Behavioral Classification Using Feature Selection in the Micro Activity Retrieval Task

Takuma Yoshimura
University of Hyogo
aa20w514@ai.u-hyogo.ac.jp

Pham HuuLong
University of Hyogo
aa19e510@ai.u-hyogo.ac.jp

Ryota Mibayashi
University of Hyogo
aa20r511@ai.u-hyogo.ac.jp

Rui Kimura
KDDI Research
ui-kimura@kddi-research.jp

Hiroaki Ohshima
University of Hyogo
ohshima@ai.u-hyogo.ac.jp

ABSTRACT

Human activity recognition has an important role in helping computers to understand human activity. Among them, micro activity recognition is required for computers to understand more detailed action. In this research, we proposed a micro activity retrieval task method to prevent over-learning for data sets with a small number of data and many feature dimensions. As the feature selection method, we use Super-LCC. Super-LCC is fast and has low information entropy loss. The proposed method reduces the feature size from 3108 dimensions to 20.75 dimensions on average. The result of the mAP evaluation of the proposed method was 0.71707. We succeeded in the task with a technique that requires fewer feature dimensions to be entered.

TEAM NAME

UHAIK

SUBTASKS

Retrieval task

KEYWORDS

human activity detection, micro-activities, feature selection, feature value selection

1 INTRODUCTION

There are many applications of activity recognition using sensor data in the field of human activity recognition. Human activity recognition include things like “walk, run, stand upright, and sit down,” and micro activity recognition like “search the Internet on a computer and edit PowerPoint on a computer.” Micro activity recognition is deserved research to recognize human activity in more detail.

However, there is a concern that if many features are used to recognize micro-activities, some of them may be redundant features. Therefore, we propose a method for recognizing micro activities while reducing feature dimensions and removing redundant features.

Section 2 describes the related works. Section 3 describes the method. Section 4 describes the experiment. Section 5 describes the result. Section 6 describes the conclusions.

2 RELATED WORK

In this section, we describe the related work to this research.

In recent years, research in human activity recognition has been popular using smartphone sensors [6–8] and wearable sensors, including smartwatches[4, 5]. There is research that performs activity recognition by data segmentation[9].

In addition, there are research on human micro activity recognition[11], research on human activity recognition using feature selection[3], and research on human activity recognition by dynamically determining the sensor to be used[1].

Super-CWC[10] is a feature selection algorithm. The algorithm selects the most important features among all of the features while making sure that information entropy loss is the lowest as possible. In this task, we use Super-CWC, or Super-LCC, which allows us to set parameters, as a method for feature selection. We noticed that it not only has a great performance in terms of accuracy, but is also extremely fast.

3 METHOD

In this section, we describe the method.

In this task, input data has 3108 features. The number of these features is large compared to the number of data (280). Too many dimensional data can cause not only slow speed in model training, but also can hurt the classifier model’s performance.

The proposed method takes the approach of performing discretization, raising the dimension of features, selecting features, and classifying them, as shown in Figure 1.

- Discretization
- Feature selection
- Classification

We also introduce an approach to selecting feature values when selecting features, as shown in Figure 2.

- Discretization + One−hot encoding
- Feature value selection
- Classification

While feature selection is selected based on which features are important, feature value selection is a method of selecting based on where the feature values are located.

Section 3.1 describes the discretization, section 3.2 describes feature selection and feature value selection, and section 3.3 describes the classification.

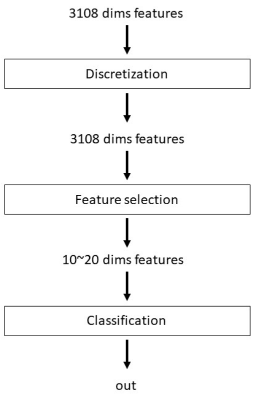

Figure 1: Overall diagram of feature selection

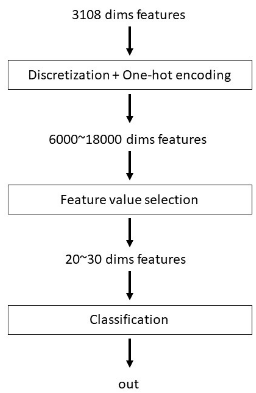

Figure 2: Overall diagram of feature value selection

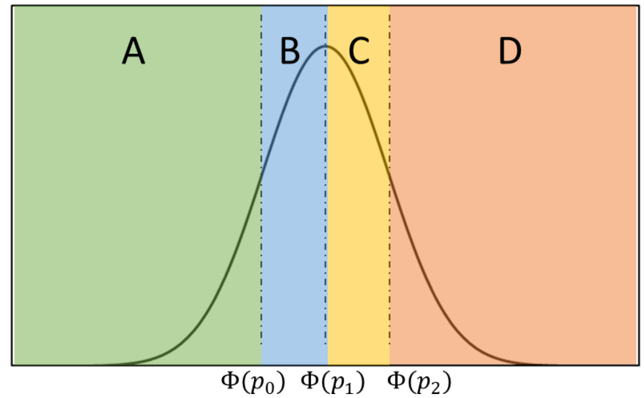

Figure 3: How to split the normal distribution, in the case of four partitions

3.1 Discretization and One–hot encoding

All 3108 features are continuous data, while the Super-LCC algorithm only accepts discrete data as input. To use Super-LCC properly, data transformation is required. Therefore, we transformed the continuous input features to the acceptable data type, in two ways. In both approaches, we first standardized all features of the input data.

For standardized, we use z-score as in the Equation 1. The Equation 1 transforms the value of the feature so that it has mean 0 variance 1. X is the value of the data, u is the mean value of X, σ is the standard deviation, and z is the value of the normalized data.

$$z = \frac{X - u}{\sigma} \tag{1}$$

The discrete values obtain in the Figure 2 are treated as 1 when the feature normalized is -0.2 with a partition number k of 4, which we will discuss later. And a normalized feature of 0.7 is a discrete value of 3. In addition, it is treated as $[0, 1, 0, 0]$ at -0.2 and $[0, 0, 0, 1]$ at 0.7 in the method in Figure 2. We describe how to discretize the feature values.

First, we describe how to divide a continuous value into k regions to convert it to a discrete value. For example, in the case of four partitions, we divide the normalized features into four regions with thresholds $\Phi(p_0)$, $\Phi(p_1)$, and $\Phi(p_2)$, as shown in the Figure 3. The threshold value $\Phi(p)$ in the Figure 3 can be obtained by the Equation 2. p is probability. $\Phi(p)$ is threshold value.

$$\Phi^{-1}(p) = \sqrt{2}\mathrm{erf}^{-1}(2p - 1), p \in (0, 1) \tag{2}$$

For example, when dividing into four parts, as shown in the Figure 3, the threshold should be set to the point where p is 0.25, 0.5, and 0.75 in the Equation 2. Thus, $\Phi(p_0)$ is -0.67, $\Phi(p_1)$ is 0, and $\Phi(p_2)$ is 0.67.

Second, we discuss how to convert continuous values to discrete values. The discrete values are given values depending on where the continuous values are located in the region. For example, if a continuous value exists in the region of A in the Figure 3, a discrete value of 0 is given. In addition, 1, 2, and 3 are given in regions B, C, and D, respectively.

Thirdly, we discuss how to discrete using the one-hot encoder after splitting the region. The one-hot encoder converts a continuous value into a vector that indicates where it lies in the region. For example, if a continuous value exists in the region of A in the Figure 3, a discrete value of $[1, 0, 0, 0]$ is given. In addition, $[0, 1, 0, 0]$, $[0, 0, 1, 0]$, and $[0, 0, 0, 1]$ are given in regions B, C, and D, respectively.

These discretized or one-hot encoded values are used to select the most appropriate feature using Super-LCC.

3.2 Feature selection and feature value selection

Super-LCC[10] was used for feature selection. It is extremely powerful and beats other advanced feature selection algorithms in both terms of accuracy and speed. We first used Super-LCC to find important features in terms of classifying, and then used only the selected features to train the classifier models. An improvement in the final model's performance with Super-LCC as feature selection algorithm was recorded.

First, we describe the method of feature selection. Feature selection is selected based on which features are important. Figure 4 shows feature selection. Feature selection is done by selecting features that are effective for classification from among the discretized features, as shown in the Figure 4.

Next, we describe the method of feature value selection. Feature value selection is a method of selecting based on where the feature values are located. The feature value selection is a method of selecting from $3108 \times k$ dimensions, or 3108×4 dimensions in Figure 5, instead of the 3108 dimensions of the candidate features for selection in feature selection by handling the values in this way. Figure 5 shows feature value selection flow. Feature value selection is done by selecting feature values that are effective for classification from among the discretized feature values, as shown in the Figure 5.

Table 1 shows the parameters of Super-LCC. The t in Table 1 is the value that determines how much information loss is acceptable.

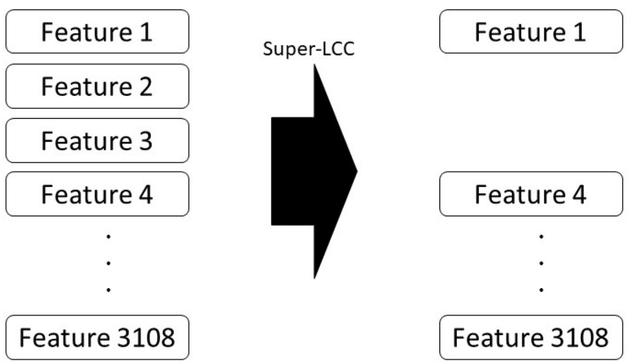

Figure 4: Feature selection flow

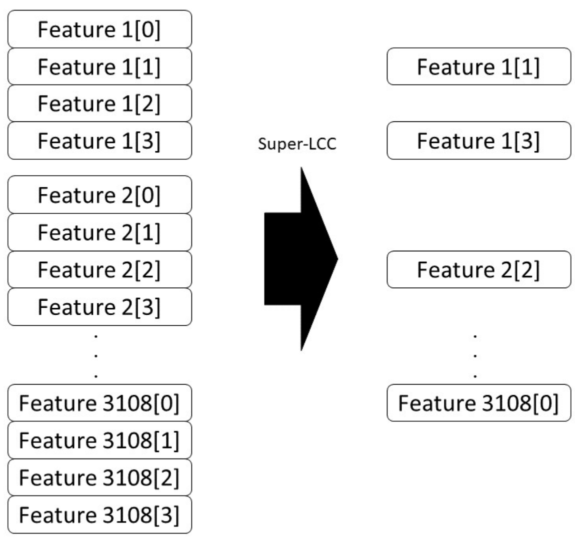

Figure 5: Feature value selection flow

Table 1: Super-LCC Parameters

slcc params	values
conversion	feature selection, feature value selection
k	2, 3, 4, 5, 6
t	0.8 to 1.0 in increments of 0.01

3.3 Classification

The classifier's output is then display in order of rank for each confidence level.

Due to the small number of data, we use four classifiers.

- SVM (Support Vector Machine)
- NB (Naïve Bayes)
- LR (Logistic Regression)
- kNN (k-Nearest Neighbor)

Table 2: Parameters of SVM, NB, LR and kNN

model	params	values
SVM	C	0.001, 0.01, 0.1, 1, 10, 100
	gamma	0.001, 0.01, 0.1, 1, 10, 100
	kernel	rbf, linear
NB		
LR	C	0.01, 0.1, 1, 10
kNN	n_neighbors	1, 3, 5, 7, 9, 11, 15, 21
	weights	uniform, distance
	p	1, 2

Table 3: Parameter tuning results from the 10-fold cross validation

model	accuracy	model params	slcc params
SVM	0.600	C=10	feature selection
		gamma=0.1	k = 6
		kernel=rbf	t=0.96
NB	0.479		feature value selection
			k=4
			t=0.97
LR	0.479	C=0.01	feature value selection
			k=4
			t=0.97
kNN	0.593	n_neighbors=5	feature selection
		weights=distance	k=6
		p=2	t=0.96

The range of possible values of these parameters is set as shown in Table 2.

4 EXPERIMENT

In this section, we describe the experimental method and parameter tuning.

In the grid search used for parameter tuning in this method, feature selection is performed by Super-LCC and the results of classification by the classifier are evaluated by accuracy. Therefore, the optimal parameters for Super-LCC vary depending on the classifier.

For parameter tuning, we used a 10-fold grid search. The parameters with the highest accuracy are shown in Table 3. Also, the parameters of Super-LCC shown in Table 1 were also tuned at the same time. The features selected for SVM, NB, LR and kNN are shown in Table 4 to Table 7.

Table 4 shows the features extracted when using SVM. From the Table 4, it can be seen that the features have decreased to 14 dimensions. In addition, according to the Table 4, features related to movement, such as hand and head acceleration and mouse movement, were selected. The Table 5 also shows the features extracted using NB. From the Table 5, it can be seen that the features have decreased to 28 dimensions. According to the Table 5 the RESNERT features are selected in addition to the features related to movement, such as hand and head acceleration and mouse movement. Table 6 shows the features extracted when using LR. From the Table 6,

Table 4: SVM selected features

selected features
data_RIGHT_ACC_MAG_std
data_RIGHT_ACC_Y_len
data_RIGHT_ACC_Y_std
data_LEFT_ACC_Y_len
data_RIGHT_ACC_Z_std
data_LEFT_ACC_MAG_std
data_RIGHT_ACC_X_std
data_MOUSE_VELOCITY_average
data_MOUSE_PIX_DIS
data_HEAD_Z_by_activity_stdTS_average
data_LEFT_ACC_X_std
data_MOUSE_PIX_DISTS_len
data_MOUSE_TIMEDIFFS_max
data_RIGHT_ACC_MAG_average

Table 5: NB selected feature values

selected features
data_RIGHT_ACC_Z_len[3]
data_MOUSE_TIMEDIFFS_average[2]
data_LEFT_ACC_Y_len[3]
data_MOUSE_TIMEDIFFS_std[2]
data_MOUSE_VELOCITY_max[2]
data_RIGHT_ACC_X_average[0]
data_MOUSE_VELOCITY_len[1]
data_RIGHT_ACC_Z_std[3]
data_RIGHT_ACC_Z_average[0]
data_HEAD_Z_by_activity_std[0]
data_LEFT_ACC_Y_std[0]
data_LEFT_ACC_MAG_std[3]
data_RIGHT_ACC_X_std[0]
data_LEFT_ACC_Z_std[0]
data_LEFT_ACC_Z_std[3]
data_RIGHT_ACC_MAG_average[1]
data_RIGHT_ACC_Y_std[1]
data_LEFT_ACC_X_std[0]
data_LEFT_ACC_X_average[0]
data_AUTOGRAPHER_RESNET_mean_envelope[1]
data_RIGHT_ACC_MAG_std[1]
data_AUTOGRAPHER_RESNET_mean_ashcan, trash can, garbage can, wastebin, ash bin, ash-bin, ashbin, dustbin, trash barrel, trash bin[1]
data_AUTOGRAPHER_RESNET_mean_cuirass[1]
data_MOUSE_VELOCITY_len[3]
data_LEFT_ACC_MAG_average[3]
data_HEAD_Z_by_activity_std[3]
data_AUTOGRAPHER_RESNET_max_envelope[1]
data_AUTOGRAPHER_RESNET_mean_totem pole[1]

Table 6: LR selected feature values

selected features
data_RIGHT_ACC_Z_len[3]
data_MOUSE_TIMEDIFFS_average[2]
data_LEFT_ACC_Y_len[3]
data_MOUSE_TIMEDIFFS_std[2]
data_MOUSE_VELOCITY_max[2]
data_RIGHT_ACC_X_average[0]
data_MOUSE_VELOCITY_len[1]
data_RIGHT_ACC_Z_std[3]
data_RIGHT_ACC_Z_average[0]
data_HEAD_Z_by_activity_std[0]
data_LEFT_ACC_Y_std[0]
data_LEFT_ACC_MAG_std[3]
data_RIGHT_ACC_X_std[0]
data_LEFT_ACC_Z_std[0]
data_LEFT_ACC_Z_std[3]
data_RIGHT_ACC_MAG_average[1]
data_RIGHT_ACC_Y_std[1]
data_LEFT_ACC_X_std[0]
data_LEFT_ACC_X_average[0]
data_AUTOGRAPHER_RESNET_mean_envelope[1]
data_RIGHT_ACC_MAG_std[1]
data_AUTOGRAPHER_RESNET_mean_ashcan, trash can, garbage can, wastebin, ash bin, ash-bin, ashbin, dustbin, trash barrel, trash bin[1]
data_AUTOGRAPHER_RESNET_mean_cuirass[1]
data_MOUSE_VELOCITY_len[3]
data_LEFT_ACC_MAG_average[3]
data_HEAD_Z_by_activity_std[3]
data_AUTOGRAPHER_RESNET_max_envelope[1]
data_AUTOGRAPHER_RESNET_mean_totem pole[1]

Table 7: kNN selected features

selected features
data_RIGHT_ACC_MAG_std
data_RIGHT_ACC_Y_len
data_RIGHT_ACC_Y_std
data_LEFT_ACC_Y_len
data_RIGHT_ACC_Z_std
data_LEFT_ACC_MAG_std
data_RIGHT_ACC_X_std
data_MOUSE_VELOCITY_average
data_MOUSE_PIX_DISTS_average
data_LEFT_ACC_X_std
data_MOUSE_TIMEDIFFS_max
data_RIGHT_ACC_MAG_average
data_LEFT_ACC_Y_std

it can be seen that the features have decreased to 28 dimensions. According to the Table 6, the RESNERT features are selected in addition to the features related to movement, such as hand and head acceleration and mouse movement. Table 7 shows the features extracted when using kNN. From the Table 7, it can be seen that the features have decreased to 13 dimensions. In addition, according to the Table 7, features related to movement, such as hand and head

Table 8: Result

model	mAP
SVM(feature selection)	0.71707
SVM(feature value selection)	0.57774
kNN(feature selection)	0.55080
kNN(feature value selection)	0.55501

acceleration and mouse movement, were selected. In addition, these tables show that RESNET features are not selected for feature selection, but motion features, including acceleration, are emphasized. The feature value selection shows that although RESNET features are selected for feature selection, they still focus on acceleration and other features.

From the Table 3, we can see that the results for SVM and kNN are good, we use SVM and kNN as classifiers.

5 RESULT

In this section, we describe the results of submitting the output of this method.

Table 8 shows the feature selection using the parameters in Table 3, and submits the output after classification by SVM and kNN. Since both SVM and kNN handled Super-LCC values as a feature selection, the results tried with feature value selection are also shown in the same way in Table 8.

From Table 8, we can confirm that the results of SVM (feature selection) and kNN (feature selection) are better for SVM (feature selection). In addition, when comparing feature selection and feature value selection, feature selection is better. These results are consistent with the results of 10-fold cross validation on train data.

According to Overview[2], the result for the first place group is 0.95037. Our result is 0.70707, but we are successful in reducing the amount of features we use as much as possible to identify micro activities.

6 CONCLUSIONS

We used feature selection to prevent over-learning and solve the problem as a classification problem for learning with a disproportionate number of features. The approach is based on the ranking of the confidence level of the features. As a feature selection method, we used Super-LCC as a fast and low information entropy loss method to eliminate redundant features. In the verification, we compared the feature selection and feature value selection with SVM and kNN in the value handling of Super-LCC, and found that the feature selection SVM was the best result. We succeeded in the task with a technique that requires fewer feature dimensions to be entered.

ACKNOWLEDGMENTS

This work was supported by JSPS KAKENHI Grant Numbers JP18H03243, JP18H03494, JP17H00762, JP16H01756.

REFERENCES

[1] Luis Gioanni, Christel Dartigues-Pallez, Stéphane Lavirotte, and Jean-Yves Tigli. 2016. Opportunistic Human Activity Recognition: A Study on Opportunity Dataset. In *Proceedings of the 13th International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services (MOBIQUITOUS 2016)*. 290–291.

[2] Healy Graham, Le Tu-Khiem, Joho Hideo, Hopfgartner Frank, and Gurrin Cathal. 2020. Overview of NTCIR-15 MART. In *Proceedings of the NTCIR-15 Conference*. 8–11.

[3] Oscar Herrera-Alcántara, Ari Barrera-Animas, Miguel Gonzalez-Mendoza, and Félix Castro Espinoza. 2019. Monitoring Student Activities with Smartwatches: On the Academic Performance Enhancement. *Sensors* 19, 1605 (2019), 18 pages.

[4] Tahera Hossain, Md Shafiqul Islam, Md Atiqur Rahman Ahad, and Sozo Inoue. 2019. Human Activity Recognition Using Earable Device. In *Adjunct Proceedings of the 2019 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2019 ACM International Symposium on Wearable Computers (UbiComp/ISWC 2019)*. 81–84.

[5] Min-Cheol Kwon and Sunwoong Choi. 2018. Recognition of Daily Human Activity Using an Artificial Neural Network and Smartwatch. *Wireless Communications and Mobile Computing* 2018 (2018), 1–9.

[6] Godwin Ogbuabor and Robert La. 2018. Human Activity Recognition for Healthcare Using Smartphones. In *Proceedings of the 2018 10th International Conference on Machine Learning and Computing (ICMLC 2018)*. 41–46.

[7] Kazushige Ouchi and Miwako Doi. 2012. Indoor-Outdoor Activity Recognition by a Smartphone. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing (UbiComp 2012)*. 600–601.

[8] José Santoyo Ramón, Eduardo Casilari-Pérez, and José Cano-García. 2018. Analysis of a Smartphone-Based Architecture with Multiple Mobility Sensors for Fall Detection with Supervised Learning. *Sensors* 18, 1155 (2018), 29 pages.

[9] Hela Sfar and Amel Bouzeghoub. 2019. DataSeg: Dynamic Streaming Sensor Data Segmentation for Activity Recognition. In *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing (SAC 2019)*. 557–563.

[10] Kilho Shin, Tetsuji Kuboyama, Takako Hashimoto, and Dave Shepard. 2015. Super-CWC and super-LCC: Super fast feature selection algorithms. In *Proceedings of the 2015 IEEE International Conference on Big Data (Big Data 2015)*. 1–7.

[11] Wei Wang, Alex Liu, Muhammad Shahzad, Kang Ling, and Sanglu Lu. 2017. Device-Free Human Activity Recognition Using Commercial WiFi Devices. *IEEE Journal on Selected Areas in Communications* 35, 5 (2017), 1118–1131.