Activity Recognition to Traditional Dances using Dimensionality Reduction

Vasileios Gavriilidis and Anastasios Tefas

Aristotle University of Thessaloniki, Department of Informatics,
Thessaloniki, Greece
vgavril@csd.auth.gr, tefas@aiia.csd.auth.gr

Abstract. Activity recognition is a complex problem mainly because of the nature of the data. Data usually are high dimensional, so applying a classifier directly to the method data is not always a good practice. A common method is to find a meaningful representation of complex data through dimensionality reduction. In this paper we propose novel kernel matrices based on graph theory to be used for dimensionality reduction. The proposed kernel can be embedded in a general dimensionality reduction framework. Experiments on a traditional dance recognition dataset are conducted and the advantage of using dimensionality reduction before classification is highlighted.

Keywords: Random Walk Kernel, Activity Recognition, Dimensionality Reduction, Support Vector Machines

1 Introduction

Activity recognition is an important and active area of computer vision research. Video surveillance and video annotation are two fields that use activity recognition of everyday actions such as walking, running and sitting. Surveys of activity recognition approaches can be found in [5, 11, 14]. The importance of generic activity recognition relies on the fact that it can be applied to many real-life problems, with most of them being human–centric [4]. An example of the activity recognition problem is dance recognition [6]. Dance recognition is a difficult problem, since it involves the movement of the body in a specific way that characterise a specific dance. Dances are performed in many cultures to express ideas or tell a story, which suggests their importance especially for countries with long history, such us Greece.

Dance recognition problems usually begin with video recordings and with labelling a video. In order for those videos to be used in a classifier, features need to be extracted. A commonly used framework for transferring the problem from video recording to a feature space, where all data have the same dimensions, is the bag–of–features approach [10]. Such methods for feature extraction from videos are *STIP*, *TRAJ* and *ISA* which have been proposed in [8, 15, 9], respectively.

Another example of feature extraction is dimensionality reduction used for transferring a feature space to a lower feature space. Dimensionality reduction is a commonly used preprocessing step in machine learning, especially when dealing with a high dimensional space of features which can enhance data separability [7]. Many methods for dimensionality reduction exist; the most common of them are embedded in the framework described in [16]. Using the aforementioned framework we can retain or avoid specific statistical or geometrical properties of our data. For this reason a graph is created, specifically a k–nearest neighbour graph.

Graph properties are described extensively in [2]. One of the properties of a graph, that will be used in this paper, is that if $\mathbf{W}$ represents the adjacency matrix between nodes, where $W(i,j) = 1$ if nodes i and j are connected and $W(i,j) = 0$ otherwise, then the ij–th element of the p–th power of adjacency matrix, $W^p(i,j)$, gives the number of paths of length p between nodes i and j. This notion can be applied to either directed or undirected graphs and can also be extended to weighted graphs, $W(i,j) \in [0, \inf)$. In this paper, we propose the use of the number of paths between two samples of the dataset as a similarity that can be embedded in a general dimensionality reduction framework as it will be explained in the following Sections.

The structure of the paper is as follows: In Section 2 we describe previous work and state the problem we solve. We then introduce our method for dimensionality reduction in Section 3, providing some theoretical background. In Section 4 we explain the way we conducted our experiments and present classification results on traditional dance recognition. We also show some interesting dimensionality reduction projections. Finally we give concluding remarks and discussion of future work in Section 5.

2 Prior work and problem Statement

Usually, activity recognition datasets consist of videos. Firstly, the bag–of–features approach [10] is typically performed and later a codeword is created by applying k–means to the extracted features. The last step is to map each recording video to a certain codebook and, thus, the original recording can be represented as a histogram of codewords. Depending on the number of centres of k–means a different codeword is produced and, hence, a different representation for each recording is created. Let the recordings represented as histograms of codewords be the data matrix $\mathbf{X}$.

Techniques for dimensionality reduction have always attracted interest in computer vision and pattern recognition. Graph embedding [16] provides a general framework for dimensionality reduction and many algorithms can be integrated into this framework. Let an undirected weighted graph $G\{X, W\}$ be defined as a vertex set X and similarity matrix W whose entries can be positive, negative or zero. Also let a diagonal matrix $\mathbf{D}$ be constructed as:

$$D_{ii} = \sum_{i \neq j} W_{ij}, \tag{1}$$

and Laplacian matrix as:

$$\mathbf{L} = \mathbf{D} - \mathbf{W}, \tag{2}$$

The aim of graph embedding is to find a procedure where desired characteristics between nodes of the graph are preserved and undesired properties of the data are suppressed after dimensionality reduction. Hence, a penalty graph is also defined, whose entries are to be suppressed in the new feature space. Graph embedding framework requires the solution to the generalized eigenvalue decomposition problem:

$$\tilde{\mathbf{L}}\mathbf{U} = l\tilde{\mathbf{B}}\mathbf{U}, \tag{3}$$

where $\tilde{\mathbf{L}} = \mathbf{L}, \mathbf{XLX}^T$ or $\mathbf{KLK}$ and $\tilde{\mathbf{B}} = \mathbf{I}, \mathbf{B}, \mathbf{K}, \mathbf{XBX}^T$ or $\mathbf{KBK}$ depending on the dimensionality reduction algorithm used. After calculating the matrix $\mathbf{U}$, we choose those eigenvectors that correspond to the smallest eigenvalues of l.

3 Proposed Dimensionality Reduction Method

In the case of LPP [3] and more specifically the Kernel version of LPP, the substitution is $\tilde{\mathbf{L}} = \mathbf{KLK}$ and $\tilde{\mathbf{B}} = \mathbf{KBK}$. This suggests that the similarities in the new space will be comparable to the similarities of the original space after the transformation of the data through the kernel function $\Phi(.)$.

Even though kernel LPP typically uses the RBF kernel, this is not mandatory and any matrix can be chosen as long as it is a kernel. There are various kernel functions that can be used; linear, polynomial, RBF and sigmoid are some examples. Another kind of kernels, are random walk kernels which were first proposed in [13] and later were used as kernel matrices for semi–supervised learning using cluster kernels [1] .

Random walk kernel based on [1] is computed in two steps. First, RBF kernel matrix is computed and then, each value is normalised by the sum of its row. The resulted matrix can also be seen as a transition matrix of a random walk on a graph. This suggests probability of starting from one point and arriving at another. Using a diagonal matrix defined as in equation (1) the transition matrix has the form of:

$$\mathbf{P} = \mathbf{D}^{-1}\mathbf{K}. \tag{4}$$

thus the matrix $\mathbf{P}^p = (\mathbf{D}^{-1}\mathbf{K})^p$ can be interpreted as transition probability after p steps. Unfortunately, matrix $\mathbf{P}^p$ is not symmetric, hence it can not be used as a kernel.

Another example of random walk kernel is introduced in [12]. Assume that we have the adjacency matrix $\mathbf{W}$, with $W_{ij} = 1$ if samples i and j are neighbours and zero otherwise, and the Normalised Laplacian:

$$\tilde{\mathbf{L}} = \mathbf{D}^{-\frac{1}{2}}\mathbf{L}\mathbf{D}^{-\frac{1}{2}}. \tag{5}$$

the p–step random walk kernel is computed as:

$$\mathbf{K} = (a\mathbf{I} - \tilde{\mathbf{L}})^p, \text{ with } a \geqslant 2. \tag{6}$$

In general $\mathbf{W}$ has no restriction about the graphs it can be applied to. In addition, parameter a ensures positive definiteness of $\mathbf{K}$.

Starting from the simplest kernel, which is inner product, we propose a random walk kernel for dimensionality reduction. The inner product, expresses the similarity between i-th and j-th sample and is defined as:

$$W(i,j) = \mathbf{x}_i^T \mathbf{x}_j. \tag{7}$$

Let i–th and j–th samples be represented as nodes in an unweighted graph with $W(i,j) = 0$ meaning samples are not similar and $W(i,j) = 1$ meaning samples are similar.

We may now propose the similarity matrix:

$$\mathbf{W}^p = \underbrace{\mathbf{W}\mathbf{W}\ldots\mathbf{W}}_{\text{p times}}. \tag{8}$$

We can say that $W^p(i,j)$ expresses the similarity between i–th and j–th samples after visiting all possible paths passing from $p-1$ in–between similar samples.

Extending the notion of the discrete values of similar and not similar (0 and 1) to continuous values, we define a relaxed definition of a weighted graph which can take values in $[0 - \inf)$, where 0 is the least similar and inf is the most similar. This way, when two samples' similarity is computed, more paths are approachable, since the only paths that are not viable are those that pass from an intermediate sample that has zero similarity. In reality, every single path is involved because even though the similarity of two samples can be small, it is rarely zero. For example, the similarity matrix passing from one intermediate sample can be computed as $\mathbf{W}^2 = \mathbf{X}^T\mathbf{X}\mathbf{X}^T\mathbf{X}$.

Without loss of generality, we assume that data matrix has zero mean, hence $\mathbf{X}\mathbf{X}^T = \mathbf{\Sigma}$, where $\mathbf{\Sigma}$ is the covariance matrix, thus $\mathbf{W}^2 = \mathbf{X}^T\mathbf{\Sigma}\mathbf{X}$ also holds. Moreover, it is straightforward to show that $\mathbf{W}^p = \mathbf{X}^T\mathbf{\Sigma}^{p-1}\mathbf{X}, \text{ with } p \geqslant 1$.

So, $\mathbf{W}^p$ is a similarity matrix and $W^p(i,j)$ expresses the similarity of two samples beginning from the i–th sample and ending at the j–th sample after passing through $p-1$ intermediate samples. The goal is to connect similar nodes by several paths. Even if $\mathbf{W}^p$ is a similarity matrix, this does not necessarily mean that it can be used as a kernel matrix. We now prove that apart from $\mathbf{W}$, which is by definition a kernel matrix, $\mathbf{W}^p$ is also a kernel matrix.

It is safe to replace $\mathbf{W}$ by $\mathbf{K}$ since $\mathbf{W}$ is positive definite. $\mathbf{K}$ has an eigenvalue decomposition $\mathbf{K} = \mathbf{U}^T\mathbf{\Lambda}\mathbf{U}$, where $\mathbf{U}$ is an orthogonal matrix and $\mathbf{\Lambda}$ is a diagonal matrix of real and positive eigenvalues, that is, $\mathbf{\Lambda} = diag(\lambda_1, \lambda_2, \dots, \lambda_D)$. So, now $\mathbf{W}^p$ can be written as:

$$\begin{aligned}\mathbf{W}^p &= \underbrace{\mathbf{K}\mathbf{K}\dots\mathbf{K}}_{\text{p times}} \\ &= \mathbf{U}^T\mathbf{\Lambda}\underbrace{\mathbf{U}\mathbf{U}^T}_{I}\mathbf{\Lambda}\underbrace{\mathbf{U}\mathbf{U}^T}_{I}\dots\underbrace{\mathbf{U}\mathbf{U}^T}_{I}\mathbf{\Lambda}\mathbf{U} \\ &= \mathbf{U}^T\underbrace{\mathbf{\Lambda}\dots\mathbf{\Lambda}}_{\text{p times}}\mathbf{U} \\ &= \mathbf{U}^T\mathbf{\Lambda}^p\mathbf{U}. \end{aligned} \tag{9}$$

Since eigenvalues $\lambda_i > 0$, $\forall i = 1, \dots, N$ then $\lambda_i^p > 0$, $\forall i = 1, \dots, N$, which leads to $\mathbf{x}\mathbf{W}^p\mathbf{x}^T \geq 0, \forall \mathbf{x}$ which is the definition of a positive definite matrix. Notice that no assumptions were made for the original kernel matrix. Thus, in general every kernel matrix elevated to any power is also a kernel matrix.

Now, $\mathbf{W}^p$ can safely be used as a Kernel. Moreover, when inner product is used as the initial kernel matrix, and assuming data have zero mean, we arrive at an interesting property. The covariance matrix, $\mathbf{\Sigma}$, is symmetric and has real values, so it has an eigenvalue decomposition that can be written as:

$$\mathbf{\Sigma} = \mathbf{U}\mathbf{D}\mathbf{U}^T. \tag{10}$$

Hence:

$$\begin{aligned}\mathbf{W}^p =& \mathbf{X}^T\mathbf{\Sigma}^{p-1}\mathbf{X} \\ =& \mathbf{X}^T\mathbf{U}\mathbf{D}^{p-1}\mathbf{U}^T\mathbf{X} \\ =& (\mathbf{D}^{\frac{p-1}{2}}\mathbf{U}^T\mathbf{X})^T(\mathbf{D}^{\frac{p-1}{2}}\mathbf{U}^T\mathbf{X}). \end{aligned} \tag{11}$$

So, kernel $\mathbf{W}^p$ can be calculated differently by multiplying data matrix $\mathbf{X}$ by $\mathbf{D}^{\frac{p-1}{2}}\mathbf{U}^T$. The calculation of inner product of the transformed data matrix with itself yields the same results as when using original data and $\mathbf{W}^p$.

RBF kernel is defined as:

$$K(\mathbf{x}_i, \mathbf{x}_j) = e^{-\frac{|\mathbf{x}_i - \mathbf{x}_j|^2}{2\sigma^2}}, \tag{12}$$

so $K^p(\mathbf{x}_i, \mathbf{x}_j), p \geqslant 1$ can be expressed as:

$$\begin{aligned}K^p(\mathbf{x}_i, \mathbf{x}_j) &= \sum_{l_1}^{N}\cdots\sum_{l_{p-1}}^{N} e^{-\frac{\left|\mathbf{x}_i - \mathbf{x}_{l_1}\right|^2}{2\sigma^2}}\dots e^{-\frac{\left|\mathbf{x}_{l_{p-1}} - \mathbf{x}_j\right|^2}{2\sigma^2}} \\ &= \sum_{l_1}^{N}\cdots\sum_{l_{p-1}}^{N} e^{-\frac{\left|\mathbf{x}_i - \mathbf{x}_{l_1}\right|^2 + \dots + \left|\mathbf{x}_{l_{p-1}} - \mathbf{x}_j\right|^2}{2\sigma^2}}. \end{aligned} \tag{13}$$

By examining equation (13) we observe that the distance between two nodes is relative to the whole structure of the graph, since in order to compute the distance of two nodes, all the nodes of the graph are taken into account which resembles a graph based distance. The property we would like to retain is for the number of all possible paths after p steps to be the same after dimensionality reduction.

Finally, kernel LPP keeps the similarities between samples the same, after the dimensionality reduction by using $\mathbf{K}$. Using this notion, we similarly use $\mathbf{K}^p$ to keep the similarities after visiting all possible paths passing through $p-1$ intermediate samples. In order to achieve this, equation (3) is used to embed our proposed method to the framework using $\tilde{\mathbf{L}} = \mathbf{K}^p\mathbf{L}\mathbf{K}^p$ and $\tilde{\mathbf{B}} = \mathbf{K}^p\mathbf{B}\mathbf{K}^p$. Notice that like kernel LPP our proposed method is unsupervised and the labels of the data are not required.

4 Experimental Results

We performed classification to a dataset of Greek traditional dances. The dataset consists of 10 videos of 5 Greek traditional dances, the *Lotzia*, the *Capetan Loukas*, the *Ramna*, the *Stankaina* and finally the *Zablitsaina*. In more detail, two professional dancing groups were recorded dancing. Each traditional dance was performed twice, once indoor by one group and once outdoor by another group. In Figure 1, two frames of two different videos are illustrated.

The 5 recordings of indoor were used for training and the outdoor recordings were used for testing. In order, to transform the video recordings to feature vectors we have extracted *ISA STIP* and *TRAJ*. Using overlapping clips of 80 to 100 frames we ended up with 78, 113, 110, 95 and 101 clips for each training video and for each testing video to 102, 107, 110, 106 and 91 clips, resulting to 496 and 516 short sequences overall, respectively. Also we have created a dataset consisting of only 8 videos out of 10, without the dance *Zablitsaina*. The different representations of the traditional dance dataset characteristics are depicted in Table 1.

(a) Stankaina dance performed by a professional group

(b) Stankaina dance performed by another professional group with different costumes

Fig. 1. Sample Frame of Two Videos.

Table 1. Representations of Traditional Dances Dataset

Extracted features	Train Samples	Test Samples	Clusters	Classes
isa.10	497	516	10	5
isa.100	497	516	100	5
stip.10	497	516	10	5
stip.100	497	516	100	5
stip.1000	497	516	1000	5
stip.2000	497	516	2000	5
traj.10	497	516	10	5
traj.100	497	516	100	5
traj.1000	497	516	1000	5
isa4.100	396	425	100	4
isa4.1000	396	425	1000	4

All representations were scaled to the interval $[0, 1]$. We conducted experiments using SVM with the linear kernel and the RBF kernel. The evaluation of parameters was performed by using grid search of 5–fold cross validation. Using the best parameters found on training set, we trained once more using the entire training dataset in order to predict the classes of the test set. More specifically, we trained with exponentially growing sequences of $C \in \{2^{-5}, \ldots, 2^{15}\}$ and $\gamma \in \{2^{-15}, \ldots, 2^{3}\}$. Obviously, inner product uses only parameter C and the RBF kernel uses both parameters C and γ.

Apart from SVM, we also need to choose a kernel for the dimensionality reduction method that was proposed. We chose to evaluate dimensionality reduction using the kernels as described in equations (8) and (13). In addition, our proposed method requires the selection of the parameter p. Thus, we used a procedure for finding automatically the parameter p similarly to grid search. Assume, we want to find the parameter p when our data is projected onto a 2–dimensional space. We produce all different projections of original data using $p = \{1, \ldots, 11\}$, then for each projection we perform grid search, looking for parameters of the SVM, while we also keep the one of 11 projections that attained the best performance on the grid search.

As illustrated in Tables 2–5, the proposed dimensionality reduction method improves the performance of SVM. For the recognition of the 5 traditional dances, regardless of which features were extracted (*ISA STIP* and *TRAJ*), and also regardless of which kernel was used for classification the best SVM result attained was 51.74%. On the other hand, projecting data first to a lower space and then using SVM the best classification performance was 58.33%. Moreover, in the smaller dataset of 4 traditional dances, the best result, using SVM was 63.53% but using our proposed dimensionality reduction technique before classification improves classification accuracy to 89.18%. This means that the structure of the data can be represented in a lower dimensional space more effectively. For example, in Figure 2 some projections are illustrated, highlighting the structure of data with different values of p. It is obvious that different val-

Table 2. Inner Product SVM + (Inner Product)^p

Dataset	SVM	Dimensions									
		1	2	3	4	5	6	7	8	9	10
isa.10	23.64	**33.53**	29.65	30.81	30.81	29.46	26.74	24.22	17.05	13.76	13.57
isa.100	40.12	**57.36**	33.53	40.31	33.33	49.61	39.53	23.45	34.30	32.95	34.30
stip.10	32.36	39.15	**40.12**	34.50	34.11	34.88	34.88	35.08	33.53	34.69	34.69
stip.100	33.72	36.43	36.24	32.75	**37.60**	35.66	35.27	31.40	32.17	22.48	23.06
stip.1000	36.05	34.30	33.72	**37.60**	30.04	30.23	29.65	29.65	29.65	29.07	35.66
stip.2000	30.43	35.66	33.14	28.68	28.29	28.29	32.56	34.11	37.79	38.18	**41.28**
traj.10	32.17	19.77	20.93	28.29	29.84	**32.36**	29.84	30.23	31.40	31.40	30.81
traj.100	32.75	20.16	27.71	34.50	30.81	33.72	34.50	34.50	35.47	36.63	**37.02**
traj.1000	**44.19**	27.52	25.19	41.47	43.22	43.41	43.80	35.85	38.57	40.89	39.92
isa4.100	54.59	30.12	37.65	48.94	56.47	68.71	71.29	76.47	31.06	31.06	**77.18**
isa4.1000	63.53	49.18	63.76	43.29	72.00	83.06	**89.18**	85.41	85.65	82.82	82.59

Table 3. RBF SVM + (Inner Product)^p

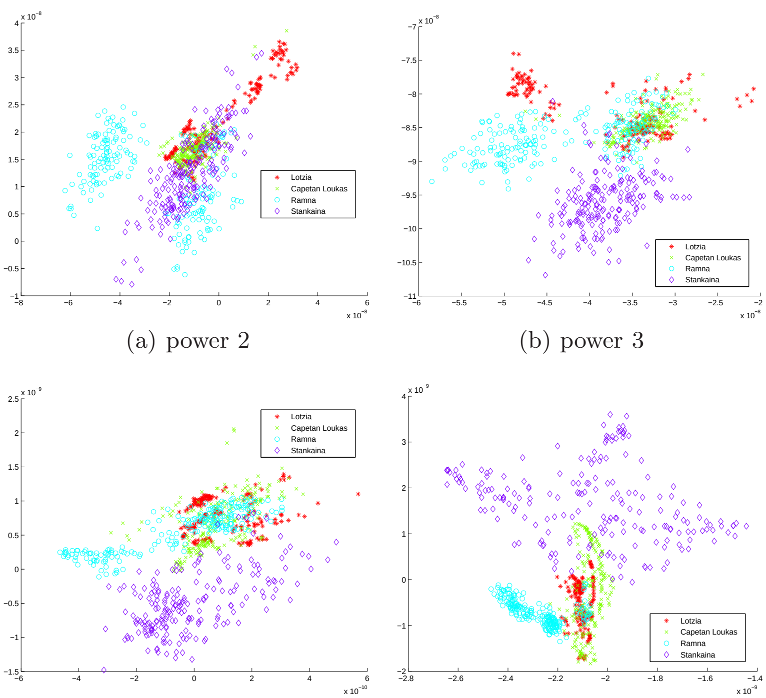
Dataset	SVM	Dimensions									
		1	2	3	4	5	6	7	8	9	10
isa.10	**40.70**	40.50	34.11	31.59	27.33	24.61	22.09	29.65	26.94	21.51	21.32
isa.100	51.74	50.58	54.26	50.97	**58.33**	43.22	38.18	27.91	27.52	27.52	18.60
stip.10	27.13	35.08	34.69	30.04	32.36	34.11	**37.21**	33.53	31.40	33.33	35.66
stip.100	21.32	**40.89**	37.21	31.59	29.46	27.91	27.33	32.56	29.65	28.88	32.56
stip.1000	22.09	26.94	**34.30**	32.17	28.68	29.84	28.29	27.71	29.26	29.07	23.26
stip.2000	22.67	21.71	28.29	29.65	29.46	31.01	31.78	29.26	25.19	31.01	**32.36**
traj.10	24.22	19.19	16.67	15.50	13.76	13.18	30.43	**36.82**	36.63	36.63	36.63
traj.100	20.74	21.51	15.12	22.09	20.54	20.74	21.71	21.12	23.06	25.97	**28.88**
traj.1000	22.67	23.26	33.33	31.59	36.63	35.27	37.40	**39.92**	**39.92**	35.08	34.69
isa4.100	54.82	21.88	24.00	49.88	**65.18**	24.24	50.59	52.94	50.59	50.12	49.18
isa4.1000	30.82	49.41	62.82	48.71	61.41	60.71	**82.82**	82.12	81.88	78.59	79.06

Table 4. Inner Product SVM + (RBF)^p

Dataset	SVM	Dimensions									
		1	2	3	4	5	6	7	8	9	10
isa.10	23.64	12.98	20.16	20.93	20.93	13.76	20.16	**27.71**	26.94	26.94	26.74
isa.100	40.12	43.99	35.47	49.81	48.26	54.46	37.98	38.37	44.77	43.99	**57.17**
stip.10	32.36	**39.73**	37.21	33.14	35.27	35.85	36.24	39.34	36.82	36.43	36.82
stip.100	33.72	37.21	35.47	36.43	35.27	36.82	35.66	42.44	**44.96**	42.25	32.36
stip.1000	36.05	34.30	36.24	**37.21**	36.63	36.43	36.43	36.63	28.68	24.81	20.54
stip.2000	30.43	36.05	**38.18**	29.84	31.01	29.46	30.04	29.65	26.16	29.65	30.23
traj.10	32.17	31.98	30.62	32.95	31.01	30.43	31.01	34.69	**34.88**	32.17	31.59
traj.100	32.75	31.78	36.24	34.69	**37.40**	36.82	34.69	34.11	33.14	30.62	28.88
traj.1000	**44.19**	20.93	35.85	23.84	29.46	34.11	26.94	27.91	26.16	26.94	25.19
isa4.100	54.59	73.88	70.59	68.00	**77.88**	54.35	56.47	56.47	57.18	57.41	57.88
isa4.1000	63.53	78.59	74.35	50.59	68.71	63.06	64.94	82.12	80.00	83.29	**88.71**

Table 5. RBF SVM + $(\mathrm{RBF})^p$

Dataset	SVM	Dimensions									
		1	2	3	4	5	6	7	8	9	10
isa.10	**40.70**	9.69	19.96	34.11	17.83	17.64	20.35	18.80	18.60	23.64	28.49
isa.100	51.74	29.65	31.01	**52.13**	45.35	49.42	24.42	48.26	25.00	27.71	35.85
stip.10	27.13	**35.47**	33.53	31.01	33.14	32.95	19.96	17.83	29.26	33.72	32.95
stip.100	21.32	33.33	**37.98**	33.91	34.88	34.69	30.04	30.43	31.78	31.78	30.81
stip.1000	22.09	36.43	36.05	33.91	**36.82**	32.95	29.07	22.09	29.65	25.97	19.96
stip.2000	22.67	24.22	23.84	36.43	**37.21**	30.62	27.13	26.74	22.87	30.23	29.26
traj.10	24.22	28.29	21.71	25.78	19.57	24.22	25.97	30.81	**31.40**	26.16	30.43
traj.100	20.74	34.30	34.30	**38.18**	36.43	37.40	32.56	35.66	35.27	32.75	29.26
traj.1000	22.67	33.72	25.58	28.10	**34.30**	32.95	34.11	33.53	34.11	32.56	31.01
isa4.100	54.82	**73.65**	64.00	64.94	72.71	34.82	32.24	29.18	29.65	24.47	30.35
isa4.1000	30.82	77.88	38.59	64.47	68.00	62.12	64.71	70.82	**78.82**	78.59	69.65



(a) power 2 (b) power 3

(c) power 5 (d) power 7

Fig. 2. isa4.100 projections using (Inner Product)^p with different values of p

ues of the parameter p of the proposed kernel result to significantly different representations with varying discriminality among the classes.

5 Conclusions

A novel kernel has been proposed which can be embedded to a dimensionality reduction framework. The proposed kernel produces representations that highlight the separability between classes. We performed classification using SVM as a classifier to a traditional dance recognition dataset and the advantage of using dimensionality reduction, before classifying, is highlighted. In addition, some interesting projections of the data were given. Future work can be focused on performing dimensionality reduction using different initial kernels.

Acknowledgements. This research has been co–financed by the European Union (European Social Fund - ESF) and Greek national funds through the Operation Program "Education and Lifelong Learning" of the National Strategic Reference Framework (NSRF) - Research Funding Program: THALIS-UOA-ERASITECHNIS MIS 375435.

References

1. Chapelle, O., Weston, J., Scholkopf, B.: Cluster kernels for semi-supervised learning. In: Advances in Neural Information Processing Systems. p. 15 (2002)
2. Chung, F.R.K.: Spectral Graph Theory (CBMS Regional Conference Series in Mathematics, No. 92). American Mathematical Society (Dec 1996)
3. He, X., Niyogi, P.: Locality preserving projections (2002)
4. Iosifidis, A., Tefas, A., Pitas, I.: View-invariant action recognition based on artificial neural networks. Neural Networks and Learning Systems, IEEE Transactions on 23(3), 412–424 (2012)
5. Ji, X., Liu, H.: Advances in view-invariant human motion analysis: A review. Trans. Sys. Man Cyber Part C 40(1), 13–24 (Jan 2010)
6. Kapsouras, I., Karanikolos, S., Nikolaidis, N., Tefas, A.: Feature comparison and feature fusion for traditional dances recognition. In: Iliadis, L., Papadopoulos, H., Jayne, C. (eds.) Engineering Applications of Neural Networks, Communications in Computer and Information Science, vol. 383, pp. 172–181. Springer Berlin Heidelberg (2013)
7. Kyperountas, M., Tefas, A., Pitas, I.: Salient feature and reliable classifier selection for facial expression classification. Pattern Recognition 43(3), 972 – 986 (2010)
8. Laptev, I., Marszalek, M., Schmid, C., Rozenfeld, B.: Learning realistic human actions from movies. In: Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on. pp. 1–8 (2008)
9. Le, Q.V., Zou, W.Y., Yeung, S.Y., Ng, A.Y.: Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. In: Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition. pp. 3361–3368. CVPR '11, IEEE Computer Society, Washington, DC, USA (2011)

10. Leung, T., Malik, J.: Representing and recognizing the visual appearance of materials using three-dimensional textons. International Journal of Computer Vision 43(1), 29–44 (2001)
11. Poppe, R.: A survey on vision-based human action recognition. Image Vision Comput. 28(6), 976–990 (Jun 2010)
12. Smola, A.J., Kondor, R.: Kernels and regularization on graphs (2003)
13. Szummer, M., Jaakkola, T.: Partially labeled classification with markov random walks. In: Advances in Neural Information Processing Systems 15. MIT Press, Cambridge, MA (2001)
14. Turaga, P., Chellappa, R., Subrahmanian, V.S., Udrea, O.: Machine recognition of human activities: A survey. Circuits and Systems for Video Technology, IEEE Transactions on 18(11), 1473–1488 (2008)
15. Wang, H., Kläser, A., Schmid, C., Liu, C.L.: Dense trajectories and motion boundary descriptors for action recognition. International Journal of Computer Vision 103(1), 60–79 (2013)
16. Yan, S., Xu, D., Zhang, B., Zhang, H.J., Yang, Q., Lin, S.: Graph embedding and extensions: A general framework for dimensionality reduction. Pattern Analysis and Machine Intelligence, IEEE Transactions on 29(1), 40–51 (2007)