Dissimilarity-Based Action Recognition with the Pair Hidden Markov Support Vector Machine

Zhen Wang
School of Computing and Communications
Faculty of Engineering and IT
University of Technology Sydney
Email: zhen.wang-3@student.edu.au

Massimo Piccardi
School of Computing and Communications
Faculty of Engineering and IT
University of Technology Sydney
Email: massimo.piccardi@uts.edu.au

***Abstract*—Human action recognition in video is highly challenging due to the substantial variations in motion performance, recording settings and inter-personal differences. Most current research focuses on the extraction of effective features and the design of suitable classifiers. Conversely, in this paper we tackle this problem by a dissimilarity-based approach where classification is performed in terms of minimum distance from templates. To measure the dissimilarity between any two action instances, we propose leveraging the Pair Hidden Markov Support Vector Machine (PHMM-SSVM) that was recently proposed for tasks of video alignment. The main advantages of PHMM-SSVM are its ability to learn optimal alignment models from training sets of manually-aligned action pairs and provide alignment scores that can be used for action classication. The experimental results over two popular action datasets show that the proposed approach has been capable of achieving an accuracy higher than many existing methods and comparable to a state-of-the-art algorithm.**

***Index terms*—** DAGSVM, k-nearest neighbours, sequence alignment, action recognition, PHMM-SSVM.

I. Introduction and related work

The rapid growth in the availability of human action videos is pushing the need for tools that can automatically classify actions in applications such as sport video analysis, automatic surveillance, social media tagging and others. The field of automatic action recognition is geared towards the design of more effective features and classifiers [1]. However, accurate action recognition remains challenging to date due to the many degrees of variations under which human actions can be performed and recorded.

The goal of this paper is to explore action recognition by a dissimilarity-based approach where classification is performed in terms of minimum distance from pre-classified templates. Our motivation comes from the studies on dissimilarity-based classification that have all reported remarkable accuracy [2], [3], [4], [5], [6]. Dissimilarity-based classification does not represent each object by a conventional feature vector: rather, it provides a *distance function* that can quantify the dissimilarity between any given object pair. Once the distance function is chosen, classification can be provided in either of two ways: a) by directly using a minimum-distance classifier (e.g., k-nearest neighbours), or b) by using the distances between the object and a given set of template samples as a distance-based feature vector for the object. In the first case, the object is assigned to the most frequent class amongst the k closest samples of a training set. In the second case, the distances between the object and the templates form a feature vector in their own right, and can be used in conjunction with any conventional classifier (SVM, deep neural networks etc). Dissimilarity-based classification has proved very successful for the classification of non-vector objects such as graphs and strings [2], [3], [4] and seems promising also for the classification of complex sequential data such as human actions.

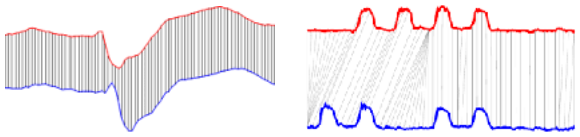

Fig. 1: Left: Example of frame-by-frame distances. Right: Example of dynamic time warping.

The crux of dissimilarity-based classification is the choice of an effective distance function. In the case of action videos, a basic idea could be to measure the distance between any two given videos simply as the sum of the distances between their frame pairs in appearance order. However, such a distance is too crude since it does not take into account local misalignments (see the left diagram in Fig. 1). For this reason, alignment algorithms such as dynamic time warping (DTW) [7], canonical time warping (CTW) [8], [9] and many others have been used to more suitably measure the distance between two videos. These algorithms are generally more effective since they compensate for temporal distortions (see the right diagram in Fig. 1). However, they are typically based on fixed cost models and they are difficult to adapt to specific applications.

Amongst alignment models, the recently-proposed Pair Hidden Markov Support Vector Machine [10] (acronymised as PHMM-SSVM from the combination of pair hidden Markov model and structural SVM), is a more sophisticated alignment

algorithm that has proved capable of remarkable accuracy, outperforming established algorithms such as DTW [7] and CTW [8], [9] on alignment metrics. PHMM-SSVM provides major advantages over conventional alignment approaches: 1) an ability to learn an optimal cost model from any set of manually-aligned video pairs; 2) a maximum-margin objective that has a strong reputation for empirical accuracy; 3) the possibility to choose arbitrary loss functions for tuning the model to specific type of data; and 4) a customisable kernel distance between frame pairs for implementing nonlinear cost models (function $\mathbb{K}$ in Equation 1). In this paper, we experiment with both classification approaches (minimum-distance classification and distance-based feature vectors) using the PHMM-SSVM as the underlying distance. As minimum-distance classifier, we have adopted the k-nearest neighbour classifier (k-NN) [11]. To curb its computational complexity, we have applied a *prototype selection technique* that selects a number of template samples, or "prototypes", from each class to abate the overall number of comparisons. For the second approach, we have computed the distances between each input sequence and the prototypes of every class, and used it as feature vector with a multiclass SVM classifier. We have tested the proposed approach in a set of experiments on action recognition over the popular KTH [12] and Olympic Sport [13] datasets. The experimental results show that the proposed approach has been able to outperform many existing approaches in terms of classification accuracy and to rank closely to the state of the art.

II. The PHMM-SSVM distance

Given the frame sequences of two videos, $s = \{s_1, ..., s_i, ..., s_{L_s}\}$ and $t = \{t_1, ..., t_j, ..., t_{L_t}\}$, an "alignment path", y, is a sequence of symbols that pairs frames from s and t. The symbols are of three types: M ("match"), S ("insert a gap on sequence s"), and T ("insert a gap on sequence t"), with the following meaning: assuming that i and j are current indices over sequences s and t, respectively, 1) symbol M pairs frames s_i and t_j, and then increments both indices; 2) symbol S pairs no frames and only increments index j; and, likewise, 3) symbol T pairs no frames and only increments index i. To illustrate the alignment, Fig. 3 shows a diagram with two input sequences and an alignment path, including matched frames and inserted gaps; Fig. 2 shows actual examples of alignments for actions from the Olympic Sports dataset.

The PHMM-SSVM model [10] is a probabilistic model that defines a joint probability, $p(s,t,y)$, for sequences s,t and their alignment path, y. This joint probability is chosen from the class of the exponential family of distributions, $p(s,t,y) \propto \exp(w^\top \psi(s,t,y))$, where w notes a parameter vector and ψ a suitable feature function. With this parametrization, a PHMM-SSVM can be trained by leveraging maximum-margin approaches that have gained a strong reputation for accuracy [14].

Like in a conventional hidden Markov model, the joint probability of a PHMM-SSVM conveniently factorises into a set of transition and emission probabilities (Fig. 3 shows the model as a graphical model). Accordingly, parameter vector w divides in two parts: transition parameters, w^{tr}, and emission parameters, w^{em}. The overall scoring function, $F(s,t) = w^\top \psi(s,t,y)$, is written as:

$$
\begin{aligned}
& w^\top \psi(s,t,y) = \sum_{k=1}^{|y|} w^{tr}_{y_{k-1},y_k} + w^{em\top} \mathbb{K}(s_i - t_j)\mathbf{I}[y_k = M] \\
& w^{tr}_{0,*} = 0; \quad \mathbf{I}[y_k = M] : i{+}{+}, j{+}{+}; \\
& \mathbf{I}[y_k = S] : j{+}{+}; \quad \mathbf{I}[y_k = T] : i{+}{+}
\end{aligned}
\tag{1}
$$

where $\mathbf{I}$ is the indicator function and indices i and j, initially set to 1, are post-incremented according to the value of label y_k. Function $\mathbb{K}$ is a generic, nonlinear kernel function which accounts for the dissimilarity between any two frames of s and t.

Once a model and two input sequences are given, the optimal alignment $\bar{y} = \operatorname{argmax}_y w^\top \psi(s,t,y)$ can be computed by an efficient dynamic programming algorithm of linear complexity akin to the Viterbi algorithm [15]. In the follwoing, we use the score of this optimal path as the inverse distance between the two input sequences. For training the PHMM-SSVM model, we have used structural SVM [14] over a set of manually-aligned video pairs (further details can be found in [10]).

III. Prototype selection with PHMM-SSVM

The main drawback of nonparametric dissimilarity-based classifiers such as k-NN is their computational complexity at run time: in principle, each test sample should be compared with every sample in the training set. While mitigation techniques such as the use of the triangle inequality and k-d trees can be exploited, the problem remains intrinsically complex, especially for large datasets. An alternative to the full run-time search is offered by *prototype selection*: in this case, the training set is replaced by a subset of representative prototypes, making the search substantially faster. In addition to reducing the run-time complexity, prototype selection typically achieves a comparable or even higher classification accuracy than the full search thanks to the removal of noisy and redundant samples [16]. The selection of the best prototypes can be performed according to a number of different criteria, including uniform distribution, centrality in the class, and others [3]. For this work, we have decided to adopt *KCentres* that selects prototypes that well reflect the sample distribution inside each class [17]. For each class, *KCentres* chooses L prototypes with these steps: 1) randomly pick an initial set of prototypes, $P = \{p_1, ...p_L\}$; 2) partition all the class' samples into L subsets, $J_1 = \{p_1\}, ..., J_L = \{p_L\}$, based on their closest prototype; 3) for each $J_l, l = 1, 2, ..., L$, find its most central element (the element whose maximum distance to all other elements is minimum); 4) replace the prototypes in P with the most central elements. Eventually, iterate steps 2-4 until convergence or a maximum number of iterations is reached.

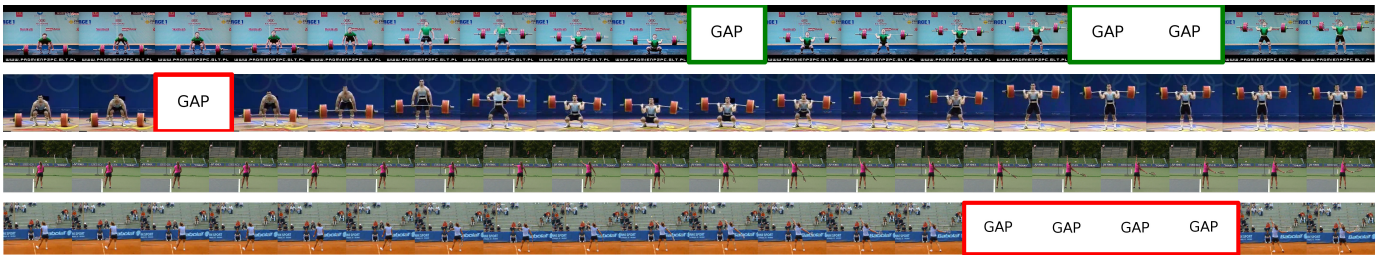

Fig. 2: Examples of PHMM-SSVM alignment from the Olympic Sport dataset. Top two rows: 14 matches and inserted gaps for two paired "clean-and-jerk" sequences. Bottom two rows: 16 matches and inserted gaps for two paired "tennis-serve" sequences (the gaps are inserted only on the second sequence because of its slower execution).

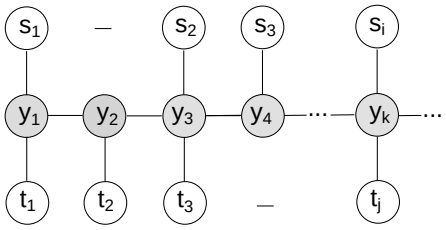

Fig. 3: The PHMM-SSVM as a graphical model.

Algorithm 1: k-NN classification with the PHMM-SSVM distance.

Input : Class set $C = \{c_1, ..., c_n\}$;
Test set $T = \{t_1, ..., t_l\}$;
Prototype set $P = \{P_1, ..., P_{n*L}\}$

Allocate l result sets $r = (r_1, \ldots, r_l)$
foreach $i \in 1 \ldots l$ **do**
  Compute the PHMM-SSVM distances between t_i and P: $D(t_i, P) = \{d(t_i, P_1), ..., d(t_i, P_{n*L})\}$
  if *there is a majority class,* c **then**
    $c -> r_i$
  end
  else
    Apply DAGSVM with BoW(t_i) to find class c
    $c -> r_i$
  end
end
Output: result sets r

IV. Classification

For classification, we have used two different approaches: 1) minimum-distance classification with a k-NN classifier and 2) a distance-based feature vector with a DAGSVM classifier. In the first approach, the k-NN classifier predicts the class of a test sample by the majority class of its k nearest neighbours. Despite its simplicity, the k-NN classifier has reported state-of-the-art accuracy over a number of benchmarks [11]. Before running k-NN, we have selected L prototypes per class and replaced the training set with the prototypes. This reduction abates the run time by the ratio between the size of these two sets. However, the prediction of a k-NN classifier is ambiguous whenever the majority class is at a parity. In this case, rather than picking a class using an arbitrary criterion, we have resorted to a "refinement classifier" to disambiguate the choice. As refinement classifier, we have used a multiclass DAGSVM with a standard bag-of-words (BoW) representation of the video as input. This idea was inspired by [5], although our implementation is simpler and faster. The overall classification procedure is summarised in Algorithm 1.

In the second approach, we have first computed a feature vector for each video consisting of the distances between the video itself and all the prototypes from all classes. Then, we have performed action classification using these feature vectors and a multiclass DAGSVM classifier [18]. A DAGSVM classifier (reading as "directed acyclic graph" SVM) is an improvement over the popular "one-vs-one" and "one-vs-all" SVM [19] that, by design, is capable of preventing classification parities. In one-vs-one and one-vs-all SVM, mutliclass classification is performed as a set of binary classifications that can often lead to classification ambiguities (samples than are classified into more than one class, or none). DAGSVM prevents these cases by organising the set of binary classifications as a decision tree: in each node of the tree, a single binary classification is performed to *exclude* one of the classes in turn. When the sample eventually reaches a leaf of the tree, its class assignment is unique. DAGSVM is also efficient since it requires only $M - 1$ binary inferences for classification over a set of M classes. As an example of the feature vector, Table I shows the PHMM-SSVM distances between a sample of the "walking" class in the KTH dataset and 10 prototypes from each of the dataset's 6 classes. To further illustrate the alignment distance, Figure 4 shows the PHMM-SSVM distances between every sample pair in the same dataset (2,391 istances) as a grey-scale matrix. The overall classification procedure is summarised in Algorithm 2.

V. Experiments and discussion

This section compares the performance of PHMM-SSVM distance-based classification against state-of-the-art methods over four experiments. To prepare the measurements for the experiments, we have first extracted dense feature descriptors

Algorithm 2: DAGSVM classification with the PHMM-SSVM-based feature vector.

Input : Class set $C = \{c_1, ..., c_n\}$;
Test set $T = \{t_1, ..., t_l\}$;
Prototype set $P = \{P_1, ..., P_{n*L}\}$

Allocate l result sets $r = (r_1, \ldots, r_l)$
foreach $i \in 1 \ldots l$ *in* T **do**
  Compute feature vector $t_i{}'$ for t_i:
  $t_i{}' = \{d(t_i, P_1), ..., d(t_i, P_{n*L})\}$
  Apply DAGSVM with $t_i{}'$ to find class c
  $c- > r_i$
end
Output: result sets r

TABLE I: Example of PHMM-SSVM distances between a *walking* sample in the KTH dataset and prototypes from the various classes. Darker colours denote higher similarity.

from each frame of all the video sequences by using the STIP extractor of [12]. We have then computed a bag-of-words with 1,000 bins for each frame using the VLFeat library [20]. After that, we have trained a PHMM-SSVM distance model for each class on manually-aligned sequence pairs (ground-truth alignments) from that class. We have not trained cross-class models, expecting that a trained class model would return the highest scores for test sequences from the same class. For the annotation of the ground-truth alignments, we have selected and matched "key frames" (i.e., apexes of actions) from the paired sequences. After the PHMM-SSVM training, prototype selection has been performed on each class using the respective PHMM-SSVM distance to select L sequences as prototypes.

The experiments have been carried out over the KTH [12] and Olympic Sports [13] datasets. KTH is a video dataset of 6 action classes staged by 25 actors in various indoor and outdoor scenarios for a total of 2,391 action instances. The Olympic Sports dataset is a sport action dataset containing 16 action classes and a total of 800 action instances. This dataset is more challenging than KTH since its samples are real videos shot under a variety of viewing conditions and from different, unknown cameras.

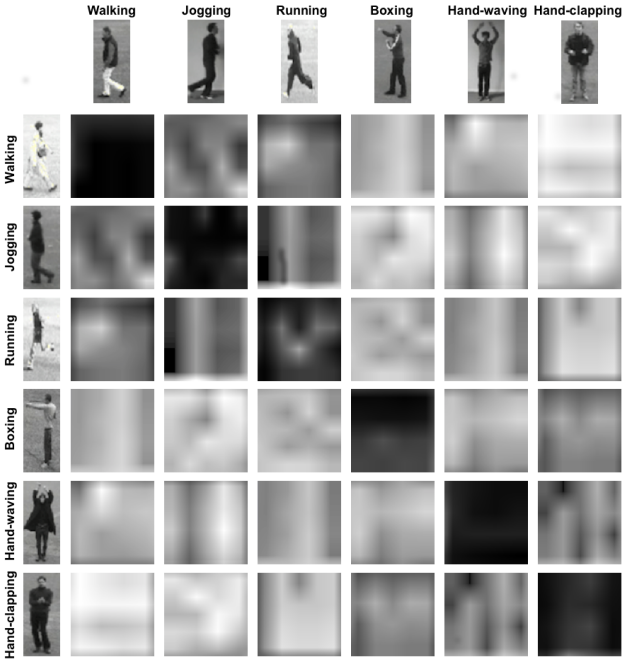

Fig. 4: The PHMM-SSVM distance matrix between action instances in KTH dataset. Darker colours denote higher similarity.

A. Results with the k-NN classifier on the KTH dataset

For this experiment, we have used the standard training and test sets provided by the dataset's authors. To train the PHMM-SVMM models, we have selected 16 training sequences per class and generated 120 (i.e., $16 * 15/2$) manually-aligned pairs using 6 key frames for each sequence. The experiments have been carried out with 1, 5, and 10 nearest neighbours (1-NN, 5-NN and 10-NN) and the number of prototypes per class, L, has been set to twice the number of the nearest neighbours. Table II shows that the accuracy from 10-NN has proved the highest, outperforming the results retrieved from the literature (including the method from Niebles *et al*. [13] by more than 23 percentage points). The lower accuracy for classes "running" and "jogging" has been likely due to their higher cross-similarity that has made it difficult for PHMM-SVMM to yield reliable alignments. Conversely, neatly characterised actions such as "boxing" and "waving" have been recognised with 100% accuracy. Figure 4 clearly shows some degree of confusion between classes "walking", "jogging" and "running", which explains the lower accuracy for these classes.

B. Results with the k-NN classifier on the Olympic Sports dataset

For the second dataset, we have used the same training and test splits of [13] and performed PHMM-SSVM training with the same procedure of the first dataset. The experiments have been carried out with 5-NN and 10-NN (1-NN did not seem promising, given its limited performance on KTH and

Action	1-NN	5-NN	10-NN
walking	83.3%	97.9%	99.3%
running	72.2%	84.0%	86.8%
jogging	72.9%	86.1%	88.2%
waving	86.1%	100%	100%
clapping	85.4%	98.6%	98.6%
boxing	86.8%	99.3%	100%

Algorithm	Avg.
Ours (10-NN)	95.5%
Niebles *et al*. [13]	71.9%
Laptev *et al*. [21]	91.8%
Wong *et al*. [22]	86.7%
Kim *et al*. [23]	95.3%
Wang *et al*. [24]	92.1%

TABLE II: Accuracy with the k-NN classifier on the KTH dataset.

Sport class	Ours 5-NN	Ours 10-NN	Niebles *et al*. [13]	Jain *et al*. [25]
high-jump	79.2%	82.5%	68.9%	**84.9%**
long-jump	82.7%	**85.2%**	74.8%	84.6%
triple-jump	**83.5%**	**85.7%**	52.3%	83.3%
pole-vault	84.3%	**88.1%**	82.0%	84.7%
vault	82.2%	**86.5%**	86.1%	82.6%
shot-put	69.3%	71.4%	62.1%	**83.6%**
snatch	82.8%	**86.1%**	69.2%	83.5%
clean-jerk	87.7%	**90.1%**	84.1%	86.6%
javelin-throw	82.6%	**85.0%**	74.6%	84.8%
hammer-throw	85.1%	**87.5%**	77.5%	86.4%
discus-throw	70.1%	73.3%	58.5%	**86.7%**
diving-platform	73.1%	78.9%	**87.2%**	86.5%
div. springboard	70.6%	74.3%	77.2%	**86.4%**
basketball	78.3%	81.6%	77.9%	**88.6%**
bowling	82.8%	86.2%	72.7%	**88.3%**
tennis-serve	77.2%	81.8%	49.1%	**83.4%**

Algorithm	Avg.
Ours (10-NN)	82.8%
Niebles *et al*. [13]	72.1%
Jain *et al*. [25]	85.3%

TABLE III: Accuracy with the k-NN classifier on the Olympic Sports dataset.

Action	1 pr.	5 pr.	10 pr.
walking	81.3%	98.9%	99.3%
running	71.9%	86.2%	87.7%
jogging	72.3%	88.0%	89.1%
waving	87.4%	100%	100%
clapping	86.2%	98.6%	99.1%
boxing	87.2%	99.3%	100%

Algorithm	Avg.
Ours (10 pr.)	95.8%
Niebles *et al*. [13]	71.9%
Laptev *et al*. [21]	91.8%
Wong *et al*. [22]	86.7%
Kim *et al*. [23]	95.3%
Wang *et al*. [24]	92.1%

TABLE IV: Accuracy with the DAGSVM classifier on the KTH dataset.

Sport class	Ours 5 pr.	Ours 10 pr.	Niebles *et al*. [13]	Jain *et al*. [25]
high-jump	77.8%	81.7%	68.9%	**84.9%**
long-jump	83.1%	**85.6%**	74.8%	84.6%
triple-jump	**84.1%**	**86.1%**	52.3%	83.3%
pole-vault	83.5%	**86.6%**	82.0%	84.7%
vault	82.6%	**86.7%**	86.1%	82.6%
shot-put	68.9%	74.1%	62.1%	**83.6%**
snatch	84.2%	**86.4%**	69.2%	83.5%
clean-jerk	88.3%	**90.5%**	84.1%	86.6%
javelin-throw	82.0%	**85.1%**	74.6%	84.8%
hammer-throw	86.6%	**88.3%**	77.5%	86.4%
discus-throw	70.5%	74.6%	58.5%	**86.7%**
diving-platform	71.8%	76.4%	**87.2%**	86.5%
div. springboard	70.1%	72.8%	77.2%	**86.4%**
basketball	80.4%	82.8%	77.9%	**88.6%**
bowling	86.4%	**88.6%**	72.7%	88.3%
tennis-serve	77.9%	83.2%	49.1%	**83.4%**

Algorithm	Avg.
Ours (10 pr.)	83.1%
Niebles *et al*. [13]	72.1%
Jain *et al*. [25]	85.3%

TABLE V: Accuracy with the DAGSVM classifier on the Olympic Sports dataset.

the more challenging videos). Table III shows that 10-NN has, again, achieved higher accuracy than 5-NN. Its per-class accuracies are comparable with those from a state-of-the-art classifier (Jain *et al*. [25]): lower in 8 cases and on average, but higher in another 8 cases. The lowest accuracies were obtained for classes "discus-throw" and "shot-put" which were often confused because of their similar appearance; class "shot-put" is also very challenging in its own right because it is performed in a variety of styles (rotational, backsliding etc). Similar misclassifications have also occurred between classes "diving-platform" and "diving-springboard". Our approach has, again, achieved the highest accuracies for distinctive actions such as "clean-and-jerk" weightlifting and "pole-vault". Overall, the average accuracy of the proposed method has proved 10.7 percentage points higher than the baseline from Niebles *et al*. [13] and has ranked closely to that of Jain *et al*. [25].

C. Results with the DAGSVM classifier on both datasets

For the experiments with the distance-based feature vector and the DAGSVM classifier we have used similar settings as with the k-NN classifier. For the KTH dataset, Table IV shows that the accuracy with 10 prototypes has proved higher than the results retrieved from the literature (including the method from Niebles *et al*. [13] by approximately 24 percentage points), and also slightly higher than with the k-NN approach (0.3 percentage points). However, the same classes (i.e., "running", "jogging") have reported lower accuracy also with this classifier. This clearly shows that the performance of the classifier is closely tied to the discriminative capability of the underlying PHMM-SSVM distance.

For the Olympic Sports dataset, the accuracies reported in Table V seem interesting and generally comparable to those of the k-NN classifier. The average accuracy is, again, slightly higher (0.3 percentage points). The DAGSVM classifier has also achieved higher scores than a state-of-the-art classifier (Jain *et al*. [25]) in 9 cases (with class "bowling" in addition to those from the previous experiment). The lowest accuracies have again been for class pairs "discus-throw" and "shot-put", and "diving-platform" and "diving-springboard", due to their evident similarity. The proposed approach has again achieved its highest accuracies for distinctive actions such as "clean-and-jerk" weightlifting and "pole-vault". Overall, the average accuracy of the proposed method has proved 11 percentage points higher than the baseline from Niebles *et al*. [13] and has ranked closely to that of Jain *et al*. [25].

VI. Conclusion

In this paper, we have presented a novel dissimilarity-based approach to action recognition in videos. The approach leverages the recently-proposed PHMM-SVMM alignment algorithm which, for every two given videos, provides an alignment path and a similarity score. In our experiments, we

have used the PHMM-SSVM similarity score as an inverse distance between the two input videos, and exploited it for action classification. We have proposed two distance-based methods for classification: 1) a k-NN classifier using the PHMM-SSVM distance; 2) a DAGSVM classifier using a PHMM-SSVM-based feature vector. Prior to applying the classifiers, we have run a step of prototype selection to select a set of prototypes for each class. For the k-NN classifier, we have replaced the training set with the prototypes' set to abate the test-time computational complexity. For the DAGSVM classifier, we have used the prototypes to obtain a distance-based feature vector for each video. The experimental results over two popular action video datasets - KTH and Olympic Sports - have showed that:

- on the KTH dataset, the proposed approaches have achieved an accuracy that is 24 percentage points higher than the classifier from Niebles *et al*. [13] and higher than the results compiled from the literature;
- on the Olympic Sports dataset, the proposed approaches have achieved an accuracy that is 11 percentage points higher than the classifier from Niebles *et al*. [13] and close to that of a state-of-the-art approach [25].

A further analysis of the per-class accuracy has shown that the proposed approach has tended to outperform the other classifiers on actions with more pronounced temporal stages. Given that the PHMM-SVMM distance is based on temporal alignment, this result is encouraging and indicates that the best application for the proposed approach are actions with neatly-outlined stages. In the future, we plan to expand the experiments by integrating the PHMM-SVMM distance with other features and classifiers.

REFERENCES

[1] P. K. Turaga, R. Chellappa, V. S. Subrahmanian, and O. Udrea, "Machine recognition of human activities: A survey." *IEEE Trans. Circuits Syst. Video Techn.*, vol. 18, no. 11, pp. 1473–1488, 2008.

[2] E. Pekalska and R. P. W. Duin, *The dissimilarity representation for pattern recognition: foundations and applications*. World Scientific Pub Co Inc, 2005, vol. 64.

[3] K. Riesen, M. Neuhaus, and H. Bunke, "Graph embedding in vector spaces by means of prototype selection," in *Proceedings of the 6th IAPR-TC-15 international conference on Graph-based representations in pattern recognition*. Springer-Verlag, 2007, pp. 383–393.

[4] K. Riesen and H. Bunke, "Graph classification based on vector space embedding," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 23, no. 6, p. 1053, 2009.

[5] H. Zhang, A. C. Berg, M. Maire, and J. Malik, "Svm-knn: Discriminative nearest neighbor classification for visual category recognition," in *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2*, ser. CVPR '06. Washington, DC, USA: IEEE Computer Society, 2006, pp. 2126–2136.

[6] K. Q. Weinberger and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification," *J. Mach. Learn. Res.*, vol. 10, pp. 207–244, jun 2009.

[7] L. Rabiner and B.-H. Juang, *Fundamentals of Speech Recognition*. Englewood Cliffs, New Jersey: Prentice-Hall Signal Processing Series, 1993.

[8] F. Zhou and F. De la Torre, "Canonical time warping for alignment of human behavior," in *Advances in Neural Information Processing Systems Conference (NIPS)*, December 2009.

[9] ——, "Generalized canonical time warping," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 2, pp. 279–294, 2016.

[10] Z. Wang and M. Piccardi, "A pair hidden Markov support vector machine for alignment of human actions," in *IEEE International Conference on Multimedia and Expo, ICME 2016, Seattle, WA, USA, July 11-15, 2016*, 2016, pp. 1–6.

[11] P. Cunningham and S. J. Delany, "k-nearest neighbour classifiers," 2007.

[12] C. Schuldt, I. Laptev, and B. Caputo, "Recognizing human actions: A local SVM approach," in *Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04) Volume 3 - Volume 03*, ser. ICPR '04. Washington, DC, USA: IEEE Computer Society, 2004, pp. 32–36.

[13] J. C. Niebles, C.-W. Chen, and L. Fei-Fei, "Modeling temporal structure of decomposable motion segments for activity classification," in *Proceedings of the 11th European Conference on Computer Vision: Part II*, ser. ECCV'10. Berlin, Heidelberg: Springer-Verlag, 2010, pp. 392–405.

[14] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun, "Large margin methods for structured and interdependent output variables," *JMLR*, vol. 6, pp. 1453–1484, 2005.

[15] M. S. Ryan and G. R. Nudd, "The Viterbi algorithm," Coventry, UK, UK, Tech. Rep., 1993.

[16] S. Ougiaroglou, L. Karamitopoulos, C. Tatoglou, G. Evangelidis, and D. A. Dervos, *Applying Prototype Selection and Abstraction Algorithms for Efficient Time-Series Classification"*. Cham: Springer International Publishing, 2015, pp. 333–348.

[17] E. Pekalska, R. P. W. Duin, and P. Paclík, "Prototype selection for dissimilarity-based classifiers," *Pattern Recogn.*, vol. 39, no. 2, pp. 189–208, feb 2006.

[18] J. C. Platt, N. Cristianini, and J. Shawe-Taylor, "Large margin dags for multiclass classification," in *Advances in Neural Information Processing Systems 12*, S. A. Solla, T. K. Leen, and K. Müller, Eds. MIT Press, 2000, pp. 547–553.

[19] C.-C. Chang and C.-J. Lin, "Libsvm: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, pp. 27:1–27:27, 2011.

[20] A. Vedaldi and B. Fulkerson, "VLFeat: An open and portable library of computer vision algorithms," http://www.vlfeat.org, 2008.

[21] C. Schmid, B. Rozenfeld, M. Marszalek, and I. Laptev, "Learning realistic human actions from movies," *2008 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–8, 2008.

[22] T.-K. Kim, S.-F. Wong, and R. Cipolla, "Learning motion categories using both semantic and structural information," *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–6, 2007.

[23] ——, "Tensor canonical correlation analysis for action classification," in *CVPR*. IEEE Computer Society, 2007.

[24] H. Wang, M. M. Ullah, A. Klser, I. Laptev, and C. Schmid, "Evaluation of local spatio-temporal features for action recognition," in *University of Central Florida, U.S.A*, 2009.

[25] A. Jain, A. Gupta, M. Rodriguez, and L. S. Davis, "Representing videos using mid-level discriminative patches," *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2571–2578, 2013.