Improved Gaussian process classification via feature space rotation

Liang Wang a,*, Christopher A. Leckie b

a National Lab of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, 95 Zhongguancun East Bd., Haidian District, Beijing 100190, PR China
b Department of Computer Science and Software Engineering, The University of Melbourne, Victoria 3010, Australia

1. Introduction

An important step in pattern classification is to properly define and choose the feature space. The extracted feature data can usually be regarded as points in a high dimensional space. Currently, there are no principled rules on how to choose the optimal features. The given feature data may not fit the assumption of a specific classification model. In order to obtain superior performance, it is generally necessary to adjust data representation according to the actual characteristics of a chosen classification model. It should be noted that in this paper we specifically focus on the Gaussian process (GP) classification model [1,9,6], which is a state-of-the-art supervised learning approach. A GP model is completely specified by its kernel function. In our context, we adopt the commonly used isotropic kernel (i.e., rotation invariant), whose advantage is simple and efficient. Hence, it is necessary to lessen the possible anisotropy that exists among the features so as to achieve better classification performance [18]. Naturally, an anisotropic kernel can also be used [25], but the complexity will substantially increase.

In general, there exist two major categories of approaches for adjusting the feature data to achieve better supervised learning results, by either selecting part of the feature data, e.g., [8,17,13,19,20,27] (i.e., feature selection), or applying a transformation to the feature data, e.g., [15,14,12,18,26,28,29] (i.e., feature transformation). Several examples of feature selection methods include Automatic Relevance Determination (ARD) [8], data partitioning [13,17] and spectral feature selection [19]. In [8], ARD was used to assign each feature a weight so that only those features that are most relevant to the prediction outputs were selected. This was carried out by making the values of the hyperparameters associated with each input adapted to the training data. The hyperparameters were estimated by optimizing the model marginal likelihood. Gramacy [17] combined the stationary GP and linear models with tree partitioning for non-stationary modeling. This combination yielded a more efficient spatial model. Kim et al. [13] partitioned the sharp changing spatial data into disjoint stationary regions where the cross region data values were assumed to be independent. Modeling within regions was handled by a full Bayesian framework. Alternatively, Zhou et al. [19,20] carried out feature selection and data partitioning by spectral analysis. Anisotropy of the data was lessened by choosing only the spectrally homogeneous features.

Examples of transformations applied to the feature data include the widely used Principal Component Analysis (PCA) [12,2], as well as deformation in the input [14,18] and output space [15]. PCA is one of the most popular tools in modern data analysis, which projects the data along the directions where the data vary the most. PCA can be used for dimensionality reduction by retaining the lower-order components of the data that contribute most to its variance, as the lower order components of the data are regarded as containing the “most important” content, although this is not always the case. As for transforms specifically
focusing on the anisotropy of the data, Schmidt et al. [14] applied an interpolation that maps the original input space to a new isotropic one. The inference of the mapping function was performed by setting a Gaussian process prior to the mapping function and Monte Carlo Markov Chain (MCMC) methods were used to obtain samples from the posterior distribution. Snelson et al. [15] proposed a warped GP applied to a non-Gaussian data output space by learning a nonlinear transformation of the GP outputs. Zhou and Suter [18] rescaled the input space to improve data isotropy by means of frequency domain analysis. A rescaling of our feature space rotation scheme are described in Section 3. In better modeling can be achieved with the isotropic GP kernel. Mainly, the consistency of the feature data along each dimension so that the alignment of the coordinate system. Instead of applying sophisticated transforms to the feature data itself, the distribution of the data can be equivalently adjusted by simply rotating the coordinate system. In this paper, we are specifically interested in improving the consistency of the feature data along each dimension so that better modeling can be achieved with the isotropic GP kernel. Main contributions of our method are summarized as follows:

i. We investigate the feature data transform in a different way, i.e., changing the data reference system (which is the coordinate system) rather than the data itself. Our approach aims to adjust the data distribution to achieve better accuracy in supervised learning (such as GP), unlike PCA which aims to retain the data with the greatest variance, but does not guarantee to enhance the performance in supervised learning.

ii. Rotation of the coordinate system is implemented elegantly. We simply map each feature to an axis in an orthogonal coordinate system, which allows the rotation of the coordinate system to be carried out separately, i.e., we can rotate two axes at a time without affecting the rest of the axes. In particular, a novel measure – the Consistency Index (CI) – is defined to guide the feature space rotation procedure effectively and efficiently towards lessening the data anisotropy and better fitting the isotropic GP kernel.

iii. Compared with existing feature data transformation or selection methods, our approach is novel, simple, well-motivated (improving supervised learning such as GP) and yet effective. Extensive experimental results on different types of data sets have shown that our approach consistently improves the classification performance.

The remainder of this paper is organized as follows. A brief introduction to GP classification is given in Section 2. Details of our feature space rotation scheme are described in Section 3. In Section 4, experimental results are presented and discussed, prior to a summary in Section 5.

2. Gaussian process classification

A Gaussian process [1] is a nonlinear kernel based learning method, focusing on modeling the posterior directly. In many applications, it outperforms most other state of the art supervised learning methods. A Gaussian processes is a collection of random variables, any finite number of which have a joint Gaussian distribution. A GP is fully specified by its mean function \(m(x)\) and kernel function \(k(x,x')\), expressed as

\[
GP \sim g(m,k)
\]

For notational simplicity, it is common to consider the mean function of the GP to be zero [1]. A GP model is a non-parametric, probabilistic (Bayesian) model in function space. One can think of a GP as defining a distribution over functions, and inference taking place directly in the space of functions.

The kernel function characterizes correlations between different data points in the Gaussian process, and can be learnt from the data. The inference can be carried out directly under the GP framework by learning a kernel function from the training data. The kernel function studied in this paper is the widely used Radial Basis Function (RBF), which is defined as

\[
k(x,x') = \sigma^2 \exp\left(-\|x-x'\|^2/(2l^2)\right)
\]

where \(x\) and \(x'\) are input vector pairs, \(l\) is the characteristic length scale and \(\sigma^2\) is the signal variance. The free parameters, i.e., \(l\) and \(\sigma^2\), are called hyperparameters of the GP model.

In order to put our work into context, we briefly overview GP classification as follows. Assume that we have a dataset \(D\) with \(n\) observations \(D = \{(x_i,y_i), i=1,2,\ldots,n\}\), where \(x\) is the input vector of dimension \(d\) and \(y\) is the class label \(+1\) or \(-1\). The input \(d \times n\) data matrix is denoted as \(X\). Predictions for new inputs \(x'\) are made out of this given training data using the GP model. As described in [1], GP binary classification is performed by first calculating the distribution over the latent function \(f\) corresponding to the test case

\[
p(f' |X,y,X) = \int p(f' |X,f)p(f |X,y) df
\]

where \(p(f' |X,y) = p(y |f') \frac{p(y |X)}{p(y |X)}\) is the latent variable posterior, \(p(f |X,f)\) is the predictive posterior with respect to possible latent functions, and values of this could lie anywhere within the range of \((-\infty, +\infty)\). So the probabilistic prediction is made by

\[
p(f') = p(y' = +1 |X,y,X) = \int sp(f |X,y,X) df
\]

where \(s\) can be any sigmoid function that ‘squashes’ the prediction output to guarantee a valid probabilistic value within the range of \([0,1]\). For the multi-class classification problem, we can treat each class as being independent from the others, and apply binary classification individually to each (single) class versus all the other classes.

As indicated in Section 1, the commonly used isotropic GP kernel is used in our work. However, the real world data vary greatly and are usually anisotropic (i.e., inconsistent properties exist among features), which gives rise to the issue of how to lower the anisotropy of the existing data to better fit into the isotropic GP kernel for the purpose of improving the classification accuracy. To solve this issue, we propose a novel and efficient “feature space rotation” method, which is detailed in the following section.

3. Feature space rotation

Feature space rotation aims to rotate the coordinate system of the feature data so as to impose an expected change on the data distribution when observed in a new rotated coordinate system. As mentioned before, we focus on adjusting the data to be more consistent among the features. There are two issues that immediately arise out of feature space rotation, namely

- How to implement feature space rotation? Arbitrary rotation in a high dimensional space will make the relevant coordinate transform very complicated.

The remainder of this paper is organized as follows. A brief introduction to GP classification is given in Section 2. Details of our feature space rotation scheme are described in Section 3. In Section 4, experimental results are presented and discussed, prior to a summary in Section 5.
Loosely speaking, an \( n \) dimensional \((n \text{ D})\) coordinate system is a system that assigns \( n \) numbers (coordinate scalars) to each point in an \( n \)-dimensional space. In our context, each “point” corresponds to a feature vector (comprising a group of features) and the “scalars” are the feature values. The feature space itself actually exists independent of any choice of coordinates. The coordinates of a space can be thought as a function mapping from the space to the scalars. When altering the coordinate system, this “mapping” changes accordingly, resulting in a different group of scalars, which in our context, are the adjusted feature vector values.

However, it should be noted that an \( n \) D space rotation involves the complicated \( n \) variable transformations \([4,5]\). Following [4,5], we give a brief introduction to the procedure of \( n \) D space rotation as follows. A general \( n \) D space rotation is implemented by rotating an angle \( \theta \) around \( (n-2)\text{ D} \) space \( \mathbf{v} \) through a series of transformations \( \mathbf{M}_k \), where \( \mathbf{v}^{(k)} = \{\mathbf{v}^{(k)}_1, \mathbf{v}^{(k)}_2, \ldots, \mathbf{v}^{(k)}_n\} \) are the \((n-2)\text{ D} \) space vertices’ coordinate matrices at each transformation step, as shown in Eqs. (5) and (6)

\[
\mathbf{v}^{(k)} = \left[
\begin{array}{cccc}
\mathbf{v}^{(k)}_1 & \mathbf{v}^{(k)}_2 & \cdots & \mathbf{v}^{(k)}_n \\
\mathbf{v}^{(k+1)}_1 & \mathbf{v}^{(k+1)}_2 & \cdots & \mathbf{v}^{(k+1)}_n \\
\mathbf{v}^{(k+2)}_1 & \mathbf{v}^{(k+2)}_2 & \cdots & \mathbf{v}^{(k+2)}_n \\
\vdots & \vdots & \ddots & \vdots \\
\mathbf{v}^{(k+n-1)}_1 & \mathbf{v}^{(k+n-1)}_2 & \cdots & \mathbf{v}^{(k+n-1)}_n
\end{array}
\right]
\]

where values in each row are the coordinates of one of the vertices \( \mathbf{v}^{(k)} = \mathbf{v}^{(k-1)} \cdot \mathbf{M}_k \), \( 1 \leq k \leq (2 \cdot \binom{n}{2} + 1) \) \( (6) \)

where \( \mathbf{M}_k = \phi(\mathbf{v}^{(k-1)}) \) is a rotation function of \( \mathbf{v}^{(k-1)} \) \([4]\). The transformation starts from firstly aligning \( \mathbf{v}^{(0)} \) with any of the main axis, then applies the desired a total of \( C(2) \) rotations around the \((n-2)\text{ D}\) space, and finally, returns the rotation axis to its original position by inverting those transformations (a total of \( C(2) \)) in a reverse order. Combination of these transformations can be expressed as

\[
M = \prod_{k=1}^{2 \cdot \binom{n}{2} + 1} \mathbf{M}_k
\]

where \( R_{a,b}(\theta) \) is the rotation matrix for rotating axis \( x_a \) by an angle of \( \theta \). Except the intersection of rows \( a \) and \( b \) with columns \( a \) and \( b \), the rest of \( R_{a,b}(\theta) \) is an identity matrix, which shows only the coordinates on \( x_a \) and \( x_b \) will change after the \( R_{a,b}(\theta) \) rotation. Since there are \( C(2) \) main planes in an \( n \) D space, rotations are setup as the composition of all the rotations in each of main planes. Fig. 1 is an illustration of main plane rotations, in which rotation is first performed on the \( X-Y \) plane, followed by another rotation on the \( X-Z \) plane.

In our work, we hope to rotate the feature space along some specific features without the impact on the rest of the features. While for the main plane rotation in Eq. (8), the axis pair choice out of \( n \) elements will cause repetitive rotations, e.g., rotation of axes \( x_a \) & \( x_b \) would change the values that are going to be used for rotations of axes \( x_a \) & \( x_c \), \( x_b \) & \( x_c \), etc. Hence, we further simplify main plane rotation by non-overlapping rotation across the axis pairs, i.e., performing rotation sequentially on \( x_a \) & \( x_b \), \( x_a \) & \( x_c \), etc.

To fit our targeted non-overlapping feature space rotation to the above mentioned main plane coordinate rotation scheme, we assume that an orthogonal coordinate system is chosen for the feature data space, i.e., each axis is orthogonal to all other axes. This differs from PCA which projects data onto an orthogonal coordinate system and focuses on dimensionality reduction, while we assume the data already lies in an orthogonal coordinate system and aims at adjusting the data to an appropriate distribution. With the “orthogonal coordinate system” assumption, the feature space rotation is converted into main plane rotation. For an \( n \) dimensional space, the features are grouped into \( \lceil n/2 \rceil \) pairs on which rotation is applied individually. The complicated high dimensional feature space rotation is thus turned to
a constrained sequence of simple 2D coordinate rotations, i.e.
\[
x' = x \cos \theta + y \sin \theta
\]
\[
y' = -x \sin \theta + y \cos \theta
\]  
(9)  
(10)
where \((x, y)\) are the coordinates of a data point on a 2D coordinate system and \((x', y')\) are the corresponding coordinates after rotating the 2D coordinate system counterclockwise by \(\theta\). It should be noted that the angle of each rotation needs to be optimized, which is nontrivial. In this paper, to examine whether our idea of feature space rotation for improved GP classification is truly feasible, we simply consider rotating the coordinate system 45° counterclockwise. This greatly simplifies the calculation and still maximizes the change from the original coordinate alignment within the 0–90° range.

3.2. Criterion of feature space rotation

As pointed out at the beginning of Section 3, another important issue for feature space rotation is how to tell if a rotation is meaningful. As indicated previously, we aim to adjust the data to be more consistent. Inspired by the signature frequency described in [18], we define a Fluctuation Index (FI) to characterize the properties of the data distribution on each feature. Furthermore, we define a measure called the Consistency Index (CI) to reflect the consistency among features.

Similar to [18], we analyze the data in the frequency domain by applying spectral analysis [22,23] to each feature, which can better reveal the underlying properties compared with direct analysis in the space domain. Since the given feature data points do not lie on a uniform grid, a Non-uniform Discrete Fourier Transform (NDFT) [16] is applied to the data on each individual feature. Suppose the feature data is denoted as a length \(N\) sequence \(x[n]\), with \(x[n]=x[t_n]\), where \(t_n\) is the irregular sampling coordinate. The NDFT of \(x[n]\) is defined as [16]
\[
X(z_i) = \sum_{n=0}^{N-1} x[n]z_i^{-n}
\]  
(11)
where \(i = 0, 1, \ldots, N-1\). \(z_0, z_1, \ldots, z_{N-1}\) are arbitrarily located points on the unit circle in the z-plane, \(z_i\) is given by \(z_i = e^{j2\pi/Ti}\), where \(j^2 = -1\), \(T\) is the sampling coordinate of \(x[N-1]\) (or the range of \(x[n]\), \(n = 0, 1, \ldots, N-1\)), \(|X(z_i)|\) is the magnitude of the NDFT coefficients, also called the “data frequency content” in our context.

It should be noted that for real world data, the sequence \(x[n]\) has a limited length which can be regarded as a finite length sampling. Applying the Fourier transform to a finite length sample of the data gives rise to the issue of “spectral leakage”, showing up as a series of side lobes of the same width, which is half of the main lobe width [22]. Further, following the scaling property of the Fourier transform [22], if the windowing function interval is multiplied by a factor \(z\), the lobes width is shrunk by \(1/z\). By adjusting (e.g., scaling) the data to make the sample intervals consistent on each feature, the corresponding lobes will also be approximately equal. In turn, this makes the main lobe widths equal and hence makes the assumed smoothness equal (i.e., better fit to the isotropic GP kernel). Therefore, it can be seen that the side lobes of the data spectrum are inherently related to the main lobe. The latter corresponds to the GP kernel modeling, i.e., if the data spectrum’s main lobe is consistent on each feature, it can be better modeled by the isotropic GP kernel. By defining a measure on the side lobes, the corresponding main lobe property can be indirectly captured. The reason for not defining a measure directly on the main lobe is that for diversified real world data frequency content, the main lobe width is hard to measure accurately.

We define such a measure called the Fluctuation Index (FI). Informally speaking, FI focuses on the shape of the data frequency content, and characterizes the main fluctuation of each feature’s data frequency content. It is estimated by applying a “double Fourier transform” to the data on each feature, i.e., we apply an NDFT to the original data and calculate the magnitudes of the NDFT coefficients, followed by applying a Fourier transform to the obtained data frequency content, which is to further estimate the energy distribution on the data frequency content. The fluctuation of the data frequency content will be reflected on the harmonics of the double Fourier transform.

\[FI_j = \arg \max_{\Omega} (F_X(\Omega))\]
(12)
where \(F_X(\Omega)<\max(F_X(\Omega))\) is the corresponding Fourier transform of \(X_j(\Omega)\) or the double Fourier transform of the original data, denoted as \(F_X(\Omega)\). \(F_j\) of the data frequency content \(X_j(\Omega)\) is defined as: the location (or harmonic index) of the second dominant peak of \(F_X(\Omega)\), which equals to the total number of lobes, i.e.

\[CI = \text{var}(F_j)\]
(13)
\(CI\) reflects the consistency among features. A smaller \(CI\) value implies more consistent features. An illustration of the fluctuation index is shown in Fig. 2. After applying the feature space rotation, the consistency indices before and after rotation will be estimated and compared. We apply the feature space rotation only when the rotation makes the consistency index smaller.

Applicability of CI: CI calculation is based on the assumption that the given data follows the GP distribution, i.e., the spectrum of the data is Gaussian. However, this GP assumption is not always true for real world data. An example is shown in Fig. 2(c), where the data spectrum does not decrease monotonically as a GP data spectrum does. Therefore, we further define a value “GPin dex” to identify how close the given data spectrum is to a GP distribution as follows:

\[GPin dex = \max(GP weight_j)\]
(14)
where \(GP weight_j\) is the sum of the spectrum weighted by the frequency on each feature respectively, i.e.

\[GP weight_j = \sum_{f=0}^{d} X_j(f)\text{ }j = 1, \ldots, d\]
(15)
where \(\Omega\) is the frequency range, \(X_j\) is the data spectrum on each feature and \(X_j(f) > 0\) (\(u\) is a threshold of the data spectrum, 0.95 – max\(X_j\) here). If \(GPin dex\) is greater than a preset threshold (which is empirically set to be 500, as a result of testing on various datasets), the given data have a strong high frequency component and is regarded as non-GP data. For non-GP data, the estimation of the CI is not applicable, and hence, no rotation will be applied to the given data.

3.3. Our algorithm of feature space rotation

Given the feature data as \(d\) features and a total of \(N\) data points, which is usually expressed as an \(N \times d\) data matrix \(X_g\). The first step of our algorithm is to check if the given data has a GP distribution. The feature space rotation steps proceed only when the given data satisfies the GP distribution criteria. Then, rotation is performed on a pair of dimensions at a time, independent of
other dimensions, until all dimensions have been considered. The coordinate values of the data points on the new rotated axes are updated, leading to the rotated data matrix \( \frac{1}{C_{138}} \). The next step is to assess whether the rotation above actually makes the rotated feature data more consistent by calculating the \( CI \) for both the original data and the rotated data. If \( CI_{\text{rotated}} \) is smaller than \( CI_{\text{original}} \) (which implies more consistent features on the rotated data), then the rotated data \( \frac{1}{C_{138}} \) are used for further classification. Otherwise, the original data \( \frac{1}{C_{138}} \) are used for classification.

Our algorithm of feature space rotation is summarized as follows:

1. **Input**: \( N \times d \) data matrix \( [x_{ij}] \), where \( i = 1, \ldots, N, j = 1, \ldots, d \), \( N \) is the total number of data points, and \( d \) is the total number of features (i.e., dimensions).
2. Calculate \( G\text{Pindex} \) using Eqs. (15) and (14).
3. if \( G\text{Pindex} > G\text{Pthres} \)
   \[ [\phi_{x_{ij}}] = [x_{ij}] \]
   end if.
4. Rotate every pair of dimensions 45° at a time using Eqs. (9) and (10) until all dimensions have been considered, resulting in the rotated data matrix \( [\phi_{x_{ij}}] \).
5. Calculate the \( FIs \) of the original data, \( F_{\text{original}} \), and the rotated data, \( F_{\text{rotated}} \), using Eq. (12).
6. Calculate \( CI_{\text{original}} \) and \( CI_{\text{rotated}} \) based on \( F_{\text{original}} \) and \( F_{\text{rotated}} \), using Eq. (13).
7. if \( CI_{\text{rotated}} < CI_{\text{original}} \)
   \[ [\phi_{x_{ij}}] = [\phi_{x_{ij}}] \]
   else
   \[ [\phi_{x_{ij}}] = [x_{ij}] \]
   end if.
8. **Output**: \( N \times d \) data matrix \( [\phi_{x_{ij}}] \).

4. Experiments and discussion

We have evaluated the effectiveness of our approach by conducting experiments on a variety of data sets, including synthetic
data, UCI machine learning data, facial expression action data [24] as well as the Weizmann human action data set [21]. GP classification results were obtained using Lawrence’s program [10,11].

4.1. Synthetic data

The proposed method was first tested on a synthetic data, where three bars of two classes were created in a 2D space as shown in Fig. 3(a). The third dimension is the class label which is represented in Fig. 3 as red and black, standing for class 1 and class 2 respectively. The created data values were deliberately made parallel to one of the axes, dimension 1 in Fig. 3(a), which causes significant overlap when the data values are projected onto the other axis, dimension 2 in Fig. 3(a). By rotating the feature data 45° as shown in Fig. 3(b), it can be seen that the data values are more evenly distributed across both axes. Classification was then performed using 10-fold cross validation. The ROC curves with respect to both original data and rotated data are shown in Fig. 3(c). It is clear that the rotated data have a better classification performance compared with the original data.

4.2. UCI machine learning data

A series of tests were also made on a total of 16 randomly chosen data files from “The UCI Repository of Machine Learning Databases and Domain Theories” as listed in Table 1. The number of features in each file ranges from 4 to 57. The first 10 files (1–10) in Table 1 are those data sets whose CI values are decreased after rotation. The other 6 (11–16) files in Table 1 is those data sets with no decreasing CI values. For the 10 UCI data sets that have decreased CI values, the GPindex values of the first 7 files (1–7) are smaller than GPthres (which is set to be 500 here) and hence are regarded as having a GP distribution. The other 3 with GPindex values greater than GPthres are identified as non-GP. Similarly, files 11–14 have a GP distribution and files 15–16 are non-GP.

The comparison of classification accuracy between the original data and the rotated data is illustrated in Fig. 4(a) for files 1–10. Among these data sets, the classification accuracy on data sets 1–7 increased or remained the same after rotation, and that on the data sets 8–10 decreased or remained the same. Such results show that feature space rotation does improve the classification accuracy on the GP distribution data sets and the GPindex value effectively filters out the non-GP distribution data sets where the

![Figure 3](image-url)
CI value does not apply for judging whether a rotation is needed. Fig. 4(b) shows the classification accuracy comparison on the rest data sets 11–16 with no increased CI values. It can be seen that the classification accuracies on these six data sets have not increased after the rotation. The results are also consistent with our rotation strategy, i.e., no rotation is needed if the CI value does not increase (whatever the GPindex value is). In a word, Fig. 4(a) and (b) shows that a combination of GPindex and CI work effectively on the UCI data sets in terms of deciding whether the data should be rotated for better classification results.

4.3. Action data

Further tests were carried out on two groups of action data, i.e., the facial expression action data in [24] and the Weizmann human action data in [21]. For both of these two data groups, spatio-temporal (ST) features were extracted using the method proposed in [24]. Regions with distinguishable motions were detected as “interest points”. At each interest point, a cuboid was extracted which contains the spatio-temporally windowed pixel values. The ST volume (or cuboids) around these interest points were represented using a gradient-based descriptor. After applying PCA to the obtained descriptors, the ST cuboids were clustered using the k-means algorithm to give each cuboid a type. In this way, a library of cuboid prototypes were created. Then, a histogram of the cuboid types was used as the feature descriptor for each action video. In our test, we used the leave-one-out cross validation scheme and reported the average accuracy.

Facial expression data: The facial expression dataset [24] includes two individuals, each performing six expressions in two lighting setups. Under each lighting setup, each individual repeats each of the six expressions eight times. In this way four data subsets are obtained. Each subset has 50 features and 48 instances. Our experiment was performed on subset 1 and subset 2, i.e., train on subset 1 and test on subset 2. The results are given in Table 2. Identified by GPindex as having a GP distribution and following a decreasing of the consistency index, the data were rotated and an obvious improvement on the classification performance can be seen on the rotated data compared with the original data.

Human action data: The Weizmann human action data set [21] contains 81 low-resolution videos from nine different people, each repeatedly performing nine actions, e.g., running, walking,

<table>
<thead>
<tr>
<th>Data file name</th>
<th># of sample</th>
<th># of dimension</th>
<th>GPindex</th>
<th>CI-original</th>
<th>CI-rotated</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Breast-cancer</td>
<td>286</td>
<td>9</td>
<td>22</td>
<td>0.138</td>
<td>0.056</td>
</tr>
<tr>
<td>2. Credit-app</td>
<td>600</td>
<td>15</td>
<td>237</td>
<td>0.130</td>
<td>0.074</td>
</tr>
<tr>
<td>3. Monks-3</td>
<td>432</td>
<td>7</td>
<td>5</td>
<td>0.099</td>
<td>0.098</td>
</tr>
<tr>
<td>4. Australian</td>
<td>2565</td>
<td>22</td>
<td>290</td>
<td>0.145</td>
<td>0.041</td>
</tr>
<tr>
<td>5. Hayes-roth</td>
<td>160</td>
<td>5</td>
<td>132</td>
<td>0.128</td>
<td>0.050</td>
</tr>
<tr>
<td>6. Crabs</td>
<td>200</td>
<td>8</td>
<td>1</td>
<td>0.090</td>
<td>0.000</td>
</tr>
<tr>
<td>7. Servo</td>
<td>167</td>
<td>4</td>
<td>305</td>
<td>0.148</td>
<td>0.125</td>
</tr>
<tr>
<td>8. Liver-disorders</td>
<td>345</td>
<td>7</td>
<td>582</td>
<td>0.067</td>
<td>0.000</td>
</tr>
<tr>
<td>9. Heart</td>
<td>303</td>
<td>14</td>
<td>1307</td>
<td>0.134</td>
<td>0.049</td>
</tr>
<tr>
<td>10. Spam</td>
<td>460</td>
<td>57</td>
<td>2601</td>
<td>0.050</td>
<td>0.025</td>
</tr>
<tr>
<td>11. Pima-diabetes</td>
<td>768</td>
<td>8</td>
<td>124</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>12. Hepatitis</td>
<td>155</td>
<td>19</td>
<td>256</td>
<td>0.050</td>
<td>0.064</td>
</tr>
<tr>
<td>13. Monks-2</td>
<td>432</td>
<td>7</td>
<td>1</td>
<td>0.067</td>
<td>0.071</td>
</tr>
<tr>
<td>14. Tic-tac-toe</td>
<td>938</td>
<td>9</td>
<td>7</td>
<td>0.060</td>
<td>0.080</td>
</tr>
<tr>
<td>15. Ionosphere</td>
<td>351</td>
<td>34</td>
<td>1187</td>
<td>0.034</td>
<td>0.038</td>
</tr>
<tr>
<td>16. Housing</td>
<td>506</td>
<td>14</td>
<td>1407</td>
<td>0.051</td>
<td>0.059</td>
</tr>
</tbody>
</table>

Table 2

Accuracy on facial expression data.

<table>
<thead>
<tr>
<th>Expressions</th>
<th>Original</th>
<th>Rotated</th>
</tr>
</thead>
<tbody>
<tr>
<td>Anger</td>
<td>0.750</td>
<td>0.875</td>
</tr>
<tr>
<td>Disgust</td>
<td>0.625</td>
<td>0.625</td>
</tr>
<tr>
<td>Fear</td>
<td>0.750</td>
<td>0.875</td>
</tr>
<tr>
<td>Joy</td>
<td>1.000</td>
<td>1.000</td>
</tr>
<tr>
<td>Sadness</td>
<td>1.000</td>
<td>1.000</td>
</tr>
<tr>
<td>Surprise</td>
<td>1.000</td>
<td>1.000</td>
</tr>
<tr>
<td>Average</td>
<td>0.854</td>
<td>0.896</td>
</tr>
</tbody>
</table>

GPindex
Consistency index 0.028 0.015

Fig. 4. The results of our approach on the UCI data sets. (a) Accuracy comparison on UCI datasets 1–10, (b) accuracy comparison on UCI datasets 11–16 and (c) accuracy comparison on UCI datasets with varied classifiers.
This is probably because that PCA transforms the space rotation method can effectively guide the feature space to rotate towards reducing the transformed axes to the directions of maximum variance, which provides dimensionality reduction, but does not guarantee for better discrimination. Also, PCA has a couple of assumptions [12] such as linearity and that big variance has important dynamics, which are not always the case for real-world data, especially for those with very low-dimensional features as used here. Therefore, it seems to be not always appropriate to apply PCA, while data rotation seems a “safe” transformation.

### 4.4. Comparison with PCA

Since PCA is a popular feature transformation method for data analysis, we have also made comparison between our feature space rotation method and PCA on the 10 UCI data sets that need rotation (i.e., data sets 1–10 in Table 1). The Nearest Neighbor (NN) [3] classifier was also applied here. Fig. 4(c) shows the comparison results, together with a detailed summarization in Table 4 for clarity. It can be seen from Fig. 4(c) and Table 4 that: (1) for all data sets, GP significantly outperforms NN on whether the original data or the rotated data; (2) GP classification was improved on most data sets (i.e., Gaussian data) after rotation; (3) data rotation makes NN classification performance slightly better on some data sets or the same on the rest; (4) the classification accuracy on these used data was reduced for both GP and NN after applying PCA transformation.

The above comparison results in Fig. 4(c) illustrate that data rotation works effectively on GP and almost makes no difference on NN, since the aim of our feature rotation method is to make the feature data more consistent, and hence to benefit isotropic GP kernels. However for NN, classification is based on the nearest neighbor which is hardly affected by the data rotation. Looking at PCA, the classification accuracy drops sharply on the PCA data for both GP and NN. This is probably because that PCA transforms the features to a small number of uncorrelated features and rotates the transformed axes to the directions of maximum variance, which provides dimensionality reduction, but does not guarantee for better discrimination. Also, PCA has a couple of assumptions [12] such as linearity and that big variance has important dynamics, which are not always the case for real-world data, especially for those with very low-dimensional features as used here. Therefore, it seems to be not always appropriate to apply PCA, while data rotation seems a “safe” transformation.

### 4.5. Comparison with non-isotropic kernels

Although isotropic kernels are popular, it is also common to use hyperparameters to enable non-isotropic kernels, such as Automatic Relevance Determination (ARD) [8]. The ARD model is implemented by making the values of the hyperparameters (estimated through optimizing the model marginal likelihood) associated with each input adapt to the training data. In this way, the kernel is made non-isotropic. As an important baseline, a comparison has been made between our proposed feature space rotation method (tuning kernels isotropic) and the ARD model (non-isotropic kernels optimization), i.e., RBF with ARD (RBF-ARD). Neural Network kernel with ARD (NN-ARD) as well as Linear kernels with ARD (LIN-ARD). Among these, RBF is a stationary kernel, i.e., shift invariant [11], while NN and LIN are non-stationary kernels. Results on the 16 UCI datasets are shown in Table 5. For our feature space rotation method, since GPin dex and CI can effectively decide whether the data should be rotated or not, values for datasets 1–7 are the classification results of the original data. Those for datasets 8–16 are the classification results of the rotated data and those for datasets 1–7 are the classification results of the rotated data and those for datasets 8–16 are the classification results of the original data. The results in Table 5 show that, overall, feature space rotation outperforms all the non-isotropic kernels, whether stationary or non-stationary.

### 5. Conclusions

A novel feature space rotation method has been proposed in this paper. By properly rotating the coordinate system, the data distribution in the high dimensional space is adjusted so as to be better modeled. By filtering out the non-GP data with GPin dex, CI can effectively guide the feature space to rotate towards reducing
the anisotropy of the GP data, so as to better fit the isotropic GP kernel. It should be noted that the proposed feature space rotation approach has been well shaped for applying to GP classification by appropriately combining the isotropic GP kernel assumption (whose spectrum is also Gaussian and isotropic) with real world data which could be either non-GP or GP data (where the non-GP data values are filtered by GPindex and the rest are guided by CI for rotation). Extensive experiments on a variety of data sets have shown that our feature space rotation method is a simple but effective way of changing the data distribution, and accordingly, the GP classification performance has been clearly enhanced.

Currently, our proposed feature space rotation is performed at a fixed rotation angle of 45°. Although such a rotation angle allows maximum change within the 0–90° range, how to choose an optimal one still remains an issue. In future work, we will study more general ways of selecting the optimized rotation angle, e.g., constructing a rotation angle parameterized kernel, with which the angle is estimated in a more principled way as a result of kernel hyperparameters optimization.

Acknowledgment

Many thanks go to Dr. Hang Zhou for her great efforts in both experiments and paper writing. Part of this work was done when working at the University of Melbourne, Australia. This work is also supported by the Fund of Hundred Talented Program of Chinese Academy of Science and National Natural Science Foundation of China (No. 61175003).

References