Novel Identification and Grouping of Growing Astronomical Supernovae

Asha Anoosheh and Howard Wu
University of California, Berkeley
December 12, 2015

***Abstract*—For years, researchers have been manually classifying images, searching for new supernovae. We propose a system for automatically detecting supernovae. Our multi-step pipeline takes in two images, a template image of the sky and a subject image, and produces a trained classifier that detects supernovae in subtracted images.**

I. Introduction

The purpose of our system is to detect supernovae early on, sooner than existing manual detection systems. In doing so, we can observe supernovae over the course of their lifetime. The advantage of additional time is to enhance existing models of supernovae light curves at infancy stages. Until recently, most supernovae have not been detected soon enough to measure characteristics at early stages.

Our work will utilize the Katzman Automatic Imaging Telescope (KAIT) at the Lick Observatory in San Jose, California. Using telescope data obtained over the course of two decades, we train our system to quickly and accurately identify new supernovae in the night sky.

II. Image Alignment with Fourier Spectrum Analysis

We develop a system for aligning images by define a template image and a subject image for alignment. Due to the possible variations in weather and exposure, the Fourier spectral content of each individual image is analyzed in place of the direct pixel content of the images. To do this, we take the two-dimensional Discrete Fourier Transform (DFT) of the images in question, given by:

$$F(k,l) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} f(i,j) e^{-i2\pi(\frac{ki}{N} + \frac{lj}{N})}$$

for an $N \times N$ image, where $f(a,b)$ represents the image in the spatial domain and $F(k,l)$ represents the corresponding Fourier space.

This technique provides more accuracy and clarity for determining rotation and translation of our subject image with respect to the template image. Our system considers analysis at a base of 0 degrees, with ± 20 degrees of freedom. Note that any arbitrarily defined rotation and degree of freedom can be defined with our system. In addition, a deterministic number of iterations is provided for the depth of analysis one wishes to perform on the subject image in comparison to the template image. The typical range for iterations is between one to three iterations. From there, the similarity of the two images are considered and a rigid transformation is applied to the subject image. We define the transformed, subject image to be our subject image with rotation and translation applied.

III. Image Subtraction with Point Spread Function (PSF) Correction

Now with a template image and transformed, subject image, we proceed to perform image subtraction. The Earth's atmosphere causes light from stars to spread in a Gaussian shape, and in varying amounts due to weather and exposure lengths. It can be approximated as a convolution, due to symmetry, on the image. To account for these possible differences in weather patterns and also exposure times between photographs, we determine a convolutional kernel K that matches with the point spread function (PSF). We define our PSF to be the following formula:

$$\sum([S \star K](x,y) - T(x,y))^2$$

where S represents our transformed, subject image, T represents our template image, and K represents our convolutional kernel. We decompose the kernel K onto basis functions in order to make this a linear least-squares problem:

$$K(u,v) = \sum_n a_n(x,y) K_n(u,v)$$

where K_n is represented by:

$$K_n(u,v) = e^{-(u^2+v^2)/2\sigma_k^2} u^i v^i$$

The idea behind this technique is that given the same star on two separate images with differing exposure times, without loss of generality, a star in the template image could be double in radius in comparison to the same star in the transformed, subject image. We can then use this technique to approximate a match between the two stars and therefore when performing image subtraction, correctly remove the matching stars.

To perform these operations, we use High Order Transform of Point Spread Function and Template Subtraction (HOTPANTS). HOTPANTS works by dividing the provided images into several regions and fitting a convolution kernel for each region. The kernel sum is used to sigma clip outliers from the distribution when solving for individual sections of regions. HOTPANTS successfully blends and subtracts the template and transformed, subject image, outputting a subtracted image.

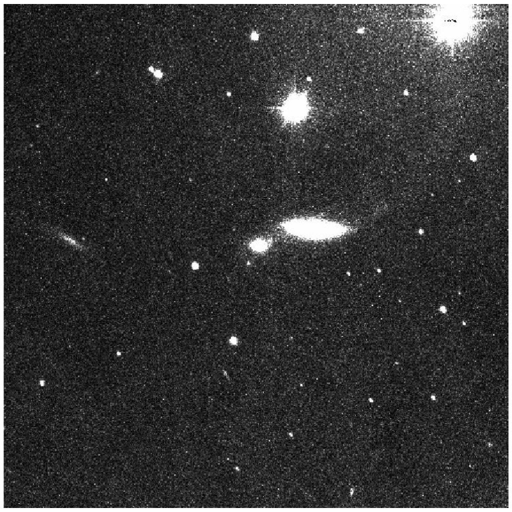

Fig. 1: Template Image

Fig. 2: Subject Image

IV. Extract Areas of Interest

With the subtracted image, our system proceeds to use Source Extractor (SExtractor) for automated detection and photometry. Our goal in using this is to detect any points of interest in the subtracted image, where interest points are meant to represent new astronomical bodies previously unaccounted for in the template image. SExtractor works by determining what the background is and proceeding to check whether pixels belong to an object or to the background. We generate an index of (x, y) coordinates which represent the points of interest. Note that each coordinate pair comes with a flux value that can be used for thresholding.

Often times, SExtractor will identify points that are false positives. These are commonly noise or artifacts left over from the image processing steps. Our system remedies this by filter out true positives by means of statistical learning.

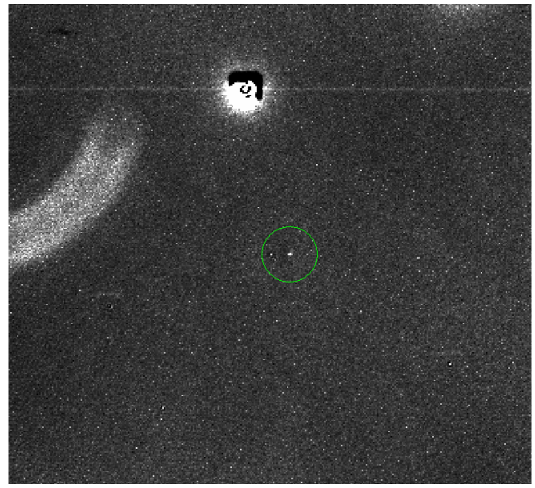

Fig. 3: Subtracted Image with Marked Supernovae

V. Supervised Classification

General supervised learning entails learning a model from a training set of data for which we provide the desired output for each training example. A set of data known as training examples must be given to the learning algorithm, from which a trained model is obtained, and this model can be used now to return desired outputs for future examples. In this case we want to designate a detection as a real transient or an artifact, and thus the problem is a supervised classification task involving two classes: true and false. Classification is a common machine-learning task that can be performed by many various algorithms.

These algorithms come in two flavors: discriminative and generative. The former attempts to find a decision boundary within the data, so that it best divides the data into two spaces, such that it minimizes the number of examples that fall on the wrong side of the boundary. The latter creates probability distributions based off the examples of each class, forming implicit decision boundaries that signify where the probability of belonging to each class is equal. The boundaries for discriminative classifiers are linear by default. To address this, special mapping functions known as kernels can be applied to the data. The details of kernels are much to elaborate on here, but, in short, they allow for nonlinear, and therefore more powerful, decision boundaries.

With more power comes more responsibility, as very powerful decision boundaries may separate the training data very well, but fail to generalize for new, unseen inputs. This is called overfitting - when the model tailors itself too well to the data at hand. Models usually have one or more free variables in their algorithms known as hyperparameters, which can be set properly to combat overfitting. These need to be tuned somehow to best fit a specific dataset; this is usually done by the user trying cross-validation on various values for each hyperparameter.

VI. Logistic Regression and Support Vector Machines (SVM)

In our case, the ideal situation would be to have - not a discrete output (0 or 1) - but continuous values indicating the probability of an example belonging to a class. A known discriminative procedure that does exactly this is known as Logistic Regression. Although it performs regression, it returns values in the range 0 to 1, indicating how likely an example is to be of a certain class, with the decision boundary returning 0.5. This would aid in knowing which examples are near the boundary and thus possibly false-positives or false-negatives that could be manually inspected if needed. Unfortunately this process is limited to a linear decision boundary that may not be powerful enough in all cases. And although its kernelized counterpart, Kernel Logistic Regression (KLR), can generate a nonlinear boundary, it is considered unreasonable performance-wise.

The solution being used here instead is the Support Vector Machine (SVM) (Cortes & Vapnik 1995.) This is a well-known classifier that can be used with any kernel, and it runs much faster than the previous methods. The downside is that SVMs do not generate probabilities as they are discriminative. A workaround is a procedure known as Platt calibration, which performs a logistic transform of the SVMs binary outputs. Given a classification function f, inputs x, and true class y, the calibration process computes:

$$P(y = 1 \mid x) = \frac{1}{1 + e^{Af(x)+B}}$$

where A and B are learned by the algorithm through maximum-likelihood estimation. Although this process involves running the SVM procedure multiple times (for cross-validation), in practice it is still faster than KLR.

SVMs have a single hyperparameter C, and the RBF (Radial Basis Function) kernel used in our model comes with γ, its own parameter. C controls the complexity of the SVM boundary, making it more rigid or flexible to change, while γ controls the influence that data points have on the decision boundary.

VII. Feature Representation

A single input for a machine-learning algorithm is merely a set of features, usually represented as a vector. The classifier uses these features to make informed decisions about the data as a whole, naturally finding features that separate the data the best.

Our algorithm cuts out a 15 by 15 pixel region of the image surrounding the center of the recorded supernova, and we use the pixel values as our features directly, by unravelling it into a vector of length 225. Our data is of small enough dimension that no dimensionality reduction is required. The experiment is run with and without normalization of data for comparison.

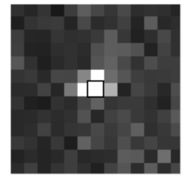

Fig. 4: Extracted Cutout of Supernovae

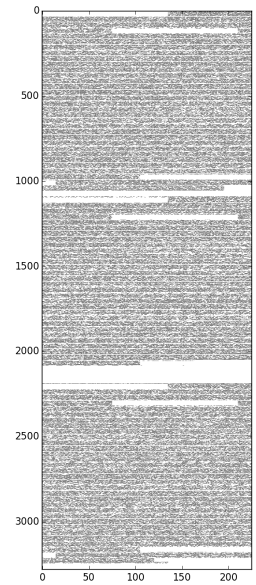

Fig. 5: Matrix Representation of Sample Training Data

VIII. Training Data

Over the past couple decades, the Katzman Automatic Imaging Telescope (KAIT) has been taking images of specific regions in the night sky. Until now, trained researchers have examined the captured images for signs of young supernovae and marked their WCS (World Coordinate System) locations, via a method similar to the template subtraction pipeline described previously. We can remove equal-sized patches surrounding the supernovae positions as positive training examples.

To acquire negative training data, patches of the same size are extracted from the image everywhere except for where the supernova resides. Since we have much more of this negative data than positive data, we can weigh the examples inversely proportional to their count to balance the classifier. This places importance on correctly classifying the positive data, otherwise, the classifier will tend to mark everything as negative simply due to sheer quantity.

IX. Pipeline

Having the absolute coordinates of the known supernovae, the templates, and the new images, we can create our own training set. By subtracting each template from its corresponding new image, we get all new light sources in the sky. And by converting the WCS coordinates into pixel values in the image, using the aperture type and telescope direction information found in the header of each image file (FITS format), we can cut out a patch of image around that pixel which will contain our supernova. We can also produce three rotated versions of this patch to feed as more training data, as supernovae are rotation-invariant.

We then feed these to SkLearn's support-vector classifier and save the trained model to disk for future use. Upon prediction of new data points, Pratt calibration with 5x-cross validation is performed to output probabilistic outputs indicating likeliness of real supernova instead of binary ones (closer to 1.0 means higher chance of being real supernova, and closer to 0.0 means higher chance of being noise or an artifact.)

X. Future Work

An interesting learning algorithm that caught our attention was the Import Vector Machine (IVM) (Zhu and Hastie 2005). This is a model that tries to solve the computational expense of Kernelized Logistic Regression by implementing it with one-step look-ahead greedy selection of the non-zero dual coefficients. We would like to compare the results of this novel algorithm with the tried-and-true SVM.

XI. Conclusion

Our system is designed to work with imaging data from most telescopes and uses machine learning to quickly identify supernovae. Extracted points of interest are efficiently processed for classification training. The system will benefit astronomers with their work on enhancing existing models of supernovae features at infancy stages.

Acknowledgment

The authors would like to graciously thank Professor Alexei Vladimir Filippenko and his Ph.D students, Weikang Zheng and Isaac Shivvers, for their assistance. In addition, the authors would also like to thank Professor Andrew Becker from the University of Washington for his work on Point Spread Function correction algorithms.

References

[1] Coates A., Huval B., Wang T., Wu D., Catanzaro B., Ng A., 2013, in Proceedings of the 30th International Conference on Machine Learning (ICML-13), pp. 13371345

[2] Cortes C., Vapnik V., 1995, Machine Learning, 20, 273

[3] Fumera G., Roli F., 2005, Pattern Analysis and Machine Intelligence, IEEE Transactions on, 27, 942

[4] Gezari S. et al., 2012, Nature, 485, 217

[5] Heller A., A Wide New Window on the Universe, S&TR, November 2005.

[6] Hodapp K. W., Siegmund W. A., Kaiser N., Chambers K. C., Laux U., Morgan J., Mannery E., 2004, in Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, Vol. 5489, Ground-based Telescopes, Oschmann Jr. J. M., ed., pp. 667678

[7] Ji Zhu and Trevor Hastie, "Kernel Logistic Regression and the Import Vector Machine", Journal of Computational and Graphical Statistics, 2005, Vol. 14(1), pp. 185-205.

[8] Kaiser N., Pan-STARRS: a wide-field optical survey telescope array, Proc. SPIE, Ground-based Telescopes, 2004.

[9] Magnier E., 2006, in The Advanced Maui Optical and Space Surveillance Technologies Conference

[10] Raina R., Battle A., Lee H., Packer B., Ng A. Y., 2007, in Proceedings of the 24th international conference on Machine learning, ACM, pp. 759766

[11] Smartt S. J. et al., 2014, The Astronomers Telegram, 6156, 1