Motion Clustering Using Machine Learning

Project-II (CS47006) report submitted to

Indian Institute of Technology Kharagpur

in partial fulfilment for the award of the degree of

Bachelor of Technology

in

Computer Science and Engineering

by

Karri Sai Satish Kumar Reddy

(15CS10018)

Under the supervision of

Partha Pratim Das

Department of Computer Science and Engineering

Indian Institute of Technology Kharagpur

Spring Semester, 2019-20

May 03 , 2019

DECLARATION

I certify that

(a) The work contained in this report has been done by me under the guidance of my supervisor.

(b) The work has not been submitted to any other Institute for any degree or diploma.

(c) I have conformed to the norms and guidelines given in the Ethical Code of Conduct of the Institute.

(d) Whenever I have used materials (data, theoretical analysis, figures, and text) from other sources, I have given due credit to them by citing them in the text of the thesis and giving their details in the references. Further, I have taken permission from the copyright owners of the sources, whenever necessary.

Date: May 03 , 2019 (Karri Sai Satish Kumar Reddy)

Place: Kharagpur (15CS10018)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR

KHARAGPUR - 721302, INDIA

CERTIFICATE

This is to certify that the project report entitled "**Motion Clustering Using Machine Learning**" submitted by **Karri Sai Satish Kumar Reddy** (Roll No. 15CS10018) to Indian Institute of Technology Kharagpur towards partial fulfilment of requirements for the award of degree of Bachelor of Technology in Computer Science and Engineering is a record of bona fide work carried out by him under my supervision and guidance during Spring Semester, 2019-20.

Date: May 03 , 2019

Place: Kharagpur

Partha Pratim Das
Department of Computer Science and Engineering
Indian Institute of Technology Kharagpur
Kharagpur - 721302, India

Abstract

Name of the student: **Karri Sai Satish Kumar Reddy** Roll No: **15CS10018**

Degree for which submitted: **Bachelor of Technology**

Department: **Department of Computer Science and Engineering**

Thesis title: **Motion Clustering Using Machine Learning**

Thesis supervisor: **Partha Pratim Das**

Month and year of thesis submission: **May 03 , 2019**

Clustering analysis has been an emerging research issue in data mining due its variety of applications.In this paper,we try to apply clustering for the domain of dance videos.We attempted to group the similar motions in a given "BharataNatyam" dance video.We have explained indetail the techniques used in our approach like optical flow,DTW and Spectral Clustering in chapter2.Then we explained our approach and concluded with the results of the approach.

Acknowledgements

I would like to express my sincere thanks to my advisor Prof.Partha Pratim Das.He was always approachable despite of his tight schedule.

I am also thankful to Himadri Buyan for helping me with a lot of valuable discussions.I express my profound thanks towards all faculty of Computer Science and Engineering department at Indian Institute of Technology, who build strong foundations for research through various courses.

Finally, I want to thank my family members and friends for their continuous support and love

Contents

Chapter 1

Introduction

1.1 Clustering

Clustering analysis has been an emerging research issue in data mining due its variety of applications.Clustering is the task of dividing data points into a number of groups such that data points in the same groups are more similar to other data points in the same group than those in other groups. In simple words, the aim is to segregate groups with similar traits and assign them into clusters.

Clustering is used in many fields like Market research,Biology,Search result grouping,Analysis of antimicrobial activity,Human genetic clustering etc.We are trying to use clustering to group similar motion segments in dance videos.In particular,we have clustered the **BharataNatyam** natta videos.

With the advent of many data clustering algorithms in the recent few years and its extensive use in wide variety of applications, including image processing, computational biology, mobile communication, medicine and economics, has lead to the popularity of this algorithms.Main problem with the data clustering algorithms is that it cannot be standardized. Algorithm developed may give best result with one type of data set but may fail or give poor result with data set of other types. Although there has been many attempts for standardizing the algorithms which can perform well in all case of scenarios but till now no major accomplishment has been achieved.

Many clustering algorithms have been proposed so far. However, each algorithm has its own merits and demerits and cannot work for all real situations.So,we have looked into different clustering algorithms and found that **Spectral Clustering** suits our purpose well.We have explained spectral clustering in the coming chapters.

1.2 Motion Representation & Similarity Measure

The problem of clustering in the domain of dance is difficult compared to other domains because there was no fixed representation for the dance motions.we have used **optical flow** to represent each motion as a feature vector.Since each motion spans different no of frames,the size of feature vector is different for different motions.

In clustering,for deciding whether two motions should go to same cluster or different clusters,we should have a measure to compare two motions.Each motion spans across different no of frames even the similar ones(for example motion1 may span 10 frames while motion2 may frame 15 frames).Depending on the frame rate,even the similar motions may have different frames.Hence we need to design a similarity measure to compare different sized feature vectors corresponding to each motion.

The similarity measure should be more for similar motions and less for different motions and it should be able to compare motions with different no of frames.In this report,we proposed a measure which deals with all these problems and giving a good accuracy.

1.3 Problem Statement

The goal of this research is to effectively represent motions and design a similarity measure to cluster the motions of the given **BharataNatyam** dance videos.Specifically we have tested our approach on bharatanatyam second Adavu **Natta**.The input given was a set of videos where a dancer will be performing a Natta and a motion annotation file representing the motion frame details in the video.

1.4 DataSet

The domain we are dealing is Indian Bharatanatyam. Adavu's are the ba- sic units of Bharatanatyam. While performing Adavu's, the dancer stamps, rubs, touches, slides on the ground in different ways in synchronization with the music.There are 15 Adavus in which **Natta** is the second one which we are dealing.In Natta,there are again 8 variations.

For each variation, there are 9 sets of data corresponding to 9 dancers, each set has a video and an RGB image, a skeleton image, a depth image and a mat file containing information of skeleton image for each frame in the video as shown in the fig 3.

(a) RGB Image (b) Depth Image (c) Skeleton Image

FIGURE 1.1: Different Channels for each frame in video

In this semester,we have worked on clustering of motions using RGB image dataset.The following picture shows the scale of the data we worked on.

Adavu	Performance1 total frames	Performance 2 total frames	Performance 3 total frames	Total motions	no of unique motions
Natta1	1546	1590	1532	32	4
Natta2	1557	1522	1545	32	4
Natta3	2680	2698	2760	64	8
Natta4	5537	5531	5504	128	8
Natta5	2580	2728	2748	64	10
Natta6	2781	2764	2729	64	12
Natta7	2828	3022	2706	64	14
Natta8	2710	2811	2752	48	11

FIGURE 1.2: Picture showing our DataSet

Along with these videos,we also have a motion annotation file with each video consisting of motion start and end frame details for each motion in the video.

1.5 Organization of this Report

Chapter 2 introduces the various techniques used in our approach.we explained the optical flow which we used it to represent motions in our scenario.we also explained the Dynamic Time Warping which is used as the similarity measure and then the Spectral Clustering.

In Chapter 3,we have outlined the approach we followed and the results we obtained on the mentioned dataset and discussed some insights on the results.Finally ,the conclusions will be summarized and the evaluation measure will described.Also possible areas for future work are stated briefly.

Chapter 2

Background, Definitions

In this chapter, we provide the reader with some material on the domain of our research work. A description of the methods, algorithms and theoretical features that one involved in this domain will give the reader a better understanding of the current work.

2.1 Related Work

In the past two decades, motion capture systems were able to track and record human motion with high spatial and temporal resolution. The extensive proliferation of motion databases urges the development of efficient techniques to index and build models of human motion. One key aspect to understand and build better models of human motion is to develop unsupervised algorithms for decomposing human motion into a set of actions and cluster those actions.

Feng Zhou[1] proposed Aligned Cluster Analysis (ACA), an extension of kernel k-means clustering that allows unsupervised clustering of temporal patterns and they worked on motion capture data. Hongbin Wang,Hua Lin[3] provide a novel spectral clustering approach to segment multiple moving objects.But they haven't attemped for clustering.There are very few papers on this field hence we attempted to solve this particular clustering problem.Also,the available papers have only attempted

to cluster using the trajectory of the motions.Hence they worked with the motion capture data.But we attempted to solve the problem using RGB videos.

2.2 Optical flow and Motion

Optical flow is the pattern of apparent motion of image objects between two consecutive frames caused by the movement of object or camera. It is 2D vector field where each vector is a displacement vector showing the movement of points from first frame to second.Optical flow has many applications in areas like Motion estimation and video compression.Consider the image below.It shows a ball moving in 5 consecutive frames. The arrow shows its displacement vector.

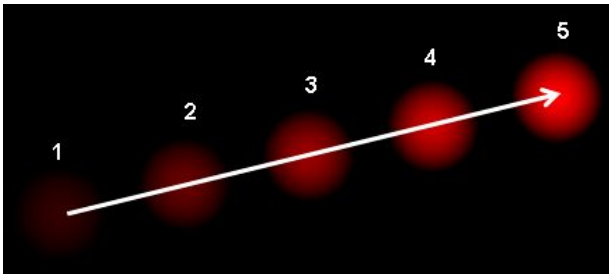


FIGURE 2.1: Arrow showing optical flow

Optical flow works on several assumptions:

1. The pixel intensities of an object do not change between consecutive frames.
2. The motion between consecutive frames is small.
3. Neighbouring pixels have similar motion.

Consider a pixel $I(x, y, t)$ in first frame.It moves by distance (dx, dy) in next frame taken after dt time. So since those pixels are the same and intensity does not change, we can say,

$$I(x, y, t) = I(x + dx, y + dy, t + dt)$$

Then taking taylor series approximation of right-hand side and neglecting higher order terms

$$I(x, y, t) = I(x, y, t) + \frac{\partial I}{\partial x}dx + \frac{\partial I}{\partial y}dy + \frac{\partial I}{\partial t}dt$$

$$\frac{\partial I}{\partial x}dx + \frac{\partial I}{\partial y}dy + \frac{\partial I}{\partial t}dt = 0$$

Dividing by dt we get

$$f_x u + f_y v + f_t = 0 \tag{2.1}$$

where

$$f_x = \frac{\partial f}{\partial x} \quad f_y = \frac{\partial f}{\partial y} \quad u = \frac{dx}{dt} \quad v = \frac{dy}{dt}$$

Above equation is called Optical Flow equation. In it, we can find f_x and f_y, they are image gradients. Similarly f_t is the gradient along time. But (u, v) is unknown. We cannot solve this one equation with two unknown variables. So several methods are provided to solve this problem and one of them is Lucas-Kanade.

2.2.1 Lucas-Kanade method

We have seen an assumption before, that all the neighbouring pixels will have similar motion. Lucas-Kanade method takes a 3x3 patch around the point. So all the 9 points have the same motion. We can find (f_x, f_y, f_t) for these 9 points. So now our problem becomes solving 9 equations with two unknown variables which is over-determined.Those 9 equations are represented in a matrix and using the concept of psuedo inverse as shown below.

$$AU = F$$
$$A^T AU = A^T F$$
$$U = (A^T A)^{-1} A^T F$$

where

$$A = \begin{bmatrix} f_{x1} & f_{y1} \\ f_{x2} & f_{y2} \\ \vdots & \vdots \\ f_{x9} & f_{y9} \end{bmatrix} \quad U = \begin{bmatrix} u \\ v \end{bmatrix} \quad F = \begin{bmatrix} -f_{t1} \\ -f_{t2} \\ \vdots \\ -f_{t9} \end{bmatrix}$$

Below is the final solution which is two equation-two unknown problem and solve to get the solution.

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \sum_i {f_{x_i}}^2 & \sum_i f_{x_i} f_{y_i} \\ \sum_i f_{x_i} f_{y_i} & \sum_i {f_{y_i}}^2 \end{bmatrix}^{-1} \begin{bmatrix} -\sum_i f_{x_i} f_{t_i} \\ -\sum_i f_{y_i} f_{t_i} \end{bmatrix}$$

The good thing was this was all already implemented in **open cv**.Finally note that this derivation is valid only for detecting small motions.For our purpose,the motion between two consecutive frames is very less.Hence this method can be safely applied.If we obtain (u, v) for every pixel in the image,then it is called dense optical flow and we have used dense optical flow in the proposed approach.

There are several other algorithms proposed to solve optical flow.you can refer them here[6]

2.3 Histogram of Optical flow

A feature descriptor is a representation of an image or an image patch that simplifies the image by extracting useful information and throwing away extraneous information.Typically, a feature descriptor converts an image of size width x height x 3 (channels) to a feature vector / array of length n.Histogram of Optical Flow (HOF) is one such descriptor.In the case of the HOG feature descriptor, the input image is of size $h * w * 3$ and the output feature vector is of length $h * w * 9/64$.

In the HOF feature descriptor, the distribution (histograms) of directions of optical flow are used as features.Since, we are trying to cluster motions,optical flow serves as good feature for capturing motion.The following shows the steps in the calculation of HOF descriptor.

Step 1 : Calculate Optical Flow

To calculate a HOF descriptor, we need to first calculate the optical flow; after all, we want to calculate the histogram of optical flow.To calculate the optical flow, use the methods described above.Now we get two components (u, v) foe every pixel by using the optical flow.

Step 2 : Calculate Histogram of Gradients in $8 * 8$ cells

In this step, the image is divided into $8 * 8$ cells and a histogram of optical flow is calculated for each $8*8$ cells.One of the important reasons to use a feature descriptor to describe a patch of an image is that it provides a compact representation.

An $8 * 8$ image patch contains $8 * 8 * 3 = 192$ pixel values. The optical flow of this patch contains 2 values (magnitude and direction) per pixel which adds up to $8 * 8 * 2 = 128$ numbers. By the end of this section we will see how these 128 numbers are represented using a 9-bin histogram which can be stored as an array of 9 numbers. Not only is the representation more compact,calculating a histogram over a patch makes this representation more robust to noise. Individual optical flow may have noise, but a histogram over $8*8$ patch makes the representation much less sensitive to noise.

The histogram is essentially a vector (or an array) of 18 bins (numbers).First convert the optical flow (u, v) of every pixel into polar coordinates.The bins can be made using following two techniques

Interval Binning

In this technique, the bins are considered as intervals 0 - 20 , 20 - 40 ... , 340 - 360. For every pixel in the $8 * 8$ patch a bin is selected based on the direction, and the vote (the value that goes into the bin) is selected based on the magnitude.

Weighted Binning

In this technique, The histogram contains 18 bins corresponding to angles 0, 20, 40 ... 160. For every pixel in the $8 * 8$. Depending on the direction, the magnitude is split across at most two bins depending on the distance of direction from bin angle.An illustration of this is shown below.

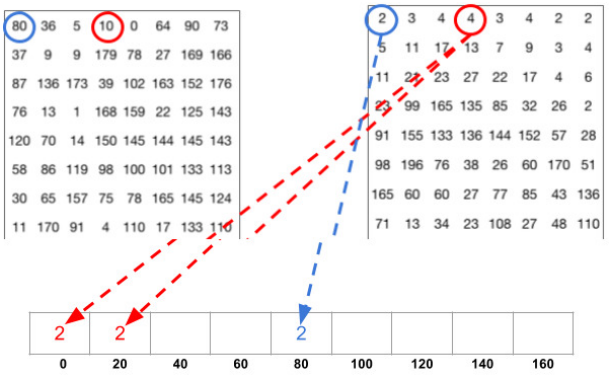


FIGURE 2.2: Weighted Binning illustration

Let's first focus on the pixel encircled in blue. It has an angle (direction) of 80 degrees and magnitude of 2. So it adds 2 to the 5th bin. The optical flow at the pixel encircled using red has an angle of 10 degrees and magnitude of 4. Since 10 degrees is half way between 0 and 20, the vote by the pixel splits evenly into the two bins.

Step 4 : $16 * 16$ Block Normalization

In order the descriptor to be independent of lighting variations, we would like to "normalize" the histogram so they are not affected by lighting variations.A $16 * 16$ block has 4 histograms which can be concatenated to form a $36 * 1$ element vector and it will be normalized.The window is then moved by 8 pixels and a normalized $36 * 1$ vector is calculated over this window and the process is repeated.After doing this,all the bins are concatenated to form the final feature vector.

But for our motion clustering, this step didn't worked well(Accuracy without this step was good).Hence we just didn't up to step 3.Let us say for each frame the HOF vector is of size x. If a motion has f frames the feature vector size will be of size $f * x$. Since it depends on number of frames,feature vector for each motion may not be of same size.

2.4 Dynamic Time Warping (DTW)

The distance between two point $x = [x_1, x_2, ..., x_n]$ and $y = [y_1, y_2, ..., y_n]$ in a n-dimensional space can be computed via the Euclidean distance:

$$dist(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\| = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \cdots + (x_n - y_n)^2}.$$

However, if the length of **x** is different from **y**, then we cannot use the above formula to compute the distance. Instead, we need a more flexible method that can find the best mapping from elements in **x** to those in **y** in order to compute the distance.

The goal of **dynamic time warping** (DTW for short) is to find the best mapping with the minimum distance by the use of DP. The method is called **"time warping"** since both x and y are usually vectors of time series and we need to compress or expand in time in order to find the best mapping.

Let t and r be two vectors of lengths m and n, respectively. The goal of DTW is to find a mapping path $\{(p_1, q_1), (p_2, q_2), ..., (p_k, q_k)\}$ such that the distance on this mapping path $\sum_{i=1}^{k} |t(p_i) - r(q_i)|$ is minimized, with the following constraint:

- The ordering should be preserved meaning if i is mapped with j then i-1 can be mapped with any index until j but not beyond j.This local constraint guarantees that the mapping path is monotonically non-decreasing in its first and second arguments. Moreover, for any given element in t, we should be able to find at least one corresponding element in r, and vice versa.

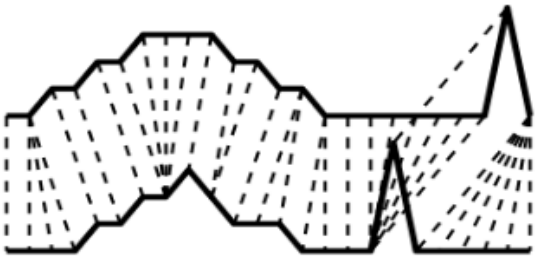


FIGURE 2.3: Figure showing the mapping obtained by DTW

2.4.1 Solving DTW

we can solve it using the following dynamic programming(DP) approach.

1. **Optimum-value function** : Define $D(i, j)$ as the DTW distance between $t(1 : i)$ and $r(1 : j)$, with the mapping path starting from $(1, 1)$ to (i, j)
2. **Recursion** :

$$D(i, j) = |t(i) - r(j)| + min \left\{ \begin{matrix} D(i-1, j) \\ D(i-1, j-1) \\ D(i, j-1) \end{matrix} \right\},$$

   with the initial condition $D(1, 1) = |t(1) - r(1)|$
3. **Final answer** : $D(m, n)$

If we want to know the optimum mapping path in addition to the minimum distance, we may want to keep the optimum fan-in of each node. Then at the end of DP, we can quickly back track to find the optimum mapping path between the two input vectors.

There are many attempts made to make DTW run faster.you can refer [5] to know about a faster version of DTW.

2.4.2 Applications

Dynamic time warping is mainly used in speech recognition,since speech is very sensitive for variations in time,but the technique has proved to be useful for many other applications like gesture recognition, robotics, data mining,hand writing recognition etc

For our purpose,we used DTW for measuring similarity between motions.As we stated earlier that each motion is represented by a feature vector of different sizes.Hence we use DTW to compare these unequal sized feature vectors.Everything remains same except the t,r vectors which in our case are not simple vectors but vector of vectors.This will be explained in detail in the coming chapters.

2.5 Spectral Clustering

The goal of spectral clustering is to cluster data that is connected but not necessarily compact or clustered within convex boundaries.It very often outperforms traditional clustering algorithms such as the k-means algorithm.

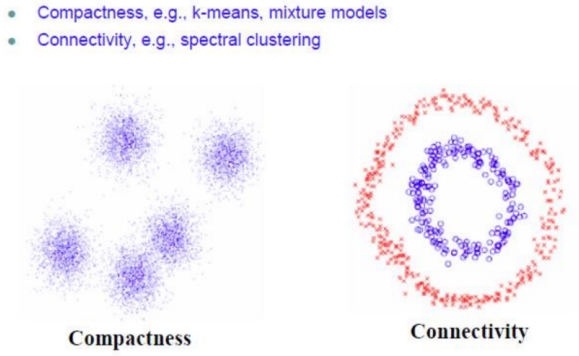


FIGURE 2.4: Example explaining two properties of data

2.5.1 Basic Idea

- Spectral clustering models the objective function as a graph spectrum problem and solves it with the known linear algebra techniques.
- It connects the objective function to one of the following property of graph laplacian and solves it in place of minimizing objective function.

For every vector $f \in R^n$ we have

$$f^{'} L f = \frac{1}{2} \sum_{i,j=1}^{n} w_{ij} \left(f_i - f_j\right)^2$$

where

$$L = D - W, \quad \text{W is the adjacency Matrix}$$

$$d_i = \sum_{j=1}^{n} w_{ij}$$

D is a diagonal Matrix with d_i as diagonal

2.5.2 Algorithm

1. **Input** : Similarity matrix $W \in R^{n*n}$, number k of clusters to construct
2. Compute the unnormalized Laplacian L.
3. Compute the first k eigenvectors $(u_1, u_2, \ldots, u_k)$ of L.
4. Let $U \in R^{n*k}$ be the matrix containing the vectors $u_1, u_2, \ldots, u_k$ as columns.
5. Let $i = 1 \ldots n$ let $y_i \in R^k$ be be the vector corresponding to the i_{th} row of U
6. Cluster the points $(y_i)_{i=1,\ldots,n}$ in R^k with k-means clustering algorithm into $C_1, \ldots, C_k$ clusters.
7. **Output** : Clusters $A_1, A_2, \ldots, A_k$ with $A_i = \{j | y_i \in C_j\}$

2.5.3 Objective Function

The following is the Objective function the spectral clustering is trying to minimize

$$\text{RatioCut}(A_1, \ldots, A_k) := \frac{1}{2} \sum_{i=1}^{k} \frac{W(A_i, \overline{A}_i)}{|A_i|} = \sum_{i=1}^{k} \frac{\text{cut}(A_i, \overline{A}_i)}{|A_i|}$$

$$\text{Ncut}(A_1, \ldots, A_k) := \frac{1}{2} \sum_{i=1}^{k} \frac{W(A_i, \overline{A}_i)}{\text{vol}(A_i)} = \sum_{i=1}^{k} \frac{\text{cut}(A_i, \overline{A}_i)}{\text{vol}(A_i)}.$$

FIGURE 2.5: Objective Function

Intuitively this function is trying to minimize the sum of crossing edges between clusters while trying to maximize the within cluster edges.Note that here edge weight represent similarity.In other words,it is trying to maximize within cluster similarity while minimizing across cluster similarity.

2.5.4 How the algorithm is minimizing the above Function

This section shows how graph laplacian is related to the above objective function and hence we can minimize the laplacian property instead of that objective function.

2.5.4.1 Approximating Ratio Cut for K = 2

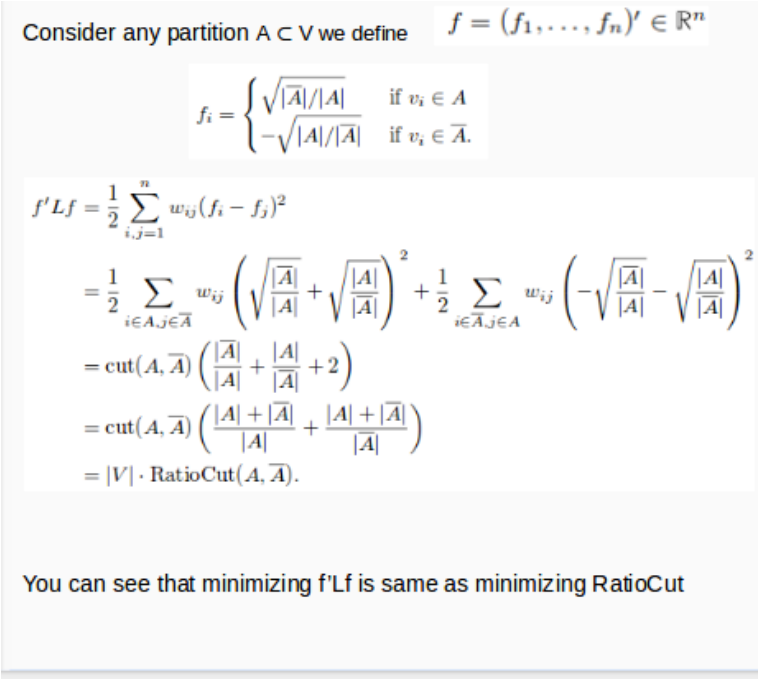


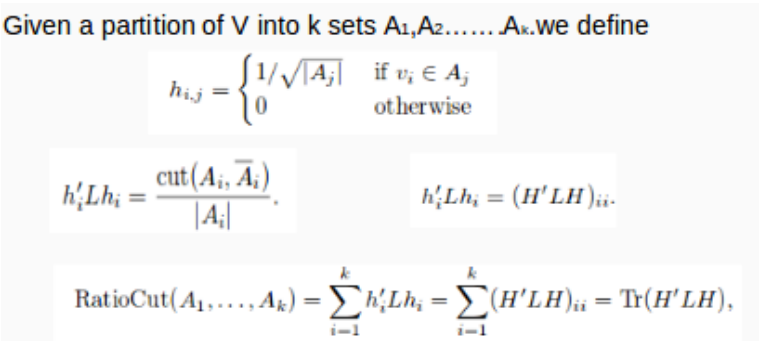
FIGURE 2.6: Figure showing how Laplacian property related to RatioCut

Finally we should minimize the below equation to minimize the objective function

$$\min_{A\subset V} f'Lf \text{ subject to } f \perp \mathbb{1},\ f_i \text{ as defined in Eq. (2)},\ \|f\| = \sqrt{n}.$$

This is a discrete optimization problem but it is NP Hard.Relaxing f to take real values makes it solvable.By the Rayleigh-Ritz theorem, the solution to this is eigenvector corresponding to the second smallest eigenvalue of L.

2.5.4.2 Approximating Ratio Cut for K > 2



The final minimization problem is

$$\min_{H \in \mathbb{R}^{n \times k}} \mathrm{Tr}(H'LH) \text{ subject to } H'H = I.$$

By the Rayleigh-Ritz theorem,the solution is given by choosing H as the matrix which contains the first k eigen vectors of L as columns.So,for this reason we are calculating first k eighen vectors in the algorithm.we can refer [4] for more indepth details on spectral clustering.

2.6 Support Vector Machine

A Support Vector Machine (SVM) is a supervised machine learning algorithm that can be employed for both classification and regression purposes. SVMs are more commonly used in classification problems.SVMs are based on the idea of finding a hyperplane that best divides a dataset into two classes, as shown in the image below.

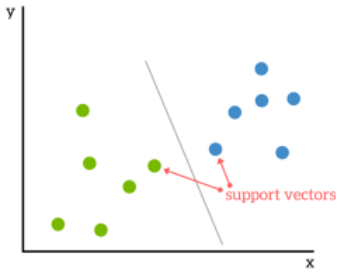


FIGURE 2.7: Figure demonstrating hyperplane

2.6.1 Support Vectors

Support vectors are the data points nearest to the hyperplane, the points of a data set that, if removed, would alter the position of the dividing hyperplane. Because of this, they can be considered the critical elements of a data set.

2.6.2 HyperPlane

As a simple example, for a classification task with only two features (like the image above), you can think of a hyperplane as a line that linearly separates and classifies a set of data.

Intuitively, the further from the hyperplane our data points lie, the more confident we are that they have been correctly classified. We therefore want our data points to be as far away from the hyperplane as possible, while still being on the correct side of it.So when new testing data is added, whatever side of the hyperplane it lands will decide the class that we assign to it.

2.6.3 Finding the right hyperplane

The distance between the hyperplane and the nearest data point from either set is known as the **margin**. The goal is to choose a hyperplane with the greatest possible margin between the hyperplane and any point within the training set, giving a greater chance of new data being classified correctly.

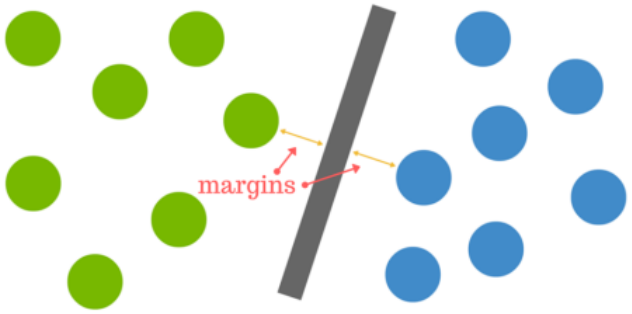


FIGURE 2.8: Figure demonstrating margin

2.6.4 Linearly non separable data

We will transform the axes and convert the data into linearly separable.The transformation of data is done using concept called **kernels**.The below figures demonstrates how the transformation is done.

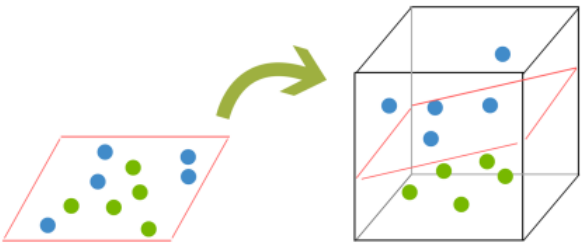


FIGURE 2.9: Figure demonstrating kernels

For mathematical foundations of SVM, refer CS229 Lecture notes.

2.7 K- Nearest Neighbour (KNN)

k-nearest neighbors algorithm (k-NN) is a non-parametric method used for classification and regression.In this supervised learning technique,there is no training phase.when a new data point come,it will be assigned the class of training data point which is closest to the new data point.The "closest" is defined as per application requirements.In our context the closest is defined using the measure DTW.

2.8 Accuracy Measure (Rand Index)

In order to evaluate the results obtained by our algorithm,we have used **rand index** as a evaluation measure for computing accuracy.

If C is a ground truth class assignment and K the clustering,then let:

- a, the number of pairs of elements that are in the same set in C and K
- b, the number of pairs of elements that are in different sets in both C and K

$$RI = \frac{a+b}{\binom{n_{samples}}{2}}$$

Chapter 3

Chapter3 Approach & Results

We have used both unsupervised learning and supervised learning techniques They can be explained in brief as follows:

Unsupervised Learning

1. Used Dense optical flow and obtained the feature vector of each motion.
2. Applied DTW as similarity measure for motions and obtained similarity matrix.
3. Used Spectral Clustering to cluster the data using the above similarity matrix.

Supervised Learning

1. Used Dense optical flow to get the features for each motion.
2. Made all vectors equal size by appending with small value(1e-5).
3. Used SVM for the classification.

3.1 Unsupervised Approaches

3.1.1 Feature vector for representing motions

From the given annotation file,we know for each motion the starting and ending frame number.Let the frame corresponding to a motion be $[s, t]$. Taking frame "s" as reference we computed dense optical flow for all the frames in the range $[s+1, t]$.

If a motion has f frames each of size 640*480 The size of feature vector will be $2 * 640 * 480 * f$.The optical flow (u, v) for each pixel so the 2 in the equation.Each motion feature vector can be visualized as follows:

$$fV[1, \ldots, f], \quad \text{where f is no of frames in the motion}$$

$$fV[i] \text{ is a 3d matrix as } V[640][480][2]$$

3.1.2 Similarity Matrix from Feature Vectors

These feature vectors are of different sizes because of different no of frames for each motion.So,used DTW approach to compare them.As $fV1[i], fV2[j]$ are not simple numbers as explained in previous chapter,so, we define the quantity $|fV1[i] - fV2[j]|$ as follows.

we first made

$$temp[i][j] = \text{euclidean distance between } fV1[i][j][2] \text{ and } fV2[i][j][2]$$

$$i = 1, \ldots, 640 \text{ and } j = 1, \ldots, 480$$

And finally,

$$|fV1[i] - fV2[j]| = \sum_{i=1}^{640} \sum_{j=1}^{380} temp[i][j]$$

Suppose there are M motions,so we define a dissimilarity matrix in this way

$$DSM[i][j] = DTW(fV[i], fV[j])$$

$$i = 1, \ldots, M \text{ and } j = 1, \ldots, M$$

Similarity Matrix is obtained as

$$SM[i][j] = \frac{1}{DSM[i][j]}$$

3.1.3 Clustering

After obtaining similarity matrix as explained above,we tried **Affinity Propagation** and **Spectral Clustering**.In Affinity Propagation, the no of clusters best for the dataset is automatically obtained.But in case of complex nattas,the no of clusters automatically obtained is not correct.So,we left Affinity Propagation and taken Spectral Clustering.You can see the results section for more details.

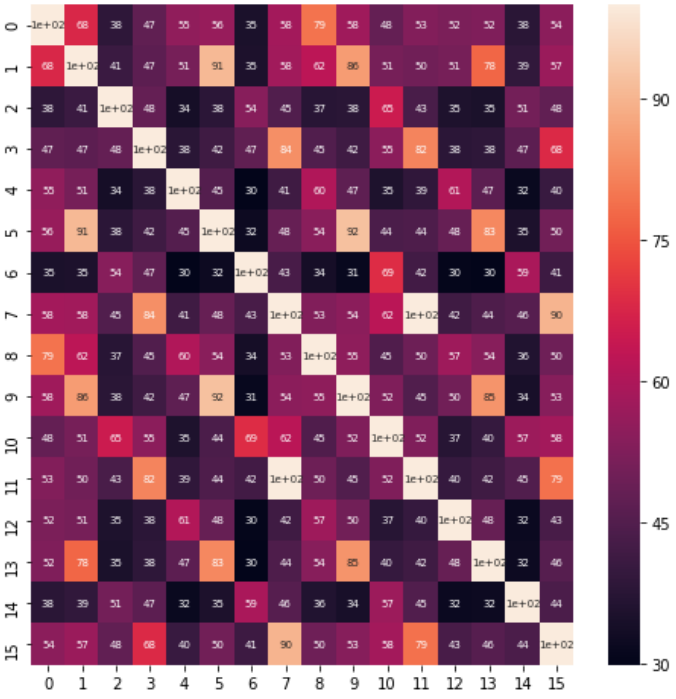
3.1.4 Results

Natta	Total motions	Total no of motion pairs	Correctly classified Pairs	Rand_Index
Natta1	32	992	992	100%
Natta2	32	992	834	84.05%
Natta3	64	4032	3111	77.17%
Natta4	128	16256	10840	66.68%
Natta5	64	4032	3210	79.61%
Natta6	64	4032	3003	74.5%
Natta7	64	4032	2163	53.66%
Natta8	48	2256	1373	60.87%

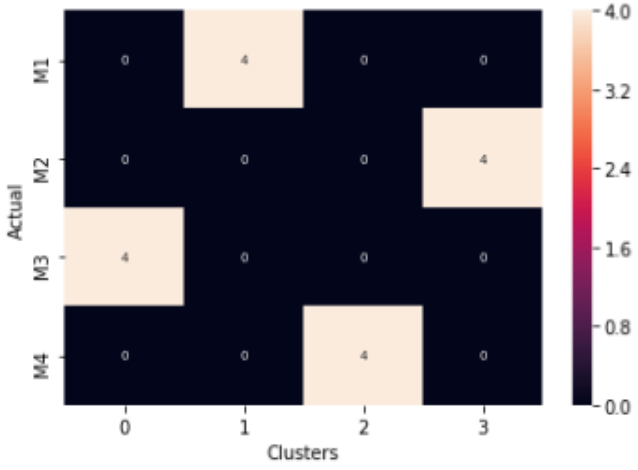
The accuracy is measured using the rand index as described in the previous chapter.We can observe the accuracy is really good for the simple motions in the initial Nattas.But as complexity of motions increases in the later nattas,the accuracy was decreased.We have analyzed reasons for less accuracy in the Analysis section.

3.1.4.1 Similarity Matrix

We can see the similarity matrix and observe the following point.Motion1 is similar with Motion5,Motion9,Motion13 which is evident by light colour in the figure.Motion0 is similar with Motion8.But because Motion4 is similar with Motion8,the Motion0,Motion4,Motion8 went into same cluster as we wanted.



3.1.4.2 Confusion Matrix



Here you can see there are four motions of type M1 and every motion went into cluster 1.The results for other nattas and dancer can be found here.

3.1.5 Different Variations tried

Generally Histogram of Optical Flow is used in place of using dense optical flow directly.Hence, we tried to use HOF in place of Dense Optical Glow in the first step.Here,we tried two variations of HOF weighted binning and Interval binning.

Natta	Optical flow Accuracy	HOF weighted binning	HOF interval binning
Natta1	100%	100%	100%
Natta2	84.05%	91.38%	91.33%
Natta3	77.17%	78%	78.15%
Natta5	79.61%	62%	63.15%
Natta7	53.66%	24.8%	24.6%
Natta8	60.87%	27.3%	26.7%

We have also tried varying binsizes to see how it affects accuracy. Increasing bin size decreases the accuracy of simple motions but increases the accuracy of complex motions.

Natta	HOF(10)	HOF(20)	HOF(40)	HOF(60)
Natta1	100%	100%	100%	100%
Natta3	79.66%	78.15%	76.7%	76.7%
Natta5	55.55%	63.15%	72%	72.36%
Natta7	19.13%	24.6%	34%	38%
Natta8	18.19%	26.7%	26%	27%

3.1.6 Reasons for Low Accuracy for Natta7

In Natta7, the number of frames for each motion differs too much.For example M1 contains 62 frames whereas M2 contains 4 frames Consider three motions M1,M2 and M1 in the next cycle Since M2 has less frames DTW is assigning less cost to M1 and M2 rather than to M1 and M1 in the next cycle.So the motions with less no of frames are affecting the similarity matrix.

Noisy Motions : The motions having less than 10 frames are causing the accuracy to be low as explained above.Also it is highly impractical for a motion to be in such less no of frames. Hence we treat motions with less frames and removed them and recalculated accuracy for the above methods.

3.1.7 Results without noisy motions

Adavu	Optical flow accuracy with noisy images	Optical flow accuracy without noisy images	HOF accuracy with noisy images	HOF accuracy without noisy images
Natta1	100%	100%	100%	100%
Natta2	84.05%	84.05%	91.33%	91.33%
Natta3	77.17%	77.17%	78.15%	78.15%
Natta4	66.68%	66.68%	60.68%	60.68%
Natta5	79.61%	79.61%	63.15%	63.15%
Natta6	74.5%	85.6%	74.5%	80%
Natta7	53.66%	78.67%	24.66%	72.64%
Natta8	60.87%	76.67%	26.7%	72.6%

3.2 Supervised Learning

As we have seen dense optical flow as the better feature compared to HOF. we used dense optical flow for the feature extraction.But the problem is feature vector are of different sizes for different motions,we made all of them equal size by appending with a small value (1e-5).we then used SVM for classification.We have trained the svm on one performance of the dancer and tested on the other performance.

3.2.1 SVM Results

Natta	Accuracy	Accuracy without noisy frames
Natta1	100%	100%
Natta2	100%	100%
Natta3	83%	83%
Natta4	60%	60%
Natta5	65%	65%
Natta6	64%	66%
Natta7	23%	30%
Natta8	29%	40%

Here note that removing noisy motions doesn't increased accuracy much because noisy motions affect only the DTW Result.

3.2.2 Approach 2 : K - Nearest Neighbour

In this approach we have taken one performance of a dancer as training data and tested it on the other performance.For each motion in test data we assign it to the most similar motion in training set.The "most similar" is measured by Dense Optical Flow and DTW.

Here the removal of noisy frames had increased accuracy since here we used DTW and the noisy motions are effecting DTW result.

Natta	Accuracy with noisy frames	Accuracy without noisy frames
Natta1	100%	100%
Natta2	94%	94%
Natta3	80%	80%
Natta4	60.56%	60.56%
Natta5	70%	70%
Natta6	62%	70%
Natta7	18%	60%
Natta8	20%	58.67%

3.3 Conclusion & Future Work

In this work we tried various approaches for clustering of motions.In the unsupervised approach, using dense optical flow we achieved an average accuracy of 81% and by using HOF we achieved 77% average accuracy.However using HOF decreased the running time of the algorithm by half since it decreases the feature vector size.we have also tried using SVM for classification but the accuracy was not so good.

In future,we want to automate the process to find optimal no of clusters in the video.we also wanted to extend this approach to skeletal videos.we aim to improve the SVM classification approach by training on more data.

Bibliography

[1] Feng Zhou, F. D. l. Torre and J. K. Hodgins, "Aligned Cluster Analysis for temporal segmentation of human motion," 2008 8th IEEE International Conference on Automatic Face & Gesture Recognition, Amsterdam, 2008, pp. 1-7. doi: 10.1109/AFGR.2008.4813468 publisher

[2] Kexue Dai, Guohui Li and Defeng Wu, "Motion clustering for similar video segments mining," 2006 12th International Multi-Media Modelling Conference, Beijing, 2006, pp. 4 pp.-. doi: 10.1109/MMMC.2006.1651368 publisher

[3] Hongbin Wang and Hua Lin, "A spectral clustering approach to motion segmentation based on motion trajectory," 2003 International Conference on Multimedia and Expo. ICME '03. Proceedings (Cat. No.03TH8698), Baltimore, MD, USA, 2003, pp. II-793. doi: 10.1109/ICME.2003.1221736 publisher

[4] von Luxburg, U. Stat Comput (2007) 17: 395. https://doi.org/10.1007/s11222-007-9033-z publisher

[5] FastDTW: Toward Accurate Dynamic Time Warping in Linear Time and Space Stan Salvador and Philip Chan Dept. of Computer Sciences Florida Institute of Technology Melbourne, FL 32901 publisher

[6] Optical Flow Estimation David J. Fleet, Yair Weiss publisher

[7] A Tutorial to understand DTW tutorial