Kernels for Periodic Time Series Arising in Astronomy

Gabriel Wachman[1], Roni Khardon[1],
Pavlos Protopapas[2,3], and Charles R. Alcock[2]

[1] Tufts University, Medford, MA USA
{gwachm01,roni}@cs.tufts.edu
[2] Harvard-Smithsonian Center for Astrophysics, Cambridge, MA USA
{pprotopapas,calcock}@cfa.harvard.edu
[3] Harvard Initiative in Innovative Computing, Cambridge, MA USA

Abstract. We present a method for applying machine learning algorithms to the automatic classification of astronomy star surveys using time series of star brightness. Currently such classification requires a large amount of domain expert time. We show that a combination of phase invariant similarity and explicit features extracted from the time series provide domain expert level classification. To facilitate this application, we investigate the cross-correlation as a general phase invariant similarity function for time series. We establish several theoretical properties of cross-correlation showing that it is intuitively appealing and algorithmically tractable, but not positive semidefinite, and therefore not generally applicable with kernel methods. As a solution we introduce a positive semidefinite similarity function with the same intuitive appeal as cross-correlation. An experimental evaluation in the astronomy domain as well as several other data sets demonstrates the performance of the kernel and related similarity functions.

1 Introduction

The concrete application motivating this research is the classification of stars into meaningful categories from astronomy literature. A major effort in astronomy research is devoted to sky surveys, where measurements of stars' or other celestial objects' brightness are taken over a period of time. Classification as well as other analyses of stars lead to insights into the nature of our universe, yet the rate at which data are being collected by these surveys far outpaces current methods to classify them. For example, microlensing surveys, such as MACHO [1] and OGLE [2] followed millions of stars for a decade taking one observation per night. The next generation panoramic surveys, such as Pan-STARRS [3] and LSST [4], will begin in 2009 and 2013, respectively, and will collect data on the order of hundreds of billions of stars. It is unreasonable to attempt manual analysis of this data, and there is an immediate need for robust, automatic classification methods.

It is this need that we address directly with our first contribution: the foundation of an automatic methodology for classifying *periodic variable stars* where

W. Buntine et al. (Eds.): ECML PKDD 2009, Part II, LNAI 5782, pp. 489–505, 2009.

a star is variable if its brightness varies over time, and periodic if the variance in brightness is periodic over time. In the data sets taken from star surveys, each example is represented by a time series of brightness measurements, and different types of stars have different periodic patterns. Fig. 1 shows several examples of such time series generated from the three major types of periodic variable stars: Cepheid, RR Lyrae, and Eclipsing Binary. In our experiments only stars of the types in Fig. 1 are present in the data, and the period of each star is given. A complete solution will automatically process an entire survey, of which a small percentage will be periodic variable stars. We are actively working on automatic methods for filtering out non-periodic variables and for identifying period, however these are outside the scope of this paper. We use the existing OGLEII periodic variable star catalog [5] to show that our classification method achieves $> 99\%$ accuracy once such processing and filtering has been done.

As our second contribution we present several insights into the use of the cross-correlation function as a similarity function for time series. Cross-correlation provides an intuitive mathematical analog of what it means for two time series to look alike: we seek the best phase alignment of the time series, where the notion of alignment can be captured by a simple Euclidean distance or inner product. We show that cross-correlation is “almost” a kernel in that it satisfies the Cauchy-Schwartz inequality and induces a distance function satisfying the triangle inequality. Therefore, fast indexing methods can be used with cross-correlation for example with the k-Nearest Neighbor algorithm [6]. We further show that although every 3×3 similarity matrix is positive semidefinite, some 4×4 matrices are not and therefore cross-correlation is not a kernel and not generally applicable with kernel methods.

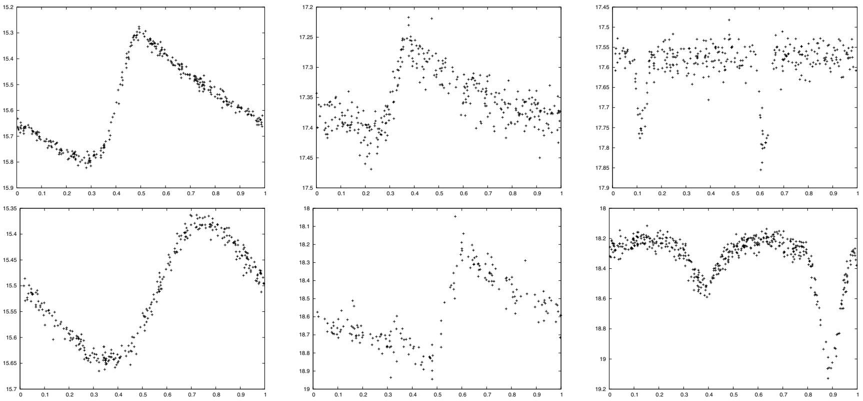

Fig. 1. Examples of light curves of periodic variable stars. Each column shows two stars of the same type. Left: Cepheid, middle: RR Lyrae, right: eclipsing binary. Examples of the same class have similar shapes but are not phase aligned. Examples are a result of folding a long sequence of observations leading to a noisy sample of one period of the light curve. The y-axis labels represent brightness in magnitude units, which is an inverse logarithmic scale (this is the convention in astronomy).

As our final contribution we introduce a positive semidefinite similarity function that has the same intuitive appeal as cross-correlation. We investigate the performance of our kernel on other data sets, both real and artificial, showing excellent performance. We show instances where the kernel outperforms all other methods as well as instances where a simple universal phasing algorithm performs comparably. Our investigation reveals that our kernel performs better than cross-correlation and that the ability to use Support Vector Machines (SVM) [7] with our kernel can provide a significant increase in performance.

The remainder of the paper is organized as follows. Section 2 investigates properties of cross-correlation, and Sect. 3 introduces the new kernel function. Related work is discussed in Sect. 4. We present our experiments and discuss results in Sect. 5. Finally, the concluding section puts this work in the larger context of fully automatic processing of sky surveys.

2 Cross-Correlation

Our examples are vectors in $\mathbb{R}^n$ but they represent an arbitrary shifts of periodic time series. We use the following notation: y_{+s} refers to the vector y shifted by s positions, where positions are shifted modulo n. We then use the standard inner product between shifted examples

$$\langle x, y_{+s} \rangle = \sum_{i=1}^{n} x_i (y_{+s})_i .$$

We define the *cross-correlation* between $x, y \in \mathbb{R}^n$ as

$$C(x, y) = \max_s \langle x, y_{+s} \rangle .$$

In the context of time series, computing the cross-correlation corresponds to aligning two time series such that their inner product, or similarity, is maximized.

2.1 Properties of Cross-Correlation

We first show that cross-correlation has some nice properties making it suitable as a similarity function:

Theorem 1

(P1) $C(x, x) = \langle x, x \rangle \geq 0$.
(P2) $C(x, y) = C(y, x)$.
(P3) The Cauchy-Schwartz Inequality holds, i.e. $\forall x, y, \; C(x, y) \leq \sqrt{C(x, x) C(y, y)}$.
(P4) If we use the cross-correlation function to give a distance measure d such that

$$d(x, y)^2 = C(x, x) + C(y, y) - 2C(x, y) = \min_s \|x - (y_{+s})\|^2$$

then d satisfies the Triangle Inequality.

In other words cross-correlation has properties similar to an inner product, and can be used intuitively as a similarity function. In particular, we can use metric trees and other methods based only on the triangle inequality [8,6] to speed up distance based algorithms using cross-correlation.

Proof. For (P1) note that by definition $C(x,x) \geq \langle x,x\rangle$. On the other hand, $C(x,x) = \sum x_i x_{i+s}$, and by the Cauchy-Schwartz inequality,

$$\sum x_i x_{i+s} \leq \sqrt{\sum x_i^2}\sqrt{\sum x_{i+s}^2} = \sqrt{\sum x_i^2}\sqrt{\sum x_i^2} = \langle x,x\rangle. \tag{1}$$

Which means $\langle x,x\rangle \geq C(x,x) \geq \langle x,x\rangle$ or $C(x,x) = \langle x,x\rangle \geq 0$.

To prove (P2) observe that since $\langle x, y_{+s}\rangle = \langle x_{-s}, y\rangle = \langle x_{+(n-s)}, y\rangle$ maximizing over the shift for y is the same as maximizing over the shift for x.

(P3) follows from K1 of Theorem 2 below (see Proposition 2.7 of [9]) but we give a direct argument here. Let $C(x,y) = \langle x, y_{+s}\rangle = \langle x, z\rangle$, where s is the shift maximizing the correlation and where we denote $z = y_{+s}$. Then by (P1), $\sqrt{C(x,x)C(y,y)} = \sqrt{\langle x,x\rangle\langle y,y\rangle} = \|x\|\|y\|$. Therefore the claim is equivalent to $\|x\|\|y\| \geq \langle x,z\rangle$, and since the norm does not change under shifting the claim is equivalent to $\|x\|\|z\| \geq \langle x,z\rangle = C(x,y)$. The last inequality holds by the Cauchy-Schwartz inequality for normal inner products.

Finally, for (P4) let $x,y,z \in \mathbb{R}^n$. Let τ_{ab} be the shift that minimizes $d(a,b)$.

$$\begin{aligned}
d(x,y) + d(y,z) &= \|(x_{+\tau_{xy}}) - y\| + \|(y_{+\tau_{yz}}) - z\| &(2)\\
&= \|(x_{+\tau_{xy}+\tau_{yz}}) - (y_{+\tau_{yz}})\| + \|(y_{+\tau_{yz}}) - z)\| &(3)\\
&\geq \|(x_{+\tau_{xy}+\tau_{yz}}) - (y_{+\tau_{yz}}) + (y_{+\tau_{yz}}) - z\| &(4)\\
&= \|(x_{+\tau_{xy}+\tau_{yz}}) - z\| &(5)\\
&\geq \|(x_{+\tau_{xz}}) - z\| = d(x,z) &(6)
\end{aligned}$$

Where (3) holds because shifting x and y by the same amount does not change the value of $\|x-y\|$, (4) holds because of the triangle inequality, and (6) holds because by definition τ_{xz} minimizes the distance between x and z. □

Since cross-correlation shares many properties with inner products it is natural to ask whether it is indeed a kernel function. We show that, although every 3x3 similarity matrix is positive semidefinite, the answer is negative.

Theorem 2

(K1) Any 3×3 Gram matrix of the cross-correlation is positive semidefinite.
(K2) The cross-correlation function is not positive semidefinite.

Proof. Let $x_1, x_2, x_3 \in \mathbb{R}$, G a 3×3 matrix such that $G_{ij} = C(x_i, x_j)$, $c_1, c_2, c_3 \in \mathbb{R}$. We prove K1 by showing $Q = \sum_{i=1}^{3}\sum_{j=1}^{3} c_i c_j G_{ij} \geq 0$.

At least one of the products c_1c_2, c_1c_3, c_2c_3 is non-negative. Assume WLOG that $c_2c_3 \geq 0$ and shift x_2 and x_3 so that they obtain the maximum alignment with x_1, calling the shifted versions $\tilde{x_1}, \tilde{x_2}, \tilde{x_3}$ noting that $\tilde{x_1} = x_1$. Now $C(x_i, x_j) = \langle \tilde{x_i}, \tilde{x_j} \rangle$ except possibly when $(i, j) = (2, 3)$, so

$$\begin{aligned}\sum_{i=1}^{3}\sum_{j=1}^{3} c_i c_j G_{ij} &= \sum_{i=1}^{3}\sum_{j=1}^{3} c_i c_j \langle \tilde{x_i}, \tilde{x_j} \rangle + 2c_2c_3(C(\tilde{x_2}, \tilde{x_3}) - \langle \tilde{x_2}, \tilde{x_3} \rangle) \\ &\geq \sum_{i=1}^{3}\sum_{j=1}^{3} c_i c_j \langle x_i, x_j \rangle \geq 0\end{aligned}$$

since $c_2c_3 \geq 0$ and $C(\tilde{x_2}, \tilde{x_3}) \geq \langle \tilde{x_2}, \tilde{x_3} \rangle$ by definition.

The negative result, K2, is proved is by giving a counter example. Consider the matrix A and the row-normalized A'

$$A = \begin{pmatrix} 0 & 1 & 2 \\ 1 & 0 & 0 \\ 2 & 1 & 2 \\ 0 & 2 & 1 \end{pmatrix} A' = \begin{pmatrix} 0 & 0.4472 & 0.8944 \\ 1 & 0 & 0 \\ 0.6667 & 0.3333 & 0.6667 \\ 0 & 0.8944 & 0.4472 \end{pmatrix}$$

where each row is a vector of 3 dimensions. This illustrates a case where we have 4 time series, each with 3 samples and the time series are normalized. Using the cross-correlation function on A', we would get the following Gram matrix

$$G = \begin{pmatrix} 1 & 0.8944 & 0.8944 & 0.8 \\ 0.8944 & 1 & 0.6667 & 0.8944 \\ 0.8944 & 0.6667 & 1 & 0.8944 \\ 0.8 & 0.8944 & 0.8944 & 1 \end{pmatrix}$$

G has a negative eigenvalue of -0.0568 corresponding to the eigenvector $c = (-0.4906, 0.5092, 0.5092, -0.4906)$ and therefore G is not positive semidefinite. In other words $cGc' = \sum_{i=1}^{4}\sum_{j=1}^{4} c_i c_j G_{ij} = -0.0568$. □

3 A Kernel for Periodic Time Series

Since the cross-correlation function is not positive semidefinite, we propose an alternative kernel function that can be used in place of the cross-correlation function with kernel methods. To motivate our choice consider first the kernel

$$K(x, y) = \sum_{i=1}^{n}\sum_{j=1}^{n} \langle x_{+i}, y_{+j} \rangle.$$

Note that here K iterates over all possible shifts, so that we no longer choose the best alignment but instead aggregate the contribution of all possible alignments.

This seems to lose the basic intuition behind cross-correlation and it is indeed not a good choice. On closer inspection we can see that

$$\begin{aligned} K(x,y) &= (x_{+1} + x_{+2} + \ldots + x_{+n})y_{+1} + \ldots + (x_{+1} + x_{+2} + \ldots + x_{+n})y_{+n} \\ &= (\sum_{i=1}^{n} x_{+i})(\sum_{j=1}^{n} y_{+j}). \end{aligned}$$

So K just calculates the product of the sums of the shifted vectors. In particular, if the data is normalized as mentioned above then this is identically zero.

Instead our kernel weights each shift with exponential function so that shifts with high correlation are highly weighted and shifts with low correlation have smaller effect.

Definition 1. *The kernel function* $K : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ *is defined as*

$$K(x,y) = \sum_{i=1}^{n} e^{\gamma\langle x, y_{+i}\rangle} \tag{7}$$

where $\gamma \geq 0$ *is a constant.*

Thus like cross-correlation the value of the kernel will be dominated by the maximizing alignment although the number of "good alignments" is also important. In this way we get positive semidefinite kernel while having the same guiding intuition as cross-correlation. Exponential weighting of various alignments of time series has been proposed previously in [10]. Despite the similarity in the construction, the proof of positive semidefiniteness in [10] does not cover our case as their set of alignments is all possible time warpings under a fixed phase and does not allow for circular shifting. Similar ideas to weight different matches exponentially have also been explored in kernels for multi-instance problems [11].

Theorem 3. *K is a positive semidefinite kernel.*

Proof. Consider the following function

$$K'(x,y) = \sum_{i=1}^{n}\sum_{j=1}^{n} e^{\gamma\langle x_{+i}, y_{+j}\rangle}.$$

By [12], $K'(x,y)$ is a convolution kernel. This can be directly shown as follows. First rewrite K' as

$$K'(x,y) = \sum_{a \in R^{-1}(x)} \sum_{b \in R^{-1}(y)} e^{\gamma\langle a,b\rangle} \tag{8}$$

where $R^{-1}(x)$ gives all shifts of x. It is well known that the exponential function $e^{\gamma\langle x,y\rangle}$ is a kernel [9]. Let $\Phi(x)$ be the underlying vector representation of the this kernel so that $e^{\gamma\langle x,y\rangle} = \langle\Phi(x),\Phi(y)\rangle$. Then

$$K'(x,y) = \sum_{a \in R^{-1}(x)} \sum_{b \in R^{-1}(y)} \langle\Phi(a),\Phi(b)\rangle = \langle(\sum_{a \in R^{-1}(x)} \Phi(a)), (\sum_{b \in R^{-1}(y)} \Phi(b))\rangle \tag{9}$$

Thus K' is an inner product in the same vector space captured by Φ with the map being the aggregate of all elements in $R^{-1}(x)$.

Note that $K'(,)$ iterates over all shifts of both x and y, hence effectively counting each shift n times. For example, observe that for the identity shift, we have $\langle x, y\rangle = \langle x_{+1}, y_{+1}\rangle = \ldots = \langle x_{+(n-1)}, y_{+(n-1)}\rangle$. Hence we need to scale K' by $1/n$ in order to count each shift exactly once. This gives us

$$K(x, y) = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{n} e^{\gamma\langle x_{+i}, y_{+j}\rangle}.$$

Since scaling a kernel (i.e. K') is also a kernel, K is a kernel. □

Previous work [13] has shown that cross-correlation can be calculated in time $O(n \log n)$ where n is the length of the time series. In particular they show that $\langle x, y_{+s}\rangle = \mathcal{F}^{-1}(\mathcal{X} \cdot \hat{\mathcal{Y}})[s]$ where $\cdot$ indicates point-wise multiplication, $\mathcal{X}$ is the discrete Fourier transform of x, and $\hat{\mathcal{Y}}$ is the complex conjugate of the discrete Fourier transform of y. Therefore cross-correlation can be calculated as $C(x, y) = \max_s \mathcal{F}^{-1}(\mathcal{X} \cdot \hat{\mathcal{Y}})[s]$ and using the fast Fourier transform we get the claimed time bound. This easily extends to our kernel by calculating $K(x, y) = \sum_s e^{\mathcal{F}^{-1}(\mathcal{X} \cdot \hat{\mathcal{Y}})[s]}$ implying:

Proposition 1. *$K(x, y)$ can be calculated in time $O(n \log n)$.*

Note that we need take the Fourier transform of each example only once. This gives a significant practical speedup over the naive quadratic time implementation.

4 Related Work

The current discoveries from the microlensing surveys such as OGLE and MACHO are predominantly transient objects such as gravitational microlensing, supernovae etc., and some periodic variable stars [14,15]. Recent work on star surveys introduced the application of semi-automatic classification techniques for periodic variable stars based on simple selection criteria over the parameter space indexed by average brightness, average difference in brightness between two spectral regions, and period, e.g [16,17]. We refer to these three parameters as *explicit features*. The semi-automatic methods require significant human intervention and hence pose an imperfect solution for a survey of even tens of millions of stars. An automatic approach has been proposed in [18]. This approach extracts explicit features from the light curves and applies machine learning methods in the resulting parameter space. Despite the similarity in terms of automation, our approach is unique in that we use the shape of the periodic time series to derive a similarity measure. Furthermore our approach is not astronomy-specific and is applicable across a range of domains.

There are many existing approaches for processing and classifying time series. A classical approach is to extract features of the time series, such as the Fourier basis, wavelets, or Hermite basis representation, and then work directly in the resulting vector space,e.g. [19]. Another major approach models the time series using a generative probabilistic model, such as Hidden Markov Models (HMM), and classifies examples using maximum likelihood or MAP estimates [20]. Our work falls into a third category: using similarity functions or distance measures for time series data [21,22]. Various similarity functions for time series have been proposed. Notably, Dynamic Time Warping (DTW) has been shown to be very effective across a large number of applications [21,23]. Such similarity functions are not phase invariant, hence they rely on a good universal phasing of the data.

Cross-correlation has been proposed precisely as an effective phase-invariant similarity function for astronomy and has been used for anomaly detection [13]. It is faster in runtime, $O(n \log n)$, than other methods that compute a maximum phase-invariant alignment. The notion of phase-invariance similarity has also been explored in the context of time series classification, specifically for time series generated from 2-d shape contours. For example, [23] present a method for applying any distance measure in a phase-invariant context. This allows for the application of Dynamic Time Warping, for instance, to data that is phase-invariant. While in general the run-time $(O(n^3))$ is as bad as brute-force methods such as in [24], they give experimental evidence that their heuristics lead to much faster, run-times in practice. We extend the work in [13] by investigating theoretical properties of cross-correlation and proposing a positive semidefinite alternative.

Several alternative approaches for working with non-positive semidefinite similarity measures exist in the literature. The simplest approach is just to use the (non-PSD) similarity function with SVM and hope for good results. Our experiments in the next section show that this does not always yield the desired performance. Another common alternative is to add a diagonal term λI to the gram matrix in order to render it positive semidefinite. More recent approaches reformulate the SVM optimization to account for the potential non-PSD kernel [25,26]. Finally, [27] show that a similarity function that meets some general requirements can be used to project examples into an explicit feature space indexed by their similarity to a fixed set of examples, and that this preserves some useful learnability properties. Unlike these generic methods, our work gives an explicit kernel construction that is useful for the time series domain.

There is significant overlap between the domain of time series classification and 2-d shape matching [23]. This is in part because a popular method for representing 2-d shapes is to create a time series from the contour of the shape. Shape classification has its own domain-specific approaches and it is beyond the scope of this paper to examine them. Nevertheless we note that shape matching is an example of a phase-invariant time series classification problem, and in fact we will present experiments from this domain.

The general issue of "maximizing alignment" appears repeatedly in work on kernels for structured objects. Dynamic Time Warping is a classic (non-positive semidefinite) example where we maximize alignment under legal potential warping of the time axes. A general treatment of such alignments, characterizing when the result is a kernel, is developed by [28]. Their results do not cover the case of cross-correlation, however. A similar alignment idea has been used for graph kernels in the application of classifying molecules, where each molecule can be seen as a graph of atoms and their bonds [29,30,31]. Here a base kernel is introduced between pairs of nodes in the two graphs. Then one can define a convolution kernel between the graphs using an equation similar to (8) where the sum ranges over all nodes in the graph [29,30]. Note that this approach does not maximize alignments, but sums over all possible alignments. A non-positive semidefinite alternative is to maximally align the two molecules by pairing their atoms in a one-to-one manner [31]. A major question is whether one could define an efficiently computable exponentially weighted version of such a (non-maximizing but PSD) graph kernel. One can show that this problem is closely related to calculating the permanent, a problem well known to be computationally hard [32,33]. As it is a special case of the permanent problem, however, where edge weights are related through the kernel function, it may be possible to calculate efficiently.

5 Experiments

In the following sets of experiments we demonstrate the performance of cross-correlation and our kernel in the context of phase-invariant time series classification.

For real-world data we use time series from astronomy surveys, and time series generated from contours of 2-d images. For artificial data, we generate examples that highlight the importance of phase invariance in an intuitive fashion. We use the same pre-processing for all time series, unless otherwise noted. The time series are smoothed as in [13,34], linearly-interpolated to 1024 evenly spaced points, and normalized to have mean of 0 and standard deviation of 1.

In all experiments we use the LIBSVM [35] implementation of SVM [7] and k-Nearest Neighbors (k-NN) to perform classification. For LIBSVM, we choose the "one-versus-one" multiclass setting, and we do not optimize the soft-margin parameter, instead using the default setting. For k-NN, we choose $k = 1$ following [23], who have published results on the shape data used in this paper[1]. When we use explicit features, we use a linear kernel. When we use cross-correlation or our kernel in addition to explicit features, we simply add the result of the inner product of the explicit features to the value of the cross-correlation or kernel[2].

We use five different similarity functions in our experiments: Euclidean Distance (ED) returns the inner product of two time series. The Universal Phasing (UP) similarity measure uses the method from [13] to phase each time series

[1] We reproduce their experiments as opposed to reporting their results in order to account for the different splits when cross-validating; our results do not differ significantly from those reported in [23].

[2] Another approach would be to perform multiple kernel learning [36] with one kernel being the cross-correlation and the other the inner product of the explicit features. However, this issue is orthogonal to the topic of the paper so we use the simple weighting.

according to the sliding window on the time series with the maximum mean, and then behaves exactly like Euclidean Distance. We use a sliding window size of 5% of the number of original points; the phasing takes place after the preprocessing explained above. In all experiments where we use K as in Equation 7, we do parameter selection by performing 10-fold cross-validation on the training set for each value of γ in $(1, 5, 10, 15, 25, 50, 80)$, then re-train using the value of γ that gave best average accuracy on the training set. When we use Dynamic Time Warping (DTW), we use the standard algorithm and do not restrict the warping window [21]. Finally we note that although cross-correlation is not positive semidefinite, we can in practice use it on some data sets with SVM.

In the first set of experiments we run on the OGLEII data set [5]. This data set consists of 14087 time series (light curves) taken from the OGLE astronomical survey. Each light curve is from one of three kinds of *periodic variable star*: Cepheid, RR Lyrae (RRL), or Eclipsing Binary (EB). We run 10-fold cross-validation over the entire data set, using the cross-correlation (CC), our kernel (K), and Universal Phasing (UP). The results, shown in the left top three rows of Tab. 1, illustrate the potential of the different similarities in this application. We see significant improvements for both cross-correlation and the kernel over Universal Phasing. We also see that the possibility to run SVM with our kernel leads to significant improvement over cross-correlation.

While the results reported so far on OGLEII are good, they are not sufficient for the domain of periodic variable star classification. Thus we turn next to improvements that are specific to the astronomy domain. In particular, the astronomy literature identifies three aggregate features that are helpful in variable star classification: the average brightness of the star, the *color* of the star which is the difference in average brightness between two different spectra, and the *period* of the star, i.e. the length of time to complete one period of brightness variation [16,17]. The right side of Tab. 1 gives the results when these features are added to the corresponding similarities. The features on their own yield very high accuracy, but there is a significant improvement in performance when we combine the features with cross-correlation or the kernel. Interestingly, while Universal Phasing on its own is not strong enough, it provides improvement over the features similar to our kernel and cross-correlation. Notice that a performance gain of 2% is particularly significant in the domain of astronomy where our goal is to publish such star catalogs with no errors or very few errors. The left confusion matrix in Tab. 2 (for SVM with our kernel plus features) shows that we can get very close to this goal on the OGLEII data. To our knowledge this is the first such demonstration of the potential of applying a shape matching similarity measure in order to automatically publish clean star catalogs from survey data. In addition, based on our domain knowledge, some of the errors reported in the left of Tab. 2 appear to be either mis-labeled or borderline cases whose label is difficult to determine.

In addition to classification, we show in Tab. 2 that the confidences produced by the classifier are well ordered. Here we do not perform any calibration and

Table 1. Accuracies with standard deviation reported from 10-fold cross-validation on OGLEII using various kernels and the cross-correlation

	1-NN	SVM		1-NN	SVM
CC	0.844 ± 0.011	0.680 ± 0.011	features + CC	0.991 ± 0.002	0.998 ± 0.001
K	0.901 ± 0.008	0.947 ± 0.005	features + K	0.992 ± 0.002	0.998 ± 0.001
UP	0.827 ± 0.010	0.851 ± 0.006	features + UP	0.991 ± 0.002	0.997 ± 0.001
			features	0.938 ± 0.006	0.974 ± 0.004

Table 2. Three confusion matrices for OGLEII, using SVM with K and features. From left to right we reject none, then the lowest 1%, 1.5% and 2%.

	Ceph	EB	RRL	Ceph	EB	RRL	Ceph	EB	RRL	Ceph	EB	RRL
Cepheid	3416	1	13	3382	1	3	3363	1	3	3352	1	0
EB	0	3389	0	0	3364	0	0	3342	0	0	3312	0
RRL	9	0	7259	1	0	7195	0	0	7166	0	0	7138

simply take the raw output of each of the 3 hyperplanes learned by the SVM[3]. To calculate the confidence in label 1, we add the raw output of the $1v2$ (the classifier separating class 1 from class 2) and $1v3$ classifiers. To calculate the confidence in label 2 we add the negative output of the $1v2$ hyperplane and the output of the $2v3$ hyperplane, etc. We can then reject the examples that received the lowest confidences and set them aside for review. When we reject the lowest 1%, for example, we reject all but 5 errors, showing that almost all of our errors have low confidences. We now have reason to believe that, when we classify a new catalog, we can reliably reject a certain percentage of the predictions that are most likely to be errors. The rejected examples can either be ignored or set aside for human review.

In the next set of experiments we use five shape data sets: Butterfly, Arrowhead, Fish, Seashells introduced in [23], as well as the SwedishLeaf data set [38][4]. These data sets were created by taking pictures of objects and creating a time series by plotting the radius of a line anchored in the center of the object as it rotates around the image [23]. As all of the pictures have aligned each object more or less along a certain orientation, we randomly permute each time series prior to classification in order to eliminate any bias of the orientation. The identification of objects from various orientations is now cast as a phase-invariant time series problem.

A natural and relatively easy problem is to use a classifier to separate the different image types from each other. In this case we attempt to separate butterflies, arrowheads, seashells, and fish. We refer to this data set as *Intershape*[5].

[3] While there are several methods in the literature to produce class-membership probabilities from SVM output [37], exploratory experiments could not confirm the reliability of these probabilities for our data. Therefore we chose to use the simple method based on raw SVM output.

[4] Detailed Information available via `www.cs.ucr.edu/~eamonn/shape/shape.htm`

[5] We treat the SwedishLeaf set differently because it has a different resolution and is not part of the same overall shape data set.

Table 3. Number of examples in each data set. For those data sets that were filtered to include 20 examples of each class, the number of examples post-filtering appears after the '/'.

	Num Examples	Num Classes	Majority Class
Arrowhead	558/474	9	0.19
Butterfly	754/312	5	0.39
Intershape	2511	4	0.30
SwedishLeaf	1125	15	0.07

We also investigate the potential to separate sub-classes of each shape type. The SwedishLeaf data has already been labeled before, and hence the sub-classes are already identified. For the other data sets that have not been explicitly labeled by class before, we generate labels as follows: for the Butterfly and Fish data set, we consider two examples to be the same class if they are in the same genus. For the Arrowhead data set, we consider two arrowheads to be the same type if they share the same type name, such as "Agate Basin", or "Cobbs." In order to make the results more statistically robust, we eliminate sub-types for which there exist fewer than 20 examples. Seashells and Fish have too few examples when processed in this way and are therefore only used in the Intershape data set. A summary of the data sets, including number of examples and majority class probability (that can be seen as a baseline) are given in Tab. 3.

For these experiments we calculate no explicit features. We run 10-fold cross-validation using k-NN with cross-correlation (1-NN CC), the kernel (1-NN K), Dynamic Time Warping (1-NN DTW), Universal Phasing (1-NN UP) and SVM with the kernel (SVM K), Universal Phasing (SVM UP), and Euclidean distance (SVM ED). The results are given in Tab. 4. We also tried using 1-NN with Euclidean Distance, but the performance was not competitive with any of the other methods so we do not include it in the comparison.

The results demonstrate that both cross-correlation and the kernel provide a significant performance advantage. It is not surprising that DTW does not do well since it only considers the one given random phasing of the data. Rather, it is surprising that it does not perform worse on this data. The only way it can expect to perform well with k-NN is if, by chance, for each example there is another example of the same class that happens to share roughly the same phase. In a large enough data set, this can happen, and this may explain why DTW does much better than random guessing. It is interesting that SVM does not always dominate k-NN and does very poorly on SwedishLeaf. It may be that the data are linearly inseparable but there are are enough examples such that virtual duplicates appear in the data allowing 1-NN to do well.

Another interesting observation is that while Universal Phasing never outperforms all methods it does reasonably well across the domains. Recall that this method phases the time series according to the maximum average brightness of a sliding window. This finds a "maximum landmark" in the data for alignment

Table 4. Performance on various shape data sets. All results are cross-validated. Data set names: A = arrowhead, B = butterfly, I = intershape, S = Swedish.

	1-NN CC	1-NN K	1-NN DTW	1-NN UP	SVM ED	SVM UP	SVM K
A	0.54 ± 0.06	0.54 ± 0.08	0.33 ± 0.06	0.49 ± 0.05	0.2 ± 0.05	0.41 ± 0.05	0.63 ± 0.04
B	0.73 ± 0.04	0.73 ± 0.04	0.59 ± 0.08	0.70 ± 0.07	0.4 ± 0.1	0.65 ± 0.08	0.76 ± 0.08
I	0.98 ± 0.01	0.98 ± 0.01	0.84 ± 0.03	0.97 ± 0.02	0.47 ± 0.03	0.8 ± 0.02	0.91 ± 0.02
S	0.84 ± 0.03	0.82 ± 0.03	0.48 ± 0.06	0.78 ± 0.04	0.08 ± 0.03	0.18 ± 0.03	0.33 ± 0.04

and is obviously not guaranteed to be informative of the class in every case. Nevertheless, it works well on the Butterfly and Intershape data sets showing that this type of landmark is useful for them.

As we show in the next set of experiments with artificial data, it is easy to construct examples where Universal Phasing will fail. We generate two classes of time series. Each example contains 1024 points. Class 1 is a multi-step function with one set of four steps beginning at time 0, as well as one spike placed randomly. Class 2 is also a multi-step function but with two sets of two steps, the first at time 0 and the second at time 665 (roughly 65% of the entire time series) and one random spike exactly as in class 1. We show two examples of each class in Fig. 2. We generate 10 disjoint training sets containing 70 examples and test sets containing 30 examples for cross-validation. We keep the training set small to avoid clobbering the results by having near-identical examples. In these experiments we normalize as above, however we do not perform smoothing.

For this type of data the random spike will always be in the center of the largest magnitude sliding-window, and hence Universal Phasing will phase each time series according to the random location of the spike. In a real world setting, the random spike could be sufficiently wide noise period in the signal, or any irrelevant feature of the time series. This is key to understanding the strength of our method: if it is easy to find a global shifting such that each example is maximally correlated with every other, our method performs identically to Euclidean Distance. On the other hand, when a global shift is not trivial to find, our method succeeds where a Universal Phasing algorithm fails. To illustrate further the performance potential of the kernel we create a second version of the data where we add noise to the labels by flipping the label of each example with probability of 0.1. When the data are completely or nearly separable, both k-NN and SVM should attain close to 100% accuracy. The noise changes the domain to make it harder to get this level of performance.

The results are shown in Tab. 5. As we expected, Universal Phasing does quite poorly in this setting. With no noise, 1-NN with cross-correlation, 1-NN with our kernel, and SVM with our kernel attain almost 100% accuracy. The results with noisy data show that SVM with our kernel is more robust to noise than 1-NN with cross-correlation or our kernel.

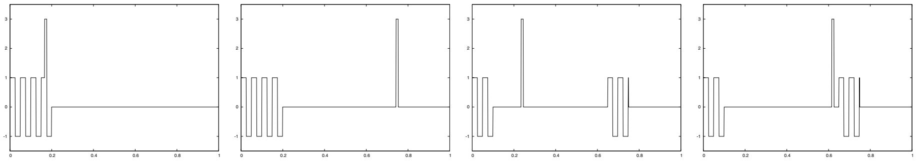

Fig. 2. Examples of artificial data. The left two examples are from class 1, the right two example are from class 2.

Table 5. Results on artificial data

	1-NN CC	1-NN K	1-NN UP	SVM UP	SVM K
Artificial	0.99 ± 0.02	1.00 ± 0.00	0.65 ± 0.04	0.50 ± 0.07	0.997 ± 0.001
Artificial w/ Noise	0.84 ± 0.14	0.84 ± 0.12	0.61 ± 0.09	0.53 ± 0.12	0.90 ± 0.05

6 Conclusion

On the OGLEII data set we have shown a basis for a completely automatic star classification algorithm. We have shown that cross-correlation is an effective similarity measure for phase-invariant time series, we proved that cross-correlation is not positive semidefinite, and we gave a positive semidefinite alternative, justifying its use in an experimental setting.

The work we have presented in the astronomy domain is a portion of our continuing effort to build an “end-to-end” automatic classification system for astronomy events. In particular we have used the work presented here to classify other star surveys such as MACHO. A complete star classification system requires several modules in addition to the classification method we have demonstrated here. For instance, the raw data from a survey contains no information about the star other than its brightness measured at specific times. To classify the star, we must first determine if it is variable, determine if it is periodic, and find its period; only then can we finally classify it. What we have shown in this paper represents the bulk of the classification portion. A manuscript detailing a methodology for the complete task and results for the MACHO catalog is currently in preparation.

As discussed in Sect. 4, using a computationally tractable approximation to the maximum alignment has potential applications in the domain of classifying graphs and other structured data. The main question is whether we can efficiently calculate an exponential weighted approximation to a maximum alignment and whether this would prove useful in an experimental setting.

Acknowledgments

This research was partly supported by NSF grant IIS-080340. The experiments in this paper were performed on the Odyssey cluster supported by the FAS

Research Computing Group at Harvard. We gratefully acknowledge Xiaoyue (Elaine) Wang, Lexiang Ye, Chotirat Ratanamahatana and Eamonn Keogh for creating the UCR Time Series Classification data repository, and for providing us with the shape data sets. We would also like to thank the Harvard Initiative in Innovative Computing for research space and computing facilities.

References

1. Alcock, C., et al.: The MACHO Project - a Search for the Dark Matter in the Milky-Way. In: Soifer, B.T. (ed.) Sky Surveys. Protostars to Protogalaxies. Astronomical Society of the Pacific Conference Series, vol. 43, p. 291 (1993)
2. Udalski, A., Szymanski, M., Kubiak, M., Pietrzynski, G., Wozniak, P., Zebrun, Z.: Optical gravitational lensing experiment. Photometry of the macho-smc-1 microlensing candidate. Acta Astronomica 47(431) (1997)
3. Hodapp, K.W., et al.: Design of the Pan-STARRS telescopes. Astronomische Nachrichten 325, 636–642 (2004)
4. Starr, B.M., et al.: LSST Instrument Concept. In: Tyson, J.A., Wolff, S. (eds.) Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, vol. 4836, pp. 228–239 (2002)
5. Soszynski, I., Udalski, A., Szymanski, M., Kubiak, M., Pietrzynski, G., Wozniak, P., Zebrun, K., Szewczyk, O., Wyrzykowski, L.: The Optical Gravitational Lensing Experiment. Catalog of RR Lyr Stars in the Large Magellanic Cloud. Acta Astronomica 53, 93–116 (2003)
6. Elkan, C.: Using the triangle inequality to accelerate k-means. In: Fawcett, T., Mishra, N. (eds.) International Conference on Machine Learning, pp. 147–153 (2003)
7. Boser, B.E., Guyon, I., Vapnik, V.: A training algorithm for optimal margin classifiers. In: Computational Learning Theory, pp. 144–152 (1992)
8. Moore, A.W.: The anchors hierarchy: Using the triangle inequality to survive high dimensional data. In: Boutilier, C., Goldszmidt, M. (eds.) Uncertainty in Artificial Intelligence, pp. 397–405. Morgan Kaufmann, San Francisco (2000)
9. Schölkopf, B., Smola, A.J.: Learning with Kernels. The MIT Press, Cambridge (2002)
10. Cuturi, M., Vert, J.P., Birkenes, O., Matsui, T.: A kernel for time series based on global alignments. In: IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP) 2007, vol. 2, pp. 413–416 (2007)
11. Gärtner, T., Flach, P.A., Kowalczyk, A., Smola, A.J.: Multi-instance kernels. In: International Conference on Machine Learning, pp. 179–186 (2002)
12. Haussler, D.: Convolution kernels for discrete structures. Technical Report UCSC-CRL-99-10, Department of Computer Science, University of California at Santa Cruz (1999)
13. Protopapas, P., Giammarco, J.M., Faccioli, L., Struble, M.F., Dave, R., Alcock, C.: Finding outlier light curves in catalogues of periodic variable stars. Monthly Notices of the Royal Astronomical Society 369, 677–696 (2006)
14. Faccioli, L., Alcock, C., Cook, K., Prochter, G.E., Protopapas, P., Syphers, D.: Eclipsing Binary Stars in the Large and Small Magellanic Clouds from the MACHO Project: The Sample. Astronomy Journal 134, 1963–1993 (2007)

15. Alcock, C., et al.: The MACHO project LMC variable star inventory. 1: Beat Cepheids-conclusive evidence for the excitation of the second overtone in classical Cepheids. Astronomy Journal 109, 1653 (1995)
16. Geha, M., et al.: Variability-selected Quasars in MACHO Project Magellanic Cloud Fields. Astronomy Journal 125, 1–12 (2003)
17. Howell, D.A., et al.: Gemini Spectroscopy of Supernovae from the Supernova Legacy Survey: Improving High-Redshift Supernova Selection and Classification. Astrophysical Journal 634, 1190–1201 (2005)
18. Debosscher, J., Sarro, L.M., Aerts, C., Cuypers, J., Vandenbussche, B., Garrido, R., Solano, E.: Automated supervised classification of variable stars. i. methodology. Astronomy and Astrophysics 475, 1159–1183 (2007)
19. Vlachos, M., Vagena, Z., Yu, P.S., Athitsos, V.: Rotation invariant indexing of shapes and line drawings. In: Herzog, O., Schek, H.J., Fuhr, N., Chowdhury, A., Teiken, W. (eds.) Conference on Information and Knowledge Management, pp. 131–138. ACM, New York (2005)
20. Ge, X., Smyth, P.: Deformable markov model templates for time-series pattern matching. In: Knowledge Discovery and Data Mining, pp. 81–90 (2000)
21. Berndt, D.J., Clifford, J.: Using dynamic time warping to find patterns in time series. In: Knowledge Discovery and Data Mining, pp. 359–370 (1994)
22. Lu, Z., Leen, T.K., Huang, Y., Erdogmus, D.: A reproducing kernel hilbert space framework for pairwise time series distances. In: Cohen, W.W., McCallum, A., Roweis, S.T. (eds.) International Conference on Machine Learning, pp. 624–631 (2008)
23. Keogh, E.J., Wei, L., Xi, X., Lee, S.H., Vlachos, M.: Lb keogh supports exact indexing of shapes under rotation invariance with arbitrary representations and distance measures. In: Dayal, U., et al. (eds.) International Conference on Very Large Databases, pp. 882–893. ACM, New York (2006)
24. Adamek, T., O'Connor, N.E.: A multiscale representation method for nonrigid shapes with a single closed contour. IEEE Trans. Circuits Syst. Video Techn. 14(5), 742–753 (2004)
25. Luss, R., d'Aspremont, A.: Support vector machine classification with indefinite kernels. In: Platt, J.C., Koller, D., Singer, Y., Roweis, S.T. (eds.) Neural Information Processing Systems. MIT Press, Cambridge (2007)
26. Ong, C.S., Mary, X., Canu, S., Smola, A.J.: Learning with non-positive kernels. In: Brodley, C.E. (ed.) International Conference on Machine Learning (2004)
27. Balcan, M.F., Blum, A., Srebro, N.: A theory of learning with similarity functions. Machine Learning 72(1-2), 89–112 (2008)
28. Shin, K., Kuboyama, T.: A generalization of Haussler's convolution kernel: mapping kernel. In: Cohen, W.W., McCallum, A., Roweis, S.T. (eds.) International Conference on Machine Learning, pp. 944–951 (2008)
29. Gärtner, T., Flach, P., Wrobel, S.: On graph kernels: Hardness results and efficient alternatives. In: Schölkopf, B., Warmuth, M.K. (eds.) COLT/Kernel 2003. LNCS (LNAI), vol. 2777, pp. 129–143. Springer, Heidelberg (2003)
30. Wachman, G., Khardon, R.: Learning from interpretations: a rooted kernel for ordered hypergraphs. In: Ghahramani, Z. (ed.) International Conference on Machine Learning, pp. 943–950 (2007)
31. Fröhlich, H., Wegner, J., Sieker, F., Zell, A.: Optimal assignment kernels for attributed molecular graphs. In: International Conference on Machine Learning, pp. 225–232 (2005)
32. Valiant, L.G.: The complexity of computing the permanent. Theor. Comput. Sci. 8, 189–201 (1979)

33. Papadimitriou, C.H.: Computational Complexity. Addison Wesley, Reading (1993)
34. Gorry, P.A.: General least-squares smoothing and differentiation by the convolution (savitzky-golay) method. Analytical Chemistry 62(6), 570–573 (1990)
35. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines (2001), `http://www.csie.ntu.edu.tw/~cjlin/libsvm`
36. Sonnenburg, S., Rätsch, G., Schäfer, C., Schölkopf, B.: Large scale multiple kernel learning. Journal of Machine Learning Research 7, 1531–1565 (2006)
37. Huang, T.K., Weng, R.C., Lin, C.J.: Generalized bradley-terry models and multiclass probability estimates. Journal of Machine Learning Research 7, 85–115 (2006)
38. Söderkvist, O.J.O.: Computer vision classification of leaves from Swedish trees. Master's thesis, Linköping University, SE-581 83 Linköping, Sweden (September 2001)