The Astronomical Journal, 152:71 (13pp), 2016 September doi:10.3847/0004-6256/152/3/71

ENSEMBLE LEARNING METHOD FOR OUTLIER DETECTION AND ITS APPLICATION TO ASTRONOMICAL LIGHT CURVES

Isadora Nun, Pavlos Protopapas, Brandon Sim, and Wesley Chen
Institute for Applied Computational Science, Harvard University, Cambridge, MA, USA
Received 2015 December 20; accepted 2016 May 8; published 2016 September 1

ABSTRACT

Outlier detection is necessary for automated data analysis, with specific applications spanning almost every domain from financial markets to epidemiology to fraud detection. We introduce a novel mixture of the experts outlier detection model, which uses a dynamically trained, weighted network of five distinct outlier detection methods. After dimensionality reduction, individual outlier detection methods score each data point for "outlierness" in this new feature space. Our model then uses dynamically trained parameters to weigh the scores of each method, allowing for a finalized outlier score. We find that the mixture of experts model performs, on average, better than any single expert model in identifying both artificially and manually picked outliers. This mixture model is applied to a data set of astronomical light curves, after dimensionality reduction via time series feature extraction. Our model was tested using three fields from the MACHO catalog and generated a list of anomalous candidates. We confirm that the outliers detected using this method belong to rare classes, like Novae, He-burning, and red giant stars; other outlier light curves identified have no available information associated with them. To elucidate their nature, we created a website containing the light-curve data and information about these objects. Users can attempt to classify the light curves, give conjectures about their identities, and sign up for follow up messages about the progress made on identifying these objects. This user submitted data can be used further train of our mixture of experts model. Our code is publicly available to all who are interested.

Key words: catalogs – methods: data analysis – methods: statistical – stars: statistics – stars: variables: general

1. INTRODUCTION

Outlier, or anomaly, detection is a growing area of research in machine learning with broad applications that range from credit card fraud detection (Ghosh & Reilly 1994) to anomaly pattern detection for disease outbreaks (Wong et al. 2003). Multiple studies have demonstrated the diverse applications of outlier detection, even in the last decade (Hodge & Austin 2004; Chandola et al. 2007; Aggarwal 2013; Kalinichenko et al. 2014; Pawar & Mahindrakar 2015). Anomalies are of interest since they can represent both artifacts of the data— for example measurement errors, sampling errors, standardization failure, faulty distributional assumptions, etc. (Osborne & Overbay 2004)—as well as novel, interpretable findings. In some cases, outliers can be more valuable to study than "normal data points" because they can drive comparisons and lead to the discovery of root causes. Consider the scenario of an entire town infected by a disease. If there is any case of a disease-negative person, studying this outlier would be more medically useful than than studying the rest of the population. For HIV-1, this hypothetical was actually the case—the discovery of one HIV-1 resistant woman in Nairobi, Kenya (Fowke et al. 1996) led to the discovery of a natural immunity and more insight into combatting the virus.

The identification of fradulent activity is another practical usage of outlier detection (Ahmed et al. 2015). Suspicious activity, such as money laundering, should have different signatures in the data than normal usage (Rohit & Patel 2015). In computer science, malware can be identified by creating baseline usage models of "safe" programs and then identifying attacks based on deviation (Tang et al. 2014, p. 109). Manufacturing requires similar identification of defects to prevent costly recalls (Worden et al. 2000). Genomic variations, such as differential gene expression in cancerous cell lines, are also anomalies when compared to benign cells (Wang & Rekaya 2010).

For our paper, we have chosen astronomy as the focus field where the detection of anomalies can often lead to the discovery of novel objects in the universe. Given the increasing number of surveys such as SDSS (York et al. 2000), Dark Energy Survey (Drinkwater et al. 2010), PanSTARRS (Hodapp et al. 2004), LSST (Tyson et al. 2002), etc., and subsequent increase of data, there is an increased need for automated analysis (Nun et al. 2014). Astronomical data often take the form of light curves, which are then labeled with the different known classes (quasars, microlensing, eclipsing binaries, etc.). Objects that do not belong to any of these categories because of a significant difference could be rare or unknown celestial events (supernovae, dwarf novae, Coronae Borealis, etc.) which deserve further study. These types of objects are usually rare and a minority in this kind of survey. This definition of outliers will be considered throughout this whole paper.

Despite these broad applications, there is a requirement for accuracy and automation. Most applications of outlier detection rely on the confidence of the classification. Other cases involve high-dimensional data sets where the standard methods become computationally infeasible. There is often a trade-off between different methods, which are better or worse at identification in different regions of the entire space. We have developed a mixture of experts model, used in conjunction with feature-space reduction, that allows for more accurate outlier detection even on large data sets. The mixture of experts model is an ensemble method model where different algorithms are combined to improve predictions. This method assumes that the data can be summarized by a collection of functions, each of which is defined over a local region of the domain. The approach attempts to allocate different experts model to summarize the data located in these different regions. The

individual responses of the experts model are nonlinearly combined using a single gating network. This allows us to have different models, which individually may be computationally simpler to fit the regions that the particular model is best at with little trade-off. We use these individual components to form the final engine.

We have chosen five different experts/algorithms models for outlier detection: two *k*-Nearest Neighbors (*k*-NN1, and *k*-NN2; Hodge & Austin 2004), Random Forest, and Joint Probability approaches (RF + JP; Nun et al. 2014), Local Correlation Integral (LoCI; Papadimitriou et al. 2002), and Learned Probability Distribution (LPD; Eskin 2000). To use them in the ensemble method, we had to make two necessary modifications. First, we adapted each method to scale freely across *n*-dimensional space. Second, we applied a sigmoid function to the scores outputted by each expert model to make individual outputs comparable to each other.

This paper is organized into the following sections. Section 2 is devoted to work related to ensemble methods for outlier detection. Section 3 describes the different classes of outlier detection methods, and we detail the ones used in our ensemble method. The adopted methodology and the details of the ensemble method are shown in Section 4. The adaptations of the original methods and our implementations to make the problem scalable and efficient are presented in Section 5. Section 6 contains the information about the data used in this work and Section 7 shows the results of the performed tests and the experiments with real data. Conclusions follow in Section 8.

2. RELATED WORK

We believe that our adaptation of the mixture of experts model to outlier detection is novel and not currently found in the literature. There are examples in the literature that address the use of ensemble methods for anomaly detection, but none refer to the particular case of a mixture of experts combining aspects of differing feature spaces. Furthermore, these examples provide basic theoretical results with implementation or application. In this paper, we provide a novel model as well as a case study in the characterization of astronomical light curves.

Bhattacharyya & Kalita (2013) analyze the advantages and disadvantages of using ensemble methods for anomaly detection. Some of the primary benefits are scalability, natural interpretation of the results, and the possibility of combining multiple weak methods into a single strong one. Among the drawbacks are the difficulty of selecting the optimal methods to combine from a very large pool of algorithms as well as the difficulty of providing real-time performance.

Zimek et al. (2014b) describe the necessary steps and challenges for developing ensemble methods for unsupervised outlier detection. They highlight the necessity of having accurate but diverse models and the difficulties of combining their scores. Gao & Tan (2006) only tackle the latter problem by presenting two methods for converting outlier scores into probabilities for later use in an ensemble method. The first approach assumes that the posterior probabilities follow a logistic sigmoid function with parameters learned from the distribution of outlier scores. The second approach models the scores as a mixture of exponential and Gaussian probability functions with posterior probabilities learned in a Bayesian fashion.

Zimek et al. (2014a) presents an integral method where they use data perturbation to introduce diversity in the individual anomaly detection algorithms. They define a perturbation as the addition of small and controlled noise to the data. Combining the individual outlier rankings allows for the construction of an outlier detection ensemble for which they also develop a method for rank accumulation.

3. BACKGROUND THEORY

The mixture of experts approaches derives its strength from combining the output of separate methods (experts), which presumably have different regimes of expertise across a diverse feature space.

The reader can find an extensive description of outlier detection methods in Hodge & Austin (2004). In that work the authors divide these methods into three different types:

1. *Type 1* is applied when there is no prior information for the data (labels). They assume that "normal" points lie in the same region of feature space and that outliers are far from this area.
2. *Type 2* is applied when labels for normal and abnormal data are available. They are usually classifiers which are trained with these data and can consequently flag the outliers that belong to the already known outlier classes.
3. *Type 3* is applied when only labels for normal data are available. They aim to define the normality boundaries, and anything that does not lie within the normal boundary is flagged as an outlier.

A more specific review of anomaly detection methods, as well as a description of existing ones for astronomical data, can be found in Nun et al. (2014).

In this paper, five outlier detection methods (some modified from previous works, others independently implemented) were applied to astronomical data. Here, we describe each one of them, their advantages, and their drawbacks.

3.1. *k-Nearest Neighbors 1* (k-*NN1*)

We use a modified version of the standard *k*-NN methods described in the outlier detection and classification literature (Hodge & Austin 2004). The classical approach is an unsupervised algorithm (type 1) which assumes that anomalies are significantly distant from the "normal" data and thus can be found by computing a distance metric.

In other words, for each point, we compute its distance to its *k*-nearest neighbors, where the distance in $\mathbb{R}^n$ is the Euclidean distance, or L^2 norm. Note that this distance metric is generalized and can be replaced by other metrics as required by the data set. The sum of the distances are used as the final score.

K-NN methods are geometric in nature, and thus they are density-based outlier detection methods. This leads to drawbacks for this type of method. For example, two classes that are very similar in density may lead to misclassification if careful tuning of parameters is not performed. Also, the curse of dimensionality strikes when higher dimensions are reached. Because of the increasing sparsity of data in higher dimensions as well as the tendency for pairs of points to become close to equidistant from each other in high dimensional space, the *k*-nearest neighbor algorithm becomes unstable when the number of dimensions becomes too large. Therefore, dimensionality

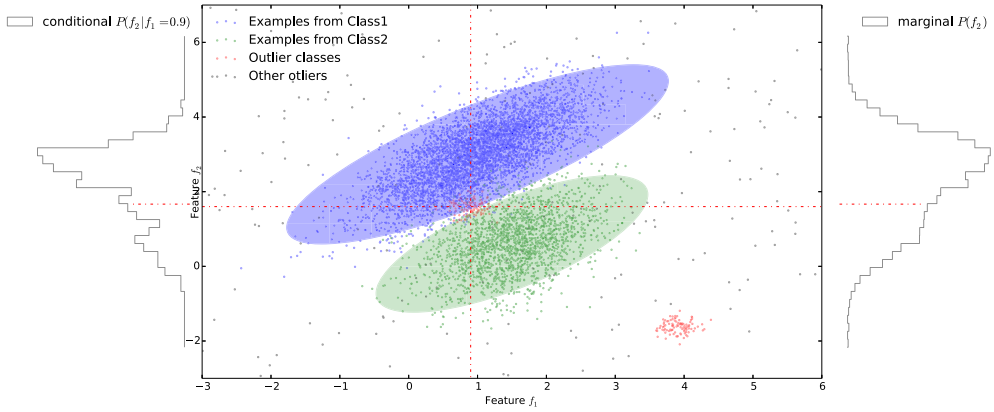

Figure 1. Simple illustration of the method. The blue and green points represent instances in a two dimensional feature space from known class1 and class 2 respectively. The shaded areas represent the boundaries learned from the RF classification. The gray points represent isolated outliers and the red points represent outlier classes.

reduction, such as principal components analysis (PCA), is often suggested before attempting outlier detection via k-nearest neighbors for high dimensional data.

3.2. k-Nearest Neighbors 2 (k-NN2)

As with the other k-NN algorithm, this version is also discussed in standard classification/outlier detection approaches (Hodge & Austin 2004). K-NN2 also belongs to type 2 algorithms. If m of the k-nearest neighbors, for $m < k$, lie within a specific distance threshold, d, then the exemplar is considered to lie in a sufficiently dense region of the data distribution and is classified as normal. However, if there are less than m neighbors inside the distance threshold, then the exemplar is an outlier.

The naive computation complexity is $O(n^2)$, but can be reduced to $O(n \log n)$ with the use of kd-trees (see details in Section 5.2). The process of labeling outliers is similar to the above: assume that scores are normally distributed, then create a standard deviation cutoff. Such a cutoff can be tuned to select a certain proportion of possible outliers.

Both nearest neighbor methods share the same weaknesses when dealing with high dimensional data.

3.3. Random Forest and Joint Probability (RF + JP)

This is a supervised learning method modified from Nun et al. (2014), and it is a type 3 algorithm. Here, a classifier is trained on data with known-class labels. Then, for each point, a membership probability vector is produced that lists the probability of belonging to each category. The joint probability for the particular combination of membership probabilities can then be calculated. Nun et al. (2014) used the random forest classifier to produce the probabilities and a Bayesian Network to construct the joint probabilities. Outliers are then identified as points that have low joint probabilities because their class membership probability vectors are not been seen often enough in the training data.

An illustration of how this method works is presented in Figure 1. In most unsupervised methods, the red points in the middle will not be considered as outliers because they are in a region with point density that is not separable. In the naivest supervised methods, anything that is outside the boundaries is considered as an outlier. For the example of the outlier class in the middle, the product of the probabilities or the sum of the

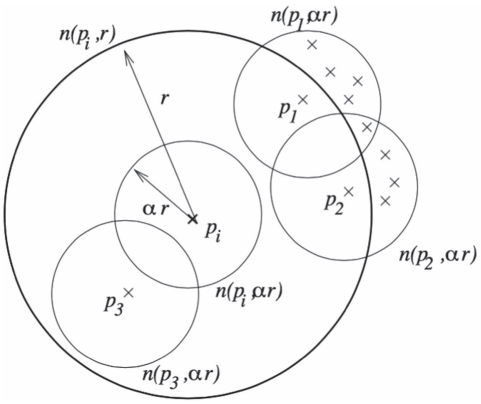

Figure 2. Simple illustration of the definitions used above. The point p_i has four r-neighbors contained within the larger circle with radius r: p_i, p_1, p_2, and p_3. p_i and p_3 each have only one αr-neighbor: themselves, since no other point lies within the circles centered at those points with radius αr. p_1, on the other hand, has 6 αr neighbors, and p_2 has 5. Then, we can figure out the MDEF and σ_{MDEF} and hence the outliers.

distances to the known classes may not be adequate as an outlier score, and therefore, the joint probability is a better measure for outliers. This case occurs when the conditional probability is lower than the marginal probability[1] as it can be seen from this simple illustration. The conditional probability shown on the left is smaller than the marginal probability shown on the right. This model will consider those objects as outliers.

3.4. Local Correlation Integral (LoCI)

LoCI is a type 1 method that determines whether or not a point is an outlier by looking at the distribution of distances between pairs of objects. The algorithm involves a multi-granularity deviation factor (MDEF), and then selects a point as an outlier if its MDEF value deviates significantly (more than 3σ) from local averages. Intuitively, the LoCI method finds points that deviate significantly from the density of points in its local neighborhood. This is formalized in the MDEF concept. Let the r-neighborhood of an object p_i be the set of objects within distance r of p_i. Then, the MDEF at radius r for a point p_i is the relative deviation of its local neighborhood density

[1] Not necessarily true for all cases.

from the average local neighborhood density in its r-neighborhood. An object with neighborhood density that matches the average local neighborhood density will have MDEF 0; outliers will have MDEFs far from 0. More formally, we have:

$$\mathrm{MDEF}(p_i, r, \alpha) = 1 - \frac{n(p_i, \alpha r)}{\bar{n}(p_i, \alpha, r)}. \quad (1)$$

Here, $n(p_i, \alpha r)$ is the number of αr-neighbors of p_i; that is, the number of points $p \in \mathbb{P}$ such that $d(p_i, p) \leqslant \alpha r$, including p_i itself, so $n(p_i, \alpha r) > 0$. Let $\bar{n}(p_i, \alpha, r)$ denote the average of $n(p, \alpha r)$ over the set of r-neighbors of p_i, and define

$$\sigma_{\mathrm{MDEF}}(p_i, r, \alpha) = \frac{\sigma_{\hat{n}}(p_i, r, \alpha)}{\bar{n}(p_i, r, \alpha)} \quad (2)$$

where

$$\sigma^2_{\bar{n}}(p_i, \alpha, r) = \frac{1}{n(p_i, r)} \sum_{p \in \mathcal{N}(p_i, r)} (n(p, \alpha r) - \bar{n}(p_i, r, \alpha))^2. \quad (3)$$

Then, to determine if a point is an outlier, we use the following algorithm: for each $p_i \in \mathbb{P}$, compute $\mathrm{MDEF}(p_i, r, \alpha)$ and $\sigma_{\mathrm{MDEF}}(p_i, r, \alpha)$. If $\mathrm{MDEF} > 3\sigma_{\mathrm{MDEF}}$, flag p_i as an outlier. If, for any $r_{\min} \leqslant r \leqslant r_{\max}$, a point p_i is flagged as an outlier via the aforementioned mechanism, then we consider that point to be an outlier. These cutoffs can be determined on a per-problem basis, but in general we use the following: We set $r_{\max} \simeq \alpha^{-1} R_p$ and $r_{\min}$ such that we have $\bar{n}_{\min} = 20$ neighbors. The number of neighbors can be tuned as necessary for the data set being analyzed. A graphical illustration of the aforementioned quantities can be found in Figure 2.

3.5. Learned Probability Distribution

Modified from Eskin (2000), LPD is also a type 1 algorithm. The main goal of this method is to model the underlying generating process for more complicated data as being drawn from a mixture of two simpler distributions. Outlier classification is achieved by first assuming that all points belong to the normal group. Conventionally, the points in the non-outlier group are assumed to be normally distributed, and a T-statistic can be computed as a measure of the probability of observing this grouping given that they are generated from a normal distribution. Then, for every order k, we iterate through the $\binom{n}{k}$ space and move k points at a time into the outlier class, which is usually assumed to follow a uniform distribution. Then, the score of the new grouping is computed, which represents the probability of observing the updated grouping. If the change in score is past a given threshold, the k points are marked as outliers in that order of space.

This means that optimizing for the threshold parameter can only be performed to target a certain number of outliers which again requires additional knowledge. For computational efficiency, the scoring is often based on log scores.

There are three main assumptions which form the basis of this method's efficacy. First, it is assumed that the non-outlier group can be fairly modeled by the assumed non -outlier probability distribution (in our case, a Gaussian distribution). Second, we assume that the anomalous elements are sufficiently distinct from the non-outlier elements. Finally, the model assumes that anomalies are few (<5% of the entire data set), otherwise the model will become distorted.

The greatest weakness of this algorithm is its dependence on the underlying simple models that form the mixture. This means that the normal points' need to follow a known distribution that can lead to computed t-statistics (such as joint distributions, etc.). When in doubt, a Gaussian model can be assumed as the underlying non-outlier distribution, but, if possible, the appropriate distribution should always be chosen.

4. METHODOLOGY

4.1. Mixture of Experts

With the mixture of experts approach, we assume that each outlier detection method performs best within a particular domain of the sample space. In this ensemble method, we combine the results from each method in a smart way so that the diversity of experts model can make up for the deficiencies of the individual methods over particular domains. Therefore, the result of each expert approach is weighted by values generated based on the location of the point in the 64-dimensional space and the approach's strength in this particular part of the feature space. Next, we explain what these parameters represent and the methodology we use to obtain them.

The gating probability g_i^x is the weight assigned to each expert i for data point $\boldsymbol{x}$. The weights are calculated using a soft-max gating network. The idea of using the soft-max procedure is to follow a "winner takes all" mechanism for choosing the best expert (Arbib 2003). This is useful in some network models for enforcing competition between different possible outputs of the network. In the soft-max procedure, a weight is assigned to each input so that all weights add to one, and the largest input receives the biggest weight. Let $I = \{I_a: a = 1, \ldots, N\}$ be the input and β be a positive parameter; the weight $w_a(I; \beta)$ for each input I_a is defined by:

$$w_a(I; \beta) = \frac{e^{\beta I_a}}{\sum_b e^{\beta I_b}}. \quad (4)$$

Since the exponential function is monotonically increasing, in the limit of $\beta \longrightarrow \infty$, the weight of the largest input tends to1, and all the other weights tend to 0. On the other hand, performing the optimization of a function that takes the maximum of two or multiple numbers might be difficult to solve. For example $f(x, y) = \mathrm{argmax}(x, y)$ has a sharp corner along the line $x = y$ and consequently is not differentiable. Therefore, an alternative to handle this is to use the soft-max function since it is differentiable everywhere.

Applying the soft-max function in the mixture of experts context, the gating probability g_i^x of each expert i for data point $\boldsymbol{x}$ is given by the following equation:

$$g_i^x = \frac{\exp(\boldsymbol{\eta}_i^T \boldsymbol{x})}{\sum_{j=1}^{k} \exp(\boldsymbol{\eta}_j^T \boldsymbol{x})}, \quad (5)$$

where $\boldsymbol{x}$ is a 64-dimensional data vector (one data point), and η_i is the gating parameter for each expert i, with a total of e experts.

As can be seen from Equation (5), to obtain the gates g_i^x, it is first necessary to calculate the parameter η_i. η_i is a matrix of dimensions—number of experts by number of features. To

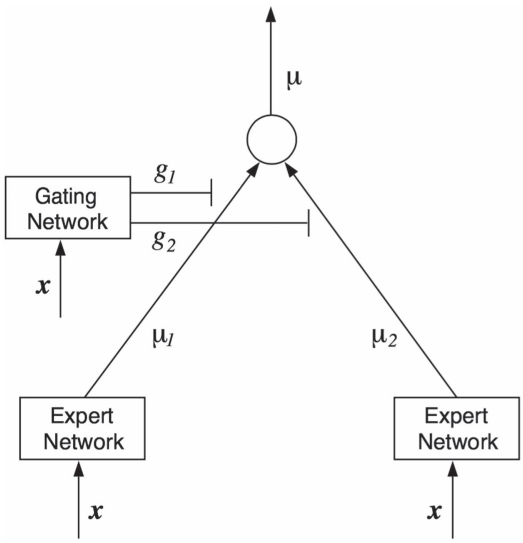

Figure 3. Two-level hierarchical mixture of experts. To form a deeper tree, each expert is expanded recursively into a gating network and a set of sub-experts.

determine their values, we perform a gradient descent optimization. Before explaining how this optimization is performed, it is first important to inquire into the meaning of the η_i variable.

The matrix η_i weights experts per feature but not per feature value. In other words, the feature value originates from x, and therefore, there is a linear relationship between the experts model and the feature values. This is not exactly the ideal scenario, and we consider this as the main drawback of the mixture of experts model approach. For example, in one-dimensional case, one expert wins, independently, all of the feature value. In the multidimensional case, there is a linear combination of expert methods; along the axes, one expert model wins regardless of the feature values, but in between the axes there is a linear combination of methods. In the ideal scenario, we would have one expert approach per feature domain or range. Unfortunately, this implies that there is an exponential number of parameters to be learned. An efficient method for this approach might be developed in a future work.

To find the optimal η, we define the following cost function I that we aim to minimize with respect to η:

$$I = \frac{1}{N}\sum_i \|L_i - y_i\|, \tag{6}$$

where y_i represents the objects labels: 0 for non-outliers and 1 for outliers and L_i is the abnormality score for each particular data point i, as

$$L_i = \sum_e g_{ie} S_{ie} \tag{7}$$

where S_{ie} is the abnormality score of expert e for object i.

As we previously mentioned, the objective function is minimized by using a stochastic gradient descent (SGD) approach. The reader can find details about the implementation in Section 5.7 and an extended explanation in the Appendix in this document. In our work, we use a two-level hierarchical mixture of experts model, in which multiple expert networks are gated according to the process described above in order to provide a final recommendation. It is also possible, as seen in Figure 3, to have each expert network be recursively expanded into its own gating network and a set of its own sub-experts. In this way, multiple expert models with an expertise in specific domains can be combined using a hierarchy of gating probabilities to form one comprehensive prediction.

5. IMPLEMENTATION

Given the large size and dimensionality of our data (see following section for details), we need all of the methods to be scalable and time efficient. On the other hand, since we must be able to compare "less of outlier" versus "more of outlier," each expert method must output a numerical score that measures the degree of abnormality of an object. To achieve this, we adjust some of our outlier detection algorithms. Next, we explain in more detail each of these modifications.

5.1. Dimensionality Reduction

As we mentioned in Sections 3.1, and 3.2, when using algorithms such as k-nearest neighbors on high dimensional data, it might be useful to first perform dimensionality reduction. Using the Feature Analysis for Time Series (FATS) library developed by Nun et al. (2015; for more details, please see Section 6), the high-dimensional astronomical light curve data was mapped to 64 astronomically descriptive features per light curve. We tried to apply in this new space other methods such as PCA. However, the principal components are chosen to pick the bulk of the data variance. Since we are looking for a few rare objects, PCA does not fit our goal. Furthermore, given that both k-NN1, and k-NN2 have a high accuracy (>90%) when testing them with known outliers, we decided not to dig in further in dimensionality reduction.

5.2. Neighbor Distances

Computing all neighbor distances is naively an $\mathcal{O}(n^2)$ problem. To reduce the running time of k-NN1, k-NN2, and LoCI, we use kd-trees (Friedman et al. 1977).[2] These structures are binary trees where tree nodes correspond to splits of space, allowing for $\mathcal{O}(n \log n)$ performance for all neighbor distances. Other spatial partitioning methods exist as well, but performance here is sufficient for our application.

5.3. Construction

RF + JP is a supervised algorithm and consequently needs to be trained. The remaining four outlier detection methods are unsupervised and thus do not need training. Nevertheless, a base "construction" is necessary for all methods. This refers to making a pass through all the data; since these algorithms are either distance based (k-NN1, and k-NN2), density based (LoCI), or distribution based (LPD), they need to "learn" the data distances, the data densities, or the data distributions.

In the ideal scenario, each algorithm would be run on the whole data set, and each object would be compared to the rest. This would be memory and time inefficient in our case. Given that the data are large enough (roughly 250,000 light curves per field, see Section 6) and that we assume that outliers are a minority, we considered that "constructing" the algorithms with a portion of the data does not affect the final results. To do so,

[2] We use the Scikit-learn library at http://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KDTree.html for all kd-tree nearest neighbor calculations.

we took 5% of the data at random and built the models with it. In the case of *k*-NN1, *k*-NN2, and LoCI, we used it to construct the *kd*-trees. In the case of LPD, we used it to estimate the normal distribution. Consequently, to make our method scalable when analyzing a new light curve, we did not run each of the methods in the whole data but used the already built algorithms.

5.4. LoCI

As we described in Section 3.4, the original algorithm considers every point that follows MDEF $> 3\sigma_{\mathrm{MDEF}}$ as an outlier. Since in our case we do not want a binary score (outlier or non-outlier), but a score that quantifies the degree of abnormality of each object, we define the LoCI's score to simply be the value of MDEF. Non-outliers will have MDEF values closer to 0, and outliers' values will be far from 0.

5.5. LPD

When using the LPD method, recall that we are modeling the generative process for the data as a mixture of two models; in particular, we use a Gaussian distribution for the normal class and a uniform distribution for the outlier class. Because of the large size of our data set, we assume for simplicity that moving one element out of the normal class and into the outlier class for testing does not change the parameters of the Gaussian distribution of the normal class. This is a reasonable assumption because of the size of our data set; no one point, even if it is an outlier, will change the distribution of the non-outlier class. Therefore, to find the outliers, the problem is simplified: we must only calculate the log-likelihood of each point belonging to the normal class (modeled as a Gaussian distribution with all the data). Then, we take the points with the smallest likelihoods and flag them as outliers.

5.6. Outlier Detection Scores

As Zimek et al. (2014b) describe in their work, there are several aspects to consider when normalizing and combining scores from different models, but the most important aspect is to make the scores comparable. Since many of the outlier detection methods used in this work do not assign probabilities to outputs, their scores are not equivalent. Consequently, a sigmoid function is applied to the outputted scores of each model.

Specifically, we combined two sigmoid functions for each method: one function for the highest scores (outliers) and one for the lowest scores (non-outliers). This is done to avoid the bimodality of the data; outliers and non-outliers have their separable distribution (see top figure in Figure 5).

To accomplish this, we follow two steps for each model. First, we obtain the ROC (receiver operating characteristic) curve (Metz 1978) of the expert by using the training data set. For a better comprehension of our method, a description of ROC curves and its characteristics is presented next. ROC curves are graphical representations that show the performance of a binary classifier under different discrimination thresholds. Specifically, the ROC curve is a plot of the true positive rate (TPR) versus the false positive rate (FPR) for a range of decision cutpoints or thresholds. Once we obtain the ROC curve, we calculate its maximum Youden's index (Youden 1950). Youden's index is defined for each point in the ROC curve and corresponds graphically to the height of the point to the diagonal (see Figure 4). It is maximum for the optimum cutoff value of the performance test.

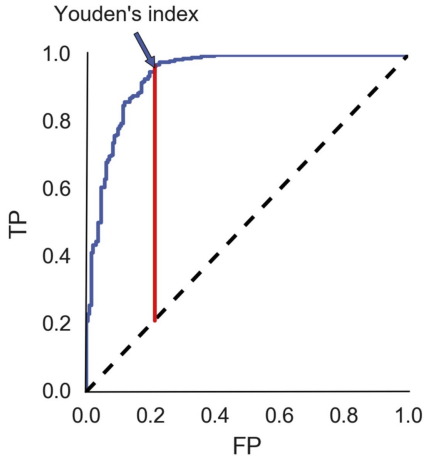

Figure 4. ROC curve and Youden's index. The vertical axis is the TPR, outliers that were correctly identified as such. The horizontal axis is the FPR, normal objects that were incorrectly labeled as outliers. The ROC curve plots the true positive rate against the false positive rate for a variety of decision thresholds and visually represents the trade-off that must be made between improving one's true positive rate while maintaining an acceptable false positive rate.

Youden's index can consequently be used as a threshold to optimally outliers separate from non-outliers: anything above S_o is considered as outlier and anything below S_n as non-outlier. This index is important because it is the index at which the F-score is maximized.

As a second step, we calculate the median absolute deviation (MAD) of the non-outliers' scores, S_n, and the median absolute deviation of the outliers' scores, S_o. These values correspond to the δ parameters (sharpness of the sigmoid function, see below) of our two sigmoid functions.

We choose to use MAD as a variability measure because of its robustness to extreme differences in scale. For some models, we found out that a small number of outliers had extremely high values. Such extreme differences in scale among scores for an expert model leads to undesirable bimodality in our new score functions, a problem alleviated by utilizing the more robust mean absolute deviation metric.

By using the value of Youden's index Y we determine the new scores function (sigmoid function) $S2$ in the following way:

$$S2(i_{i\in\{S(i)>Y\}}) = \frac{1}{1+e^{\frac{-(S(i)-Y)}{\delta_o}}} \tag{8}$$

$$S2(i_{i\in\{S(i)<Y\}}) = \frac{1}{1+e^{\frac{-(S(i)-Y)}{\delta_n}}} \tag{9}$$

where δ_o is the MAD value for S_o and δ_n is the MAD value for S_n.

Figure 5 shows the difference between the old and the new distribution before and after pre-processing. In particular, it corresponds to the results obtained from testing on a balanced test set, which has the same amount of outliers and non-outliers in order to avoid the accuracy paradox (Zhu 2007). As expected, $S2$ has values from 0 to 1.

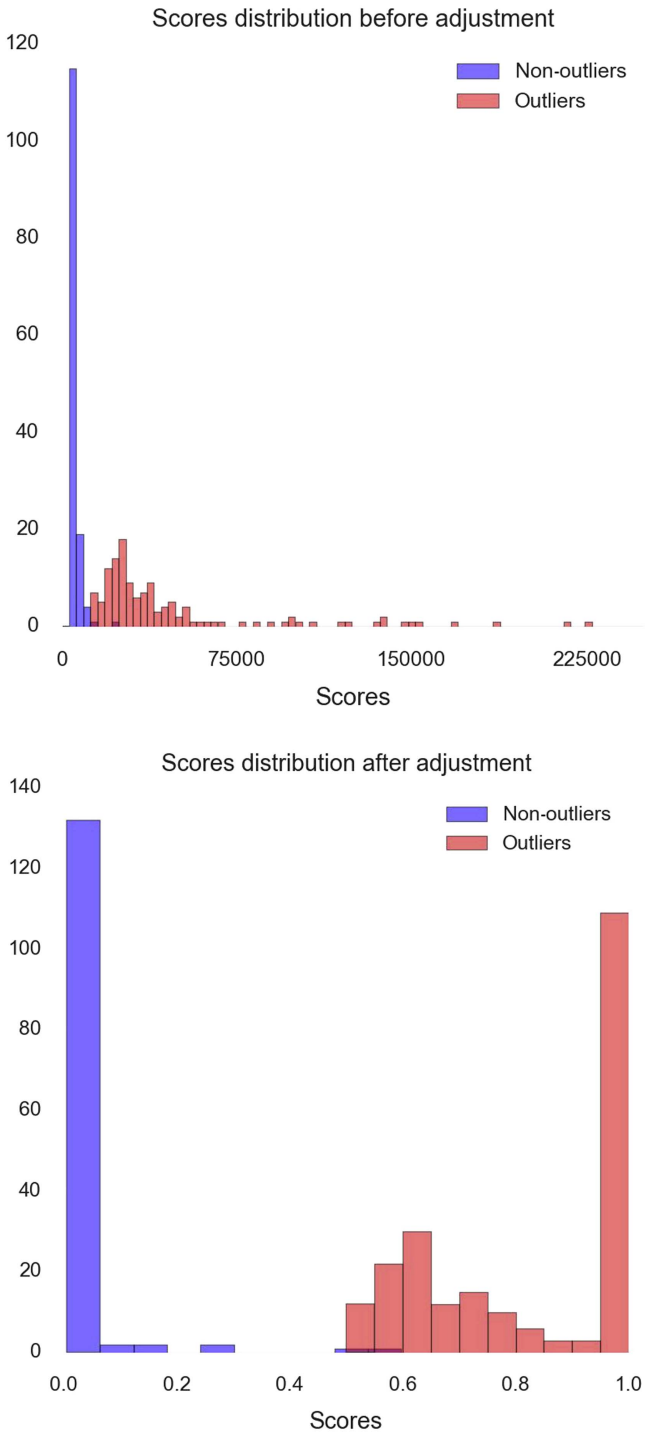

Figure 5. Scores processing for *k*-NN1

5.7. Stochastic Gradient Descent

SGD is an optimization method developed by Bottou (1998) which seeks for the minimization of a loss function $f(\hat{y}, y)$ that measures the cost of predicting $\hat{y}$ when the actual answer is y (Bottou 2012). It is an iterative algorithm in which each step moves in the negative direction of the function gradient. To do so, in each iteration, a set of parameters are updated to minimize the loss function. While a standard gradient descent approach requires running through all the samples in the training set to perform a single update for a parameter in a particular iteration, SGD estimates the gradient using only one randomly picked example from the training set. It can be shown using the Robbins–Siegmund theorem that, for an appropriately chosen (decreasing) learning rate η and under other mild assumptions, SGD almost always converges (Bottou 2012).

During the optimization process of the mixture of expert models, we encountered overflow and underflow limitations. The values of $\exp(\boldsymbol{\eta}_i^T \boldsymbol{x})$ (see Equation (5)) are either too large or too little to be handled by a 64-bit machine (i.e., they cannot be represented in a floating point number). To deal with this, for every overflow or underflow case, we subtract from $\exp(\boldsymbol{\eta}_i^T \boldsymbol{x})$ its range (the difference between the largest and smallest value). This process does not change the results of the optimization since we subtract it from the nominator and the denominator in Equation (5) and consequently the operation cancels out.

Table 1
MACHO Training Set Composition

	Class	Number of Objects
1	Non variable	3969
2	Quasars	58
3	Be-stars	127
4	Cepheid	78
5	RR Lyrae	288
6	Eclipsing Binaries	193
7	MicroLensing	574
8	Long Period Variable	359

6. DATA

The data used in this paper correspond to roughly twenty million light curves obtained from the massive compact halo objects (MACHO) database. MACHO is an astronomical survey started in 1992 whose goal was to detect dark matter in the form of massive compact halo objects. Stars were monitored in the Large Magellanic Cloud (LMC), the Small Magellanic Cloud, and the Galactic Bulge for several years. In order to reach all these areas of the sky, the observational target space was divided into 200 observation fields of view.

With the aim of reducing the dimensionality of the data while conserving the individual characteristics of each object, we calculated 64 descriptive features for each light curve in the database. This was achieved by using the package FATS developed by Nun et al. (2015). These features were tested and proved to be suitable for classifying light curves of different classes with a very high accuracy as the reader can see in http://isadoranun.github.io/tsfeat/FeaturesDocumentation.html.

For the supervised outlier detection method and for parameter tuning, we used the MACHO training set (Kim et al. 2011), whose composition is presented in Table 1. Variable stars were collected from the MACHO variable catalog.[3] The catalog is comprised of variable sources from several MACHO variable studies (Alcock et al. 1996, 1997d, 1997e, 1999), the MACHO microlensing studies (Alcock et al. 1997a, 1997b, 1997c; Thomas et al. 2005), and the LMC long-period variable study (Wood 2000). Quasars in the training set are collected from several studies (Blanco & Heathcote 1986; Schmidtke et al. 1999; Dobrzycki et al. 2002; Geha

[3] http://vizier.u-strasbg.fr/viz-bin/VizieR?-source=II/247

et al. 2003). Be-stars are obtained from private communication with Geha et al (2003). The non-variables are randomly chosen from the MACHO LMC database, and any previously known MACHO variables are removed from the nonvariable set.

Similar observational artifact among the light curves in the same field of view is expected to be found occasionally. For example, if a cloud was interfering with the observation of a certain field of view, it is likely that all the light curves in that field will present a simultaneous period of darkness. It is important to consider this when looking for outliers: an object that might appear as an anomaly when compared against the whole database might be normal in its own field of view.

To overcome this problem, we constructed, and ran our algorithm separately by field of observation. Recall that, by "construction," we refer to using a portion of the data set for building the *kd*-trees in the case of *k*-NN1, *k*-NN2, and LoCI, and learning the normal distribution in the case of LPD. The idea of applying the algorithm by field is to find the anomalies within every field and avoid false positives that could be associated with running the algorithm on the whole data set at the same time. To do so, we used as a construction set the set addition of the MACHO training set and a randomly selected set of light-curves from the studied field. The MACHO training set was used to avoid finding known MACHO classes as outliers (quasars, eclipsing binaries, etc.). On the other hand, by selecting random light curves from the field, we assumed that outliers are a minority in our data set and that our selection represented on average normal objects.

Once the "construction" was complete, we found the parameters for the processing of each expert model's scores (see Section 5.6) and also performed optimization of the gating parameters. To do so, we needed a set of "normal" light curves and a set of anomalous ones. Recall that during the optimization, the idea is to maximize the experts model's abnormality score for outliers and minimize it for "normal" light curves. For the normal data set, we used again a random subset of light curves from the studied field combined with the MACHO training set. The anomalous set was built from a selection of known MACHO outliers and artificially created outliers. The set of known outliers was obtained from the results of the paper Nun et al. (2014). We manually inspected the top 1000 outliers and selected the ones that did not look like artifacts. The artificial outliers were created by randomly permuting features of light curves and by adding noise (a factor of the standard deviation of the permuted feature).

7. RESULTS

In the following section, we present the results obtained after applying our method to *Field 77* of observation in the MACHO data set.

For each field, the algorithm was constructed and ran in parallel by using the Harvard Odyssey cluster.

Our first step was to perform a test of our outlier detection algorithms. After building each one of them, we wanted to verify if some methods were able to identify outliers that others could not and to observe the intersection of their results. To do so, we constructed our algorithms as explained in Section 5.3 and tested them on a toy data set composed of random light curves from the studied field (non-outliers), the set of known outliers, and the artificially created outliers. The results are presented in Table 2 and a graphical representation is shown in Figure 6.

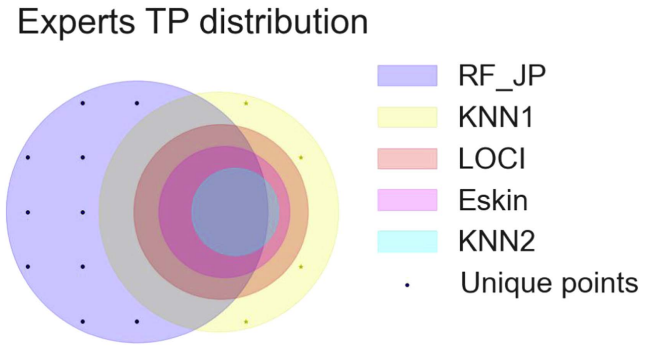

Figure 6. Intersection of the TPR found by each expert on the toy data set. *k*-NN1 and RF+ JP find outliers that the rest of the methods do not.

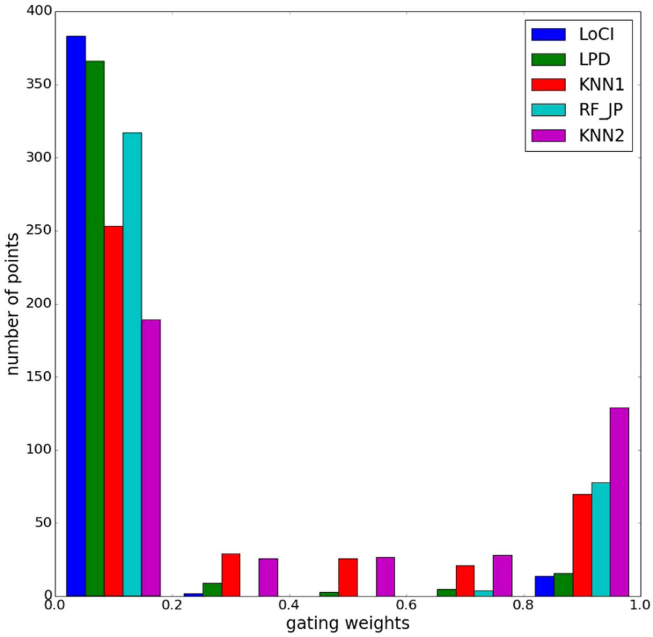

Figure 7. Distribution of the gating parameters. The values of the parameters concentrate in 0 and 1 because of the soft-max approach.

As we can see, none of the algorithms were capable of detecting all the outliers of our toy set, while the union of them resulted in the detection of 100% of the objects. It is also important to note that RF, JP, and *k*-NN1 detected unique outliers that the other methods did not flag as outliers. This verifies our hypothesis which said that some algorithms have a specific area of performance and justifies the idea of creating an ensemble method for finding their regions of specificity.

We then proceeded with the optimization of the parameters by applying the SGD method. We iterated until the value of the objective function reached a plateau of 0.035. The obtained distribution of the gating values for each of the outlier detection algorithms is presented in Figure 7. As we expected because we generated the weights by using soft-max, the values of the gates are mostly concentrated in the range between 0 and 1. The results of applying the obtained model on the toy set are presented in the last row of Table 2. The mixture of experts method detected 99% of the outliers and had a very small number of false positives and false negatives compared to the simple union of the expert methods. In other words, there was a significant improvement of the accuracy using our model of the mixtureof experts.

Table 2
Performance of Each Expert for the Training Set

Expert	TP (222)	FP	TN (140)	Unique
RF and JP	203	7	133	7
KNN1	215	34	106	2
KNN2	102	2	138	0
LoCI	132	20	120	0
LPD	123	2	138	0
Union	222	36	104	...
Mixture of experts	220	2	138	...

Once the parameters were optimized and the model was tested, we ran the ensemble method in some of the other fields and obtained a list of the top outliers for each one of them.

7.1. Top Outliers

Next, we present a small selection of the outliers we obtained for *Field 77*. We visually inspected the top 500 outliers among a total of 100,000 objects and cross-matched them with publicly available astronomical catalogs. Some of these catalogs are collections of known types. For example, LMC Long Period Variables (Fraser et al. 2008) is a collection of long period variables from LMC. On the other hand, catalogs like *XMM-Newton* (Watson et al. 2009) contain X-ray information, which can be useful to further understand the nature of the candidates. We did this in order to check whether there was any information about these objects and, if so, to determine if they belonged to a rare class. Nevertheless, the scope of this paper is not to analyze each of the detected anomalies. In the next section, we detail the second stage of this work where we aim to elucidate the nature of our outliers.

As an example of our results, we show in Figure 8 the light curves of some of the most interesting objects we detected.

MACHO_77.7667.1117, as well as a large portion of our top outliers, belong to the "blue variable class" (Keller et al. 2002). This is a generic class without a unified light curve morphology, features, or underlying physics. Thus, further studies need to be conducted on these objects. We found out that MACHO_77.7552.49 is an Asymptotic Giant Branch Star (He-burning)[4] and MACHO_77.7546.1541 is a Nova.[5] MACHO_77.7918.68 is a hot emission-line star (Reid & Parker 2012). MACHO_77.7306.17 is a red giant (Olivier & Wood 2005).

Figure 9 shows some of the outliers that did not present any information when cross-matching them with known catalogs.

7.2. Outliers Website

As we previously mentioned, the scope of this paper is not to find and identify every outlier in the MACHO catalog, but rather to show the efficiency and efficacy of our method. As we showed above, some of the objects have no information about their class, and it would take significant effort to find this information. Thus, to better understand the nature of the discovered outliers, we created an online catalog with our results http://iacs-courses.seas.harvard.edu/courses/TSC/OutliersWWW/Home.php. A diagram and screenshot of the website are shown in Figure 10. The goal is to clarify in a collaborative way the identity of unusual, rare, or unknown types of astronomical objects or phenomena that have not been yet classified. A user can contribute with information or possible conjectures about the presented outliers. A user can also provide outliers to be included in the catalog by sending them to any of the contact emails and later receive updates with the posted comments. By collecting this feedback, we also expect to select the most "interesting" outliers for a telescope follow-up observation and possibly solve some of the mysteries behind these anomalies.

[4] http://simbad.u-strasbg.fr/simbad/sim-id?Ident=SV*+HV+ 12048
[5] http://wwwmacho.anu.edu.au/Novae/index.html

8. CONCLUSIONS

Although the mixture of experts architecture has been commonly used for classification and regression problems, we believe there is no existing application in the literature where it has been used for outlier detection. In this paper, we proposed a mixture of experts model for anomaly detection in large databases. We combined five different outlier detection methods and ran our mixture of experts model on a subset of the MACHO data set in search of rare stellar objects. As we expected, the mixture of experts allowed us to find outliers that some of the individual algorithms would have ignored. Furthermore, empirical evaluation shows that our model scales linearly with the size of the input and has good scalability as the number of dimensions in the data increases.

We obtained a list of outliers which we published on the website http://iacs-courses.seas.harvard.edu/courses/TSC/OutliersWWW/Home.php with the idea of collaborating on their identification. In future work, we will follow up on the most interesting objects and try to determine what makes them anomalies. We will also explore the use of a different way of combining the expert models (other than soft-max) such that each of them is an expert per feature domain or range and not an expert on the complete space of a feature.

The code of this work is available from https://github.com/isadoranun/OutlierDetection.

We are grateful to Rahul Dave for many discussions related to this work. The computations in this paper were run on the Odyssey cluster supported by the FAS Science Division Research Computing Group at Harvard University.

APPENDIX

Stochastic gradient descent and most optimizations methods work with the derivatives of the function to be minimized. Minimization packages such as *theano* (Bastien et al. 2012) have advanced the field by functionally estimating the derivatives and therefore speed up the optimization process. However, knowing the analytic expressions of these derivatives can help debug and control the overflows and underflows that functions like soft-max. Here we present the derivatives of soft-max. These derivatives were used in our implementation.

Let n_f being the number of features and n_e the number of experts and n_i the number of objects. Then η has dimensions of $[n_f \times n_e]$, $\boldsymbol{x}$ has $[n_i \times n_f]$.

The gate weight for expert e and object i is:

$$g_{ie} = \frac{e^{\boldsymbol{x}_i \cdot \boldsymbol{\eta}_e}}{\sum_e e^{\boldsymbol{x}_i \cdot \boldsymbol{\eta}_e}} = \frac{e^{\sum_f x_{if} \eta_{fe}}}{\sum_e e^{\sum_f x_{if} \eta_{fe}}} \quad (10)$$

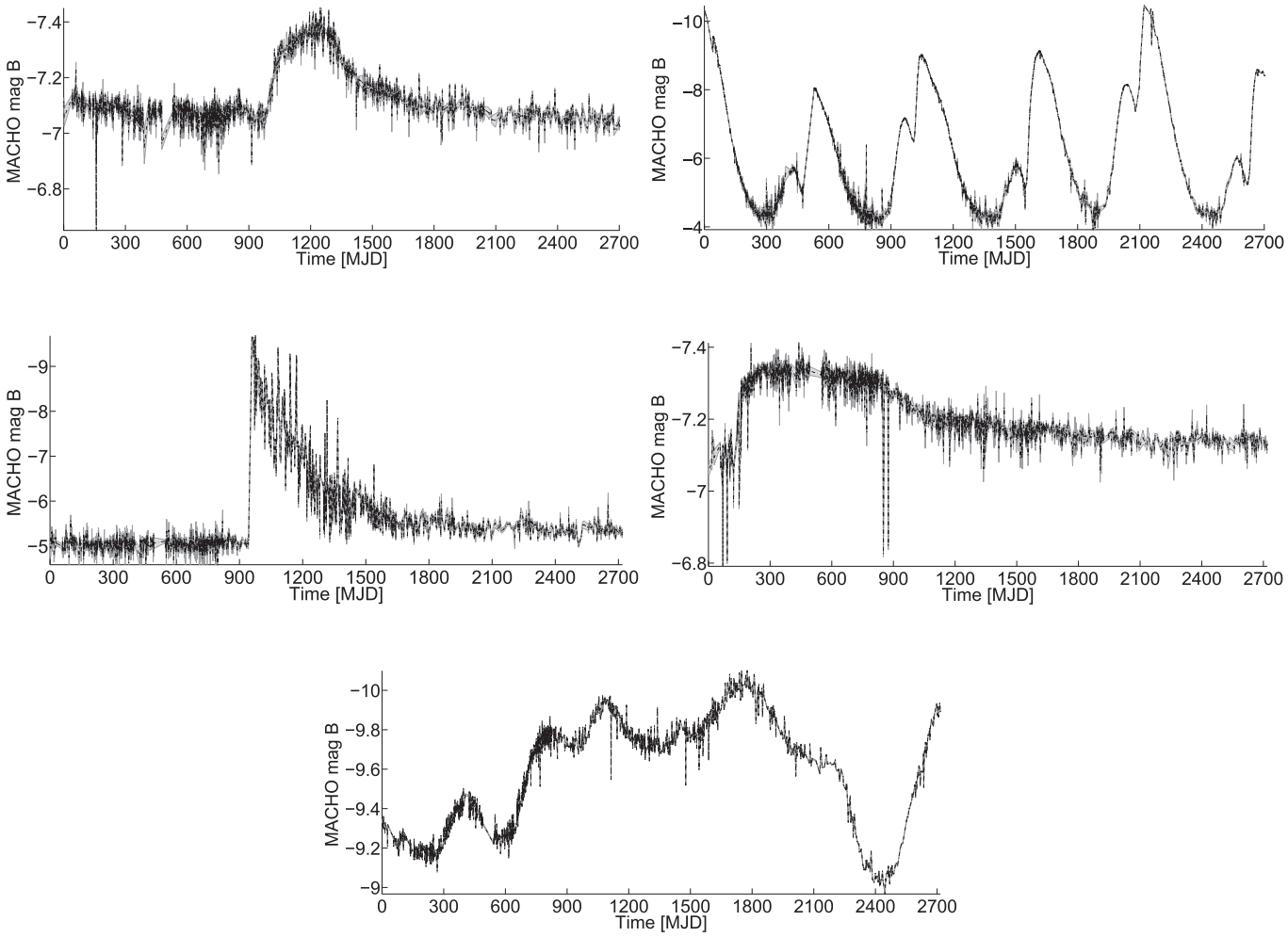

Figure 8. Examples of interesting light curves detected by our method that had positive cross-matching with public catalogs. Top left panel (MACHO_77.7667.1117) is a blue variable, top right panel (MACHO_77.7552.49) shows an Asymptotic Giant Branch Star, bottom left panel (MACHO_77.7546.1541) is a Nova, and bottom right panel (MACHO_77.7918.68) presents a hot emission-line star. The bottom light curve (MACHO_77.7306.17) is a red giant.

We define:

$$\hat{g}_{ie} \equiv e^{\boldsymbol{x}_i \cdot \boldsymbol{\eta}_e}$$

$$\bar{g}_i \equiv \sum_e e^{\boldsymbol{x}_i \cdot \boldsymbol{\eta}_e}$$

Let S_{ie} be the score of expert e for object i, then the total score L_i considering all experts is

$$L_i = \sum_e S_{ie}\, g_{ie}$$

The objective function is:

$$I = \frac{1}{N} \sum_i \| L_i - y_i \|$$

where y_i are the true labels.

The derivatives of the objective function with respect to a specific $\eta_{\varphi\varepsilon}$ is simply

$$\frac{\partial}{\partial \eta_{\varphi\varepsilon}} I = \frac{1}{N} \sum_i \left\| \frac{\partial}{\partial \eta_{\varphi\varepsilon}} L_i - y_i \right\|$$

The derivative of L only depends on the derivative of g's which can be calculated as:

$$\frac{\partial}{\partial \eta_{\varphi\varepsilon}} g_{ie} = \frac{\partial}{\partial \eta_{\varphi\varepsilon}} \frac{\hat{g}_{ie}}{\sum_e \hat{g}_{ie}}.$$

We look at the derivatives of $\hat{g}$ and $\bar{g}$,

$$\frac{\partial}{\partial \eta_{\varphi\varepsilon}} \hat{g}_{ie} = \frac{\partial}{\partial \eta_{\varphi\varepsilon}} e^{\sum_f x_{if} \eta_{fe}} = \delta_{e\epsilon}\, x_{i\phi}\, e^{\sum_f x_{if} \eta_{fe}} = \delta_{e\epsilon}\; x_{i\phi}\; \hat{g}_{i\epsilon}$$

$$\Rightarrow \frac{\partial}{\partial \eta_{\varphi\varepsilon}} \hat{g}_{ie} = \delta_{e\epsilon}\; x_{i\phi}\; \hat{g}_{i\epsilon}$$

$$\frac{\partial}{\partial \eta_{\varphi\varepsilon}} \bar{g}_i = \frac{\partial}{\partial \eta_{\varphi\varepsilon}} \sum_e \hat{g}_{ie} = \frac{\partial}{\partial \eta_{\varphi\varepsilon}} \sum_e e^{\sum_f x_{if} \eta_{fe}}$$

$$= \sum_e \frac{\partial}{\partial \eta_{\varphi\varepsilon}} e^{\sum_f x_{if} \eta_{fe}} = x_{i\phi}\, e^{\sum_f x_{if} \eta_{f\epsilon}}$$

$$\Rightarrow \frac{\partial}{\partial \eta_{\varphi\varepsilon}} \bar{g}_i = x_{i\phi} \hat{g}_{i\epsilon}$$

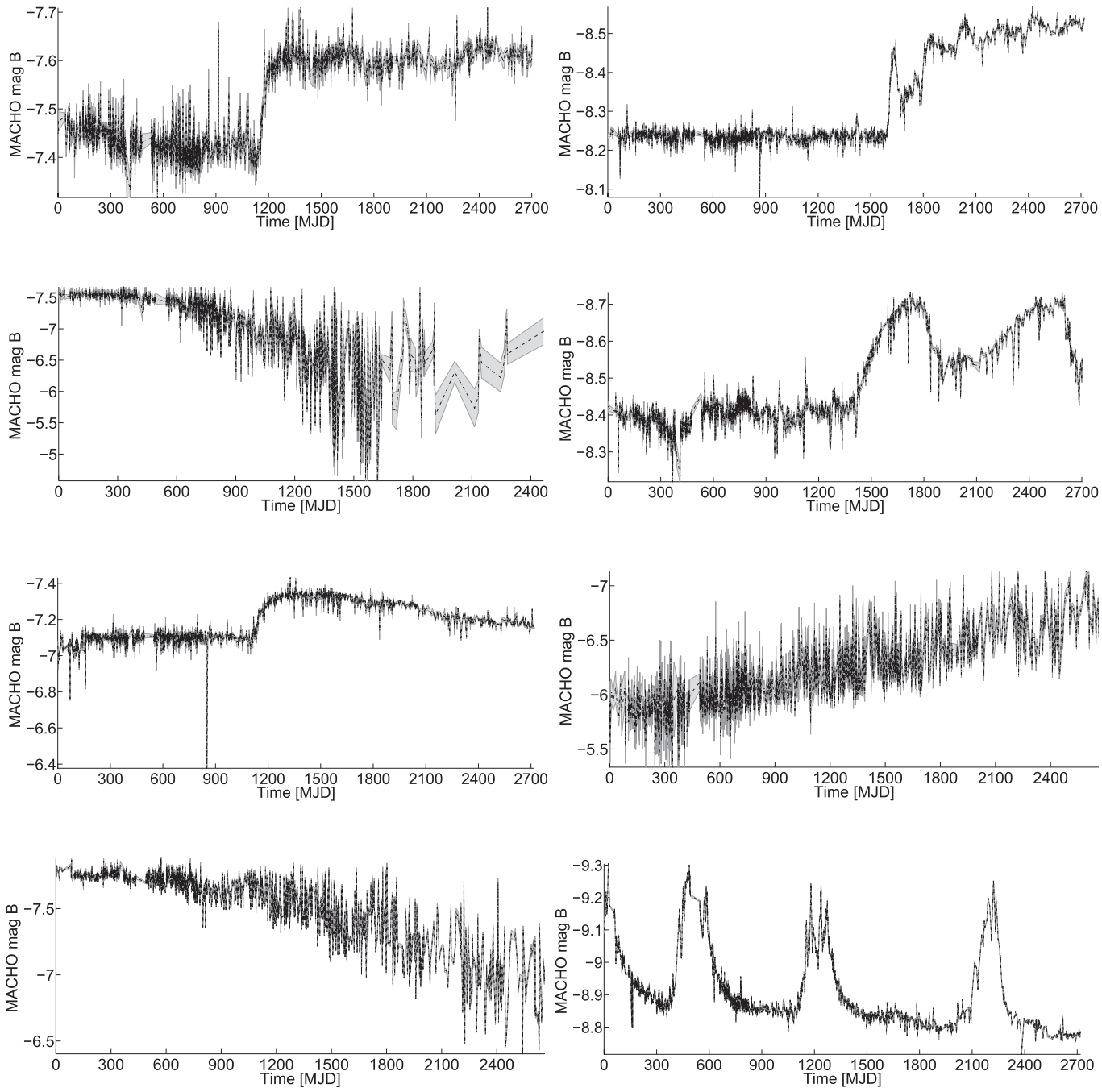

Figure 9. Example of interesting light curves detected by our method that did not present results when cross-matching with public catalogs. From left to right, and top to bottom: MACHO_77.7187.327, MACHO_77.7674.53, MACHO_77.7067.121, MACHO_77.7432.75, MACHO_77.8039.110, MACHO_77.7672.836, MACHO_77.7672.22, and MACHO_77.7554.25.

And therefore

$$\frac{\partial}{\partial \eta_{\varphi\varepsilon}} g_{ie} = \frac{\partial}{\partial \eta_{\varphi\varepsilon}} \frac{\hat{g}_{ie}}{\bar{g}_i} = \frac{\delta_{e\epsilon}\, x_{i\phi}\, \hat{g}_{ie} \bar{g}_i - x_{i\phi} \hat{g}_{i\epsilon} \hat{g}_{ie}}{\bar{g}_i^2}$$
$$= \delta_{e\epsilon} x_{i\phi} g_{ie} - x_{i\phi} g_{ie} \frac{\hat{g}_{i\epsilon}}{\bar{g}_i}$$

The derivative of the objective function is then

$$\frac{\partial}{\partial \eta_{\varphi\varepsilon}} L_i = \sum_e S_{ie} \frac{\partial}{\partial \eta_{\varphi\varepsilon}} g_{ie} = S_{i\epsilon} x_{i\phi} g_{i\epsilon} - x_{i\phi} \frac{\hat{g}_{i\epsilon}}{\bar{g}_i} \sum_e g_{ie}$$

$$\frac{\partial}{\partial \eta_{\varphi\varepsilon}} I = \frac{1}{N} \sum_i \left\| S_{i\epsilon} x_{i\phi} g_{i\epsilon} - x_{i\phi} \frac{\hat{g}_{i\epsilon}}{\bar{g}_i} \sum_e g_{ie} - y_i \right\|$$

Figure 10. Website with the detected anomalies. *Catalog of astronomical outliers*: http://iacs-courses.seas.harvard.edu/courses/TSC/OutliersWWW/Home.php.

REFERENCES

Aggarwal, C. C. 2013, Outlier Analysis (Berlin: Springer)
Ahmed, M., Mahmood, A. N., & Islam, M. R. 2015, Future Generation Computer Systems (Amsterdam: Elsevier)
Alcock, C., Allsman, R. A., Alves, D., et al. 1997a, ApJL, 491, L11
Alcock, C., Allsman, R. A., Alves, D., et al. 1997b, ApJ, 479, 119
Alcock, C., Allsman, R. A., Alves, D., et al. 1997c, ApJ, 486, 697
Alcock, C., Allsman, R. A., Alves, D., et al. 1997d, ApJ, 482, 89
Alcock, C., Allsman, R. A., Alves, D., et al. 1997e, AJ, 114, 326
Alcock, C., Allsman, R. A., Alves, D. R., et al. 1999, AJ, 117, 920
Alcock, C., Allsman, R. A., Axelrod, T. S., et al. 1996, AJ, 111, 1146
Arbib, M. A. 2003, The Handbook of Brain Theory and Neural Networks (Cambridge, MA: MIT Press)
Bastien, F., Lamblin, P., Pascanu, R., et al. 2012, in Theano: New Features and Speed Improvements, Deep Learning and Unsupervised Feature Learning NIPS 2012 Workshop
Bhattacharyya, D., & Kalita, J. 2013, Network Anomaly Detection: a Machine Learning Perspective (London: Chapman and Hall)
Blanco, V. M., & Heathcote, S. 1986, PASP, 98, 635
Bottou, L. 1998, On-line Learning in Neural Networks (New York: Cambridge Univ. Press)
Bottou, L. 2012, Neural Networks: Tricks of the Trade, Vol. 1 (Berlin: Springer)
Chandola, V., Banerjee, A., & Kumar, V. 2009, CSUR, 41, 15
Dobrzycki, A., Groot, P. J., Macri, L. M., & Stanek, K. Z. 2002, ApJL, 569, L15
Drinkwater, M. J., Jurek, R. J., Blake, C., et al. 2010, MNRAS, 401, 1429

Eskin, E. 2000, Proc. of the Seventeenth International Conf. on Machine Learning, Anomaly Detection over Noisy Data Using Learned Probability Distributions, 255
Fowke, K. R., Nagelkerke, N. J., Kimani, J., et al. 1996, The Lancet, 348, 1347
Fraser, O. J., Hawley, S. L., & Cook, K. H. 2008, AJ, 136, 1242
Friedman, J. H., Bentley, J. L., & Finkel, R. A. 1977, ACM Transactions on Mathematical Software (TOMS), 3, 209
Gao, J., & Tan, P.-n. 2006, in Sixth Int. Conf. Data Mining (ICDM'06), Converting Output Scores from Outlier Detection Algorithms into Probability Estimates (Piscataway, NJ: IEEE), 212
Geha, M., Alcock, C., Allsman, R. A., et al. 2003, AJ, 125, 1
Ghosh, S., & Reilly, D. L. 1994, in Proc. Twenty-Seventh Hawaii Int. Conf. 3, System Sciences (Piscataway, NJ: IEEE), 621
Hodapp, K. W., Kaiser, N., Aussel, H., et al. 2004, AN, 325, 636
Hodge, V. J., & Austin, J. 2004, Artificial Intelligence Review, 22, 85
Kalinichenko, L., Shanin, I., & Taraban, I. 2014, Methods for Anomaly Detection: A Survey
Keller, S. C., Bessell, M. S., Cook, K. H., Geha, M., & Syphers, D. 2002, AJ, 124, 2039
Kim, D.-W., Protopapas, P., Byun, Y.-I., et al. 2011, ApJ, 735, 68
Metz, C. E. 1978, in Seminars in Nuclear Medicine (Amsterdam: Elsevier)
Nun, I., Pichara, K., Protopapas, P., & Kim, D.-W. 2014, ApJ, 793, 23
Nun, I., Protopapas, P., Sim, B., et al. 2015, arXiv:1506.00010
Olivier, E., & Wood, P. 2005, MNRAS, 362, 1396
Osborne, J. W., & Overbay, A. 2004, Practical Assessment, Research Evaluation, 9, 1
Papadimitriou, S., Kitagawa, H., Gibbons, P., & Faloutsos, C. 2002, Loci: Fast Outlier Detection Using the Local Correlation Integral, Tech. Rep. IRP-TR-02-09 (Pittsburgh, PA: Intel Research Laboratory), 02
Pawar, A. M., & Mahindrakar, M. S. 2015, IJCA, 119, 17
Reid, W. A., & Parker, Q. A. 2012, MNRAS, 425, 355
Rohit, K. D., & Patel, D. B. 2015, International Journal for Innovative Research in Science and Technology, 1, 129
Schmidtke, P. C., Cowley, A. P., Crane, J. D., et al. 1999, AJ, 117, 927
Tang, A., Sethumadhavan, S., & Stolfo, S. J. 2014, Research in Attacks Intrusions and Defenses (Berlin: Springer)
Thomas, C. L., Griest, K., Popowski, P., et al. 2005, ApJ, 631, 906
Tyson, J. A., Collaboration, L., Labs, B., Technologies, L., & Hill, M. 2002, Astronomical Telescopes and Instrumentation, 10
Wang, Y., & Rekaya, R. 2010, Biomarker insights, 5, 69
Watson, M. G., Schröder, A. C., Fyfe, D., et al. 2009, A&A, 493, 339
Wong, W.-K., Moore, A., Cooper, G., & Wagner, M. 2003, ICML, 808
Wood, P. R. 2000, PASP, 17, 18
Worden, K., Manson, G., & Fieller, N. 2000, JSV, 229, 647
York, D. G., Adelman, J., Anderson, J. E., Jr, et al. 2000, AJ, 120, 1579
Youden, W. J. 1950, Cancer, 3, 32
Zhu, X. 2007, Knowledge Discovery and Data Mining: Challenges and Realities (Hershey, PA: IgI Global)
Zimek, A., Campello, R., & Sander, J. 2014a, in Proc. 26th International Conf. on Scientific and Statistical Database Management, Data Pertbation for Outlier Detection Ensembles (New York: ACM), 13
Zimek, A., Campello, R. J., & Sander, J. 2014b, ACM SIGKDD Explorations Newsletter, 15, 11