical data into account, a large number of mining approaches are applied and compared to create the quasar targeting catalog. Each method has its merits and demerits.

INTRODUCTION

Quasars are among the most luminous, powerful, and energetic objects known in the Universe and show a very high redshift. High redshift quasars are taken as the powerful probe of structure formation in the very early Universe and important for understanding the formation and evolution of galaxies and supermassive black holes (SMBHs) in the early Universe. The most distant quasars place constraints on the reionization epoch (Fan et al. 2006). Quasars are also tracers of structure at large scales and small scales (Kirkpatrick et al. 2011 and references therein). The number of quasars has continually increased due to large sky surveys (e.g. 2DF, SDSS). The large number of quasars are helpful to the study of the luminosity function of quasars (Boyle et al. 2000), as well as that of baryon acoustic oscillations in the distribution of Lyα absorption (Ross et al. 2011).

Quasars can be detected over the entire observable electromagnetic spectrum including radio, infrared, optical, ultraviolet, X-ray and even gamma rays. In addition, they have variation in luminosity on a variety of time scales. Only from morphology, quasars looked like single points of light (i.e., point sources), indistinguishable from stars, except for their peculiar spectra. Based on these characteris-

tics of quasars, various methods for preselecting quasar candidates have been applied, including their nonstellar colors in ugriz broadband photometry (Richards et al. 2002), KX method (Warren et al. 2000), support vector machines and learning vector quantization (Zhang & Zhao 2003), Kernel Density Estimation (Richards et al. 2004, 2009), UV-excess (Smith et al. 2005), Support Vector Machines and KDTree (Gao et al. 2008), color space cutoff (Wu & Jia 2010), Artificial Neural Networks (Yeche et al. 2010), Probabilistic Principal Surfaces and Negative Entropy Clustering (D'Abrusco et al. 2009), Difference Boosting Neural Network (Abraham et al. 2010), likelihood estimator (Kirkpatrick et al. 2011), combination of quite a few methods (Ross et al. 2011).

With the exponential growth of astronomical data, astronomy changes from data-driven science to data-intensive science and further steps into astroinformatics era. Data mining and machine learning in astronomy are hot issue (see the review of Ball & Brunner 2010). There have been many successful applications of data mining in astronomy. In this paper, we experiment a number of classification approaches in WEKA used for discriminating quasars from stars, based on large sky survey databases SDSS and UKIDSS.

The Sample

The Sloan Digital Sky Survey (SDSS) is one of the most ambitious and influential surveys in the history of astronomy (York et al. 2000). The SDSS used a dedicated 2.5-meter telescope at Apache Point Observatory, New Mexico, equipped with two powerful special-purpose instruments. SDSS data have been released to the scientific community and the general public in annual increments, with the final public Data Release 7 from SDSS-II occurring in October 2008. Meanwhile, SDSS is continuing with the Third Sloan Digital Sky Survey (SDSS-III), a program of four new surveys using SDSS facilities. SDSS-III began observations in July 2008 and released its first public data as Data Release 8 to emphasize its continuity with previous SDSS releases. SDSS-III will continue operating and releasing data through 2014. The photometric imaging total area in the SDSS DR7 covers 11663 deg^2 mainly in the Northern hemisphere in five specifically designed bands $ugriz$, but the spectroscopic total area just covers 9380 deg^2. The catalog of DR7 derived from the images includes more than 350 million celestial objects, and spectra of 930,000 galaxies, 120,000 quasars, and 460,000 stars.

The UKIRT Infrared Deep Sky Survey (UKIDSS) is the near-infrared sky survey which began in May 2005 (Lawrence et al. 2007). UKIDSS is being carried out using the UKIRT Wide Field Camera (WFCAM), which is the largest IR astronomical instrument to date. UKIDSS will be the true near-infrared counterpart to the Sloan survey, and will produce as well a panoramic clear atlas of the Galactic plane. It will survey 7500 square degrees of the Northern sky, extending over both high and low Galactic latitudes with five survey components covering various combinations of the filter set $ZYJHK$ and H_2. The limiting magnitude of UKIDSS is three magnitudes deeper and twelve times larger in effective volume than the 2MASS survey.

In this study, we use the public Data Release 7 and query the SDSS spectroscopic database to obtain point sources of all spectral types. The point sources meeting $zconf \geq 0.95$ and $16 < psfMag\_i < 22$ are picked out and these sources include 434280 stars and 112425 quasars. In order to keep the sample balanced, 112425 stars are randomly selected from the whole star sample. Culling the records with missing values, the final sample consist of 112289 stars and 112004 quasars. This sample is regarded as SDSS sample. All magnitudes are dereddened according to Schlegel et al. (1998). The adopted parameters are i', u'-g',g'-r',r'-i',i'-z'.

Another sample is the cross-match result of SDSS DR7 and UKIDSS DR3 by finding the nearest counterparts within 3 arcsec radius. The detail selecting criterion is described in Section 2 of Wu and Jia (2010) including the equations of conversion between the SDSS AB magnitudes and UKIDSS Vega magnitudes. Getting rid of the records with missing values, the SDSS_UKIDSS sample include 8996

stars and 8496 quasars. As for this sample, Vega magnitudes are adopted. The input pattern is i,u-g,g-r,r-i,i-z,z-Y,Y-J,J-H,H-K.

ALGORITHMS

Based on the above samples, we have tried a range of classification algorithms to seperate quasars from stars. Classification belongs to supervised learning which forms a model based on training data and uses this model to classify new data. We make use of the following algorithms:

- Naive Bayes
- Bayes Network
- Logistic Regression
- RBF network
- SMO
- LibSVM
- Voted Perceptron
- Bagging
- LogitBoost
- Decision Table
- ADTree
- Decision Stump
- NBTree
- Random forest
- IB1

These algorithms are all provided and integrated in Weka. The Waikato Environment for Knowledge Analysis, or Weka for short, is a collection of machine learning algorithms for data mining tasks (Hall et al. 2009). The algorithms can either be applied directly to a dataset or called from your own Java code. Weka contains tools for data pre-processing, classification, regression, clustering, association rules, and visualization. It is also well-suited for developing new machine learning schemes. it is available as Java source code at http://www.cs.waikato.ac.nz/ml/weka/. Weka is open source software issued under the GNU General Public License. About the principles and application of these algorithms refer to the book named "Data Mining: Practical Machine Learning Tools and Techniques" and written by Ian H. Witten, Eibe Frank and Mark A. Hall.

Based on Bayes, Naive Bayes and Bayes Network are applied. A Naive Bayes classifier is a simple probabilistic classifier based on applying Bayes' theorem (from Bayesian statistics) with strong (naive) independence assumptions that the presence (or absence) of a particular feature of a class is unrelated to the presence (or absence) of any other feature. Bayes Network is a probabilistic graphical model that represents a set of random variables and their conditional dependencies via a directed acyclic graph (DAG).

About function, Logistic Regression, radial basis function (RBF) network, SMO, LibSVM and Voted Perceptron are tried. Logistic Regression is used for prediction of the probability of occurrence of an event by fitting data to a logit function logistic curve. A radial basis function network is an artificial neural network that uses radial basis functions as activation functions. It is a linear combination of radial basis functions. The Sequential Minimal Optimization (SMO) algorithm was developed as a faster, more scalable Support Vector Machine (SVM). These improvements are related to increasing the speed of training and as such classification is performed as with standard SVM. SVM constructs a hyperplane or set of hyperplanes in

a high- or infinite-dimensional space, which can be used for classification, regression, or other tasks. Intuitively, a good separation is achieved by the hyperplane that has the largest distance to the nearest training data points of any class (so-called functional margin), since in general the larger the margin the lower the generalization error of the classifier. The parameters of the maximum-margin hyperplane are derived by solving the optimization. There exist several specialized algorithms for quickly solving the QP problem that arises from SVMs, mostly reliant on heuristics for breaking the problem down into smaller, more-manageable chunks. A common method for solving the QP problem is Platt's Sequential Minimal Optimization (SMO) algorithm, which breaks the problem down into 2-dimensional sub-problems that may be solved analytically, eliminating the need for a numerical optimization algorithm. LIBSVM is short for a library for Support Vector Machines. It is an integrated software for support vector classification, (C-SVC, nu-SVC), regression (epsilon-SVR, nu-SVR) and distribution estimation (one-class SVM). In the voted perceptron algorithm, more information is stored during training and then this elaborate information is used to generate better predictions on the test data. The information maintained during training is the list of all prediction vectors that were generated after each and every mistake. For each such vector, the number of iterations is counted, and it "survives until the next mistake is made; this count is regarded as the "weight of the prediction vector. To calculate a prediction, the binary prediction of each one of the prediction vectors is computed and all these predictions are combined by a weighted majority vote. The weights used are the survival times described above. This makes intuitive sense as "good prediction vectors tend to survive for a long time and thus have larger weight in the majority vote.

Considering meta algorithms, AdaBoost, Bagging and LogitBoost are adopted. Meta algorithms such as boosting or bagging can improve accuracy by combining multiple weaker classifiers into one strong classifier. Boosting is a process used to increase the performance of weak learning algorithms. It can also be used on strong algorithms, but improvements are less dramatic. Boosting works by combining the classifiers produced by the learning algorithm over a number of distributions of the training data. AdaBoost can be used in conjunction with many other learning algorithms to improve their performance. AdaBoost is adaptive in the sense that subsequent classifiers built are tweaked in favor of those instances misclassified by previous classifiers. Combining the decisions of different models means amalgamating the various outputs into a single prediction. The simplest way to do this in the case of classification is to take a vote (perhaps a weighted vote); in the case of numeric prediction it is to calculate the average (perhaps a weighted average). Bagging adopts this approach and the models receive equal weight. LogitBoost is a boosting algorithm. Specifically, if one considers AdaBoost as a generalized additive model and then applies the cost functional of logistic regression, one can derive the LogitBoost algorithm.

Given decision rule, Decision Table is selected. Like flowcharts and if-then-else and switch-case statements, Decision Table is a precise yet compact way to model complicated logic, associates conditions with actions to perform, but in many cases does so in a more elegant way.

In terms of tree method, ADTree, Decision Stump, NBTree and Random forest are implemented. An alternating decision tree (ADTree) generalizes decision trees and has connections to boosting. ADTree consists of decision nodes and prediction nodes. Decision nodes specify a predicate condition. Prediction nodes contain a single number. ADTrees always have prediction nodes as both root and leaves. An instance is classified by an ADTree by following all paths for which all decision nodes are true and summing any prediction nodes that are traversed. A decision stump is a one-level decision tree. That is, it is a decision tree with one internal node

(the root) which is immediately connected to the terminal nodes. A decision stump makes a prediction based on the value of just a single input feature. Sometimes they are also called 1-rules. NBTree is a hybrid of a decision tree classifier and a Naive Bayes classifier. Designed to allow accuracy to scale up with increasingly large training datasets. The NBTree model is best described as a decision tree of nodes and branches with Bayes classifiers on the leaf nodes. Random forest is an ensemble classifier that consists of many decision trees and outputs the class that is the mode of the class's output by individual trees.

Talking about lazy learning, IB1 is used. Lazy learning methods defer processing of training data until a query needs to be answered. This usually involves storing the training data in memory, and finding relevant data in the database to answer a particular query. This type of learning is also referred to as memory-based learning. IB1 uses a simple distance measure to find the training instance closest to the given test instance, and predicts the same class as this training instance. If multiple instances are the same (smallest) distance to the test instance, the first one found is used.

With various classification algorithms, how to decide which one has better performance is important. It needs measurement to judge the performance of classifier. We use metrics such as Accuracy, True Negative Rate, True Positive Rate, Precision, Recall, and F-measure (FM) to evaluate the performance of classification algorithms. These metrics have been widely used for comparison of different classifiers. All these metrics are functions of the confusion matrix as shown in Table 1. A false positive (FP) is when the outcome is incorrectly predicted as yes (or positive) when it is actually no (negative). A false negative (FN) is when the outcome is incorrectly predicted as negative when it is actually positive. The true positive rate is TP divided by the total number of positives, which is TP + FN; the false positive rate is FP divided by the total number of negatives, which is FP + TN. The overall success rate (Accuracy) is the number of correct classifications divided by the total number of classifications. Recall is the fraction of actual positive cases that were correct, and Precision is the fraction of the predicted positive cases that were correctly identified. For any classifier, there is always a trade off between Recall and Precision. The F-measure can be interpreted as a weighted average of the precision and recall. These metrics are commonly used in the information retrieval area as performance measures. We will adopt all these measurements to compare our methods with different patterns. Ten-fold cross-validation is carried out to obtain all the performance metrics. The rows of the matrix are actual classes, and the columns are the predicted classes. Based on Table 1, the above-mentioned metrics are defined as follows:

$$Accuracy(Acc) = \frac{TP + TN}{TP + FP + TN + FN} \tag{1}$$

$$True\ Positive\ Rate(Acc^{+}) = \frac{TP}{TP + FN} = Recall \tag{2}$$

$$True\ Nagative\ Rate(Acc^{-}) = \frac{TN}{TN + FP} \tag{3}$$

$$Precision = \frac{TP}{TP + FP} \tag{4}$$

$$F - measure(FM) = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{5}$$

RESULTS AND DISCUSSION

Table 1: Confusion matrix.

	Predicted Positive Class	Predicted Negative Class
Actual Positive class	TP (True Positive)	FN (False Negative)
Actual Negative class	FP (False Positive)	TN (True Negative)

In order to assure the reliability of comparison result, we apply the same sample and input pattern for the above mentioned algorithms. As for SDSS sample, the detailed accuracy for each method is shown in Table 2; accuracy and training time to build a model are indicated in Table 3. From Tables 2-3, it is obvious that the algorithms except Naive Bayes, Logistic Regression and RBF network have good performance; LibSVM, NBTree, Bagging, Random Forest and IB1 have better performance. When considering the efficiency of a classifier, we also take the time to train and predict into account. Usually the time to built a model is much longer than that of prediction. Nevertheless, lazy learning algorithms are exception because these algorithms delay the computation until prediction. So IB1 spends little time to learning while it takes more than half a day to prediction. Local weighted learning and Kstar spend more than several days to build models for the SDSS sample. Due to slow speed, we don't list the performance of these two methods here. Of those algorithms with better performance, the time to build models for Bagging and Random Forest is about a minute. In terms of accuracy and speed, Bagging and Random Forest are better choice.

For simplicity, we only give accuracy and training time by different methods based on SDSS_UKIDSS sample in Table 4. Comparing Table 3 with Table 4, the classifier performance rank is nearly same. LibSVM, NBTree, Bagging, Random Forest and IB1 still have better performance. Considering both accuracy and speed, Bagging and Random Forest are better methods. For the same method, the performance in Table 3 is better than that in Table 4. When the training sample become smaller, the training speed accelerates sharply although the number of dimensionality is larger. For our case, accuracy doesn't improve when adding infrared information from UKIDSS. When data are from more bands, the size of sample become much smaller and thus this leads to incomplete sample. Therefore we can't simply consider much better performance of a classifier with data from more bands. Circumstances alter cases. So far astronomy has stepped into huge sample era and become data-intensive astronomy. Only when adding information from more bands and the variation in size of a sample is much smaller or same, the performance of a classifier possibly improves.

Many factors influence the performance of a classifier, such as data characteristics, data preprocessing, feature selection/extraction, model parameter optimization, model principle. The above algorithms adopt the default settings in WEKA. Perhaps the training models haven't arrive at the optimal. Model parameter modulation contributes to good performance (Gao et al. 2008). Although the rank of classifier performance is the same for these two samples, the performance rank based on different samples is usually different compared to the work of Zhao & Zhang 2008. During data preprocessing, we only remove the records with missing values. Considering feature selection/extraction, some methods will improve their performances, different methods along with appropriate feature selection/extraction methods may obtain good results (Zheng & Zhang 2008). Moverover different feature combination also affects the accuracy (D'Abrusco et al. 2009). With the volume of astronomical data becoming larger and larger, how to high-efficiently store, move, handle, mine and analysis so huge data is hot issue. For those algorithms with high accuracy and slow speed, parallel technology and GPU technology are good answers. For example, CUDA-based k-nearest neighbors and CUDA-based SVM are applied in classification of celestial objects (Pei et al. 2010; Peng et al. 2010). Meanwhile the model principle of each algorithm is the leading role of its performance. The advantage and

Table 2: Detailed Accuracy by Different Methods Based on SDSS Sample.

Method	Class	TP Rate	FP Rate	Precision	Recall	F-Measure
Naive Bayes	stars	0.726	0.091	0.889	0.726	0.799
	quasars	0.909	0.274	0.768	0.909	0.832
Bayes Network	stars	0.929	0.07	0.93	0.929	0.929
	quasars	0.93	0.071	0.929	0.93	0.92
Logistic Regression	stars	0.855	0.081	0.914	0.855	0.884
	quasars	0.919	0.145	0.864	0.919	0.891
RBF network	stars	0.861	0.094	0.902	0.861	0.881
	quasars	0.906	0.139	0.866	0.906	0.886
SMO	stars	0.917	0.077	0.923	0.917	0.92
	quasars	0.923	0.083	0.918	0.923	0.921
LibSVM	stars	0.976	0.033	0.967	0.976	0.972
	quasars	0.967	0.024	0.976	0.967	0.971
Voted Perceptron	stars	0.928	0.079	0.921	0.928	0.925
	quasars	0.921	0.072	0.927	0.921	0.924
AdaBoostM1	stars	0.954	0.099	0.906	0.954	0.93
	quasars	0.901	0.046	0.952	0.901	0.926
Bagging	stars	0.981	0.02	0.98	0.981	0.981
	quasars	0.98	0.019	0.981	0.98	0.981
LogitBoost	stars	0.968	0.091	0.914	0.968	0.941
	quasars	0.909	0.032	0.966	0.909	0.937
Decision Table	stars	0.976	0.056	0.946	0.976	0.961
	quasars	0.944	0.024	0.975	0.944	0.959
ADTree	stars	0.908	0.034	0.964	0.908	0.935
	quasars	0.966	0.092	0.913	0.966	0.939
Decision Stump	stars	0.952	0.099	0.906	0.952	0.929
	quasars	0.901	0.048	0.95	0.901	0.925
NBTree	stars	0.978	0.022	0.978	0.978	0.978
	quasars	0.978	0.022	0.978	0.978	0.978
Random forest	stars	0.984	0.021	0.979	0.984	0.982
	quasars	0.979	0.016	0.984	0.979	0.982
IB1	stars	0.978	0.024	0.976	0.978	0.977
	quasars	0.976	0.022	0.978	0.976	0.977

disadvantage of algorithms in common use are summarized in Table 2 of Ball & Brunner 2009.

CONCLUSION

Many classification algorithms in WEKA have been tried to separate quasars from stars using data from SDSS and UKIDSS databases. Just for our case, LibSVM, NBTree, Bagging, Random Forest and IB1 have obtained better performance. These approaches can be trained to build models to preselect quasar candidates from large sky survey databases. In order to improve the reliability of candidates, we may implement a number of high-efficient classification algorithms to preselect quasar candidates separately and then take the cross-result from the results of different methods. WEKA

Table 3: Accuracy and Training Time by Different Methods Based on SDSS Sample.

Method	Accuracy	Training Time
Naive Bayes	81.7025%	0.87s
Bayes Network	92.9302%	3.76s
Logistic Regression	88.7379%	6.95s
RBF network	88.3353%	20.99s
SMO	92.0412%	13.39s
LibSVM	97.1403%	1293.96s
Voted Perceptron	92.4206%	9.02s
AdaBoostM1	92.7679%	13.24s
Bagging	98.0766%	54.98s
LogitBoost	93.8687%	23.23s
Decision Table	95.9945%	25.47s
ADTree	93.7078%	34.95s
Decision Stump	92.6868%	1.24s
NBTree	97.8002%	146.98s
Random forest	98.1832%	63.6s
IB1	97.6963%	0.07s

Table 4: Accuracy and Training Time by Different Methods Based on SDSS_UKIDSS Sample.

Method	Accuracy	Training Time
Naive Bayes	69.4375%	0.04s
Bayes Network	88.1489%	0.09s
Logistic Regression	82.8665%	0.5s
RBF network	75.5545%	1.56s
SMO	83.2724%	0.45s
LibSVM	94.9863%	14.63s
Voted Perceptron	81.2314%	2.68s
AdaBoostM1	89.3551%	0.45s
Bagging	96.6899%	1.61s
LogitBoost	90.6186%	0.82s
Decision Table	93.9344%	1.71s
ADTree	91.0302%	1.53s
Decision Stump	87.6058%	0.04s
NBTree	96.0439%	9.01s
Random forest	97.2216%	1.66s
IB1	96.5241%	0.01s

provides us a good testbed of various algorithms for classification, clustering, data preprocessing, feature selection and an overview of performance for various methods. Before we don't know which method to choose, we may experiment the sample with WEKA. WEKA will give us a good guidance. When the complexity of astronomical data increases, handling them is more difficult. Efficiently mining useful knowledge from data ocean needs sincere communion and close collaboration of specialists

from various fields, such as statisticians, database professionals, software and hardware experts, astronomers, IT experts, computer scientist, data mining experts and so on. Moreover a new generation of astronomers and technologists fit for data science are in urgent requirement. With the deployment and development of ground- and space-based large facilities (e.g. SDSS, LSST, PanStars, LAMOST), careful preparation of input catalogues and followup data processing become more important and urgent. The tools and technologies to suit this situation must be developed as soon as possible. The scientific achievements from other fields may be refereed and transformed into astronomy. To our happiness, many experts from other fields have joined the astronomical field and lots of related projects (e.g. Virtual Observatory) are in bloom, which push astronomy forward smoothly.

Acknowledgements This paper is funded by National Natural Science Foundation of China under grant No.10778724 and No.11033001, by the Natural Science Foundation of Education Departement of Hebei Province Grant No. ZD2010127and by the Young Researcher Grant of National Astronomical Observatories, Chinese Academy of Sciences. We acknowledgment the SDSS database and UKIDSS database.

REFERENCES

Abraham S, Philip N, et al. 2010, MNRAS, astro-ph1011.2173
Ball, N., Brunner, R., 2010, Int.J.Mod.Phys, D19, 1049
Boyle, B. J., Shanks, T., Croom, S. J., et al. 2000, MNRAS, 317, 1014
D'Abrusco, R., Longo, G., Walton, N.A., 2009, MNRAS, 396, 223
Fan, X. H., Carrilli, C. L., Keating, B., 2006, ARA&A, 44, 415
Gao, D., Zhang, Y., Zhao, y., 2008, MNRAS, 386, 1417
Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H., 2009, The WEKA Data Mining Software: An Update, SIGKDD Explorations, 11(1), 10
Kirkpatrick, J. A., Schelegel, D. J., Ross, N. P., et al. 2011, astro-ph1104.4995v1
Lawrence, A., Warren, S.J., Almaini, O., et al., 2007, MNRAS, 379, 1599
Pei, T., Zhang, Y., Zhao, Y., 2010, Proceedings of the SPIE, 7740, 77402G
Peng, N., Zhang, Y., Zhao, Y., 2011, ADASS Proceeding, in press
Richards G. T., et al. ApJ, 2002, 123, 2945
Richards G.T. et al. 2004, ApJS, 155, 257
Richards G. T., et al. 2009, AJ, 137, 3884
Ross, N.P. et al. 2011, astro-ph1105.0606v1
Schlegel, D. J., Finkbeiner, D. P., & Davis, M., 1998, ApJ, 500, 525
Smith et al. 2005, MNRAS, 359, 57
Warren S.J. et al. 2000, MNRAS. 312, 827
Wu, X., and Jia, Z., 2010, MNRAS, 406, 1583
Yeche Ch, et al. 2010, A&A, 523, A14
York, D.G., et al., 2000, AJ, 120, 1579
Zhao, Y., Zhang, Y., 2008, Advances in Space Research, 41(12), 1955
Zhang, Y., Zhao, Y., 2003, PASP, 115, 1006
Zheng, H., Zhang, Y., 2008, Advances in Space Research, 41(12), 1960