Mean Polynomial Kernel and Its Application to Vector Sequence Recognition

Raissa Relator[1,a) Yoshihiro Hirohashi[1] Eisuke Ito[1] Tsuyoshi Kato[1,b)

Abstract: Classification tasks in computer vision and brain-computer interface research have presented several applications such as biometrics and cognitive training. However, determining suitable data representation has been challenging, and recent approaches have deviated from the familiar form of one vector for each data sample. This paper considers a kernel between vector sets, the mean polynomial kernel, motivated by recent studies where data are approximated by linear subspaces, in particular, methods on Grassmann manifolds. The kernel supports vector sequences as input. We discuss how the kernel can be associated with the Grassmann Projection kernel, and provide experimental results showing how it outperforms existing subspace-based methods on Grassmann manifolds.

Keywords: kernel methods, support vector machines, Grassmann distance and kernels, face recognition, brain-computer interface, vector sequence, binary classification

1. Introduction

Among currently trending fields, research efforts particularly related to computer vision and brain-computer interface (BCI) have been aimed at modeling data either as a low-dimensional subspace or a sequence of vectors. There have been studies in these areas dedicated to algorithms for such type of input [1], [3], [7], [9], [17], [21], [25], [29], [30]. For computer vision, this approach may be motivated by the presence of abundant material derived from videos and sets of image sequences [3], [7], [9], [21], [25], [29], [30] such as in Figure 1a. Each video image extracted is represented by a vector, while the whole vector sequence, concatenated as a matrix, approximates the video for a given time frame. As for BCI adopting a similar approach, this may be induced by the nature of the data, which is commonly time series, such as electroencephalography (EEG) signals collected while subjects perform motor tasks or during induction of visual stimuli [1], [17]. EEG data is generated by placing several sensors accordingly on the head of the subject as shown in Figure 2, and each sensor records neural activity depicted by the signals. The vector sequence illustrated in Figure 1b corresponds to the signals collected from all sensors.

Appropriate data representation has been considered as one of the most important challenges in dealing with classification tasks. Vector form may be the simplest and most common representation of samples in existing literatures, especially when using popular techniques such as support vector machines (SVM) and kernel methods. However, this may not be the best representation to encompass significant, if not all, attributes and information useful for discrimination. To address this problem, new modes of data representation are constantly being explored [2], [5], [6], [8], [10], [11], [12], [13], [14], [15], [18], [19], [24], [25], [26], [27]. Along with this, various feature extraction techniques and discrimination methods are also being investigated, and several studies have proven kernel methods to be a flexible technique in supporting various data structures, such as graphs [8], [12], [13], [18], [24], [26], strings [14], [15], [19], [27], and even subspaces and sets of vectors [2], [5], [6], [10], [11], [25].

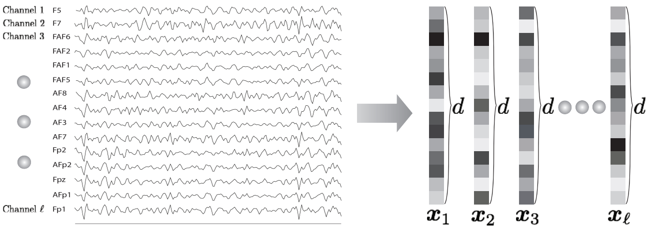
Kernel-based algorithms [22], [23] have come a long way since their introduction. Aside from the fact that kernel functions have provided algorithms a bridge between linearity and nonlinearity, their performance have been proven comparable to, if not better than, existing algorithms in various areas where they have been applied. Moreover, applying the so-called 'kernel trick' is very straightforward and new kernels can be easily derived from old kernels. Compared to other methods, the dimension of the feature space can also be treated more lightly since the technique can be reduced to simply performing inner products between data images on the space, thus making the algorithm computationally inexpensive.

In this paper, we focus on data represented as sets of vectors. Different algorithms have been formulated in such a way that data are approximated by low-dimensional linear subspaces [3], [6], [7], [9], [21], [25], [29], [30]. However, as previously pointed out [7], the task of appropriately handling data has become an issue, such as inconsistency in strategy when feature extraction is done in a Euclidean space while non-Euclidean metrics are used. For this purpose, they proposed a unified framework for subspace-based approaches by formulating the problem on the Grassmann manifold, a space of linear subspaces with a fixed dimension. On the other hand, these methods involve dimension reduction, and even with the use of the usual dimension reduc-

---

[1] Department of Computer Science, Graduate School of Science and Engineering, Gunma University
a) relator-raissa@kato-lab.cs.gunma-u.ac.jp
b) katotsu@cs.gunma-u.ac.jp

(a) Video sequence data.



(b) EEG signals.

Fig. 1: Examples of data modeled as vector sequences. (a) For video sequences, each image frame extracted is represented as a vector of pixel intensity values. The vector sequence are usually concatenated to represent the vector set input. (b) For BCI, EEG signals are recorded over a certain time interval using several channels or sensors. Each vector in the sequence corresponds to a channel used in the procedure, and vector entry represents an instantaneous signal intensity.

tion techniques such as Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA), there is always a possibility of information loss. This makes the selection of the subspace dimension a crucial step. Furthermore, methods such as PCA and LDA usually employ eigendecomposition, and hence, may be very time consuming especially for high dimensional data.

With the aforementioned issues in mind, the goal of this paper is to examine a kernel function, which we refer to as the mean polynomial kernel, that can retain data information while being computationally inexpensive. Also, as a more general approach than kernels for subspaces, we treat data as a common collection of vectors, instead of a linear subspace. The kernel is invariant of the permutation order of the vectors in the set. In addition, we present an interesting relationship between this kernel and the Projection kernel, which is a known Grassmann kernel. We give emphasis to face recognition and BCI applications posed as binary classification problems, which are of particular interest due to their practicality in various areas, biometrics and cognitive training and improvement, among others. Experimental results using real data modeled as vector sequences show that, aside from being computationally efficient, the performance of the mean polynomial kernel is comparable to methods employing kernels in the Grassmann manifold and subspace methods using Grassmann distances.

2. Preliminaries

Consider a set of data $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_\ell \in \mathbb{R}^d$, where ℓ is the number of data points. Let us denote the hth entry in the ith data point $\boldsymbol{x}_i$ by $x_{h,i}$. A sample statistic, $\frac{1}{\ell}\sum_{i=1}^{\ell}\prod_{h=1}^{d} x_{h,i}^{p_h}$, is said to be the *qth order moment* if the d-dimensional vector $\boldsymbol{p} \in (\mathbb{N} \cup \{0\})^d$ satisfies $p_1 + \cdots + p_d = q$. The *uncentered covariance matrix* defined by $\frac{1}{\ell}\sum_{i=1}^{\ell}\boldsymbol{x}_i\boldsymbol{x}_i^\top$ contains all the second order moments. Indeed, the (h,k)th entry in the uncentered covariance matrix is the the second order moment with $\boldsymbol{p} = \boldsymbol{e}_h + \boldsymbol{e}_k$, where $\boldsymbol{e}_h$ is a unit vector whose hth entry is one and the rest of the entries are zero. Let $\bar{\boldsymbol{x}} = [\bar{x}_1, \ldots, \bar{x}_d]^\top$ be the mean vector of the data points. With the d-dimensional vector $\boldsymbol{p}$ satisfying $p_1 + \cdots + p_d = q$, the *qth order central moment* is defined as $\frac{1}{\ell}\sum_{i=1}^{\ell}\prod_{h=1}^{d}(x_{h,i} - \bar{x}_h)^{p_h}$. Every second order central moment is included in the *central covariance matrix* $\frac{1}{\ell}\sum_{i=1}^{\ell}(\boldsymbol{x}_i - \bar{\boldsymbol{x}})(\boldsymbol{x}_i - \bar{\boldsymbol{x}})^\top$, which is usually referred to simply as the *covariance matrix*.

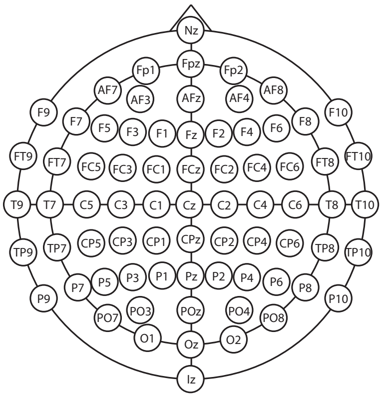


Fig. 2: Sample position map of sensors for EEG.

For succeeding sections, we refer to a matrix $\boldsymbol{U}$ as *orthonormal* if $\boldsymbol{U}^\top\boldsymbol{U} = \boldsymbol{I}$, and define the *vectorization* of an $m \times n$ matrix $\boldsymbol{A}$ as the column vector $\mathrm{vec}(\boldsymbol{A}) = [a_{11}, a_{12}, \ldots, a_{1n}, \ldots, a_{m1}, a_{m2}, \ldots, a_{mn}]^\top$.

3. Grassmann Kernels and Related Methods

We give a concise discussion of the Grassmann kernels [6], [7], [28], their analogy with the mean polynomial kernel, and some related methods.

A Grassmann manifold, or Grassmannian, is defined as a set of linear subspaces with a fixed number of dimensions, say, m. Several metrics used in literatures have been specified in this manifold, mostly incorporating principal angles or angles between subspaces in their characterization [3], [6], [7], [21], [25], [28], [29], [30]. Moreover, kernels over these manifolds have also been introduced. In particular, we are interested in the following kernels:

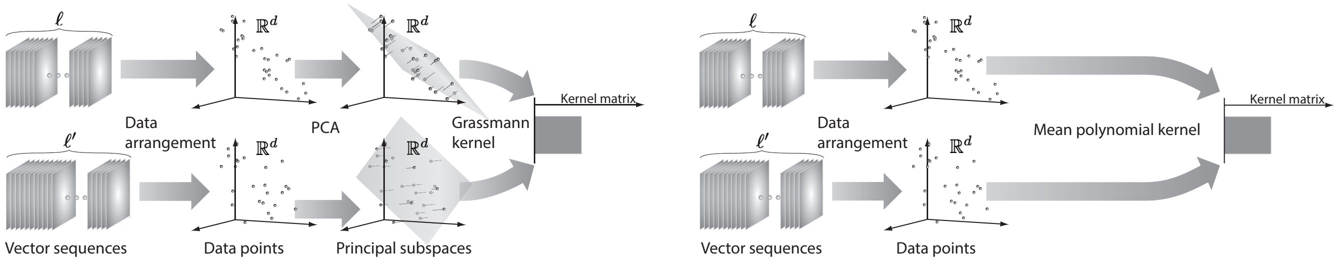
Definition 1. Let $\boldsymbol{U}_x$ and $\boldsymbol{U}_y$ be orthonormal matrices whose columns are bases of linear subspaces. The Projection kernel is defined as

$$k_{\mathrm{PROJ}}(\boldsymbol{U}_x, \boldsymbol{U}_y) = \left\|\boldsymbol{U}_x^\top\boldsymbol{U}_y\right\|_F^2,$$

where $\|\cdot\|_F$ denotes the Frobenius norm, and the Binet-Cauchy kernel is given by

$$k_{\mathrm{BC}}(\boldsymbol{U}_x, \boldsymbol{U}_y) = (\det \boldsymbol{U}_x^\top\boldsymbol{U}_y)^2 = \det \boldsymbol{U}_x^\top\boldsymbol{U}_y\boldsymbol{U}_y^\top\boldsymbol{U}_x.$$

Many existing problems can be realized on nonlinear manifolds such as the Grassmannian. This being said, various methods in the Grassmannian setting have been proposed. One such technique is the use of Grassmann kernels in conjunction with support vector machines (GK-SVM) [25]. This approach entails the computation of kernel matrices, which then proceed as the



(a) Flow for Grassmann kernels.

(b) Flow for Mean Polynomial kernel.

Fig. 3: Flow of methodology for computing values for Grassmann kernels and the mean polynomial kernel. Grassmann kernels are defined on a Grassmann manifold which is a set of linear subspaces. When employing these kernels, each vector sequence, represented by a set of data points on space, is approximated by a principal subspace obtained via PCA. However, this poses a threat of some degree of information loss, and is more likely to consume more time due to eigendecomposition. The mean polynomial kernel, on the other hand, can be directly applied to compute the kernel value between the sets of data points. It can avoid information loss while being more time efficient.

SVM input. Analogously, the mean polynomial kernel given in Section 4 is applied in this manner when SVM is the classifier. Figure 3 gives a general illustration of the flow of computation of the Grassmann kernels and the mean polynomial kernel, and also highlights the difference between the two kernels.

Another comparable method is the Grassmann Distance Mutual Subspace Method (GD-MSM) [25]. This technique integrates the Grassmann metrics in the Mutual Subspace Method (MSM) [30]. Furthermore, the task of subspace classification can be approached in two ways. The first one, which is referred to as the *subject-wise dictionary*, is done by assuming that one subject or object corresponds to one principal subspace. During the training stage, the total of principal subspaces calculated is the same as the number of subjects. These serve as the bases to which the unlabeled principal subspaces of test subjects are compared to, and the subspace with the minimal Grassmann distance from the unlabeled subspace is determined. The second approach is done by assuming one principal subspace per class. The principal subspaces, which in this case is referred to as the *class-wise dictionary*, are derived from each class among the training data. This being said, we have only two principal subspaces in the case of binary classification, regardless of the number of subjects. In the testing stage, unlabeled principal subspaces are classified according to which subspace they are closer to in terms of metric.

The score function can be considered for the two aforementioned mutual subspace methods. The SVM score can serve as a confidence level. Namely, a higher score may provide higher certainty of assigning the data to the positive class. For the *class-wise dictionary*, the difference between the distance to the subspace of the negative class, d_-, and the distance to the subspace of the positive class, d_+, represents how confidently unknown labels are classified as positive. Hence, we define the score function as $d_- - d_+$. For the *subject-wise dictionary*, we define the score function by the difference between the minimal distance to negative class subspaces and the minimal distance to the positive class subspaces.

4. Mean Polynomial Kernel

In this section, we discuss the details of the *mean polynomial kernel*, which can be directly applied to data in the form of vector sets.

Consider two sets of vectors $\mathcal{X} = \{\boldsymbol{x}_i\}_{i=1}^{\ell}$ and $\mathcal{Y} = \{\boldsymbol{y}_j\}_1^{\ell'}$, where $\boldsymbol{x}_i, \boldsymbol{y}_j \in \mathbb{R}^d$. To define a kernel for such types of data, we introduce a notation of a set of vector sequences as $\mathcal{S} = \{\{z_i\}_{i=1}^n \mid n \in \mathbb{N} \text{ and } \forall i \in \mathbb{N}_n,\ z_i \in \mathbb{R}^d\}$, where $\mathbb{N}$ is the set of natural numbers, and $\mathbb{N}_n = \{i \in \mathbb{N} \mid i \leq n\}$, such that $\mathcal{S}$ is the input domain for the kernel defined as follows.

Definition 2. Let $k_q : \mathcal{S} \times \mathcal{S} \to \mathbb{R}$ such that

$$k_q(\mathcal{X}, \mathcal{Y}) = \frac{1}{\ell\ell'} \sum_{i=1}^{\ell} \sum_{j=1}^{\ell'} \langle \boldsymbol{x}_i, \boldsymbol{y}_j \rangle^q,$$

where $\mathcal{X}, \mathcal{Y} \in \mathcal{S}$ and $q \in \mathbb{N}$. We shall refer to k_q as the *qth order mean polynomial kernel.*

It can be shown that this kernel is a special case of the multi-instance kernels [4] when instances involve linear kernels or polynomial kernels with constant $c = 0$. With regards to its characterization, we can easily confirm that for the case $q = 2$, the covariance matrix is directly used as a feature vector. For instance, consider two matrices, $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_\ell]$ and $\boldsymbol{Y} = [\boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_{\ell'}]$, for the set of vectors $\mathcal{X}$ and $\mathcal{Y}$, respectively. Then their respective uncentered covariance matrices are given by $\boldsymbol{\Sigma}_x = \frac{1}{\ell} \sum_{i=1}^{\ell} \boldsymbol{x}_i \boldsymbol{x}_i^\top$ and $\boldsymbol{\Sigma}_y = \frac{1}{\ell'} \sum_{j=1}^{\ell'} \boldsymbol{y}_j \boldsymbol{y}_j^\top$. By defining a feature map $\boldsymbol{\phi}(\boldsymbol{X}) = \mathrm{vec}(\boldsymbol{\Sigma}_x)$, we have

$$\begin{aligned}
&\langle \boldsymbol{\phi}(\boldsymbol{X}), \boldsymbol{\phi}(\boldsymbol{Y}) \rangle = \langle \mathrm{vec}(\boldsymbol{\Sigma}_x), \mathrm{vec}(\boldsymbol{\Sigma}_y) \rangle = \mathrm{tr}(\boldsymbol{\Sigma}_x \boldsymbol{\Sigma}_y) \\
=&\frac{1}{\ell\ell'} \sum_{i=1}^{\ell} \sum_{j=1}^{\ell'} \mathrm{tr}(\boldsymbol{x}_i \boldsymbol{x}_i^\top \boldsymbol{y}_j \boldsymbol{y}_j{}^\top) = \frac{1}{\ell\ell'} \sum_{i=1}^{\ell} \sum_{j=1}^{\ell'} \mathrm{tr}(\boldsymbol{y}_j{}^\top \boldsymbol{x}_i \boldsymbol{x}_i^\top \boldsymbol{y}_j) \\
=&\frac{1}{\ell\ell'} \sum_{i=1}^{\ell} \sum_{j=1}^{\ell'} \langle \boldsymbol{x}_i, \boldsymbol{y}_j \rangle^2. \qquad (1)
\end{aligned}$$

Hence, the Euclidean inner product of vectorized covariance matrices is precisely the second order mean polynomial. Furthermore, all information contained within the uncentered matrices are preserved and can be exploited.

If we rewrite the definition of the kernel as

$$\bar{k}_q(\boldsymbol{X}, \boldsymbol{Y}) = \frac{1}{\ell\ell'} \sum_{i=1}^{\ell} \sum_{j=1}^{\ell'} \langle \boldsymbol{x}_i - \bar{\boldsymbol{x}}, \boldsymbol{y}_j - \bar{\boldsymbol{y}} \rangle^q, \qquad (2)$$

where $\bar{\boldsymbol{x}}$ and $\bar{\boldsymbol{y}}$ are the mean vectors of $\boldsymbol{X}$ and $\boldsymbol{Y}$, respectively, then the kernel is the inner product among centered covariance matrices when $q = 2$.

More generally, we can say that the qth order mean polynomial kernel contains all qth order moments as feature vectors. Indeed, if we let $\mathbb{P}_q = \{\boldsymbol{p} \in (\mathbb{N} \cup \{0\})^d | \boldsymbol{p}^\top \boldsymbol{1} = q\}$ and $x_{h,i}$ be the hth entry in $\boldsymbol{x}_i$, enumerating all qth order moments allows us to define

$$\phi_{\boldsymbol{p}}(\boldsymbol{X}) = \frac{1}{\ell}\sqrt{\frac{q!}{p_1! \cdots p_d!}} \sum_{i=1}^{\ell} \prod_{h=1}^{d} x_{h,i}^{p_h}.$$

By using the feature map given by $\boldsymbol{\phi}(\boldsymbol{X}) = [\phi_{\boldsymbol{p}}(\boldsymbol{X})]_{\boldsymbol{p} \in \mathbb{P}_q}$, we can derive the following equality

$$k_q(\boldsymbol{X}, \boldsymbol{Y}) = \langle \boldsymbol{\phi}(\boldsymbol{X}), \boldsymbol{\phi}(\boldsymbol{Y}) \rangle, \tag{3}$$

as given in the journal version of this paper [20]. Existence of a feature vector ensures the positive semidefiniteness of the mean polynomial kernel. Similarly for the centered version of the mean polynomial kernel, the features can be explicitly expressed as a set of all the qth order central moments [20].

5. Mean Polynomial Kernel and Projection Kernel Relationship

We aim to establish a relationship between the mean polynomial kernel and the Projection kernel. In principle, Grassmann kernels are considered as kernel functions for principal subspaces. Eigendecomposition of two symmetric matrices $\boldsymbol{\Sigma}_x$ and $\boldsymbol{\Sigma}_y$ is essential for the computation of the Projection kernel value between two vector sequences $\boldsymbol{\mathcal{X}}$ and $\boldsymbol{\mathcal{Y}}$. Moreover, it can be shown that the bases of the principal subspaces are exactly the m major eigenvectors. To obtain the value of the Projection kernel between two subspaces $\boldsymbol{\mathcal{X}}$ and $\boldsymbol{\mathcal{Y}}$, the m eigenvectors are initially stored in the matrices $\boldsymbol{U}_x$ and $\boldsymbol{U}_y$. Let us define $\boldsymbol{\Sigma}'_x = \boldsymbol{U}_x \boldsymbol{U}_x^\top$ and $\boldsymbol{\Sigma}'_y = \boldsymbol{U}_y \boldsymbol{U}_y^\top$, the uncentered covariance matrices where the m major eigenvalues are replaced with ones and the rest of the eigenvalues are disregarded. Then the two kernels are related by the equality

$$k_{\mathrm{PROJ}}(\boldsymbol{U}_x, \boldsymbol{U}_y) = \langle \mathrm{vec}(\boldsymbol{\Sigma}'_x), \mathrm{vec}(\boldsymbol{\Sigma}'_y) \rangle. \tag{4}$$

Details of the derivation are given in [20].

An assessment of both equations (1) and (4) suggests that while the second order mean polynomial kernel preserves every bit of information in the uncentered covariance matrices, the Projection kernel possesses the possibility to disregard and lose information of each dimension of the principal subspaces, and all the information on their orthogonal complements. A similar case can be said for the centered version of the mean polynomial kernel (2) versus the Projection kernel, by using the centered covariance matrices. Although the first dilemma of the Projection kernel has been addressed by Hamm and Lee [6] by extending the kernel, resulting to the scaling of information of each dimension in linear subspaces and their preservation, data on the orthogonal complement are still overlooked. As with all dimension reduction techniques, there is always a risk of losing information when employing the Grassmann kernel. Though the hope is to retain the dimensions that are most discriminant, dimension number selection must be done with care and has become a critical stage in the implementation process. Furthermore, implementation via eigenvalue decomposition adds to the computational cost of k_{PROJ}, and also k_{BC}, giving the mean polynomial kernel an efficiency advantage, especially when presented with very high dimensional data.

6. Experiments and Results

We evaluate the performance of the mean polynomial kernel in binary classification tasks using data with underlying subspace structures. Techniques using the Grassmann kernels and Grassmann Distance Mutual Subspace method (GD-MSM) were also performed for comparison.

6.1 Face Membership Authentication

An important application of face recognition is face membership verification. The goal of this operation is to determine whether a subject is a 'member' or not. Moreover, we can also extend this to determining whether the given query is the authorized user or owner, which are common situations in accessing secured buildings or offices, logging on to computers, unlocking mobile phones, availing of online services, and other access control systems. The task can easily be modeled as a binary classification problem. For this purpose, we attempt to classify image sequences extracted from videos. The data was from the MOBIO database [16], and contains videos data taken from 152 persons, each having 12 video sessions divided into two: 6 sessions for Phase 1, and 6 sessions for Phase 2. Only data from 25 subjects and the 6 sessions from Phase 1 were used for the face membership verification task. Each session contains 21 image sequences of varying length. For the experiments, we set the sequence length to 25 images, where each image is a cropped face image of the subject, obtained using a face detection program, transformed to gray scale and resized to 25 × 25 pixels. Among the 25 subjects, 10 were randomly selected and labeled as 'member' (+1), and the remaining 15 as 'nonmember' (−1).

Two methods were employed: one using kernels with SVM and the other one using GD-MSM. For the first method, three types of kernel functions were utilized: the Grassmann kernels, Projection (PROJ) and Binet-Cauchy (BC) kernels, and the mean polynomial kernel (MP). For the GD-MSM, eight metrics were used for comparison: average distance, Binet-Cauchy metric, Geodesic distance, maximum correlation, minimum correlation, Frobenius norm based Procrustes distance, 2-norm based Procrustes distance, and Projection metric, as defined in [25]. For the SVM setting, 6-fold cross-validation was employed to evaluate the performance of the kernels such that one session per subject is used as test data while the remaining five sessions are used for training. On the other hand, class-wise (GDMSM-CD) and subject-wise (GDMSM-SD) dictionaries were implemented for the GD-MSM, as described in Section 3.

As for the parameters of the kernel methods, the value of q for the MP kernel was varied from 1 to 5, while the dimension of the subspace, m, was varied from 1 to 10. The regularization parameter C for SVM was varied over the set $\{10^0, 10^1, 10^2, 10^3, 10^4, 10^5\}$. To optimize the tuning of the said

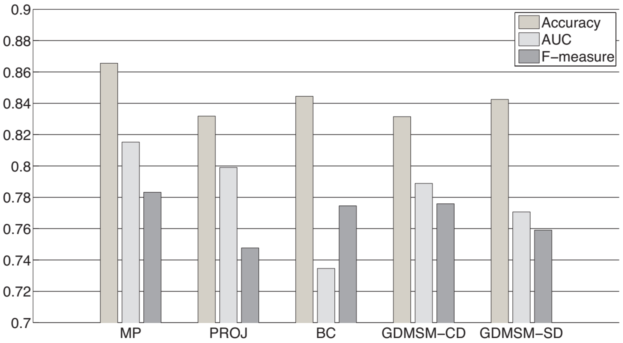


Fig. 4: Average performance of all methods for the face membership authentication task. The bar plot represents the average accuracy, average AUC, and average F-measure values computed.

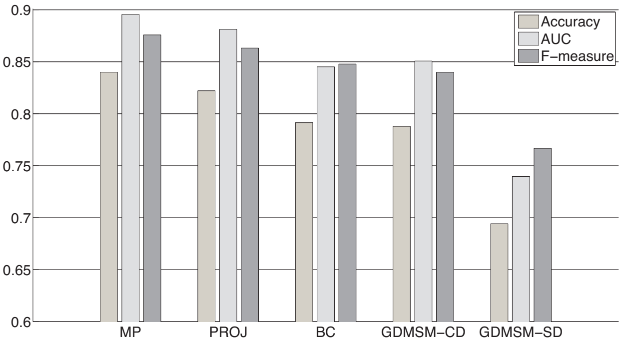


Fig. 5: Average performance of all methods for the EEG signal task classification. The bar plot represents the average accuracy, average AUC, and average F-measure values computed.

parameters, we implemented a 3-fold cross-validation grid search of the pairs (q, C) and (m, C) on the training data, for each cross-validation set. Values of the pairs were chosen such that the highest accuracy value is obtained. Variation and selection of the value of m for GDMSM was also done in a similar manner. The area under the ROC curve (AUC), accuracy, and F-measure values were considered for evaluating the performance of each method.

Figure 4 illustrates the average accuracy, AUC, and F-measure values of each method for all 6 cross-validation sets. From the graph, it is evident that the MP kernel outperforms the other methods on all three benchmarks (with accuracy, AUC and F-measure values of 81.5%, 0.866, and 0.783, respectively). Meanwhile, PROJ, BC, and GDMSM-CD obtained the second best accuracy (79.9%), AUC (0.845), and F-measure (0.776), respectively. The values presented here for the two GDMSM's are the highest obtained among all eight metrics used, which, interestingly, is the maximum correlation. We can therefore conclude that the method employing the MP kernel plus SVM is better than the GD-MSM regardless of the selected metric.

6.2 EEG Signal Task Classification

We also compared the performances of MP, PROJ, BC, GDMSM-CD and GDMSM-SD on the BCI competition III-IVa dataset [1]. The data contains recorded measurements of five subjects (aa, al, av, aw, and ay) during motor imagery tasks (right hand and right foot movement) using 118 channels of electrodes. The EEG signals were recorded for 3.5 seconds with 1000 Hz sampling rate for each trial. However, we used the available downsampled version (at 100 Hz) of the data, and utilized the 0.5 to 3.5-second interval from the visual cues for each trial, resulting to a time range of 3.0 sec per trial. For data preprocessing, frequency band selection was done, and data was filtered between frequencies of 10 to 35 Hz. For each subject, 140 trials were conducted for each task, for a total of 280 trials per subject. Settings similar to the previous application were applied to the experiments using BCI data, including the approach on parameter selection.

The average values of the performance evaluators for all subjects over the 5 cross-validation sets are given in Figure 5. As expected, the MP kernel bests the other approaches, with accuracy, AUC, and F-measure values of 84.0%, 0.896, and 0.876, respectively. This is followed by the PROJ method, with values 82.2%, 0.881, and 0.863, respectively. The GD-MSM results are also of the best performing metric, which in this case is also the maximum correlation. Hence, in a parallel logic to the previous experiments, we also conclude that the proposed method surpasses the GD-MSM approach for this task, irrespective of the metric used.

6.3 Efficiency Comparison

We investigate the time complexity of the MP kernel, and compare it with the Grassmann kernels. Suppose we are given n_{tra} number of training samples, and n_{sv} number of support vectors. For simplicity, we will assume that every (feature) vector sequence has length ℓ, and that each vector has length d. Moreover, we denote the dimension of the principal subspace as m for the Grassmann kernels, and let $k = \min(\ell, d)$. In Table 1, we give the computation time for each step in the calculation of the kernels. From this table, we conclude that the MP kernel is not only better in terms of performance, but it is also more efficient in terms of computational cost compared to the Grassmann kernels. This was confirmed empirically, as the average CPU time recorded for the MP kernel, for any value of q, is around 383 seconds for the MOBIO data, and 58 sec for the EEG data. On the other hand, computation of both Grassmann kernel matrices is around 1.21×10^4 sec when $m = 5$, and 1.24×10^4 sec for PROJ, and 1.25×10^4 sec for BC when $m = 10$, using the MOBIO data. On the EEG data, CPU time of PROJ is about 1.20×10^3 for any m, while the BC takes 1.22×10^3 and 1.23×10^3 when $m = 5$ and $m = 10$, respectively. It is also worth mentioning that should the number of features d increase, the computational time for the Grassmann kernels will drastically increase, whereas the increase with the MP kernel is only linear.

7. Conclusion

We have examined mean polynomial kernel as a kernel for binary classification of data modeled as vector sets or sequences. Analogy and connection to related methods, Grassmann Projection kernel in particular, have also been drawn. The effectiveness

Table 1: Time complexity comparison of the kernels.

<table>
<tr><th></th><th>Training Stage (For kernel matrix computation)</th><th>Testing Stage (For prediction of a single sequence)</th></tr>
<tr><td>MP</td><td>$O(n_{\mathrm{tra}}^2\ell^2 d\log_2 q)$</td><td>$O(n_{\mathrm{sv}}\ell^2 d\log_2 q)$</td></tr>
<tr><td rowspan="6">PROJ</td><td colspan="2">For covariance matrix computation</td></tr>
<tr><td>$O(d^2\ell n_{\mathrm{tra}})$</td><td>$O(d^2\ell)$</td></tr>
<tr><td colspan="2">Eigendecomposition</td></tr>
<tr><td>$O(k^3 n_{\mathrm{tra}})$</td><td>$O(k^3)$</td></tr>
<tr><td colspan="2">Kernel value computation</td></tr>
<tr><td>$O(dm^2 n_{\mathrm{tra}}^2)$</td><td>$O(d^2 m n_{\mathrm{sv}})$</td></tr>
<tr><td rowspan="6">BC</td><td colspan="2">For covariance matrix computation</td></tr>
<tr><td>$O(d^2\ell n_{\mathrm{tra}})$</td><td>$O(d^2\ell)$</td></tr>
<tr><td colspan="2">Eigendecomposition</td></tr>
<tr><td>$O(k^3 n_{\mathrm{tra}})$</td><td>$O(k^3)$</td></tr>
<tr><td colspan="2">Kernel value computation</td></tr>
<tr><td>$O(m^3 n_{\mathrm{tra}}^2)$</td><td>$O(d^2 m n_{\mathrm{sv}})$</td></tr>
</table>

of the MP kernel was empirically supported using data of face image sequences, and motor imagery EEG recordings. Furthermore, we present a comparison of computational costs between methods, and some interesting extensions of the MP kernel by considering the probabilistic distribution of the data. In brief, the mean polynomial kernel excels known methods from literature, both in performance and efficiency. In addition to the performed experiments, application to data vector sets in a multi classification problem setting may prove to be an interesting direction.

References

[1] Blankertz, B., Müller, K., Krusienki, D., Schalk, G., Wolpaw, J., Schlögl, A., Pfurtscheller, G., Millán, J., Schröder, M. and Birbaumer, N.: The BCI competition III: Validating alternative approaches to actual BCI problems, *IEEE Trans. Neural Sys. Rehab. Eng.*, Vol. 14, No. 2, pp. 153–159 (2006).

[2] Desobry, F., Davy, M. and Fitzgerald, W.: A class of kernels for sets of vectors, *Proc. 13th ESANN*, pp. 461–466 (2005).

[3] Fukui, K. and Yamaguchi, O.: Face recognition using multi-viewpoint pattern for robot vision, *Proc. Int. Symp. Robotics Research*, pp. 192–201 (2003).

[4] Gärtner, T., Flach, P., Kowalczyk, A. and Smola, A.: Multi-instance kernels, *Proc. 19th ICML*, pp. 179–186 (June 2002).

[5] Grauman, K. and Darrell, T.: The pyramid match kernel: Discriminative classification with sets of image features, *Proc. IEEE ICCV*, Vol. 2, Beijing, China, pp. 1458–1465 (October 2005).

[6] Hamm, J. and Lee, D.: Extended Grassmann kernels for subspace-based learning, *Proc. NIPS*, pp. 601–608 (2008).

[7] Hamm, J. and Lee, D.: Grassmann dicriminant analysis: a unifying view on subspace-based learning, *Proc. 25th ICML*, pp. 376–383 (2008).

[8] Kashima, H., Tsuda, K. and Inokuchi, A.: Kernels for graphs, *Kernels and Bioinformatics*, Cambridge, MA, USA, MIT Press, pp. 155–170 (2004).

[9] Kim, T.-K., Kittler, J. and Cipolla, R.: Discriminative learning and recognition of images set classes using canonical correlations, *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. 29, pp. 1005–1018 (2007).

[10] Kim, T.-K., Kittler, J. and Cippolla, R.: Discriminative learning and recognition of image set classes using canonical correlations, *IEEE Trans. Pattern Anal. Mach. Intell.*, pp. 1005–1018 (2007).

[11] Kondor, R. and Jebara, T.: A kernel between sets of vectors, *Proc. 20th ICML*, pp. 361–368 (August 2003).

[12] Kondor, R. and Lafferty, J.: Diffusion kernels on graphs and other discrete input spaces, *Proc. 19th ICML*, pp. 315–322 (July 08-12, 2002).

[13] Kuboya, T., Hirata, K. and Aoiki-Kinoshita, K.: An efficient unordered tree kernel and its application to glycan classification, *Proc. PAKDD*, pp. 184–195 (2008).

[14] Leslie, C., Eskin, E. and Noble, W.: The spectrum kernel: A string kernel for SVM protein classification, *Proc. of the Pacific Symposium on Biocomputing*, pp. 564–575 (2002).

[15] Lodhi, H., Saunders, C., Shawe-Taylor, J., Cristianini, N. and Watkins, C.: Text classification using string kernels, *J. Mach. Learn. Res.*, Vol. 2, pp. 419–444 (2002).

[16] McCool, C., Marcel, S., Hadid, A., Piettikainen, M., Matejka, P., Cernocky, J., Poh, N., Kittler, J., Larcher, A., Levy, C., Matrouf, D., Bonastre, J.-F., Tresadern, P. and Cootes, T.: Bi-modal Person Recognition on a mobile phone: Using mobile phone data, *Proc. IEEE ICME Workshop on Hot Topics in Mobile Multimedia*, pp. 635–640 (2012).

[17] Müller-Gerking, J., Pfurtscheller, G. and Flyvbjerg, H.: Designing optimal spatial filters for single-trial EEG classification in movement task, *Clinical Neurophysiology*, Vol. 110, No. 5, pp. 787–798 (1999).

[18] Neumann, M., Patricia, N., Garnett, R. and Kersting, K.: Efficient graph kernels by randomization, *Proc. ECML/PKDD*, pp. 378–393 (2012).

[19] Qiu, S., Lane, T. and Buturovic, L.: A randomized string kernel and its application to RNA interference, *Proc. AAAI*, pp. 627–632 (2007).

[20] Relator, R., Hirohashi, Y., Ito, E. and Kato, T.: Mean polynomial kernel and its application to vector sequence recognition, *IEICE Trans. Inf. and Syst.*, Vol. 97-D, No. 7, pp. 1855–1863 (2014).

[21] Sakano, H. and Mukawa, N.: Kernel mutual subspace method for robust facial image recognition, *Proc. Int. Conf. on Knowledge-Based Intell. Eng. Sys. And App. Tech*, pp. 245–248 (2000).

[22] Schölkopf, B. and Smola, A.: *Learning with kernels: Support vector machines, regularization, optimization, and beyond*, MIT Press, Cambridge, MA, USA (2001).

[23] Shawe-Taylor, J. and Cristianini, N.: *Kernel methods for pattern analysis*, Cambridge University Press New York, NY, USA (2004).

[24] Shi, Q., Petterson, J., Dror, G., Langford, J., Smola, A. and Vishwanathan, S.: Hash kernels for structured data, *J. Mach. Learn. Res.*, Vol. 10, pp. 2615–2637 (2009).

[25] Shigenaka, R., Raytchev, B., Tamaki, T. and Kaneda, K.: Face sequence recognition using Grassmann Distances and Grassmann Kernels, *Proc. IJCNN*, pp. 2630–2636 (2012).

[26] Vishwanathan, S., Schraudolph, N., Kondor, R. and Borgwardt, K.: Graph kernels, *J. Mach. Learn. Res.*, Vol. 11, pp. 1201–1242 (2010).

[27] Vishwanathan, S. and Smola, A.: Fast kernels for string and tree matching, *Advances in Neural Info. Proc. Sys.*, Vol. 15, pp. 569–576 (2003).

[28] Vishwanathan, S. and Smola, A.: Binet-Cauchy kernels, *Proc. NIPS*, pp. 1441–1448 (2004).

[29] Wolf, L. and Shashua, A.: Learning over sets using kernel principal angles, *J. Mach. Learn. Res.*, Vol. 4, pp. 913–931 (2003).

[30] Yamaguchi, O., Fukui, K. and Maeda, K.: Face recognition using temporal image sequence, *Proc. Int. Conf. Automatic Face and Gesture Recognition*, pp. 318–323 (1998).