Gesture Recognition using Scalable Short Term Pattern Analysis

Skanda Suresh, Trilok Kumar Tourani, Samyuktha Sridhar,

Preethi Raman, Raghavendhar Thiruvoipadi, Nithish Moudhgalya

Abstract

Gesture recognition has become an infamous application of statistics and machine learning in recent years. Sensor readings from different parts of the body are collected whilst making a specific gesture and then several techniques are employed that range from simple sequence matching to machine learning based time series classification. In this project, we aim to perform gesture recognition by analysing the short term patterns. To do so, we split the time series readings in each sensor to a fixed size pattern or word. These patterns are then used to compute higher level features like Term Frequency and Inverse Document Frequency. Using these features, we calculate vector similarity using different distance metrics and report the 10 closest gestures to a given query gesture. Several alternative techniques were tried and results corresponding to the same are discussed. Different members of the team performed various tasks and employed different approaches and implementations for each task and pooled the results for analysis and discussions.

Keywords - gesture recognition, pattern analysis, windowing

1 Introduction

Gesture recognition is an active research field which tries to integrate the gestural channel in Human Computer Interaction. It has applications in virtual environment control, but also in sign language translation, robot remote control or musical creation. Recognition of human gestures comes within the more general framework of pattern recognition. In this framework, systems consist of two processes: the representation and the decision processes. The representation process converts the raw numerical data into a form adapted to the decision process which then classifies the data. Gesture recognition systems inherit this structure and have two more processes: the acquisition process, which converts the physical gesture to numerical data, and the interpretation process, which gives the meaning of the symbol series coming from the decision process. In device-based systems, the acquisition of gestures is made by a physical device that directly measures some characteristics of the gesture, generally the different joint bending angles.

In this work, we use a dataset constructed from a mixture of data seen in the task ChaLearn Looking at People wherein the gestures are recorded by using 20 different sensors positioned at 20 different spots in the human body. Each file has a multivariate time series recorded across multiple sensors. The objective is to use this multivariate time series data to retrieve 10 closest resembling gestures to a query gesture file. The details of the proposed solution can be seen in the following section.

1.1 Terminology

1.1.1 Term Frequency

Term Frequency (tf) measures the number of times a particular pattern or word occurs in a document as opposed to the total number of words in the document. In our case it calculates the ratio of frequency of occurrence of a word to the total number of words in each sensor in each file. Equation 1 shows the tf formula used in the phase.

$$tf(w, f, s) = \frac{F(w, file\_id = f \ \& \ sensor\_id = s)}{(file\_id = f \ \& \ sensor\_id = s)} \tag{1}$$

1.1.2 Inverse Document Frequency

Inverse Document Frequency (idf) measures the number of documents a particular pattern or word occurs in as opposed to the total number of documents. In our case it calculates the log of ratio of total number of documents to number of docuements the given word occurs in each sensor. Equation 2 shows the idf formula used in the phase.

$$idf(w, s) = log\left(\frac{num\_total\_files}{N_{\forall f \in files}(w)}\right) \tag{2}$$

1.1.3 Inverse Document Frequency 2

' Inverse Document Frequency 2 (idf2) measures the number of sensors a particular pattern or word occurs in as opposed to the total number of sensors. In our case it calculates the log of ratio of total number of sensors to the number of sensors the given word occurs in each file. Equation 3 shows the idf2 formula used in the phase.

$$idf2(w, f) = log\left(\frac{num\_total\_sensors}{N_{\forall s \in sensors}(w)}\right) \tag{3}$$

where,
F - function that calculates frequency of word w in file f and sensor s
C - function that counts the total number of words in a file f and sensor s
N - function that counts the number of entities in which word w was found in
w - the word
f - variable for file if
s - variable for sensor id

1.1.4 Normlaization

Normalization is a process in which we compress/expand the range of values of a series into a fixed range (usually 0-1) so as to ensure the features are scaled equivalently. Normalization helps a lot in understanding visually how each feature contributes to the performance of machine learning models as they all have the same range of inputs. In our case we normalize the data in each sensor in each file into range [-1,1]. Equation 4 shows the formula used for normalization.

$$data = \frac{(data - data.min()) * (new\_max - new\_min)}{(data.max() - data.min())} + new\_min \tag{4}$$

1.1.5 Quantization

Quantization is the process of digitising or discretizing continuous values. In our case, we discretize the continuous time series floating point values into 2r discrete values using bands calculated using a Gaussian distribution. The bands are calculated using the formula below,

$$length_i = \frac{\sum_{x=(i-r)/r}^{(i-r-1)/r} G(x)}{\sum_{x=-1}^{1} G(x)} \tag{5}$$

where, $G(x)$ - is a function that computes Gaussian of x

1.1.6 Cosine Distance

Cosine distance is calculated as the cosine of the angle between two vectors. Based on the range of values it takes, we can say two vectors are similar when they have a high cosine value of 1 and dissimilar when values are lower (0 or -1). Equation 6 shows the formula used to compute cosine similarity between vectors A and B.

$$cosine(A, B) = \frac{A.B}{\|A\| \, \|B\|} \tag{6}$$

1.1.7 Euclidean distance

Euclidean distance measures the straight line distance between 2 vectors. It is a special form of L_p norms where p=2. Equation 7 shows the formula used to calculate euclidean distance between two vectors A and B.

$$euclidean(A, B) = \sqrt{(A - B).(A - B)} \tag{7}$$

1.2 Goal Description

The goal of this project is to use the statistical measures of short-term patterns to estimate the 10 similar gestures from the database to a query gesture. The goal of the entire project can be divided into 4 smaller milestones for each task as follows,

1. Task1 - normalize, quantize and window the time series into small patterns/words
2. Task2 - calculate tf, idf and idf2 for each word in each sensor in each file
3. Task3 - plotting heat maps for each file for a particular feature of choice
4. Task4 - finding 10 most similar files to a query file using any feature of choice using different similarity measures

1.3 Assumptions

While going through each task me make the following assumptions,

- A word occurring in a particular sensor is treated differently from the same pattern in another sensor. This is done because the sensors measure motions on different parts of body and cannot by unified and seen as a single pattern.
- When creating plots for task3, we don't constrain the range of the heatmap to be constant giving a better view of each file and the distribution of each word in them
- When calculating similar files in task4, we calculate the distances across all sensors as a unique dimension rather than averaging across the dimensions. This is done to ensure that each sensor word dimension can be treated separately and pattern matching would require every such dimension to match.

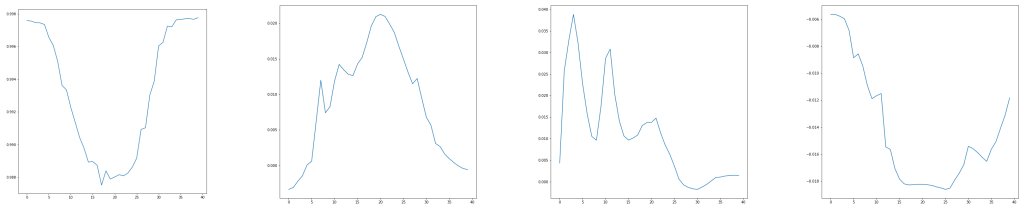
2 Proposed solution

In this project we approach the gesture recognition task by statistically analysing the occurrences of short term patterns in the data across multiple sensors. The dataset consists of 60 CSV files each with 20 rows representing each sensor reading taken for different gestures. Each file represents a gesture and the readings of each sensor follows a univariate time series. The solution can be split into 4 separate tasks for better understanding.

2.1 Task1

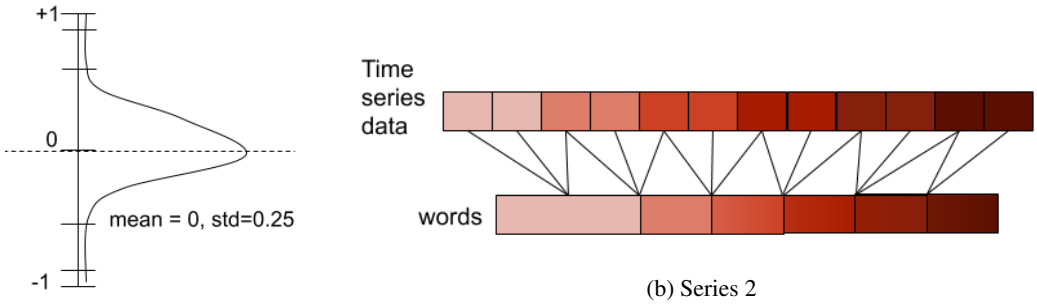
In this task, we try to convert the univariate time series data in each csv file into a word-based short term patterns. An example of an univariate time series from one of the gestures is shown in Figure 1. First, we normalize the values in each sensor between the range [-1,1]. Then we quantize the data series using a Gaussian-based band segmentation technique. The diagram for a Gaussian segmentation is shown in Figure 2. Following the quantization process, we use a sliding-window

technique to capture short term patterns in each univariate series as a combination of their quantized values. An example of such short term patterns is shown in Figure **??**.



(a) Series 1 (b) Series 2 (c) Series 3 (d) Series 4

Figure 1: Sample time series figures



(a) Series 1 (b) Series 2

Figure 2: Gaussian Bands

2.2 Task2

In this task, we calculate statistical features using the words/patterns we created in the previous task. We calculate Term Frequency(TF), Inverse Document Frequency(IDF) and a modified version of IDF that we call as IDF2 in this work. These values are calculated using equations 1, 2 and 3 respectively.

2.3 Task3

In this task we generate some heatmap plots for visualizing the tf, idf and idf2 values we computed in the previous task. To do so, we first replace the original univariate values with the the sequences of words and their tf, idf or idf2 values based on user preference. We then generate heatmaps for all sensors in a single gesture thus plotting a 2d matrix. The heatmap shows the distribution of these statistical features as a pattern along the gesture sequence. The heatmaps for the test gesture files are shown below. Figure 3 shows the tf values, Figure 4 shows the tf*idf values and finally Figure 5 shows the tf*idf2 values. In the calculation, we consider each word occurring in each sensor differently as the motions causing the patterns in each sensor are recorded from different parts of the body. This increases the total vocabulary size but gives a better distinction of the patterns and their occurrences across the dataset and gestures.

2.4 Task4

In this final task, we use vectors to represent a file using the features extracted in task2 and use several vector similarity measures to compare and find 10 closest resembling gestures from the dataset. We use cosine similarity usually to calculate the angle between any 2 vectors and hence here it gives us a good picture of how the features representing one file is aligned with the vector of another file. We can employ other similarity measures like euclidean distance to calculate similarity. To construct the vector to represent each file, we create a vector of size |w| where w is the total number for unique

(a) File 61 (b) File 62 (c) File 63

(d) File 64 (e) File 65 (f) File 66

Figure 3: TF values

(a) File 61 (b) File 62 (c) File 63

(d) File 64 (e) File 65 (f) File 66

Figure 4: TF-IDF values

(a) File 61 (b) File 62 (c) File 63

(d) File 64 (e) File 65 (f) File 66

Figure 5: TF-IDF2 values

words found in all gestures across all sensors. We then using a unique indexing technique to allot a single a fixed index for each word and replace that dimension for that file with the tf, tf*idf or tf*idf2

value of that word. By doing so we construct a uniform vector that can be compared with each other in the same vector space. We then employ the similarity measures and find 10 closest files/gestures to a given query file. The results for 6 test file using cosine similarity is shown in Tables 1-6 and using euclidean distance is shown in Tables 7-12. Do note, the test files are renamed into 61-66 for ease of logic in these tables.

Table 1: Cosine distances from File 61

File No	TF		TF-IDF		TF-IDF2	
	Closest Files	Distance Value	Closest Files	Distance Value	Closest Files	Distance Value
61	46	0.65	17	0.94	51	0.94
	30	0.62	27	0.94	30	0.94
	51	0.6	53	0.94	57	0.93
	27	0.59	30	0.93	45	0.92
	55	0.59	48	0.93	53	0.92
	45	0.58	57	0.92	7	0.92
	48	0.55	56	0.92	5	0.91
	58	0.54	55	0.92	27	0.91
	43	0.54	24	0.92	17	0.9
	53	0.53	50	0.92	37	0.9

Table 2: Cosine distances from File 62

File No	TF		TF-IDF		TF-IDF2	
	Closest Files	Distance Value	Closest Files	Distance Value	Closest Files	Distance Value
62	46	0.66	48	0.96	50	0.93
	55	0.63	50	0.96	48	0.93
	50	0.62	56	0.95	46	0.91
	29	0.62	46	0.94	56	0.9
	48	0.61	55	0.94	22	0.9
	26	0.61	17	0.93	49	0.9
	28	0.6	22	0.93	32	0.9
	59	0.6	25	0.93	25	0.9
	27	0.6	29	0.92	27	0.89
	58	0.59	24	0.92	55	0.89

Table 3: Cosine distances from File 63

File No	TF		TF-IDF		TF-IDF2	
	Closest Files	Distance Value	Closest Files	Distance Value	Closest Files	Distance Value
63	45	0.55	53	0.94	7	0.93
	46	0.44	27	0.94	41	0.93
	27	0.43	45	0.94	48	0.9
	57	0.41	50	0.93	5	0.9
	20	0.41	20	0.93	37	0.89
	51	0.4	17	0.93	53	0.89
	30	0.4	23	0.92	30	0.88
	55	0.4	48	0.92	27	0.87
	15	0.4	22	0.92	4	0.87
	17	0.38	57	0.91	45	0.87

Table 4: Cosine distances from File 64

File No	TF		TF-IDF		TF-IDF2	
	Closest Files	Distance Value	Closest Files	Distance Value	Closest Files	Distance Value
64	45	0.56	27	0.96	53	0.91
	46	0.45	45	0.96	7	0.91
	27	0.43	53	0.96	45	0.9
	57	0.42	17	0.94	27	0.9
	30	0.41	50	0.93	4	0.88
	51	0.41	48	0.93	35	0.88
	15	0.41	30	0.92	5	0.88
	20	0.39	28	0.92	13	0.88
	55	0.39	57	0.91	30	0.87
	59	0.39	58	0.91	51	0.87

Table 5: Cosine distances from File 65

File No	TF		TF-IDF		TF-IDF2	
	Closest Files	**Distance Value**	**Closest Files**	**Distance Value**	**Closest Files**	**Distance Value**
65	45	0.55	27	0.95	7	0.93
	46	0.45	30	0.93	41	0.92
	27	0.44	56	0.93	5	0.9
	57	0.42	24	0.93	34	0.9
	51	0.42	57	0.93	30	0.9
	30	0.42	48	0.92	13	0.9
	55	0.41	15	0.92	4	0.89
	15	0.41	6	0.92	51	0.89
	20	0.4	46	0.92	37	0.88
	59	0.4	53	0.92	27	0.88

Table 6: Cosine distances from File 66

File No	TF		TF-IDF		TF-IDF2	
	Closest Files	**Distance Value**	**Closest Files**	**Distance Value**	**Closest Files**	**Distance Value**
66	45	0.55	27	0.95	7	0.94
	46	0.45	53	0.94	53	0.93
	27	0.44	45	0.93	30	0.93
	57	0.41	48	0.93	51	0.92
	51	0.41	17	0.92	45	0.92
	30	0.41	30	0.92	41	0.92
	15	0.41	55	0.92	27	0.91
	20	0.4	50	0.91	48	0.91
	55	0.4	23	0.91	17	0.9
	59	0.39	56	0.91	5	0.9

3 Interface specifications

The project is divided into 4 independent modules namely - task1, task2, task3 and task4. The interface of each task are as follows.

1. Task1
   - Input - The directory that contains the csv files along with hyper parameters - window_size(w), number_of_bands(r) and stride/shift (s).
   - Output - word files that contain list of entities each of structure (idx,win) where idx = (file_id, sensor_id, start_time) and win is the window of quantized values stored in the same directory as the input
2. Task2
   - Input - The directory that contains the wrd files created in previous task.
   - Output - text file that contains sequence of 3 vectors for each sensor in each file carrying the tf, tfidf, tfidf2 values
3. Task3
   - Input - The file and the feature (tf, tfidf, tfidf2) to plot as heatmap.
   - Output - a heatmap of the vectors formed by each sensor in that file for that feature
4. Task4
   - Input - the file id for query and the feature to use for similarity.
   - Output - 10 most closest files to the query file in the database of 60 files

4 System requirements

The proposed system requires a functional python 3 environment and a few popular libraries mentioned below. The programs take user inputs for tasks 1, 3 and 4 to run.

1. numpy
2. scipy

3. pandas

4. matplotlib

5. seaborn

To run the codes, we need to first place every python file in the parent directory of the data folder and then we can just call $python\ < task\_name > .py$

5 Related work

Various exemplar-based methods are therefore proposed to circumvent the difficulties of model learning, by leveraging invariant visual representations and direct matching of example gestures. Among those visual representations, local spatio-temporal features [1–3] are the most widely exploited, though most of them are used in human action recognition. Other descriptors include motion trajectories [4], spatio-temporal gradients [5] and global histograms of optical flow [6]. However, most of these methods try to directly match the exemplars, without offering a scalable solution for efficient matching when the exemplar database is large. A few others have adopted the bag-of features framework with local spatio-temporal features for human action recognition. Meanwhile, the extraction of these features is generally slow, though some efforts toward real-time extraction and recognition are being made. Therefore, an efficient visual representation for real-time feature extraction and scalable gesture matching over large exemplar databases is still highly desirable.

6 Results and conclusion

We have already seen standard results for tasks 3 and 4. In Tables 1-6 we can see the set of 10 closest files to each test file calculated using cosine distance. It can be observed that when we use different features we get different sets of similar files. This can be attributed to the fact that, each of these features measure a different characteristic of the patterns, tf measures the frequency of the pattern in a sensor within the file, idf measures the frequency of pattern across the dataset, in other words it describes the distribution of each pattern across the entire dataset and lastly, idf2 measures the distribution of the pattern within the gesture among other sensors to catch similarities amongst them. Hence, each of these feature gives a different set of similar files/gestures. However euclidean distance captures the straight line distance between two vectors, and so it is much easier to see how it can calculate the similarity between files using different features in a similar manner. This can be seen from Tables 7-12.

Table 7: Euclidean distances from File 61

File No	TF		TF-IDF		TF-IDF2	
	Closest Files	Distance Value	Closest Files	Distance Value	Closest Files	Distance Value
61	9	1.21	9	1.08	9	0.7
	65	1.56	16	1.49	64	0.96
	42	1.56	47	1.53	65	0.96
	40	1.6	3	1.54	63	0.96
	66	1.62	13	1.55	16	1.03
	63	1.62	44	1.56	66	1.04
	36	1.62	63	1.57	35	1.07
	64	1.64	7	1.57	40	1.09
	4	1.65	34	1.57	47	1.09
	1	1.66	40	1.58	34	1.13

Table 8: Euclidean distances from File 62

File No	TF		TF-IDF		TF-IDF2	
	Closest Files	Distance Value	Closest Files	Distance Value	Closest Files	Distance Value
62	33	1.81	7	1.7	41	1.54
	42	1.82	41	1.71	5	1.55
	51	1.85	16	1.71	37	1.55
	65	1.85	47	1.72	35	1.56
	37	1.86	5	1.75	18	1.59
	61	1.87	37	1.76	7	1.59
	66	1.89	66	1.76	43	1.63
	18	1.91	13	1.76	10	1.66
	11	1.91	63	1.77	21	1.67
	63	1.91	3	1.77	39	1.67

Table 9: Euclidean distances from File 63

File No	TF		TF-IDF		TF-IDF2	
	Closest Files	Distance Value	Closest Files	Distance Value	Closest Files	Distance Value
63	66	0.62	66	1.08	65	0.0
	64	0.77	16	1.19	64	0.0
	65	0.77	13	1.21	66	0.52
	44	1.33	7	1.25	35	0.6
	36	1.46	3	1.27	16	0.66
	38	1.48	47	1.28	13	0.79
	13	1.49	40	1.3	47	0.82
	11	1.5	64	1.3	40	0.84
	40	1.52	44	1.32	44	0.85
	1	1.55	34	1.33	18	0.87

Table 10: Euclidean distances from File 64

File No	TF		TF-IDF		TF-IDF2	
	Closest Files	Distance Value	Closest Files	Distance Value	Closest Files	Distance Value
64	66	0.62	66	1.02	65	0.0
	63	0.77	16	1.19	63	0.0
	65	0.77	13	1.21	66	0.52
	44	1.3	7	1.24	35	0.6
	36	1.46	47	1.25	16	0.66
	38	1.46	63	1.3	13	0.79
	13	1.49	44	1.31	47	0.82
	11	1.51	3	1.32	40	0.84
	40	1.54	5	1.32	44	0.85
	1	1.55	41	1.32	18	0.87

Table 11: Euclidean distances from File 65

File No	TF		TF-IDF		TF-IDF2	
	Closest Files	Distance Value	Closest Files	Distance Value	Closest Files	Distance Value
65	66	0.65	66	1.15	63	0.0
	63	0.77	16	1.33	64	0.0
	64	0.77	13	1.35	66	0.52
	44	1.26	63	1.37	35	0.6
	36	1.39	64	1.37	16	0.66
	11	1.41	7	1.4	13	0.79
	38	1.43	47	1.41	47	0.82
	1	1.44	44	1.43	40	0.84
	39	1.48	5	1.43	44	0.85
	40	1.48	3	1.44	18	0.87

Table 12: Euclidean distances from File 66

File No	TF		TF-IDF		TF-IDF2	
	Closest Files	**Distance Value**	**Closest Files**	**Distance Value**	**Closest Files**	**Distance Value**
66	63	0.62	64	1.02	65	0.52
	64	0.62	63	1.08	63	0.52
	65	0.65	65	1.15	64	0.52
	44	1.29	16	1.19	35	0.72
	36	1.44	13	1.21	16	0.75
	38	1.45	7	1.25	13	0.87
	11	1.47	47	1.27	47	0.89
	13	1.49	44	1.3	40	0.9
	1	1.51	10	1.31	44	0.92
	40	1.52	3	1.32	10	0.94

7 References

[1] Hasanuzzaman, M., Ampornaramveth, V., Zhang, T., Bhuiyan, M.A., Shirai, Y. and Ueno, H., 2004, August. Real-time vision-based gesture recognition for human robot interaction. In 2004 IEEE International Conference on Robotics and Biomimetics (pp. 413-418). IEEE.

[2] Sombandith, V., Walairacht, A. and Walairacht, S., 2017, June. Hand gesture recognition for Lao alphabet sign language using HOG and correlation. In 2017 14th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON) (pp. 649-651). IEEE.

[3] Murthy, G.R.S. and Jadon, R.S., 2010, February. Hand gesture recognition using neural networks. In 2010 IEEE 2nd International Advance Computing Conference (IACC) (pp. 134-138). IEEE.

[4] Sánchez-Nielsen, E., Antón-Canalís, L. and Hernández-Tejera, M., 2004. Hand gesture recognition for human-machine interaction.

[5] Chen, F.S., Fu, C.M. and Huang, C.L., 2003. Hand gesture recognition using a real-time tracking method and hidden Markov models. Image and vision computing, 21(8), pp.745-758.

[6] New, J.R., Hasanbelliu, E. and Aguilar, M., 2003, March. Facilitating user interaction with complex systems via hand gesture recognition. In Proceedings of the 2003 Southeastern ACM Conference, Savannah, GA.

8 Appendix

In this section, we briefly describe the individual contributions of each member of our group. Every member had a very unique and distinct outlook of the tasks. Intense debates and sessions of discussion lasting over hours resulted in few consensus over the exact procedure the team members would follow for the tasks. Every member had diverse and countering points towards the understanding, implications and meanings of the task description. As this phase was an individual project but spread through group there isn't much to say about who performed which tasks but we certainly can rate each member on a scale of 0-100 based on their contributions to the discussions.

1. Skanda Suresh - 100
2. Trilok Kumar Tourani - 100
3. Preethi Raman - 100
4. Samyuktha Sridhar - 100
5. Raghavendhar Thiruvoipadi - 100
6. Nithish Moudhgalya - 100