Partial Least Squares Kernel for Computing Similarities between Video Sequences

Siddhartha Chandra
CVIT, IIIT Hyderabad
siddhartha.chandra@research.iiit.ac.in

C. V. Jawahar
CVIT, IIIT Hyderabad
jawahar@iiit.ac.in

Abstract

Computing similarities between data samples is a fundamental step in most Pattern Recognition (PR) tasks. Better similarity measures lead to more accurate prediction of labels. Computing similarities between video sequences has been a challenging problem for the PR community for long because videos have both spatial and temporal context which are hard to capture. We describe a novel approach that employs Partial Least Squares (PLS) regression to derive a measure of similarity between two tensors (videos). We demonstrate the use of this tensor similarity measure along with SVM classifiers to solve the tasks of hand gesture recognition and action classification. We show that our methods significantly outperform the state of the art approaches on two popular datasets: Cambridge hand gesture dataset and UCF sports action dataset. Our method requires no parameter tuning.

1. Introduction and Prior Work

Many pattern recognition tasks involve assigning labels to unlabeled samples. Traditionally, we solve these tasks by acquiring ground truth labels for some of the data samples. Some measure of similarity between the seen and unseen samples is then used to predict the labels of the unseen samples. The similarity measure used is thus a crucial component of any Pattern Recognition problem. A lot of similarity kernels for real valued and symbolic data such as text can be found in literature [15]. However, there are few similarity kernels for videos (tensors). Devising good discriminative kernels for videos is challenging because they have both spatial and temporal context. In this paper, we take a step forward in this direction.

Quantitative similarity measures between videos can be applied to solve various pattern recognition tasks such as hand gesture recognition and action classification. These find applications in Human Computer Interaction (HCI) [10] and video surveillance [13]. Hand gesture recognition is also widely used for sign language interpretation [4]. Studies have been conducted over the years to develop systems that perform these tasks accurately.

Some of the earlier methods for hand gesture recognition have used neural networks to recognize spatio-temporal actions [16]. Others describe videos using spatial [10] and temporal models [1]. Few methods have also used Hidden Markov Models and its variants [17]. More recently, graph matching approaches have been used for gesture recognition [14]. The most notable recent approach to hand gesture recognition [5] combines Canonical Correlation Analysis (CCA) with discriminant functions and SIFT features to extract discriminative pair-wise spatio temporal features (for pairs of videos) that perform robust gesture recognition.

There has been a furore of activity in the action classification community too. Methods that use the knowledge of the geometry of the tensor space [9] for action classication factor tensors using a modified High Order Singular Value Decomposition (HOSVD) and each factorized space is recognized as a Grassmann manifold; and classification is done on this manifold. Motivated by this approach, [8] represents tensors (videos) as a tangent bundle on a Grassmann manifold and canonical distances between these tangent spaces are then used for action classification. CCA has been extended [6] for multidimensional data arrays to inspect joint space-time linear relationships of two videos and acquire similarity features of the two videos that are both flexible and descriptive. This is achieved by representing third order tensors (videos) as a set of 2-D matrices and using CCA on each of these matrices. This method further uses a discriminative feature selection scheme and a nearest neighbour classifier for action classification.

Our method is similar to [6] in the sense that we too flatten the videos (third order tensors) to get three matrices (second order tensors) per video (these three matrices are referred to as the three joint shared modes of a tensor in [6]). However, unlike [6] that uses CCA, we use PLS regression to compute similarity between the corresponding second order tensors of a video. Finally, we build classification kernels using these similarity measures and use an SVM for classification.

2. Partial Least Squares

PLS [12] is a technique for modeling relations between sets of observed variables using latent variables. PLS assumes that observed data is generated by processes that use latent variables. PLS generates orthogonal score vectors (latent vectors) using the existing correlations between two sets of random variables while preserving most of the variance of both sets. The key difference between PLS and CCA is that CCA maximizes the correlation while PLS maximizes the covariance between two sets of variables.

In this paper, we use PLS regression to model the relationship between two sets of random variables. Although PLS can tackle sets of random variables with different dimensionalities, our method uses same sized random variables. Let $\mathbf{X}$ and $\mathbf{Y}$ be two sets of observed random variables (data). In our case, both $\mathbf{X}$ and $\mathbf{Y}$ are matrices of size $n \times m$ where n is the number of random variables and m is the dimensionality of each random variable. It is to be noted that both $\mathbf{X}$ and $\mathbf{Y}$ are preprocessed to ensure they are both zero mean matrices. PLS models the relations between these two data matrices by decomposing them into:

$$\mathbf{X} = \mathbf{T}\mathbf{P}^{\mathbf{T}} + \mathbf{E} \tag{1}$$

$$\mathbf{Y} = \mathbf{U}\mathbf{Q}^{\mathbf{T}} + \mathbf{F} \tag{2}$$

where $\mathbf{T}$, $\mathbf{U}$ are $n \times p$ matrices containing p extracted latent vectors (also called scores), $\mathbf{P}$ and $\mathbf{Q}$ are $m \times p$ matrices of the loadings while $\mathbf{E}$ and $\mathbf{F}$ are the $n \times m$ matrices of residuals. In PLS regression, a linear inner relation between $\mathbf{U}$ and $\mathbf{T}$ is assumed:

$$\mathbf{U} = \mathbf{T}\mathbf{B} + \mathbf{H} \tag{3}$$

where $\mathbf{B}$ is the $p \times p$ diagonal matrix of regression coefficients. $\mathbf{H}$ is the matrix of residuals. Hence, equation (2) can be rewritten as:

$$\mathbf{Y} = \mathbf{T}\mathbf{B}\mathbf{Q}^{\mathbf{T}} + (\mathbf{H}\mathbf{Q}^{\mathbf{T}} + \mathbf{F}) \tag{4}$$

The sum of the regression coefficients in $\mathbf{B}$ serves as the quantitative measure of similarity between sets X and Y.

The PLS method, which is most commonly implemented using the nonlinear iterative partial least squares (NIPALS) algorithm [18], constructs a set of weight vectors $W = \{w_1, w_2, \ldots, w_p\}$ such that

$$[cov(t_i, u_i)]^2 = \max_{|w_i|=1} [cov(\mathbf{X}w_i, \mathbf{Y})]^2 \tag{5}$$

where t_i, u_i are the i^{th} column of matrices $\mathbf{T}$ and $\mathbf{U}$ respectively and $cov(t_i, ui)$ is the sample covariance between latent vectors t_i and u_i. After the extraction of t_i and u_i, the matrices X and Y are deated by subtracting their rank-one approximations based on t_i and u_i. This process is repeated until convergence.

3. PLS Similarity Kernels for Videos

3.1 Joint Shared Modes

In section 2 we described a similarity measure between two sets of random variables. However, in this paper, our goal is to classify videos, which are third order tensors. Thus, we need a way to convert the third order tensors to matrices (which are second order tensors). We achieve this by flattening the video tensor to a matrix. To understand this, we first consider the 2-D case: a matrix can be flattened to a 1-D vector by a simple row-wise (or column-wise) ordering of its elements. In the same way, a third order tensor can be flattened to a matrix in three ways, depending on which two dimensions are reordered into a 1-D vector.

A third order tensor $\mathbf{V} \in \mathbb{R}^{x \times y \times t}$ can be seen as a three dimensional matrix with three modes (dimensions): axes of space (x and y) and time (t). Assuming that we have videos of uniform size ($x \times y \times t$), as described above, there are three ways to flatten the video into a matrix: by re-ordering x, y or x, t or y, t. Thus, for any video, there are three distinct corresponding sets of matrices or random variables. These corresponding matrices have been referred to as the joint shared modes of a tensor in [6]. We call these the xy, xt and yt joint shared modes and denote them by $\mathbf{V}_{xy}$, $\mathbf{V}_{xt}$ and $\mathbf{V}_{yt}$ respectively. Intuitively, by using three different joint shared modes, we are trying to encode the 3-D spatial and temporal context into 3 sets of random variables.

3.2 PLS Kernel

The PLS Kernel $\kappa(\mathbf{U}, \mathbf{V})$ gives a quantitative measure of similarity between two videos $\mathbf{U}$ and $\mathbf{V}$. As described in section 2, the quantitative similarity between two matrices (sets of random variables) is given by the sum of the regression coefficients in the diagonal matrix $\mathbf{B}$. We denote this similarity between two matrices $\mathbf{P}$ and $\mathbf{Q}$ by $\beta(\mathbf{P}, \mathbf{Q})$.

In this paper, each joint shared mode is treated as a set of random variables and contributes to the overall similarity between two videos. We compute the PLS regression coefficients between the corresponding modes for each pair of videos in the dataset. Since we have three joint shared modes corresponding to a video, essentially we can find three similarity values between each pair of videos, one corresponding to each joint shared mode. The PLS Kernel is given by:

$$\kappa(\mathbf{U}, \mathbf{V}) = \beta(\mathbf{U}_{xy}, \mathbf{V}_{xy}) + \beta(\mathbf{U}_{xt}, \mathbf{V}_{xt}) + \beta(\mathbf{U}_{yt}, \mathbf{V}_{yt})$$

Thus, the similarity between two videos is simply the sum of the similarities between their corresponding joint shared modes.

3.3 Discussion

PLS regression is an extension of the multiple linear regression model on which a number of multivariate methods such as discriminant analysis, principal components regression, and CCA are based. Multivariate methods impose two restrictions: (a) latent variables are computed using the $\mathbf{X}^T\mathbf{X}$ and $\mathbf{Y}^T\mathbf{Y}$ matrices; cross-product matrices of $\mathbf{X}$ and $\mathbf{Y}$ variables are not used, and (b) the number of prediction functions is always smaller than the number of $\mathbf{X}$ and $\mathbf{Y}$ variables. In contrast, PLS extracts prediction functions from the $\mathbf{Y}^T\mathbf{X}\mathbf{X}^T\mathbf{Y}$ matrix. The number of prediction functions may be more than the number of $\mathbf{X}$ and $\mathbf{Y}$ variables. PLS can thus be used when the predictor variables outnumber the observations, unlike traditional multivariate methods.

4. Experiments and Results

Here we demonstrate the superiority of PLS similarity kernels over state-of-the art approaches on tasks of hand gesture recognition and activity classification.

4.1 Hand gesture recognition on Cambridge dataset

The popular Cambridge hand gesture data set [6] contains 900 video sequences of 9 gesture classes, defined by 3 primitive hand shapes and 3 primitive motions (see Figure 1). Each class contains 100 video sequences; these 900 video sequences are partitioned into five different illumination setting subsets: $Set1$, $Set2$, $Set3$, $Set4$, $Set5$, each containing 180 videos. As in [8], we reduce the size of the video frames to 20×20 pixels and extract the middle 32 frames for classification. Thus, all the video sequences in the dataset were resized to $20 \times 20 \times 32$. The experimental protocol followed in [8, 6, 9] was used. According to this protocol, $Set5$ was used for training while $Set1$, $Set2$, $Set3$, $Set4$ were used for testing.

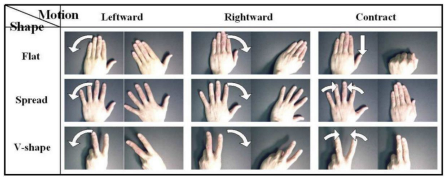

Figure 1. Cambridge hand gesture dataset

Training involved first computing the PLS Kernel Matrix containing the similarities between every pair of training video tensors. We used this Kernel Matrix to train a one-vs-rest SVM classifier [3] per gesture class. Testing involved computing the PLS Kernel Matrix containing the similarities between every pair of a training sample video and a testing sample video. This kernel matrix was used to generate SVM scores for each test sample. The test sample was assigned the class label of the classifier that gave the maximum score.

The hand gesture recognition accuracies can be seen in Table 1. We compare our method with the state-of-the-art approaches described in [8, 9, 6, 5] (Section 1). Our method significantly outperforms the other methods on all illumination settings.

4.2 Action classification on the UCF Sport dataset

The UCF sport action dataset [11] contains 150 video sequences partitioned over ten human action categories like driving, kicking, walking, swinging golf clubs (see Figure 2). Each category has a different number of videos, from 6 to 22. This dataset is challenging because of the non-uniform backgrounds and relative motion between the camera and subject in some actions.

As in [8], we resize all the video sequences to the same size $32 \times 32 \times 64$. We choose the 64 middle frames from each video, and apply linear interpolation between frames for videos with less than 64 frames. We use the leave-one-out cross validation protocol just like in

Figure 2. UCF Sports Action dataset

[8, 2, 7]. The classification setup remains the same as in our experiments on the Cambridge dataset. We trained a one-vs-rest SVM for each action class and the test video sequence was assigned the class label of the classifier with the maximum score. The classification results can be seen in Table 2. We have also compared results with [7] and [2]. While [7] learns the most discriminative space-time feature neighbourhoods for an activity using local motion and appearance features, [2] computes rich features from point trajectories, combine local descriptors to combat background noise and use a novel feature selection scheme. Here again, our method significantly outperforms the state-of-the-art approaches.

Table 1. Hand-gesture recognition accuracy (%) on the Cambridge-Gesture Dataset

Method	Set1	Set2	Set3	Set4	Total
PLS	96%	92%	96%	93%	**94 ± 2.1%**
TB [8]	93%	88%	90%	91%	91 ± 2.4%
PM [9]	89%	86%	89%	87%	88 ± 2.1%
DCCA [5]	-	-	-	-	85 ± 2.8%
TCCA [6]	81%	81%	78%	86%	82 ± 3.4%

Table 2. Leave one out cross validation on the UCF Sports Dataset

PLS	TB [8]	HDN [7]	OMD [2]
93.2%	88%	87.27%	86.9%

Discussion: Our PLS similarity kernel approach is superior to the previous best [8] for these tasks. Our method is both more intuitive (based on maximizing covariance) and straight-forward, thus easily implementable. Compared to [6], PLS is more general (sections 2, 3.3) and the use of SVM classifiers (as opposed to a nearest neighbour scheme) with our similarity kernels boosts the classification performance.

5. Conclusion

In this paper, we devised a method that employs PLS regression to derive a scalar similarity measure between two sets of random variables. We extended this technique to find quantitative similarity measures between two videos. We employed discriminative kernel matrices constructed using pair-wise similarities between the data samples to solve the tasks of hand gesture recognition and human activity classification. Our method outperforms the state-of-the-art methods on the Cambridge hand gesture dataset and the UCF Sports dataset. Our model involves no parameter tuning. A further understanding of PLS regression could lead us to investigating other interesting properties of pairs of tensors.

References

[1] R. Bowden and M. Sarhadi. Building temporal models for gesture recognition. In *BMVC*, 2000.

[2] M. Bregonzio, J. Li, S. Gong, and T. Xiang. Discriminative topics modelling for action feature selection and recognition. In *BMVC*, 2010.

[3] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2011.

[4] N. Gamage, Y. C. Kuang, R. Akmeliawati, and S. Demidenko. Gaussian process dynamical models for hand gesture interpretation in sign language. In *Pattern Recognition Letters*, 2011.

[5] T.-K. Kim and R. Cipolla. Gesture recognition under small sample size. In *ACCV (1)*, 2007.

[6] T.-K. Kim, S.-F. Wong, and R. Cipolla. Tensor canonical correlation analysis for action classification. In *CVPR*, 2007.

[7] A. Kovashka and K. Grauman. Learning a hierarchy of discriminative space-time neighborhood features for human action recognition. In *CVPR*, 2010.

[8] Y. M. Lui and J. R. Beveridge. Tangent bundle for human action recognition. In *FG*, 2011.

[9] Y. M. Lui, J. R. Beveridge, and M. Kirby. Action classification on product manifolds. In *CVPR*, 2010.

[10] V. Pavlovic, R. Sharma, and T. Huang. Visual interpretation of hand gestures for humancomputer interaction: A review. In *IEEE Trans. Patt. Anal. Mach. Intell., Vol. 19*, 1997.

[11] M. D. Rodriguez, J. Ahmed, and M. Shah. Action mach: a spatio-temporal maximum average correlation height filter for action recognition. In *CVPR*, 2008.

[12] R. Rosipal and N. Kramer. Overview and recent advances in partial least squares. In *Lecture Notes in Computer Science*, 2006.

[13] D. Schonfeld. Motionsearch: Context-based video retrieval and activity recognition in video surveillance. In *AVSS*, 2009.

[14] A. Shamaie and A. Sutherland. Graph-based matching of occluded hand gestures. In *Proc. of the Applied Imagery Pattern Recognition*, 2001.

[15] J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.

[16] M. Su, H. Huang, C. Lin, and C. Huang. Application of neural networks in spatio temporal hand gesture recognition. In *Proc. of the IEEE World Congress on Computational Intelligence*, 1998.

[17] A. D. Wilson and A. Bobick. Parametric hidden markov models for gesture recognition. In *IEEE Trans. Patt. Anal. Mach. Intell*, 1999.

[18] H. Wold. Path models with latent variables: The NIPALS approach. In *Quantitative Sociology: International perspectives on mathematical and statistical model building*, 1975.