Unsupervised Spectral Dual Assignment Clustering of Human Actions in Context

Simon Jones, Ling Shao
Department of Electronic and Electrical Engineering
The University of Sheffield, Sheffield, S1 3JD, UK
simon.m.jones@sheffield.ac.uk, ling.shao@sheffield.ac.uk

Abstract

A recent trend of research has shown how contextual information related to an action, such as a scene or object, can enhance the accuracy of human action recognition systems. However, using context to improve unsupervised human action clustering has never been considered before, and cannot be achieved using existing clustering methods. To solve this problem, we introduce a novel, general purpose algorithm, Dual Assignment k-Means (DAKM), which is uniquely capable of performing two co-occurring clustering tasks simultaneously, while exploiting the correlation information to enhance both clusterings. Furthermore, we describe a spectral extension of DAKM (SDAKM) for better performance on realistic data. Extensive experiments on synthetic data and on three realistic human action datasets with scene context show that DAKM/SDAKM can significantly outperform the state-of-the-art clustering methods by taking into account the contextual relationship between actions and scenes.

	Cycling	Soccer	Basketball
Indoor Court	✗	✓	✓
Field	✓	✓	✗
Driveway	✓	✗	✓

Figure 1: Above: various stills from videos, showing human actions happening in different scene contexts. Below: a simple model example of an action/scene relationship table. The information in this table can be directly inferred from training data in supervised learning, but in unsupervised learning it must be estimated simultaneously with the action/scene categories.

1. Introduction

Much recent research in the field of computer vision has focused on the representation and recognition of human actions from varied sources, such as YouTube videos and Hollywood films. In these realistic videos, the actions usually have a considerable amount of context – in particular, the place it is performed in, or the object it is performed with. This context information can be integrated into an action recognition system to help disambiguate between similar classes, and thereby improve classification results, as demonstrated in Marszałek et al. [7]. If an action's scene context is recognised as a basketball court, for instance, this informs us that the action to be classified is more likely to be "playing basketball" - Figure 1 illustrates this concept.

While context for supervised action recognition has been explored, however, no previous research has considered using context for unsupervised human action *clustering*. This is important to consider, as accurate unsupervised and semi-supervised clustering of human actions is crucial to many practical tasks, such as automatic annotation of video databases or fast content-based video retrieval. Action clustering can also assist in determining the semantic similarity of two videos' contents, which can be used, for example, to enhance the recommendation systems of video databases. However, it is not straightforward to apply existing action context research to action clustering. In existing work, labeled training data is typically available for both the

actions and the context, which permits direct inference of the relationship between the action categories and their contexts – it is straightforward to construct a contextual recognition model based on this relationship. For the goal of action clustering with context, on the other hand, no training labels are provided, and so the action/context relationship cannot be learned directly. It is instead necessary to simultaneously estimate the action clustering, the context clustering, and the action/context relationship together.

In this paper, to perform unsupervised action clustering with context, we propose the idea of dual assignment clustering and the novel Dual Assignment k-Means clustering algorithm (DAKM). This algorithm learns two clusterings of a dataset according to two views of the dataset, using the relationship between the views to improve both clusterings. We first demonstrate the theoretical applicability of DAKM on synthetic data, then combine it with a spectral representation to show state-of-the-art results on several realistic human action datasets (using actions and scenes as the two views).

The rest of this work is structured as follows. We outline previous works in clustering and contextual human action recognition in Section 2. Section 3 defines dual assignment clustering, and details the Dual Assignment k-Means (DAKM) clustering algorithm as well as its spectral extension. Experiments on synthetic data and three realistic human action datasets are given in Section 4. We conclude with a discussion of our findings in Section 5.

2. Previous Work

Many recent works on human action recognition have considered the effect of context. Marszalek et al. [7] demonstrate how two classifiers can be trained – one for actions and one for scenes – and then used in combination to improve the classification results with a set of weights associating the two classifiers. Ikizler-Cinbis and Sclaroff [3] go further, combining object, scene and action information in a multiple instances learning framework, to improve the classification performance of YouTube videos. Prest et al. [10] use a weakly supervised framework to learn the interaction between human actions and the objects in the scene, in particular learning the spatial relationship between actions and objects. All of these techniques rely upon training data, however, to learn the relationship between actions and context.

Other works have focused on fully unsupervised clustering of human actions. Yang et al. [14] demonstrate that a global action descriptor and a temporal matching algorithm provide superior results to local feature based methods for clustering. Niebles et al. [9] use pLSA and LDA – techniques originating from natural language processing – to cluster the actions based on the intermediate topics associated with them. Wang et al. [13] show the effectiveness of spectral clustering, using a linear programming technique to find the distance between pairs of action images.

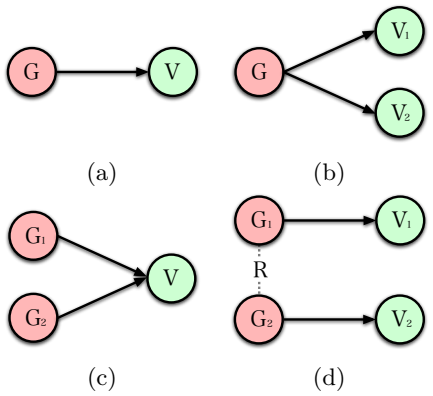

Figure 2: A visualisation of various clustering approaches, showing the dependence relationship between latent categorisations of the dataset, G, and the observable views on that dataset, V. (a) Ordinary Clustering. (b) Multiview Clustering for two views. (c) Alternative Clustering for two solutions. (d) Dual Assignment Clustering.

Recent advances in general-purpose data clustering should also be considered. In particular, multi-view clustering [1, 4] and alternative clustering [2] bear some similarity to dual-assignment clustering. We visualise how these concepts compare to our own in Figure 2.

Multi-view clustering uses multiple views of the same dataset, rather than just one view, to improve clustering performance. It assumes that there is a single, true clustering of the dataset, and that the mutual information between the views can be used to find this clustering.

Alternative clustering assumes that there are multiple valid clustering solutions for a single dataset. It then finds these multiple clustering solutions based on a single view of the dataset, maximising the optimality of each individual clustering, but also maximising the dissimilarity/orthogonality between all the clusterings.

In our algorithm, dual assignment clustering, we assume that there are two valid clusterings of the dataset (similar to alternative clustering) but we also have two views on the data (similar to multi-view clustering). Each valid clustering is associated with one view. We estimate the mutual information between the two clusterings and use it to improve the results of both clus-

terings simultaneously.

3. Dual Assignment Clustering

In this section, we define the problem of dual assignment clustering, describe one possible multi-objective optimisation approach, and then describe the Dual Assignment k-Means algorithm (DAKM) as an approximation to this optimisation.

3.1. Definition

We define the specific dual assignment clustering problem as follows. We wish to cluster a set of videos into discrete groups of similar videos. We assume that there are two separate, valid clusterings of the videos: the first is based on a video's scene; the second clustering is according to the action of the video. We also assume that these two video clusterings are not independent – if the scene is known, this provides information as to the probability of the action occurring in that video. Finally, we assume that there are two views of each video – one view (derived from motion features) is generated by the action of the video, and the other view (derived from static features) is generated by the scene of the video. The aim is to produce both an action clustering and a scene clustering, estimating the relationship between actions and scenes to enhance the accuracy of both solutions.

In realistic scenarios, the relationship between actions and scenes is many-to-many. That is, a single scene can be associated with multiple actions (e.g., both cycling and walking a dog can occur in a park), and a single action can be associated with multiple scenes (e.g., basketball can be played either indoors or outdoors). Additionally, in realistic datasets certain action/scene pairs are more likely than others, and certain combinations are impossible (e.g., playing basketball in a swimming pool). We wish to capture the full complexity of this many-to-many relationship, distinct from the one-to-one assumption implicitly made in multi-view clustering, where one action corresponds to exactly one scene.

We can model this relationship using a correlation matrix, $\mathbf{R}$. $\mathbf{R}$ captures correlation information using the joint and marginal probability distributions:

$$\mathbf{R} \equiv \frac{p(A,S)}{p(A)p(S)} \tag{1}$$

If the labels of the actions and scenes are known, the joint distribution $p(A,S)$ can be approximated using the relative contingency table F, where each entry $F_{a,s}$ indicates the percentage of videos in the dataset containing both action a and scene s. $p(A)$ and $p(S)$ are calculated as the relative marginal frequencies of the actions and scenes in the whole dataset, represented by M_a and M_s respectively. $\mathbf{R}$ is thus calculated as:

$$\mathbf{R} = F \oslash (M_a \otimes M_s) \tag{2}$$

where $\otimes$ and $\oslash$ indicates Kronecker product and Hadamard division operations respectively. Thus, $\mathbf{R}_{a,s} = 1$ indicates that a and s have no correlation. Similarly $\mathbf{R}_{a,s} > 1$ shows a positive correlation, and $\mathbf{R}_{a,s} < 1$, a negative correlation.

3.2. Optimisation Problem

To provide further insight into the clustering problem, we define it as an optimisation problem. First, given a set of observations $(\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n)$, the basic k-means hard clustering algorithm optimises the following:

$$\arg\min_{\mathbf{C}} \sum_{i=1}^{n}\sum_{j=1}^{k} \mathbf{C}_{i,j}||\mathbf{x}_i - \mu_j||^2 \tag{3}$$

In hard k-means clustering, $\mathbf{C}$ is a binary cluster-membership matrix, where each element $\mathbf{C}_{i,j}$ indicates whether observation $\mathbf{x}_i$ belongs to the jth cluster, and each observation belongs to only one cluster. μ_j is the jth cluster centroid.

In the dual assignment problem, the goal is to cluster two related sets of observations (or views), $(\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n)$ and $(\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_n)$, into k^x and $k^\mathbf{y}$ sets $\mathbf{C^x}$ and $\mathbf{C^y}$ respectively, where corresponding pairs $\mathbf{x}_i$ and $\mathbf{y}_i$ co-occur, and there is an unknown (but non-zero) correlation between them. We propose an modification to the original k-means problem, making a multi-objective optimisation problem over $\mathbf{C^x}$ and $\mathbf{C^y}$. The first objective function is:

$$\arg\min_{\mathbf{C}^x,\mathbf{C}^\mathbf{y}} \sum_{i=1}^{n}\sum_{j=1}^{k^\mathbf{x}}\sum_{l=1}^{k^\mathbf{y}} \mathbf{C}^x_{i,j}||\mathbf{x}_i - \mu^\mathbf{x}_j||^2 \mathbf{C}^\mathbf{y}_{i,l}||\mathbf{y}_i - \mu^\mathbf{y}_l||^2 \tag{4}$$

This objective is essentially identical to that of the original k-means problem, extended for two sets of observations. It is intended to reduce the sum of distances-to-cluster-centroids for both $\mathbf{x}$ and $\mathbf{y}$. As this objective takes the product of the two distances rather than the sum, we do not need to account for any scale variation between $\mathbf{x}$ and $\mathbf{y}$. The second objective is:

$$\arg\min_{\mathbf{C^x},\mathbf{C^y}} -\sum_{j=1}^{k^\mathbf{x}}\sum_{l=1}^{k^\mathbf{y}} \mathbf{R}_{j,l}\log(\mathbf{R}_{j,l}) \tag{5}$$

$\mathbf{R}$ is calculated via Equation 2 using $\mathbf{Cx}$ and $\mathbf{Cy}$. Equation 5 roughly corresponds to maximising the mutual information between $\mathbf{Cx}$ and $\mathbf{Cy}$. We include this

objective, as we are interested in finding a sparse relationship between the clusters of $\mathbf{x}$ and those of $\mathbf{y}$. When Equation 5 is maximal, $\mathbf{R}$ is a uniform matrix, and no information is shared between the two clusterings – in this case, we would rather apply two individual clusterings for improved time efficiency. As $H(\mathbf{R})$ (the joint entropy between $\mathbf{x}$ $\mathbf{y}$) decreases, $\mathbf{R}$ approaches a one-to-one correspondence (or when $k^{\mathbf{x}} \neq k^{\mathbf{y}}$, a many-to-one correspondence) between the clusters in $\mathbf{x}$ and $\mathbf{y}$, and a great deal of information is shared between the two clusterings. If $H(\mathbf{R})$ is too low, however, $\mathbf{R}$ might be distorted by noise, or may be too sparse to accurately represent the relationship between $\mathbf{x}$ and $\mathbf{y}$, so we balance it with the first objective. Our regularisation method to balance these two objectives is detailed below.

3.3. Dual Assignment k-Means Algorithm (DAKM)

Simple Expectation-Maximization clustering initially seems an ideal solution to directly optimise the objectives above. This is similar to the method used for alternative clustering by minimising mutual information in [2]. This approach is intractable, however, due to our calculation of the second objective which introduces dependence between every row of $\mathbf{Cx}$ and $\mathbf{Cy}$. Instead, we devise an iterative update scheme that approximately optimises both objectives in Equations 4 and 5. The full method is shown in Algorithm 1.

First, the cluster memberships of both datasets are initialised separately using the original k-means algorithm (or using another clustering algorithm with better guarantees, such as k-means++). Then, $\mathbf{R}$, $\mathbf{C}^{\mathbf{x}}$, $\mathbf{C}^{\mathbf{y}}$, $\mu^{\mathbf{x}}$ and $\mu^{\mathbf{y}}$ are updated iteratively in a procedure also inspired by the original k-means.

The first iterative step is to estimate $\mathbf{R}$. We calculate $\mathbf{R}$ according to Equation 2, and then normalise it so all elements sum to one. Then we get $\mathbf{R}'$ according to Equation 6:

$$\mathbf{R}'_{j,l} = \frac{\log(1+\lambda \mathbf{R}_{j,l})}{\sum_{j,l} \log(1+\lambda \mathbf{R}_{j,l})} \tag{6}$$

where λ is a user-defined parameter controlling the uniformity of $\mathbf{R}'$ and $\lambda \geq 1$.

The second iterative step updates the membership variables $\mathbf{C}^{\mathbf{x}}$ and $\mathbf{C}^{\mathbf{y}}$. For each pair of samples $(\mathbf{x}_i, \mathbf{y}_i)$, we calculate the distance to every pair of clusters $j \in [1..k^{\mathbf{x}}], l \in [1..k^{\mathbf{y}}]$, and divide by $\mathbf{R}'_{j,l}$. Then we find the values of j and l that minimise the following:

$$\arg\min_{j,l} \frac{||\mathbf{x}_i - \mu^{\mathbf{x}}_j|| + ||\mathbf{y}_i - \mu^{\mathbf{y}}_l||}{\mathbf{R}'_{j,l}} \tag{7}$$

Algorithm 1: Dual Assignment k-Means (DAKM)

Data: Two sets of observations, $(\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n)$ and $(\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_n)$, where $\mathbf{x}_i$ and $\mathbf{y}_i$ co-occur
$k^{\mathbf{x}}$, $k^{\mathbf{y}}$, the number of clusters in each dataset
λ, a parameter controlling the final sparsity of $\mathbf{R}$
Result: Membership vectors $\mathbf{C}^{\mathbf{x}}$ and $\mathbf{C}^{\mathbf{y}}$

```
begin
    (C^x, μ^x) ← Kmeans (x, k^x)
    (C^y, μ^y) ← Kmeans (y, k^y)
    repeat
        R ← UpdateRelationships(C^x,C^y,λ)
        (C^x,C^y) ← UpdateMemberships(x,y,μ^x,μ^y,R)
        μ^x ← UpdateCentroids(x,C^x,k^x)
        μ^y ← UpdateCentroids(y,C^y,k^y)
    until C^x and C^y don't change
Function UpdateRelationships(C^x,C^y,λ)
    R ← zeroes(k^x,k^y), M^x ← zeroes(k^x)
    M^y ← zeroes(k^y)
    for i ← 1 to n do
        R_{C^x_i,C^y_i} ← R_{C^x_i,C^y_i} + 1/n
        M^x_{C^x_i} ← M^x_{C^x_i} + 1/n
        M^y_{C^y_i} ← M^y_{C^y_i} + 1/n
    R ← R ⊘ (M^x ⊗ M^y) (Eqn 2)
    R ← log(1+λR) / Σ_R log(1+λR) (Eqn 6)
    return R
Function UpdateMemberships(x,y,μ^x,μ^y,R)
    for i ← 1 to n do
        for j ← 1 to k^x do
            for l ← 1 to k^y do
                dists(j,l) ← (||x_i − μ^x_j|| + ||y_i − μ^y_l||) / R_{j,l}
        (C^x_i, C^y_i) ← arg min_{j,l} dists
    return (C^x, C^y)
Function UpdateCentroids(d,C,k)
    for i ← 1 to k do
        μ_i ← mean({d|(d,c) ∈ {(d,C)} ∧ c = i})
    return μ
```

As $\mathbf{R}'$ is the divisor, more points tend to be assigned to frequent cluster-pairs, and fewer points will be assigned to rarer cluster-pairs. Over many iterations, this effect implicitly minimises the second objective – the entropy of $\mathbf{R}$ – shown in Equation 5.

The final step is to update the cluster means of both datasets independently. This is identical to that in ordinary k-means. The algorithm terminates when $\mathbf{C}^{\mathbf{x}}$ and $\mathbf{C}^{\mathbf{y}}$ stop changing.

The parameter λ acts as a regularisation parameter – it serves to balance the two objectives of the optimisation problem. For higher values of λ, $\mathbf{R}'$ will tend to have a higher entropy, a more uniform distribution, and a higher weight is placed on minimising the total distance to cluster centroids; for lower values of λ, $\mathbf{R}'$ tends to lower entropy, or high sparsity, and places more weight on utilising the mutual information between the datasets. In practical use, a lower λ results in better performance but less robustness to noise – therefore, λ should be high on noisy datasets or when there is a complex relationship between $\mathbf{x}$ and $\mathbf{y}$ – if λ is too low, poor performance (below the initial baseline) will occur. However, lower values of λ result in a better clustering solution when the dataset is clean and $\mathbf{R}$ is relatively sparse.

There are two drawbacks to DAKM that may be improved in future work. Firstly, the computational complexity of DAKM is $O(k^{\mathbf{x}}k^{\mathbf{y}}n)$ for each iteration, where n is the number of items in the dataset. As such, DAKM is most practical for low values of $k^{\mathbf{x}}$ and $k^{\mathbf{y}}$. A hierarchical extension of DAKM could be considered as a more suitable alternative for larger cluster numbers. Secondly, as DAKM only approximates the objective functions given above, it does not provably find a local optimum with respect to each of them. In an extended later work we expect to demonstrate the convergence of a modified DAKM to a single, regularised objective function. Despite this, in the experiments below DAKM reliably terminates with a good result in fewer than 100 iterations in all cases, and we show consistently superior accuracy in our comparisons.

3.4. Spectral Representation

The method presented above extends the original k-means algorithm. However, k-means does not always result in the best clustering results on natural data – especially when pairs of clusters are not linearly separable. This is evident in human action clustering, where Yang et al. [14] use spectral clustering and Niebles et al. [9] use natural language techniques LDA and pLSA, rather than opting for k-means. To gain the advantages of spectral clustering for our own algorithm, we adapt DAKM to combine dual assignment clustering with a spectral representation. First, let us observe that step 5 of the spectral clustering algorithm in Ng et al. [8] is ordinary k-means clustering. It is therefore straightforward to perform steps 1 through 4 of the algorithm in [8] separately on two views of the dataset, and replace step 5 with DAKM to exploit the mutual information between the two spectral representations. We refer to this modified algorithm as Spectral DAKM (SDAKM), and we apply it to the human action clustering experiments in Section 4.2.

4. Experiments

In this section we detail several clustering experiments. To compare experimental results to the ground truth, we use the clustering accuracy metric provided in [14]. If each cluster c contains datapoints $x_1, .., x_n$, and each datapoint is associated with a ground truth label $l_1, .., l_n$, the label l_c of cluster c is:

$$\arg\max_{l_c} \sum_{i=1}^{n} \begin{cases} 1 & \text{if } l_c = l_i \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

We then calculate percentage of datapoints across the whole dataset that have the same label as their assigned cluster.

4.1. Synthetic Data Clustering

We create several synthetic datasets to demonstrate the applicability of DAKM.

4.1.1 Synthetic Data Generation

To generate the artificial dataset, we first set the total number of clusters for $\mathbf{x}$ and $\mathbf{y}$ ($k^{\mathbf{x}} = 12$, $k^{\mathbf{y}} = 8$). For relationships between $\mathbf{x}$ and $\mathbf{y}$, we randomly generate a ground truth relationship matrix $\mathbf{R}_{\text{ground}}$, where an entry of 1 indicates that two clusters can co-occur, whereas an entry of 0 indicates the opposite. We ensure there is at least one positive entry per row and column. Each cluster in $\mathbf{x}$ and $\mathbf{y}$ is represented by a 2-dimensional Gaussian distribution, where the mean is chosen randomly between a range of values, the covariance is a diagonal matrix, and the entries of the covariance vary between a range of values. Then, 10000 samples are generated – for each sample, two clusters from $\mathbf{x}$ and $\mathbf{y}$ are chosen simultaneously in accordance with $\mathbf{R}_{\text{ground}}$, then two vectors are generated from the clusters' Gaussian distributions.

These synthetic data allow us to test how various dataset properties affect the performance of DAKM – in particular, we focus on the effect of: 1) the number of relationships in $\mathbf{R}_{\text{ground}}$, and 2) how well the clusters are separated. We compare results with ordinary k-means clustering. For all of the synthetic experiments we set $\lambda = 1$.

4.1.2 Synthetic Results

We first look at the effects of varying the number of relationships between $\mathbf{x}$ and $\mathbf{y}$, which we achieve by controlling the number of non-zero entries in $\mathbf{R}_{\text{ground}}$. We show the difference in clustering performance between ordinary k-means clustering and DAKM on a synthetic

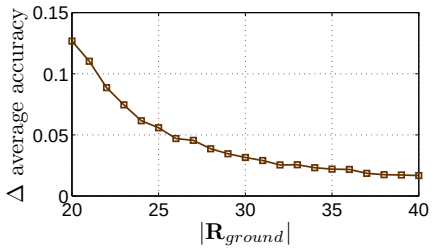

Figure 3: How improvement in DAKM accuracy (averaged over $\mathbf{x}$ and $\mathbf{y}$) changes with the number of relationships between $\mathbf{x}$ and $\mathbf{y}$.

Table 1: Performance of DAKM when view difficulty is varied.

Task Diff.	k-means		DAKM		Δ	
	$\mathbf{x}$	$\mathbf{y}$	$\mathbf{x}$	$\mathbf{y}$	$\mathbf{x}$	$\mathbf{y}$
H $\mathbf{x}$, H $\mathbf{y}$	56.7	60.0	59.3	62.4	2.7	2.4
H $\mathbf{x}$, E $\mathbf{y}$	56.5	92.8	63.3	95.9	6.7	3.2
E $\mathbf{x}$, E $\mathbf{y}$	83.0	92.8	90.9	97.5	7.9	4.7

dataset over a range of values for $|\mathbf{R}_{\text{ground}}|$ in Figure 3. As expected, as $|\mathbf{R}_{\text{ground}}|$ increases, the performance improvement decreases, as there is less mutual information to exploit between $\mathbf{x}$ and $\mathbf{y}$.

We next consider how DAKM is affected by the difficulty of the individual clustering tasks. We generate 4 views, based on how difficult they are to cluster accurately: a hard (H) $\mathbf{x}$ and hard $\mathbf{y}$, an easy (E) $\mathbf{x}$ and easy $\mathbf{y}$. The hard clustering tasks are generated in such a way that there is greater overlap between the clusters than in the easy tasks. We show the results of clustering various combinations of these views in Table 1. The accuracies displayed are for k-means and DAKM, and we additionally show the difference between the two. As can be seen, DAKM results in significant improvements over ordinary k-means in all cases. This shows the robustness of DAKM: one might expect, for instance, that the noise from a more difficult task would degrade the performance of an easier task, but this is demonstrably not the case.

4.2. Human Action Clustering

4.2.1 Datasets

We use three real-world datasets for experimentation: UCF YouTube [5], UCF Sports [11], and Hollywood-2 [7]. We report only on the accuracy for action clustering, as the scene ground truths are not provided. The UCF YouTube dataset consists of 1168 videos of various physical human activities. The UCF Sports dataset consists of 150 videos of various sports videos, taken from various broadcast sources. The Hollywood-2 dataset consists of 1707 action videos collected from Hollywood films.

4.2.2 Setup

To extract a motion representation of the actions, we use the publicly available code for dense trajectory feature extraction as presented in Wang et al. [12], with the default settings of the software. We then process the features for a bag-of-words type representation. We perform PCA and then k-means clustering on each of the descriptors separately (HOG, HOF, MBH, Tr), where we capture 95% of the variance from PCA and $k = 2000$ for the clustering. This results in a 8000-dimensional (4×2000) frequency histogram of motion features. For scene representations, we use SIFT features [6], extracted from each video at intervals of 10 frames. Once again, PCA and k-means are performed, with $k = 2000$, resulting in a 2000-dimensional frequency histogram of static features. We normalise the histograms of both the motion and static features.

To get a spectral representation, we first find the pairwise distance between all histograms in a set using the histogram intersection:

$$S(a, b) = \sum_{i=1}^{n} \min(a_i, b_i) \tag{9}$$

We apply Equation 9 to get a similarity graph for both actions and scenes. We then perform steps 2-4 of Ng et al. [8] on the actions and scenes separately to get two independent spectral representations. We then perform DAKM on the resulting vectors. We set k_a (the number of action clusters) to the number of action categories in the dataset. As the ground truths are not provided for scenes of the datasets, we have no prior knowledge of k_s, but preliminary experiments shows that a relatively high number of scenes works best – we set $k_s = 40$ for all datasets. As all algorithms are stochastic, all experiments are run 10 times and the results averaged.

4.2.3 Human Action Clustering Results

The results for motion clustering over all three datasets are summarised in Table 2. We compare the following clustering methods: Spectral Single-View (SSV), applying Ng et al. [8] to motion features only; Spectral Concatenated Views (SCV), applying Ng et al. [8] to a concatenated motion and static histogram; Co-Trained Spectral Multi-View (CMV), a recent multi-view clustering method presented in Kumar and Daume III [4], treating motion and static features as two views of the

Dataset	Clustering Accuracy (%)				
	SSV	SCV	CMV	SD1k	SDO
YouTube	39.2	40.7	38.1	41.8	**43.9**
UCF Sports	68.0	67.2	64.2	72.9	**76.0**
Hollywood-2	35.6	32.4	31.5	**36.5**	**36.5**

Table 2: Clustering performance of various methods on realistic datasets.

action; SDAKM with $\lambda = 1000$ (SD1k); SDAKM with λ set to the optimal value for each dataset individually (SDO). For SDO, we consider the following values for λ: 1, 5, 10, 50, 100, 500, 1000, 5000, 10000 and 50000.

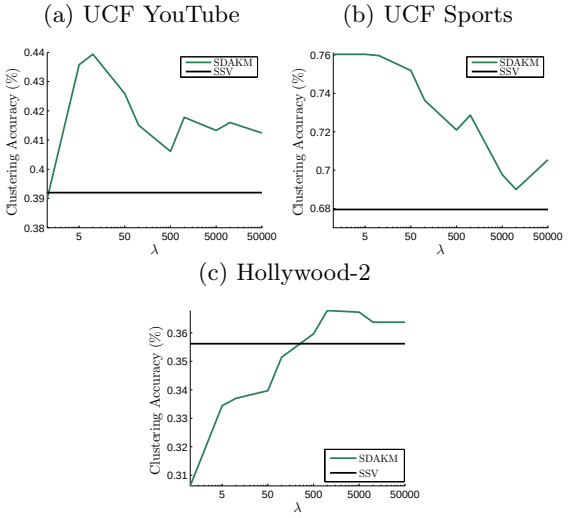

Figure 4: Performance of SDAKM with various λ.

As can be seen from the table, SDAKM with optimal λ (SDO) gives the highest accuracy on all three datasets – we propose this is because it most effectively utilises the complex relationship between actions and scenes. CMV performs far worse than the baseline clustering method (SSV) on all datasets, demonstrating that the multi-view assumption is not applicable to the relationship between motion and static features. Concatenating motion and static vectors (SCV) also negatively impacts accuracy for two of the three dataset.

SDAKM gives the greatest performance increase over the baseline on the UCF Sports dataset, which we attribute to two properties of the dataset: a highly sparse relationship $\mathbf{R}_{ground}$, and easy-to-cluster scenes. Alternatively, the weakest performance is seen on the Hollywood-2 dataset, observing only a 0.9% increase in accuracy, even with the optimal λ. This is unsurprising: Marszalek et al. [7] used context to enhance recognition accuracy on the Hollywood-2 dataset, using training data to directly infer the relationship between actions and scenes, but only observed a 1.1% improvement over baseline performance.

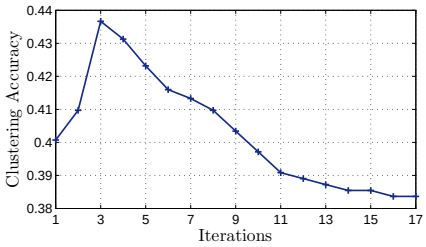

Figure 5: SDAKM on the YouTube dataset with $\lambda = 1$ – a high peak performance after few iterations, but eventually the algorithm terminates with poor performance.

We show the effect of varying parameter λ on the performance of SDAKM on the different datasets in Figure 4. Peak performance is observed at a different λ for each dataset – this is expected, as the sparsity of the unknown R_{ground} is likely to differ between datasets. In future work it would therefore be beneficial to estimate the optimal value of λ automatically using the mutual information between the two views of the dataset. However, there is still a significant improvement over SSV for all datasets when $\lambda \geq 1000$.

To understand the effect of λ further, we consider clustering on the YouTube dataset when $\lambda = 1$. We propose that under this condition, $\mathbf{R}'$ tends to become more sparse than the true relationship between actions and scenes, which will distort the clustering results. Figure 5 illustrates this situation, showing the iteration-by-iteration performance of clustering the YouTube dataset with $\lambda = 1$. Iteration by iteration, $\mathbf{R}'$ grows more sparse. Initially, this results in more accurate clustering – 3.6% better than the initial solution after the third iteration – but when the algorithm terminates, it has fallen to 1.7% below the initial accuracy, suggesting that $\mathbf{R}'$ no longer reflects the true relationship between scenes and actions.

Although none of the datasets have ground truths for the scenes, in Figure 6 we provide a few examples of discovered scene categories on the UCF YouTube dataset. Several key scene categories clearly arise: shots including the sky; playing courts; fields; swimming pools; trampolines. These are each more or less commonly associated with certain actions in the dataset, as one might expect. For instance, the sky scenes typically show golf, tennis or soccer juggling, but the playing court scenes more usually show volleyball, tennis, or basketball. Furthermore, the swimming pool and trampoline scenes have a nearly one-to-one

Figure 6: Examples of scene categories discovered in the UCF YouTube dataset, and their most strongly associated actions below.

correspondence with diving and jumping respectively. It is clear in these cases how extra information from the scene category may aid in clustering the actions.

5. Discussion

In this paper a new algorithm has been introduced – Dual Assignment k-Means clustering, or DAKM – for generating two clustering solutions according to two co-occurring views of a dataset. Unlike previous methods, it is suitable for use when there are two co-occurring views of a dataset, and a separate clustering solution associated with each view – similar previous clustering methods have either only been suitable to generate multiple clustering solutions from a single view (alternative clustering) or to generate a single clustering from multiple views (multi-view clustering). We have shown DAKM can significantly improve clustering results on synthetic data and realistic human action/scene datasets. This performance improvement is apparent even when the clusters in both views are poorly separated, demonstrating the robustness of DAKM/SDAKM.

Our further work will focus on determining λ automatically, which we believe can be calculated as a function of the mutual information between the two views of the dataset. Also, while the algorithm presented here is restricted, for complexity reasons, to considering dual-assignments only, in future we plan to consider multiple-assignment clustering.

References

[1] S. Bickel and T. Scheffer. Multi-view clustering. In *Int. Conf. Data Mining*, pages 19–26, 2004. 2

[2] X. H. Dang and J. Bailey. Generation of alternative clusterings using the cami approach. In *SIAM Int. Conf. Data Mining*, pages 118–129, 2010. 2, 4

[3] N. Ikizler-Cinbis and S. Sclaroff. Object, scene and actions: combining multiple features for human action recognition. In *Proc. European Conf. Comput. Vision*, pages 494–507, 2010. 2

[4] A. Kumar and H. Daume III. A co-training approach for multi-view spectral clustering. In *Proc. Int. Conf. Mach. Learning*, pages 393–400, 2011. 2, 6

[5] J. Liu, J. Luo, and M. Shah. Recognizing Realistic Actions from Videos "in the Wild". In *Proc. IEEE Conf. Comput. Vision and Pattern Recognition*, pages 1996–2003, June 2009. 6

[6] D. G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *Int. J. Comput. Vision*, 60:91–110, 2004. 6

[7] M. Marszalek, I. Laptev, and C. Schmid. Actions in context. *Proc. IEEE Conf. Comput. Vision and Pattern Recognition*, (i):2929–2936, June 2009. 1, 2, 6, 7

[8] A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *Advances Neural Inform. Process. Syst.*, pages 849–856, 2001. 5, 6

[9] J. C. Niebles, H. Wang, and L. Fei-Fei. Unsupervised Learning of Human Action Categories Using Spatial-Temporal Words. *Int. J. Comput. Vision*, 79(3):299–318, Mar. 2008. 2, 5

[10] A. Prest, C. Schmid, and V. Ferrari. Weakly supervised learning of interactions between humans and objects. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34(3):601–614, 2012. 2

[11] M. D. Rodriguez, J. Ahmed, and M. Shah. Action MACH: a spatio-temporal maximum average correlation height filter for action recognition. In *Proc. IEEE Conf. Comput. Vision and Pattern Recognition*, 2008. 6

[12] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu. Action Recognition by Dense Trajectories. In *Proc. IEEE Conf. Comput. Vision and Pattern Recognition*, pages 3169–3176, 2011. 6

[13] Y. Wang, H. Jiang, M. S. Drew, Z.-N. Li, and G. Mori. Unsupervised discovery of action classes. In *Proc. IEEE Conf. Comput. Vision and Pattern Recognition*, pages 1654–1661, 2006. 2

[14] Y. Yang, I. Saleemi, and M. Shah. Discovering motion primitives for unsupervised grouping and one-shot learning of human actions, gestures, and expressions. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(7):1635–1648, 2013. 2, 5