4I2-2

Labeling Method for Acceleration Data Using Multi-Layer *k*-means++ Approach

k-means++の階層化による加速度データのラベリング手法

Shoto Isano 伊佐野 勝人　Nobuyuki Osawa 大澤 伸行

Mitsubishi Electric Information Systems Corporation
三菱電機インフォメーションシステムズ株式会社

Most of IoT (Internet of Things) data is not labeled and cannot be used as training data for Deep Learning. We propose a semi-automatic labeling method for acceleration data using multi-layer *k*-means++ approach. From the evaluation results, the proposed method gave a labeling accuracy of 88.6% on average. Also we used labeled data that is generated by the proposed method as training data and generated a model with Deep Learning which parameters were optimized with Genetic Algorithm. And the models gave a recognition accuracy of 98.9% on average.

1. Introduction

In recent years, the ecosystems of IoT and Deep Learning have been rapidly developed. Various business tasks are automated more efficiently when IoT data analytics is tied together with Deep Learning technology, but there are few actual cases. One of the major reasons is that most of IoT data is not labeled and cannot be used as training data for Deep Learning. And it requires a huge cost to create labeled data manually.

In this paper, we propose a semi-automatic labeling method for acceleration data. Acceleration data is generated by acceleration sensors which are mounted on various devices and they are applicable over commercial and industrial area. As shown in **Figure 1**, the proposed method automatically performs clustering using multi-layer *k*-means++ approach. The fragmented clusters are gradually integrated and continuous clusters are finally extracted as numerical labels. The numerical labels are correlated with the activity labels manually. Observers do this task by checking the visualized acceleration data from their memory or record.

2. Related Work

An automatic labeling method for acceleration data have been proposed by performing segmentation, clustering and labeling in this order [Murao 14]. However, this method has the following problems in each process. In the segmentation process, Spectral Transition Measure finds change points of activity from acceleration data and segments it. But this method is easy to make mistakes in short activities, as a result, the labeling and recognition accuracies are lowered. In the clustering process, COSINE [Peat 97] integrates clusters ignored minorities into fewer clusters. But there are some activities that minority clusters are the majority of clusters of an activity. In that case, there is a possibility that the labeling accuracy is lowered. In the labeling process, Dynamic Programming classifies the integrated clusters into seven activities. But the labeling accuracy is lowered in case of an activity sequence recorded by a subject is incorrect.

In our proposed method, segmentation and labeling are naturally realized by performing clustering with multi-layer *k*-means++. And it also solves the above problems.

3. Proposed Method

The proposed method consists of three processes, clustering of data sequences, clustering of probability distribution of clusters, and extraction of labels. In the clustering of data sequences process, sessions are created by cutting acceleration data in a fixed width sliding fixed frames apart, and they are clustered with *k*-means++. By this process, we can grasp momentary movements as clusters. In the clustering of probability distribution of clusters process, the cluster sequence created in the previous process is used as new input data, and new sessions are created in the same process. In the next step, the probability distribution of clusters for each session is calculated to create new sessions, and they are clustered with *k*-means++ in the same process. By this process, we can grasp the components of momentary movements as clusters. And by repeating this process, below the second clustering layers are created and we can grasp the components of larger movements as clusters. In the extraction of labels process, sets of continuous clusters and their sessions are extracted. Before and after continuous clusters are cut in any width because they contain noise. The clusters become numerical labels as it is. In this chapter, we explain each process in detail.

3.1 Clustering of Data Sequences

Input data is defined as l numbers of a numerical sequence $\boldsymbol{x}$. A numerical sequence $\boldsymbol{x}$ is cut in width n sliding u frames apart to create sessions $\boldsymbol{Y}$ that are defined as eq. (1), where $\boldsymbol{y}_i = [y_{i1} \ \cdots \ y_{in}]$, $m = (u + l - n) / u - (u + l - n) \% u$, $1 \leq i \leq m \leq l$, $1 \leq j \leq n \leq l$.

$$\boldsymbol{Y} = \begin{bmatrix} \boldsymbol{y}_1 \\ \vdots \\ \boldsymbol{y}_i \\ \vdots \\ \boldsymbol{y}_m \end{bmatrix} = \begin{bmatrix} y_{11} & \cdots & y_{1j} & \cdots & y_{1n} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ y_{i1} & \cdots & y_{ij} & \cdots & y_{in} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ y_{m1} & \cdots & y_{mj} & \cdots & y_{mn} \end{bmatrix} \tag{1}$$

When sessions $\boldsymbol{Z}$ equal sessions $\boldsymbol{Y}$, sessions $\boldsymbol{Z}$ are defined as eq. (2).

Contact: Kamimachiya 325, Kamakura City, Kanagawa Pref., 0467-41-3876, isano-shouto@mdis.co.jp

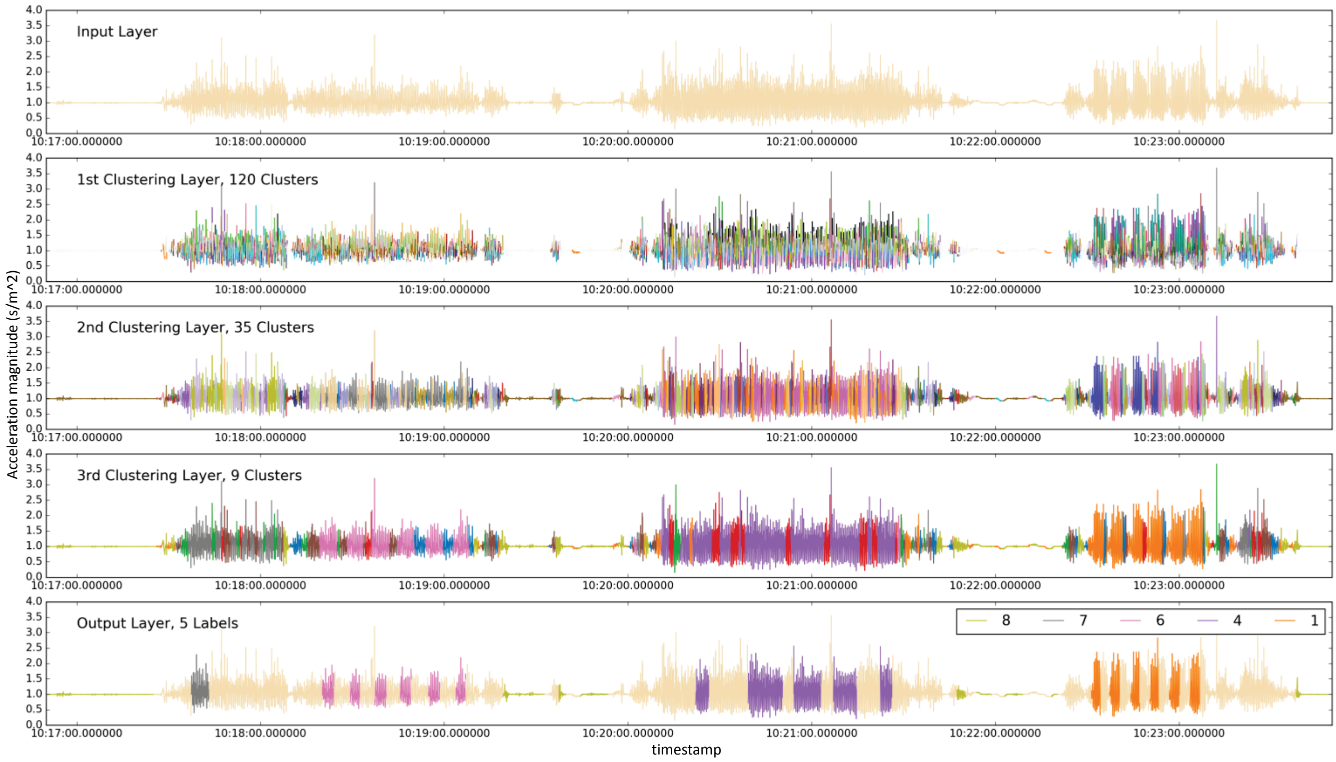

Figure 1 Semi-automatic labeling method for acceleration data using multi-layer *k*-means++.

$$\boldsymbol{Z} = \begin{bmatrix} \boldsymbol{z}_1 \\ \vdots \\ \boldsymbol{z}_i \\ \vdots \\ \boldsymbol{z}_m \end{bmatrix} = \begin{bmatrix} z_{11} & \cdots & z_{1j} & \cdots & z_{1n} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ z_{i1} & \cdots & z_{ij} & \cdots & z_{in} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ z_{m1} & \cdots & z_{mj} & \cdots & z_{mn} \end{bmatrix} \tag{2}$$

In the next place, sessions $\boldsymbol{Z}$ are clustered with *k*-means++ to find sets $\boldsymbol{S}$ that satisfy eq. (3), where $1 \le p \le k$, $\boldsymbol{S} = \{\boldsymbol{s}_1, \cdots, \boldsymbol{s}_k\}$, $\boldsymbol{\mu}_p$ is the mean of points in $\boldsymbol{S}_p$. And m numbers of a numerical sequence $\boldsymbol{p} = [p_1 \quad \cdots \quad p_m]$ is created.

$$\underset{\boldsymbol{S}}{\operatorname{argmin}} \sum_{p=1}^{k} \sum_{z \in s_p} \left\| \boldsymbol{Z} - \boldsymbol{\mu}_p \right\|^2 \tag{3}$$

A numerical sequence $\boldsymbol{p}$ is the clusters of sessions $\boldsymbol{Z}$.

3.2 Clustering of Probability Distribution of Clusters

A numerical sequence $\boldsymbol{p}$ is replaced with $\boldsymbol{x}$ and its length m with l, and eq. (1) is created in the same process 3.1. Where $i = q$, $1 \le q \le m$, probability $z_{qj} = P(p_q = y_{qj})$, probability distribution is calculated for each row and eq. (2) is created. A numerical sequence $\boldsymbol{p}$ is also calculated in the same process 3.1.

3.3 Extraction of Labels

For continuous clusters in a numerical sequence $\boldsymbol{p}$, each length *f1* and *f2* is cut from start and end of them, if the remaining number of clusters is more than length *f*, they are extracted as numerical labels.

4. Evaluation

In this chapter, we evaluate effectiveness of the proposed method.

4.1 Data

For the experiment, we develop an iOS application that acquires acceleration data at 20 Hz sampling frequencies from a 3-axis acceleration sensor of iPhone 7. We prepare a iPhone 7 installed this application. Eight subjects (P1 - P8), 4 male and 4 female between the ages of 25 and 55, carry it in their right pocket. Their clothes, shoes and the weather during data acquisition differ depending on the subjects. There are eight activities, Sitting, Standing, Going Up Elevator, Going Down Elevator, Walking Indoor, Walking Outdoor, Going Up Stairs and Going Down Stairs. In order to acquire acceleration data from their natural movements, we ask the subjects to move the same route in the following activity order. But there is not Standing if the elevator arrives quickly.

Sitting → Walking Indoor → Going Up Stairs → Walking Indoor → Standing → Walking Indoor → Going Down Elevator → Walking Indoor → Walking Outdoor → Walking Indoor → Standing → Walking Indoor → Going Up Elevator → Walking Indoor → Going Down Stairs → Walking Indoor → Sitting

The 3-dimensional acceleration data is converted into the 1-dimensional acceleration magnitude by calculating a Euclidean distance per timestamp. However, in pre-experiments, the proposed method is difficult to recognize the four activities, Sitting, Standing, Going Up Elevator and Going Down Elevator with 1-dimensional acceleration magnitude. Therefore, these four activities are all categorized into Stationary. From the above, we evaluate the five activities, Stationary, Walking Indoor, Walking Outdoor, Going Up Stairs and Going Down Stairs.

Table 1 Deep learning parameters and values.

Parameter	Values
Number of Hidden Layer	1, 2, 3
Number of Batch	5, 10, 20, 30, 50, 100
Number of Epoch	1, 10, 50, 100
Number of Hidden Layer Node	10, 30, 50, 70, 100
Weight	zeros, ones, random, truncated
Weight Standard Deviation	0.1, 0.01, 0.001, 0.0001
Bias	zeros, ones
Activation Function	none, relu, tanh, softmax
Training Optimization Function	Adam, Gradient Descent
Training Rate	0.1, 0.01, 0.001, 0.0001

4.2 Experiments

(1) Labeling

We perform semi-automatic labeling and manual labeling. And we calculate semi-automatic labeling accuracy.

・Semi-Automatic Labeling

The input data is the sessions that are created by cutting acceleration data in 20 frames (1 second) width sliding 1 frame apart. The number of significant digits is one digit after the decimal point. Clustering with *k*-means++ is performed for three layers, and the number of clusters for each layer is reduced to 120, 35, 9-12 as it goes to a lower layer. The reason why the number of clusters in the third layer has the range of values is to adjust the subjects' characteristics and the environment during data acquisition. This is the only parameter configured manually for each subject. In the output layer, clusters continued for 20*4 frames (4 seconds) or more are extracted as numerical labels. Therefore, the number of labels is smaller for semi-automatic than for manual. The labeling accuracy is calculated by eq. (4). *p* is an activation label.

$$Labeling\ Accuracy(p) = \frac{n(Semi\ Automatic\ Label\ p\ \cap Manual\ Label\ p)}{n(Semi\ Automatic\ Label\ p)} \quad (4)$$

・Manual Labeling

An observer checks the visualized acceleration data, segments the activities, and labels them.

(2) Recognition

We use the labeled acceleration data as training data, train a Feedforward Neural Network with Deep Learning, and calculate recognition accuracy for each subject.

・Training and Test Data

Test data of 10 sessions is selected at random from each activity label. The range of test data is where both semi-automatic and manual labels exist. Training data is other than test data. For semi-automatic labels, activities that the number of sessions is less than 100 are excluded from the evaluation.

・Deep Learning

The number of nodes in the input layer is 20, which is the session width. And the number of nodes in the output layer is 4 or 5, which is the number of activity labels. Loss function is the cross-entropy between the training labels and the softmax

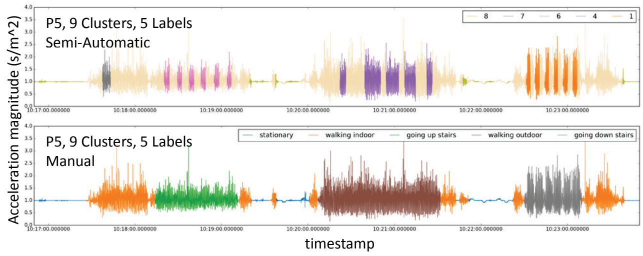

Figure 2 Semi-automatic and manual labeling result of P5.

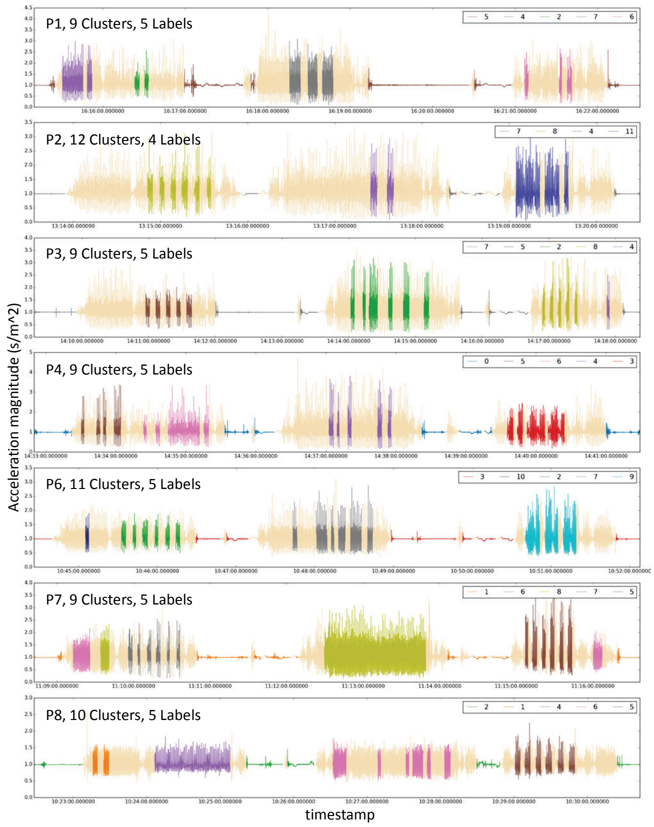

Figure 3 Semi-automatic labeling result of P1 - P4, P6 - sP8.

activation function applied to the model predicted labels. The model is trained to minimize the loss for each batch. The other parameters are selected from the values in **Table 1**. Also, when the model predicted labels are a matrix $\boldsymbol{A}$ and the test labels are a matrix $\boldsymbol{B}$, they satisfy eq. (1). For each of the matrix $\boldsymbol{A}$ and $\boldsymbol{B}$, vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ are calculated by finding the column j having the maximum value for each row i. For the two m-dimensional vectors $\boldsymbol{a}$ and $\boldsymbol{b}$, an m-dimensional vector $\boldsymbol{c}$ is created with 1 for the matching dimensions and 0 for the other dimensions. At this time, the recognition accuracy is calculated by eq. (5).

$$Recognition\ Accuracy(\boldsymbol{c}) = \frac{1}{m}\sqrt{\sum_{i=1}^{m} \boldsymbol{c}_i^2} \quad (5)$$

・Genetic Algorithm

Deep learning parameters are optimized with Genetic Algorithm. For **Table 1**, values of Weight Standard Deviation are not selected in case of Weight is zeros or ones. And a value of Activation Function in output layer is only none. Population size is 40, the number of generation is 25 and elitism rate is 0.25. Except for the first generation, Genetic Algorithm executes two-point crossover or mutation with 50% until population size is reached to the expected value for each generation. The optimization means maximizing test accuracy, in the second place, minimizing training time.

4.3 Results and Remarks

(1) Labeling

Figure 2 shows the results of semi-automatic and manual labeling of P5. Both methods classify activities into five and semi-automatic labels are almost within the range of manual labels. When both labels are correlated, 8 is Stationary, 7 is Walking Indoor, 6 is Going Up Stairs, 4 is Walking Outdoor, and 1 is Going Down Stairs. Since continuous clusters are extracted, we can confirm that there is no label where before and after the behavior change or parts the pace is disturbed. In Walking Indoor, the subjects walk a narrow hallway and need to be conscious of collision with people. Therefore, their pace is easy to be disturbed, and the number of the label is few. Also, in Going Up/Down Stairs, walking at stair landing is separated and not labeled. This is useful training data to recognize Going up/Down Stairs and Walking Indoor/Outdoor properly. Stationary has many noises immediately after its start. But, as shown in **Figure 1**, the clusters corresponding to Stationary are already continuous in the first layer, so it can be avoided by extracting it as labels from there.

Figure 3 shows the results of semi-automatic labeling of the other subjects. They show a similar tendency as P5. The number of clusters in the third layer is 12 for P2, 11 for P6, 10 for P8 and 9 for the others. And labels are classified into five activities as manual except for P2 whose Walking Indoor is not extracted. For P7, Walking Indoor/Outdoor are labeled in the range of Walking Indoor actually. The reason is that semi-automatic labeling classifies into Walking Slowly/Fast not Walking Indoor/Outdoor. This labeling accuracy lowers by eq. (4), but this data can be applied to training data.

Table 2 shows semi-automatic labeling accuracy. Stationary is low in common to all subjects and it's 44.3% on average. For the above reasons, Walking Indoor of P2 and P7 are NaN and 0.921 respectively. However, the others are perfect at 1.000. Walking Indoor of P2, P3 and P6 are excluded from the recognition evaluation because the numbers of those are 0, 48, 51 respectively.

(2) Recognition

Table 3 shows recognition accuracy using training data created by semi-automatic and manual labeling. The average accuracy is 98.9% for semi-automatic labels and 94.2% for manual labels. Since the features are automatically extracted with Deep Learning and its parameters are optimized with Genetic Algorithm, the recognition accuracy for each method is so high. The proposed method, semi-automatic, is a slightly higher accuracy than manual. The reasons are that noise is excluded from training data, and there are many subjects whose training data of Going Up/Down Stairs not including walking at stair landing. However, there are only 10-session test data in this experiment. So we need to prepare more test data and evaluate it again.

Table 2 Semi-automatic labeling accuracy.

	P1	P2	P3	P4	P5	P6	P7	P8	Ave.
Stationary	0.416	0.567	0.474	0.392	0.512	0.429	0.318	0.434	0.443
Walking Indoor	100.0	NaN	1.000	1.000	1.000	1.000	0.921	1.000	0.989
Walking Outdoor	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
Going Up Stairs	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
Going Down Stairs	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
Ave.	0.883	0.892	0.895	0.878	0.902	0.886	0.848	0.887	0.886

Table 3 Recognition accuracy.

	P1	P2	P3	P4	P5	P6	P7	P8	Ave.
Semi-Automatic	1.000	0.975	1.000	0.980	0.980	1.000	1.000	0.980	0.989
Manual	0.980	0.975	0.920	0.900	0.980	0.860	0.960	0.960	0.942

5. Conclusion

In this paper, we proposed semi-automatic labeling method for acceleration data using multi-layer *k*-means++ approach. From the evaluation results, the proposed method gave a labeling accuracy of 88.6% and a recognition accuracy of 98.9% on average. All we need to do manually is adjusting a value of a cluster for each subject and correlating numerical labels with activity labels. We confirmed that the proposed method creates useful training data for Deep Learning from unlabeled IoT data at very little cost.

In the future, we will evaluate data other than acceleration and confirm that the proposed method is a technology generally usable in IoT and Deep Learning.

Trademarks

iPhone is a trademark of Apple Inc. iOS is a trademark of Apple Inc.

References

[Murao 14] Murao, K., Torii, Y., Terada, T. and Tsukamoto, M.: Labeling method for acceleration data using an execution sequence of activities, Journal of Information Processing, Vol. 55, No. 1, pp.519-530 (2014).

[Peat 91] Peat, H. J. and Willett, P.: The limitations of term co-occurrence data for query expansion in document retrieval systems, Journal of the American Society for Information Science, Vol. 42, No. 5, pp.378-383 (1991).