LPiTrack: Eye movement pattern recognition algorithm and application to biometric identification

Subhadeep Mukhopadhyay¹ · Shinjini Nandi¹

Received: 3 June 2015 / Accepted: 10 June 2017 / Published online: 29 June 2017
© The Author(s) 2017

Abstract A comprehensive nonparametric statistical learning framework, called LPiTrack, is introduced for large-scale eye-movement pattern discovery. The foundation of our data-compression scheme is based on a new Karhunen–Loève-type representation of the stochastic process in Hilbert space by specially designed orthonormal polynomial expansions. We apply this novel nonlinear transformation-based statistical data-processing algorithm to extract temporal-spatial-static characteristics from eye-movement trajectory data in an automated, robust way for biometric authentication. This is a significant step towards designing a next-generation gaze-based biometric identification system. We elucidate the essential components of our algorithm through data from the second Eye Movements Verification and Identification Competition, organized as a part of the 2014 International Joint Conference on Biometrics.

Keywords LP Transformation · LP comoments · LP moments · Non-Gaussian nonlinear modeling · Nonparametric copula density modeling · Vector autoregression · Eye movement trajectory · Biometric identification

Editor: Cynthia Rudin.

Current research was the winner of the 2014 IEEE International Biometric Eye Movements Verification and Identification Competition.

Electronic supplementary material The online version of this article (<https://doi.org/10.1007/s10994-017-5649-1>) contains supplementary material, which is available to authorized users.

✉ Subhadeep Mukhopadhyay
deep@temple.edu

Shinjini Nandi
shinjini.nandi@temple.edu

¹ Department of Statistical Science, Temple University, Philadelphia, PA 19122, USA

1 Goals and motivation

The study of eye-movement patterns is gaining importance as a modes of scientific inquiry. It has been used as a primary tool for answering important research questions arising in various fields, from marketing (Pieters 2008; Teixeira et al. 2012), psychology (Frank et al. 2014) to neuroscience (Crutcher et al. 2009; Anderson and MacAskill 2013; Pereira et al. 2014) and biometrics (Kasprowski and Ober 2004a). This generated a great amount of interest (in both industry and academia recently) in developing a comprehensive eye-movement pattern recognition technology that can be utilized by researchers from various disciplines. Besides being of immense scientific value, eye-movement signal processing is a challenging statistical learning problem, and we feel it deserves more attention from the data science community.

In particular, the following scientific question motivated this research:

How can we use human eye-movement dynamics to establish the identity of an individual? How can we design such an automated recognition system that is computationally efficient, highly scalable (for large-scale implementation) and has a strong theoretical basis?

Using automated methods to authenticate identity based on an individual's eye movement patterns—which are difficult to imitate and forge, as they capture both physical and behavioral aspects—is a rapidly growing field. The other advantages that an eye-movement-based biometric identifier offer in contrast to classical modalities (like fingerprints, iris, or face recognition) is its greater reliability and robustness, as discussed in Kasprowski and Ober (2004b) and Rigas et al. (2012).

The discovery process of eye-movement-based ‘personalized biomarker’ critically depends on the underlying information processing algorithm that can learn spatio-temporal patterns from the large and complex eye-movement data. This article proposes a *generic* nonparametric statistical modeling scheme called LPiTrack for eye-movement trajectory data analysis and compression. We then apply that general algorithm for automated recognition (to differentiate, identify, or authenticate individuals) of people by mining distinctive eye movement patterns. Our emphasis is on developing systematic (not ad hoc) and efficient (theoretically rigorous and practically viable) data exploration strategies for eye-movement pattern recognition.

The rest of the paper is organized as follows. Section 2 is devoted to the fundamentals of nonparametric signal processing. We outline the concept of LP transform coding, which is at the heart of our compression engine to tackle the trajectory data. In Sect. 3 we explain the core modeling techniques in a systematic manner. A detailed description of various interconnected components of our proposed LPiTrack algorithm is given. We emphasize the science behind this algorithm (why and how it works) along with a real eye-tracking data-based illustration. Section 4 illustrates how LPiTrack incorporates the fundamentals of nonparametric representation learning, an emerging paradigm of modern machine learning. Detailed performance comparisons of LP Machine Learning (MLP) algorithms are reported in Sect. 5 and finally, in Sect. 6, we end with the conclusion and future works.

2 LP transform coding and signal processing

The foundation of our algorithm is based on LP orthonormal representations of stochastic processes in Hilbert space.

Theorem 1 *Let $\{X(t), t \in \mathbb{T}\}$ be a general (discrete or continuous) zero-mean random process with finite second-order moments, defined on the index set $\mathbb{T}$. Then $\{X(t)\}$ can be*

represented as follows (that converges in L_2 sense in the corresponding Hilbert functional space):

$$X(t) = \sum_{j=1}^{\infty} \text{LP}[j; X(t)] T_j[X](t), \quad \text{LP}[j; X(t)] = \mathbb{E}[X(t)T_j(X(t))] \quad (2.1)$$

where $\{T_j[X](t), j \in \mathbb{N}\}$ are the LP polynomial basis functions, especially designed rank-transform based orthogonal function for the underlying process.

Figure 1a shows the shape of LP score polynomials for EMVIC data set (discussed in Sect. 3.1). The name ‘LP’ is derived from the observation that the overall shape of these custom-designed functions resembles Legendre Polynomials (although the properties are very different) as shown in Fig. 1b.

Definition 1 To describe the recipe for constructing LP orthogonal polynomials associated with an arbitrary random variable X , it is helpful to start with the definition of mid-distribution transformation:

$$F^{\text{mid}}(x; X) = F(x; X) - .5p(x; X), \quad p(x; X) = \Pr[X = x], \quad F(x; X) = \Pr[X \leq x]. \quad (2.2)$$

Definition 2 Define $T_j[X](t)$ the j th orthonormal LP score functions by taking Gram Schmidt orthonormalization of powers of $T_1[X](t)$:

$$T_1[X; F](t) = \frac{F^{\text{mid}}(X(t); X) - .5}{\sigma(F^{\text{mid}}(X(t); X))}, \quad (2.3)$$

where $\sigma(X; X)$ denotes the standard deviation of random variable X . By construction, LP bases satisfy the following orthonormality conditions with respect to the measure F :

$$\mathbb{E}[T_j(X; F)] = 0, \quad \mathbb{E}[T_j^2(X; F)] = 1, \quad \text{and} \quad \mathbb{E}[T_j(X; F)T_k(X; F)] = 0 \quad \text{for } j \neq k.$$

Note that for $X(t)$ continuous process we have $T_j[X; F](t) = \text{Leg}_j[F \circ X(t)]$, where Leg_j denotes j th shifted orthonormal Legendre polynomials $\text{Leg}_j(u)$, $0 < u < 1$ (which can be constructed by applying the Gram-Schmidt orthogonalization to the basis $\{1, u, u^2, \dots\}$; see Fig. 1(b)). For given random sample $X_1, \dots, X_n$ we compute ‘empirical’ orthogonal rank polynomials $\{T_j(X; \tilde{F})\}$ with respect to the underlying sample distribution $\tilde{F}(x; X)$.

We would like to emphasize (which differentiates our approach from other traditional discrete signal processing approaches) that our LP representation of the stochastic process (2.1) expresses the given process as a polynomial of $F(X; X)$, the rank transform, or probability integral transform of X , ensuring robustness and stability. Furthermore, we will show (in Sect. 3) that LP-Transformation-based signal processing allows us to extract *simultaneously* the temporal-spatial-static eye-movement patterns using a single unified computational setting, which to the best of our knowledge is the first reported attempt in that direction.

Historical significance The close connection between stochastic processes and orthogonal polynomials has long been recognized. The idea of approximating a stochastic process by a linear combination of orthogonal polynomials with random coefficients was first proposed by Wiener (1938), and is thus famously known as Wiener polynomial chaos and further generalized by Kosambi (1943), Loeve (1945) and Karhunen (1947). Wiener used Hermite polynomials to model Gaussian process. Hermite polynomial-based approximation for Gaussian process achieves the optimal convergence (Cameron–Martin theorem); in fact,

Fig. 1 **a** The *left panel* shows the shape of the first four ‘piecewise-constant’ empirical LP orthogonal polynomials $\{T_j(X; \tilde{F})\}$ for the $\{\Delta Y(t)\}$ process of the eye-movement trajectory for sub id #19 in the EMVIC data set, as discussed in Sect. 3.1 and Fig. 2; **b** *Right panel* the corresponding shapes of the Legendre polynomials, which can be shown to be the LP basis for continuous process (no ties). Note the similarity in the overall shape of our custom-designed polynomials and the Legendre polynomials, which is the motivation for calling them ‘LP’ score polynomials

the rate is exponential. If, however, we have a non-Gaussian process, this representation is non-optimal, and the convergence rate may be substantially slower. A generalized version of the ‘parametric’ family of polynomials known as Askey-scheme of polynomials, used by [Xiu and Karniadakis \(2002\)](#) for stochastic process representation. The word ‘parametric’ is used to stress the fact that selecting the appropriate polynomial from the Askey library requires one to know the underlying distribution of the process. For example, if the underlying process is Poisson then Askey-scheme selects Charlier polynomials, for Gamma process, the choice will be Laguerre polynomials and so on. The main drawback of this approach is that in real-world we have no prior distributional knowledge about the underlying process. The distribution of the process under consideration may not necessarily be a member of the narrow collection of (pre-defined) parametric families of distributions associated with the Askey polynomial scheme. Ideally, we would like to devise a ‘nonparametric’ custom-designed orthonormal polynomials for a given process (as illustrated in Fig. 1), which are easy to compute and guarantee optimal convergence for an arbitrary random variable. To address this problem we have introduced the concept of *LP universal representation*. This concept is based on Hilbert space embedding and the corresponding orthonormal representation result (Theorem 1), which can be thought of as *Karhunen–Loève-type expansion* of the square-integrable random function $X(t)$ (requiring minimum number of orthogonal components or terms in the series).

Presentation style Rather than presenting the theory, methodology and numerical study in separate sections (which is the conventional approach), here we will present them concurrently to introduce the *new* concepts and tools just in time, to reduce the disconnect between theory, computation (R-code) and real application.

3 LPiTrack: theory, algorithm, and application

This section describes the theory and the main steps of the LPiTrack algorithm using a real eye-movement dataset. We will highlight all the modeling challenges and propose how to handle each one of them on a step-by-step basis, and by doing so, explore several modern nonparametric statistical techniques and concepts whose significance goes beyond the scope of the present paper. Each component of our algorithm will be presented along with the associated R-functions, to illustrate how the theory can be implemented in practice. The details on the R-scripts are available in the “Appendix” section.

3.1 Data

We use the dataset that was made available to the participants of the second Eye Movements Verification and Identification Competition (EMVIC) organized as a part of the 2014 International Joint Conference on Biometrics (IJCB). An excellent detailed description of the competition is given in [Kasprowski and Harezlak \(2014\)](#).

Data collection The dataset is composed of 1430 samples recorded for 34 subjects. Every sample consists of eye-movement recordings registered when a person observed an image, more specifically photograph of a face. All measurements were made using a head mounted JazzNovo eye tracker registering eye positions with 1kHz frequency. Every subject looked at 20 to 50 different photographs. The subjects had to observe the photographs and decide whether they knew the face by pressing yes/no button, and were not limited in time, so that *every observation length could be different*. Each such task was recorded as a separate sample. The length of the trajectories varies from 891 to 22,012 ms in the dataset. For comparable

Table 1 EMVIC data structure

Sub. ID	Z	Eye gaze points							
s1	1	x	703.72	702.63	...	−200.82			
		y	−259.22	−261.6	...	−84.94			
s2	0	x	−60.23	−58.61	...	69.38	...	88.58	15.56
		y	−839.59	839.29	...	155.07	...	94.16	−339.91
⋮	⋮			⋮	...	⋮	...	⋮	
⋮	⋮			⋮	...	⋮	...	⋮	
s34	1	x	−582.25	−581.16	...	−52.36	...	274.11	
		y	−1169.1	−1168.5	...	−783.43	...	−575.76	

It is important to note that trajectories recorded for different participants are of different lengths. The lengths vary from 891 to 22,012. x_i denotes the i -th value of the recorded horizontal eye gaze point, y_i the i -th value of the recorded vertical eye gaze point. The values are 0 for points in the middle, positive for points on the right or lower side of the screen and negative for points on the left or upper side of the screen

Fig. 2 The eye-movement trajectories for two subjects are shown: the *left plot* for sub id #19 and *right plot* for sub id #15. The LPiTrack algorithm aims to learn the trajectory patterns to identify the individuals

results, photographs (face images) of different people were used as stimuli. Moreover, every face appearing on the screen was cropped in a way that ensured the same location of eyes for every picture. [Kasprowski and Harezlak \(2014\)](#) referred to this dataset as “one of the most challenging datasets used for eye-movement biometrics so far.”

Data structure For each participant, the discrete random variable I denotes the subject identifier (it can take values between 1 and 34), Z a binary variable specifying the familiarity of the observed image of a face, and a set of horizontal and vertical coordinates (X_i, Y_i) of the eye gaze points. The value $(0, 0)$ is at the middle. Table 1 shows the structure of the data. The modeling task is to predict a correct identification of subjects based on this trajectory data, as shown in Fig. 2.

Remark 1 In the present format (see Table 1) none of the standard machine learning classification tools are directly applicable. The solution lies in efficiently representing these non-standard trajectory data by means of “statistical parameters”—representation learning.

3.2 Existing techniques

The development of smart computational models for eye-movement signal processing is still in its infancy—mostly based on primitive features (Crutcher et al. 2009; Holland and Komogortsev 2013; Kasprowski and Ober 2004b), such as total looking time, total number of fixations (i.e. the total number of fixations that met the ≥ 100 ms criterion), average velocity, average spatial location, average horizontal and vertical velocities, etc. Simola et al. (2008) described temporal techniques based on hidden Markov models usually applied to gaze velocity/acceleration. Rigas et al. (2012) and Cantoni et al. (2015) performed graph-based analysis to capture the spatial layout of the eye movements.

There exist striking disconnects, conceptually and algorithmically, among these various ‘isolated’ naive methods; thus, not surprisingly, despite more than a decade of research, very little is known about *unified statistical algorithms* for designing an automated biometric recognition system. More importantly, we are interested in how can we develop a nonparametric algorithm that can systematically learn the representations of eye-movement trajectories. In what follows, we address this intellectual challenge via LP signal processing.

3.3 Transformation

Drawing upon the fundamental stochastic representation (2.1), we begin our modeling process by taking LP-transformation, which converts (nonlinearly) the given time series $X(t)$ to multiple robust time series $\text{Vec}(T[X])(t) = \{T_1[X](t), \dots, T_m[X](t)\}$. The ‘robust’ aspect stems from the fact that each of the time series components $T_j[X; F](t)$ is polynomials of *rank-transform* of $X(t)$ —a unique aspect of our approach. The number of canonical components m in the LP-Transformation depends on the magnitude of the LP-fourier coefficients $\text{LP}[j; X(t)]$ in the expansion (2.1). One may choose m to be the minimum number (to balance efficiency and parsimony) satisfying $\sum_{j=1}^m |\text{LP}[j; X(t)]|^2 > 0.9$, i.e., the cumulative energy content (or variability) of a signal is at least 90%.

It has been empirically observed that LP-Transformation also helps to reduce the inherent ‘excess’ non-stationarity (we have used Augmented Dickey-Fuller t-statistic) of a time series, which proved to be specially effective for the tackling the series $[\Delta X(t), \Delta Y(t)]$. This stationarity promoting aspect could be an extremely useful property that can be exploited for general time series modeling. In summary, LP Transformation provides protection from excess non-stationarity, ensures robustness and creates nonlinear functions to uncover the underlying complex temporal dependency.

The R-function `LPTrans(x, m)` converts the input time series $X(t)$ into order m LP-Transformed (LPT) series. The associate R-code of this function is provided in the “Appendix” section.

3.4 Temporal modeling

We start by developing tools that capture the temporal structure of the eye movement trajectories. The challenge arises from the fact that most of these series are extremely non-Gaussian and non-linear, which makes it difficult (if not inappropriate) for traditional time series techniques (based on Gaussian and linearity assumptions) to be useful for this type of dataset.

LP Diagnostics Figure 3 shows the distribution of the process $(\Delta X(t), \Delta Y(t))$ for an eye-trajectory of the sub id #19. The histograms strongly indicate the non-Gaussian nature, suggesting Hermite polynomials based the Wiener expansion is not ‘optimal’. The correlo-

Fig. 3 Evidence of non-Gaussian distribution and non-linearity of $(\Delta X(t), \Delta Y(t))$ process is shown for the eye-trajectory of sub id #19 as depicted in Fig. 2a. Histograms are shown in the *first* row, which indicates the distribution is highly skewed and long-tailed; Correlogram of the LP-transformed series are shown in the next *two* rows. LP-correlogram provides quick diagnostic to check the presence of nonlinear autocorrelation

gram of $T_j[\Delta X](t)$ and $T_j[\Delta Y](t)$ for $j = 1, 2$ as shown in the Fig. 3 finds a prominent non-linear autocorrelation pattern. Thereby we have a situation where classical time series modeling based on Gaussian and linearity assumptions miserably fails. The question remains how to model this kind of *non-standard* time series?

As $X(t)$ can be characterized (and orthogonally decomposed) by the LP-Transformed series (Eq. 2.1), it is natural to jointly model these new multivariate times series $\text{Vec}(T[X])(t) = \{T_1[X](t), \dots, T_m[X](t)\}$ to capture the temporal information of the original series. Based on this intuition, we describe an approach for analyzing the dynamics of possibly nonlinear time series that entails LP-transformation.

Definition 3 LPTime procedure models nonlinear process by specifying ℓ th order vector autoregressive (VAR) of the following form:

$$\text{Vec}(T[X])(t) = \sum_{k=1}^{\ell} A(k; \ell) \text{Vec}(T[X])(t - k) + \epsilon(t), \quad (3.1)$$

where $A(k; \ell)$ are $(m \times m)$ time-invariant coefficient matrices and $\epsilon(t)$ is multivariate centered Gaussian white noise with covariance Σ_ϵ .

This system of equations *jointly* describe the *dynamics of the multivariate nonlinear process* and how it evolves over time. We apply LPTIME individually to all the three series (1) $\{r(t), \Delta r(t)\}$; (2) $\{\Delta X(t), \Delta Y(t)\}$ and (3) $\{\Delta^2(X(t), \Delta^2 Y(t)\}$. The estimated autoregressive coefficients capture the temporal pattern. Hence, LPTIME non-Gaussian time series modeling automatically produces parameters (time-invariant coefficient matrices: $A(k; \ell)$) that can be used as a ‘temporal signature’ for the subsequent supervised learning.

The R-function `LPTIME(z, m, l)` fits the ℓ -th order lagged LP-multivariate autoregressive model (3.1) for the input time series $Z(t)$ (where $Z(t)$ could be multivariate e.g., $\{\Delta X(t), \Delta Y(t)\}$) and returns the estimated time-invariant coefficient matrices. The associated R-code can be found in the “Appendix” section.

3.5 Spatial fixation pattern

In this section, we provide a modern nonparametric procedure to extract the spatial domain characteristics (thus we will suppress the time dimension ‘ t ’ for simplicity of presentation) from eye-tracking data. We capture the eye-fixation or clustering patterns (Fig. 4) using *copula density* that compactly represent the spatial dependence between the (X, Y) co-ordinates without the influence of marginal distributions in a scale invariant manner.

We first introduce the concept of the “normed joint density”, pioneered by Hoeffding (1940), as the joint density divided by product of the marginal densities. We denote this by “dep” to emphasize that it is a measure of dependence (and independence):

Fig. 4 The spatial distribution of the X and Y co-ordinates for a selected eye-trajectory of sub id #19. The eye fixation, or the ‘rest’ times (to inspect the minute details in the shown photograph) are denoted by the dark gray spot (high-density region), resulted in considerable ties in the (X, Y) data. The underlying fixation map density is characterized by the LP-comoments Eq. (3.7)

$$\text{dep}(x, y; X, Y) = f(x, y; X, Y)/f(x; X)f(y; Y). \quad (3.2)$$

Definition 4 For both (X, Y) either discrete or continuous, define the copula density function as

$$\text{cop}(u, v; X, Y) = \text{dep}(Q(u; X), Q(v; Y); X, Y), \quad 0 < u, v < 1, \quad (3.3)$$

where $Q(u; X)$ and $Q(v; Y)$ are the quantile function of the random variables X and Y respectively. When X and Y are jointly continuous (no ties are present), Copula density function is the joint density of rank transform variables $U = F(X; X)$, $V = F(Y; Y)$ with joint distribution function $F(u, v; U, V) = F(Q(u; X), Q(v; Y); X, Y)$, denoted by $\text{Cop}(u, v; X, Y)$ and called Copula (connection) function, pioneered in 1958 by Sklar ([Schweizer and Sklar 1958](#); [Sklar 1996](#)).

We seek to estimate the copula density to model the spatial fixation density map. There is a vast literature on parametric families of copulas ([Nelsen 2006](#)), which is inadequate for our purpose as we would like to allow arbitrary dependence of the underlying spatial random process $Z(x, y)$. On the other hand, developing a more flexible nonparametric copula estimation algorithm (under the scenario where ties are present, as demonstrated in Fig. 4) is known to be challenging. To address this statistical modeling problem we introduce a *new* estimation scheme based on a novel decomposition result (3.4) of copula density using LP-basis functions. It allows us to characterize the underlying fixation copula density using few polynomial coefficients known as LP-comoments. These nonparametric ‘spatial signatures’ can then be used for subsequent predictive learning stage.

The following LP representation of copula density plays a fundamental role in the second stage of our LPiTrack algorithm.

Theorem 2 *Square integrable copula density admits the following representation as infinite series of LP product score functions*

$$\text{cop}(u, v; X, Y) - 1 = \sum_{j, k > 0} \text{LP}(j, k; X, Y) T_j[Q(u; X); X] T_k[Q(v; Y); Y], \quad 0 < u, v < 1, \quad (3.4)$$

where $Q(u; \cdot)$ denotes the quantile function. Note that the LP copula expansion equivalently entails

$$\text{LP}[j, k; X, Y] = \iint_{[0, 1]^2} \text{cop}(u, v; X, Y) T_j[Q(u; X); X] T_k[Q(v; Y); Y] du dv. \quad (3.5)$$

Definition 5 Define LP-comoments spatial matrix by the orthogonal copula expansion coefficients $\text{LP}[j, k; X, Y]$, which can be expressed as cross-covariance between higher-order orthonormal LP score functions $T_j(X; X)$ and $T_k(Y; Y)$,

$$\text{LP}[j, k; X, Y] = \mathbb{E}[T_j(X; X) T_k(Y; Y)] \quad \text{for } j, k > 0. \quad (3.6)$$

For data analysis, we select finite numbers of $\text{LP}[j, k; X, Y]$ for $1 \leq j, k \leq m$ to characterize the underlying ‘smooth’ spatial pattern. We use LP-comoments for $(r, \Delta r)$, (X, Y) and $(\Delta X, \Delta Y)$ to construct ‘spatial signatures’ in our LPiTrack algorithm.

Sample LP-comoment matrix that characterizes the spatial layout of eye movements shown in Fig. 4 is given by

$$\widehat{\text{LP}}[X, Y; m = 4] = \begin{bmatrix} 0.686 & 0.046 & 0.168 & 0.069 \\ -0.149 & 0.626 & 0.359 & -0.231 \\ 0.069 & 0.171 & 0.099 & 0.403 \\ 0.185 & -0.134 & 0.265 & 0.110 \end{bmatrix} \quad (3.7)$$

The R-function `LP.comoment(x, y, m)` computes the LP-comoments (3.6) that capture the underlying nonparametric copula-based spatial fixation pattern for the input series $(X(t), Y(t))$. The output of this function is the LP-comoment matrix as shown in the aforementioned example (3.7). The required R-scripts are given in the “Appendix” section.

3.6 Static shape detector

Figure 5 shows the histograms of the X and Y coordinates for two different eye movement trajectories. It is clear that the (static) shape of the underlying distributions of coordinates has a significant predictive value for biometric classification. Given $X_1, X_2, \dots, X_n$ a random sample from unknown F , our goal here is to develop nonparametric tools that can summarize and uniquely characterize the *shape of the distribution* in a robust way.

We do not recommend to compute sample moments for three (well-known) reasons: First higher-moments may not exist due to the heavy-tailed nature of the underlying distributions

Fig. 5 The shape of the histogram of the X and Y coordinates of the eye-gaze shown in Fig. 2. Important discriminatory information is hidden in the shape of the distributions

Table 2 The LP-moments for the Y co-ordinate distribution for Subject id #19 and #15 shown in bottom panel of Fig. 5. LP-moments captures various shapes of distribution

Trajectory	LP[1; Y]	LP[2; Y]	LP[3; Y]	LP[4; Y]	LP[5; Y]	LP[6; Y]
Sub id #19	0.894	−0.302	0.19	−0.124	0.132	0.035
Sub id #15	0.847	0.309	0.362	0.222	0.035	0.063

Fig. 6 Flowchart of LP-representation based learning algorithm

such as shown in Fig. 5; second, moments are highly susceptible to extreme observations, not a robust signature. Third and probably the most fundamental reason, moments do not *uniquely* characterize (Heyde 1963) the probability distribution. To overcome these difficulties, we introduce the concept of LP-moments, which accurately store ‘shape’ information of a probability distribution in a way that avoids all of the aforementioned problems.

Definition 6 For a random variable X define LP moments by

$$\text{LP}[j; X] \equiv \text{LP}(j, 0; X, X) = \text{Cor}[X, T_j(X; X)], \text{ for } j > 0, \quad (3.8)$$

which are the “coordinates” of the random variable X in the LP Hilbert functional space. We define $\text{LP}[0; X] = \mathbb{E}[X]$.

LP moments contain a wealth of information about the underlying probability law. $\text{LP}[1; X]$ can be shown to be a measure of scale that is related to the Downton’s estimator (Downton 1966) and can be interpreted as a modified *Gini mean difference coefficient*. Likewise, $\text{LP}(2; X)$ and $\text{LP}(3; X)$ are measures of skewness and kurtosis. The key point here is to note that LP moments can be used for the purpose of nonparametric identification and characterization of probability law for X discrete (ties present) or continuous.

For the EMVIC eye-movement experiment, we compute LP moments for the series $r(t)$, $X(t)$, $Y(t)$ and their first and second differences. These LP-moments (as shown in Table 2) will act as a robust static ‘shape’ detector for biometric classification.

The R-function `LP.moment(y, m)` computes the LP-moments (3.8) of the random variable Y . The output of this function is a vector of LP-moments as shown in the aforementioned example in Table 2. The associated R-code is provided in the “Appendix” section.

4 LPiTrack learning algorithm

Our algorithm proceeds by representing the data in new functional space (whose basis functions are *data-adaptive not fixed*) for efficient (easier and faster) learning, as shown in Fig. 6. This is close in spirit to the philosophy proposed in Mukhopadhyay et al. (2012) and Parzen and Mukhopadhyay (2013), where they examined the modeling question: “What comes first—a parametric model or sufficient statistics?”

Remark 2 As a result of recent unprecedented technological advances for eye-tracking devices (such as google glasses¹ and smartphones) the size of the recorded datasets have

¹ <http://www.scientificamerican.com/article/eye-tracking-in-google-glass-a-window-into-the-soul/>.

Algorithm 1 LPiTrack Learning Algorithm

1. LP-Transformation: Compute mid-rank based nonlinear orthogonal polynomial transformation of the original input series (here $X(t)$ can be a multivariate process) as follows

$$X(t) \mapsto \text{Vec}(T[X])(t) = \{T_1[X](t), \dots, T_m[X](t)\},$$

where $T_j[X] := T_j[X; \tilde{F}]$ are the empirical orthonormal basis of the LP-Hilbert space of the underlying (possibly non-Gaussian) process $X(t)$.

2. Temporal Modeling: To extract the joint (horizontal and vertical coordinate movements together) nonlinear dynamics of eye-movement pattern we estimate the vector autoregressive coefficients from the following model

$$\text{Vec}(T[X])(t) = \sum_{k=1}^l A(k; l) \text{Vec}(T[X])(t-k) + \epsilon(t).$$

3. Spatial Modeling: Compute the LP-comoments by

$$\text{LP}[j, k; X, Y] = \mathbb{E}[T_j(X; X) T_k(Y; Y)],$$

which is a generalized covariance matrix that can capture the non-linear spatial dependence via nonparametric LP copula modeling as described in Eq. (3.4).

4. Static Distribution Modeling: Compute LP-moments given by

$$\text{LP}[j; X] \equiv \text{LP}(j, 0; X, X) = \text{Cor}[X, T_j(X; X)],$$

which characterize uniquely the shape of the probability distribution of the random variable X . LP-moments are applicable for X discrete (with ties, which happens to be the case for EMVIC data) or continuous.

5. Combine and return the LP-feature representation for the input eye-movement trajectory that can now be used for subsequent prediction modeling (classification or recognition task in our case).

become truly massive. Our LPiTrack can tackle this emerging big eye-movement datasets as our learning strategy allows *data-parallelism* (applying computation methods *independently* to each data item of a massive number of trajectories). This allows LPiTrack to leverage the power of distributed and parallel computing architecture (Dean and Ghemawat 2008), which makes it more scalable for large-scale studies. Our algorithm performs ‘LP nonparametric representation learning’ to systematically construct three classes of features (temporal, spatial and static) from the given eye-movement trajectories at each node terminal. In that case Algorithm 1 (which yields the ℓ^2 LP representation of L^2 trajectory) will act as a ‘Map’ function in the framework of MapReduce.

5 LP machine learning: performance comparison

It is important to recall that the non-standard structure of the eye-movement data, as described in Table 1, prevented us from directly applying machine learning tools at the initial stage. We outlined a nonparametric strategy to represent the given trajectory data using LP signal processing tools. Our learning algorithm utilizes the LP-transform coefficients (Algorithm 1) as the input to classification models—LP Machine Learning.

The problem of human identification for the EMVIC data is a classification task with 34 classes. We compared the performance of several state-of-the-art classification algorithms

such as multinomial logit regression, random forest (Breiman 2001), gradient boosting (Freund and Schapire 1997; Friedman 2001), naive Bayes, and support vector machine (Cortes and Vapnik 1995; Wahba 1999) on the LP-transform of the eye-movement trajectories. For the SVM classifier we have used a radial basis function kernel. By transforming the eye-trajectory signal into a new generalized frequency domain, LPiTrack algorithm generates a large number of features in an automated way. To build a sparse model (with the goal of achieving further compression in the transform domain) based on specially-designed high-dimensional LP-features, model selection is must to ensure better predictive performance. We used ℓ_1 -regularization or the Lasso (Least absolute selection and shrinkage operator) penalty (Tibshirani 1996) for multinomial logistic regression with 34 classes. 10-fold cross-validation was used for selecting the tuning regularization parameter. The `gbm` R package was used for implementing gradient boosting with the following choice of tuning parameters (which gives the best performance): `n.trees= 250`, `cv.folds= 10`, `depth= 5`, `bag.fraction= 0.7`.

Which type of LP features were most useful? The classification was performed separately on (i) temporal LP features; (ii) spatial LP-comoments; (iii) static LP-shape detectors; and (iv) combined to quantify how much of an improvement each piece yields individually. For each one of these cases, we randomly sampled 25% of the data to form a test set. This entire process was repeated 100 times, and test set accuracy rates (% times algorithm correctly identifies the individual based on eye-movement trajectory data) are shown in Fig. 7, which is unquestionably a promising result for biometric identification. For EMVIC data, the LP-static shape information turns out to be very effective for biometric identification. However, this may not always be the case for other general eye-tracking data, as noted in Kasprowski and Harezlak (2014). *Our LPiTrack algorithm is capable of automatically extracting and fusing information from spatial, temporal and static domains in a data-adaptive manner so that data scientists do not have to guess which parts of the data contain the most information.* LP-logistic regression is clearly the best-performing classifier. LP-random forest and LP-SVM show similar performances overall, and emerged as the second-best algorithm. For the combined case, we also noted which variables were selected at each run by ℓ_1 -logistic regression model. We found 48% of these features are temporal, roughly 33% of these features are static and 19% are spatial features. This further validates the importance of static and temporal features in identification of the subjects for EMVIC experiment.

A fixed hold-out (unlabeled) test set representing 41% of the original data was retained for evaluation in the official 2014 Eye Movement Verification and Identification contest, organized by the IEEE Biometrics Council. The goal was to predict the subject identifiers for unlabeled test data. The evaluation metric to determine the top 3 winners was based on the number of test samples classified correctly among 82 competing algorithms. Our proposed LPiTrack² algorithm was able to correctly identify 81.5% of the individuals in the test sample. The accuracy of the other two winning algorithms were, respectively, 82.3 and 86.4%³—overall a performance quite comparable with our model.

² The current version is more enhanced and refined compared to the one we submitted in the contest.

³ It utilizes multivariate Wald–Wolfowitz type runs test whose computational complexity is $O(n^3)$ and requires one to store *all the data* in-memory, which is impractical for large datasets. On the contrary, our approach represents the large trajectories by few LP-signatures. LP compressive signal processing allows fast data-processing by eliminating storage cost.

Fig. 7 The performance of the LPiTrack algorithm on the EMVIC dataset for multi-class ($k = 34$ classes) classification. Lasso-logistic regression, *RF* random forest, *SVM* support vector machine, *NB* naive Bayes, and *GB* gradient boosting classification models are applied to the LP-representation of eye-movement trajectories. The vertical axis denotes the % correct classification or accuracy based on 100 random training and test-data partitions

6 Conclusion and future work

This article develops a new data-driven orthogonal polynomials (LP-orthonormal basis) based transform coding scheme for big trajectory data compression and characterization, which is suitable for pattern recognition. At the heart of our approach lies a *new stochastic representation* of eye-movement signal using LP-orthonormal basis. This new expansion allows one to represent the trajectory (L^2 function) in terms of a finite number of LP-coefficients in a way that captures most of the ‘patterns’ (temporal-spatial-static) of the original signal, thus leading to efficient storage and fast processing. We outlined the theory, algorithm, and a real application to illustrate the proposed LP Statistical Signal Processing framework, and established its utility for designing a high-quality classifier for biometric identification. Our top performing eye-movement pattern recognition algorithm, LPiTrack, has a strong theoretical foundation, which extends the work of [Mukhopadhyay and Parzen \(2014\)](#) and [Mukhopadhyay and](#)

Parzen (2016) on ‘Nonparametric Data Science.’ In conclusion, our EMVIC data analysis shows that there is a genuine promise of using eye-movement characteristics as a biometric feature. These ‘personalized unique’ eye-movement patterns can be exploited for other data discovery purposes, such as early detection of Alzheimer’s disease or measuring the effectiveness of TV commercials, and many others. Our algorithm is implemented in the R software package `LPTIME` (Mukhopadhyay and Nandi 2015) that researchers from different fields can routinely use for mining eye-movement data.

It is worth pointing out that our proposed `LPiTrack` technology is also applicable for analyzing other types of trajectory data. One interesting application domain that we would like to explore in the future is insurance telematics (Handel et al. 2014), which aims to analyze the driving trajectory data (say, data collected using personal mobile devices) to decide the premium based on pay-as-you-drive (PAYD) scheme. The second line of future work will extend the proposed 2D learning scheme to a 3D setting that has widespread applications in motion trajectory recognition systems (e.g., robotics, virtual reality, medical research, computer games, and sign-language learning).

The proposed algorithm is implemented in the R package `LPTIME`, which is available on CRAN. We also provide the EMVIC eye-movement dataset as a supplementary material.

Acknowledgements The authors would like to extend special thanks to Pawel Kasproski for organizing and coordinating the EMVIC-2014 data challenge, and to Richard Heiberger for providing us with many valuable computational tips to make `LPiTrack` a powerful, fast algorithm. We also thank the MLJ action editor and anonymous reviewers for their valuable comments.

Appendix: `LPiTrack` R-scripts

In this note we will describe the `LPiTrack` data processing scheme and its R-implementation in a step-by-step basis.

• **Data pre-processing** The EMVIC eye-tracking data (discussed Sect. 3.1 and available as supplementary material) is available in the format of trajectory $(x_1, y_1, x_2, y_2, \dots, x_n, y_n)$, where (x_i, y_i) is the coordinate of the horizontal and vertical gaze points. The following R-function `Find.xy()` converts each trajectory to a matrix with first column corresponds to x coordinates and second column corresponds to y coordinates

```
Find.xy <- function(z){
  matrix(z, byrow =TRUE, ncol =2)
}
```

Argument: z is any vector representing a trajectory with $z_i = (x_i, y_i)$.

Return value: A matrix of order $n \times 2$ is returned whose i th row is (x_i, y_i) .

• **Transformation** The following function performs the LP-Transformation

$$Y(t) \mapsto \{T_1[Y](t), \dots, T_m[Y](t)\},$$

as explained in Sect. 3.2.

```
LPTrans <- function(y,m){
  u <- (rank(y,ties.method = c("average")) - .5)/length(y)
  m <- min(length(unique(u))-1, m)
  S.mat <- as.matrix(poly(u,df=m))
  return(as.matrix(scale(S.mat)))
}
```


Arguments: y is a input time series whose LP-Transformation is required.

m specifies the number of orthonormal transformations required.

Return value: A matrix of order $n \times m$ is returned. The j th column corresponds to the j th order orthonormal LP score function $T_j[Y](t)$.

• **Temporal modeling** `LPTime()` function implements temporal modeling of nonlinear non-Gaussian process as described in Sect. 3.3. It fits Vector Autoregressive (VAR) on the LP-transformed time series with a possibility of having an exogenous variables. The VAR model is fitted using a function `VAR()`, which is similar to a function of same name in the R-package `vars`. The structure of the function and its arguments are explained below.

```
LPTime <- function(z, exo=NULL, m, lag.order){
  Z.mat <- matrix(apply(z, 2, LPTrans, m), nrow = nrow(z))
  if(!is.null(exo))
    E.mat <- matrix( apply(as.matrix(exo), 2, LPTrans,m),
                    nrow = nrow(as.matrix(exo)) )
  else
    E.mat <- NULL
  VAR( Z.mat, exogen = E.mat, lag.order)
}
```

Arguments: z : Vector of trajectory coordinates on which temporal modeling is to be done.

exo : Exogen if required for the VAR model; default is NULL.

m : User supplied value denoting the number of required orthonormal LP-transformation for the given trajectory.

$lag.order$: An integer specifying the lag order.

Return value: A vector of estimated coefficients obtained from LP-VAR temporal modeling as described in Eq. (3.1).

• **Spatial fixation pattern** This algorithm captures spatial domain characteristics using copula density. The following function computes the LP-comoments or LP-covariance between (X, Y) , which are L^2 -Fourier coefficients of the orthogonal copula expansion based on product LP-basis functions, as describe in Sect. 3.4.

```
LP.comoment <- function(x, y, m){
  Tx <- LPTrans(x ,m)
  Ty <- LPTrans(y ,m)
  lx <- cor(x,Ty)
  ly <- cor(y,Tx)
  LPcom <- c( lx, ly, cov(Tx, Ty) )
}
```

Arguments: x : The x coordinates of a trajectory.

y : The y coordinates of a trajectory.

m : The order of the LP co-moment matrix.

Return values: vector of LP-comoments.

• **Static shape detector** The following function computes LP-moments of a random variable X . LP-moments provides a robust characterization of the shape of the probability distribution for X discrete or continuous as described in Sect. 3.5.

```

LP.moment <- function(x, m){
  Tx <- LPTrans(x,m)
  LPmom <- c(mean(x), cor(x,Tx))
}

```

Argument(s): x : a numeric data vector.

m : The number of LP moments to be found.

Return value: Vector of LP moments.

References

- Anderson, T. J., & MacAskill, M. R. (2013). Eye movements in patients with neurodegenerative disorders. *Nature Reviews Neurology*, 9(2), 74–85.
- Breiman, L. (2001). Random forests. *Machine Learning*, 42(1), 5–32.
- Cantoni, V., Galdi, C., Nappi, M., Porta, M., & Riccio, D. (2015). Gant: Gaze analysis technique for human identification. *Pattern Recognition*, 48(4), 1027–1038.
- Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20(3), 273–297.
- Crutcher, M. D., Calhoun-Haney, R., Manzanares, C. M., Lah, J. J., Levey, A. I., & Zola, S. M. (2009). Eye tracking during a visual paired comparison task as a predictor of early dementia. *American journal of Alzheimer's Disease and Other Dementias*, 24(3), 258–266.
- Dean, J., & Ghemawat, S. (2008). Mapreduce: Simplified data processing on large clusters. *Communications of the ACM*, 51(1), 107–113.
- Downton, F. (1966). Linear estimates with polynomial coefficients. *Biometrika*, 24(3), 129–141.
- Frank, M. C., Amso, D., & Johnson, S. P. (2014). Visual search and attention to faces in early infancy. *Journal of Experimental Child Psychology*, 118, 13–26.
- Freund, Y., & Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1), 119–139.
- Friedman, J.H. (2001). Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29(5), 1189–1232.
- Handel, P., Skog, I., Wahlstrom, J., Bonawiede, F., Welch, R., Ohlsson, J., et al. (2014). Insurance telematics: Opportunities and challenges with the smartphone solution. *IEEE Intelligent Transportation Systems Magazine*, 6(4), 57–70.
- Heyde, C.C. (1963). On a property of the lognormal distribution. *Journal of the Royal Statistical Society, Series B*, 25, 392–393.
- Hoeffding, W. (1940). Massstabinvariante korrelationstheorie. *Schriften des Mathematischen Seminars und des Instituts für Angewandte Mathematik der Universität Berlin*, 5(3), 179–233.
- Holland, C.D. & Komogortsev, O. V. (2013). Complex eye movement pattern biometrics: Analyzing fixations and saccades. In *International Conference on Biometrics (ICB)*, 2013 . IEEE (pp. 1–8).
- Karhunen, K. (1947). Über linear methoden in der wahrscheinlichkeitsrechnung. *Annales Academiae Scientiarum Fennicae*, 37, 1–79.
- Kasprowski, P., & Harezlak, K. (2014). The second eye movements verification and identification competition. *IEEE International Joint Conference on Biometrics*, 2014, 1–6.
- Kasprowski, P. & Ober, J. (2004a). Eye movements in biometrics. In *Biometric authentication*. Springer (pp. 248–258).
- Kasprowski, P. & Ober, J. (2004b). Eye movements in biometrics. In *Biometric authentication*. Springer (pp. 248–258).
- Kosambi, D. (1943). Statistics in function space. *Journal of Indian Mathematical Society*, 7(1), 76–88.
- Loeve, M. (1945). Fonctions aléatoires du second ordre. *Comptes Rendus Academy of Science Paris*, 220, 469.
- Mukhopadhyay, S. & Nandi, S. (2015). *LPTIME: LP nonparametric approach to non-Gaussian non-linear time series modelling*. R package version 1.0-2, <http://CRAN.R-project.org/package=LPTIME>.
- Mukhopadhyay, S. & Parzen, E. (2014). *LP approach to statistical modeling*. Preprint. [arXiv:1405.2601](https://arxiv.org/abs/1405.2601).
- Mukhopadhyay, S. & Parzen, E. (2016). *Nonlinear time series modeling by LPTIME, nonparametric empirical learning*. [arXiv:1308.0642](https://arxiv.org/abs/1308.0642).
- Mukhopadhyay, S., Parzen, E., & Lahiri, S. N. (2012). From data to constraints. Bayesian inference and maximum entropy methods. In *31st International Workshop on Science and Engineering, Waterloo, Canada*, 1443, 32–39.
- Nelsen, R. B. (2006). *An introduction to copulas* (2nd ed., Vol. 139). Berlin: Springer.

- Parzen, E. & Mukhopadhyay, S. (2013). United statistical algorithms, LP comoment, copula density, nonparametric modeling. In *59th ISI World Statistics Congress (WSC), Hong Kong*.
- Pereira, M. L. F., Marina von Zuben, A. C., Aprahamian, I., & Forlenza, O. V. (2014). Eye movement analysis and cognitive processing: Detecting indicators of conversion to alzheimer's disease. *Neuropsychiatric Disease and Treatment*, 10, 1273.
- Pieters, R. (2008). A review of eye-tracking research in marketing. *Review of Marketing Research*, 4, 123–147.
- RIGAS, I., Economou, G., & Fotopoulos, S. (2012). Human eye movements as a trait for biometrical identification. In *IEEE 5th international conference on biometrics: Theory, applications and systems (BTAS), 2012*. IEEE (pp. 217–222).
- Schweizer, B., & Sklar, A. (1958). Espaces métriques aléatoires. *Comptes Rendus Acadamy of Science Paris*, 247, 2092–2094.
- Simola, J., Salojärvi, J., & Kojo, I. (2008). Using hidden markov model to uncover processing states from eye movements in information search tasks. *Cognitive Systems Research*, 9(4), 237–251.
- Sklar, A. (1996). Random variables, distribution functions, and copulas : A personal look backward and forward. *IMS Lecture Notes Monograph Series, Institute of Mathematical Statistics (Hayward)* (pp. 1–14).
- Teixeira, T., Wedel, M., & Pieters, R. (2012). Emotion-induced engagement in internet video advertisements. *Journal of Marketing Research*, 49(2), 144–159.
- Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of Royal Statistical Society: Series B*, 58, 267–288.
- Wahba, G. (1999). Support vector machines, reproducing kernel hilbert spaces and the randomized gacv. *Advances in Kernel Methods-Support Vector Learning*, 6, 69–87.
- Wiener, N. (1938). The homogeneous chaos. *American Journal of Mathematics*, 60, 897–936.
- Xiu, D., & Karniadakis, G. E. (2002). The wiener-askey polynomial chaos for stochastic differential equations. *SIAM Journal on Scientific Computing*, 24(2), 619–644.