Vol.2011-UBI-31 No.23
2011/7/15

Regular Paper

People Identification Based on Sitting Patterns

Nguyen Gia,[†1] Jin Nakazawa,[†1] Kazunori Takashio[†1]
and Hideyuki Tokuda[†1]

This paper provide a people identification method based on the sitting patterns. This method uses weak evidences from pressure sensor, accelerometer sensor, and light sensor placed on a chair to recognize who is sitting on the chair without any psychological and physical burden on users. We propose how we have implemented the system using softmax regression model, gradient descent algorithm and nearest neighbor search algorithm. Our experimental result show that this method can be used in places which has private properties such as home or small office.

1. Introduction

Nowadays, there are several biometric people identification methods such as fingerprint based, iris based or by the using of vein. These biometric identifiers are strong and suitable for applications that request a high accuracy such as security applications. However these identifiers annoy user with the request for specific actions. For example, in the case of fingerprint, user has to properly touch a fingerprint scanner or in the case of retina, user has to look at retina scanner for a while, which might cause a psychological and physical burden on users. These methods also need delicate and expensive devices such as fingerprint scanner or retina scanner. In such situation as inside of a house or a small office with small number of user, we do not need high accuracy as those available with strong identifiers. For example, in a small office, which employees come from some different country, an employee comes to office, sits on a public chair and turns on a public computer. And then, a greeting sound of his/her country comes out of the speaker and that computer's language will be automatically change to his/her native language. For the other scenario, an office has a meeting but the boss is in a business trip so he use a robot for teleconference. The robot stands in the middle of meeting room and when the boss want to talk with one of his employees, he only have to let the robot know the employee's name instead of rotate robot by hand. Both of these scenario can be done with one of above people identification methods, but using biometric identifier for this scene is wasteful and unnecessary. Any mistake of people recognition in these scene is not a big problem, user can easily overcome it by simple actions. So, it is acceptable to inference to the user who is sitting beforehand based on weaker evidence. Collecting weak evidence also can be done without any psychological burden of users.

This paper provides a easy deployment and inexpensive people identification method that uses weak evidence from pressure sensors, accelerometer sensors and light sensor placed on a chair. The reason of using pressure sensor is the difference of weight among users. Also, we think that the sitting pattern is different between users so we use accelerometer sensor to recognize the movement of the chair when user sit in it. The light sensor is use to measure the coverage of user in the chair. We have used softmax regression model, a supervised learning algorithm and gradient descent algorithm, an algorithm to solve optimization problem to inference who is sitting in the chair

Remainder of this paper is organized as follows. Section 2 describes the design and implementation of system. The softmax regression model, gradient descent algorithm, nearest neighbor search algorithm and how they were used are discussed in section 3. Section 4 show the result of our experiment while section 5 is about related work. Conclusions and future work are written in section 6.

2. Design and Implementation

2.1 Hardware

We used SunSPOT for accelerometer sensor and light sensor. SunSPOT (Sun Small Programmable Object Technology) as shown in figure 1(a) is a wireless sensor network mote developed by Sun Microsystems. One SunSPOT device has three sensors including accelerometer sensor, light sensor and temperature sensor. In this case, we only use one SunSPOT device to sense accelerometer and light data. This data can be sent to computer where data is processed through a base

[†1] Keio University - Faculty of Environment and Information Studies

station as shown in figure 1(b).

(a) SunSPOT device

(b) SunSPOT base station

Fig. 1 SunSPOT

We also attached to the chair four pressure sensors. We used FSR406 for pressure sensor and figure 2 show how it can be viewed in fact. We want to use as least as possible sensors so we can reduce the cost of the system and we think that four is a good number. It is enough for people recognition issue based on weak evidence.

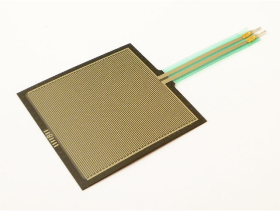

Fig. 2 FSR406 Pressure Sensor

Figure 3 show how sensors was placed in the chair. The light sensor is used for measure the coverage of user in the chair so it should be placed in the side of the chair.

2.2 Software

The software diagram of this system is shown in figure 4 . The data receiver module receives data from sensors and forwards it to the data processing module. In here, the data is normalized for used in the learning module. In the learning module, system use softmax regression model and gradient descent algorithm to inferences to the user who are sitting beforehand, output result and get confirmation from user through the user interaction module.

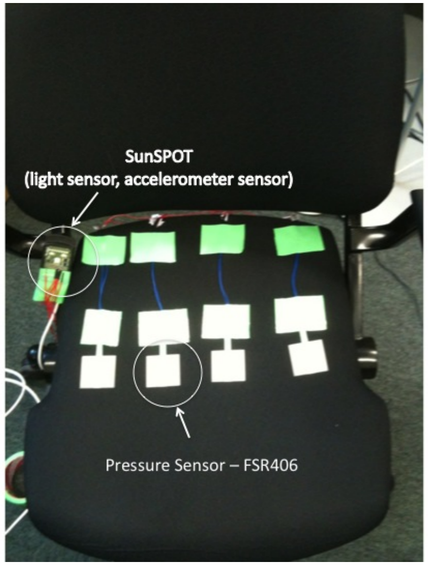

Fig. 3 Sensors placed on a chair

3. Approach

We consider the people identification problem with small number of user as a classification problem. The system classifies the data from sensors into groups. A group represents a user so the number of group equal to the number of user there are in data training set. When a user sitting to the chair, a set of data will be created. The system will determine which of group that this data set belongs to. That way, the system can recognize the user who is sitting beforehand. We used Nearest Neighbor Search Algorithm to resolve this classification problem. The "weight" used in nearest neighbor search algorithm are determined by Soft-

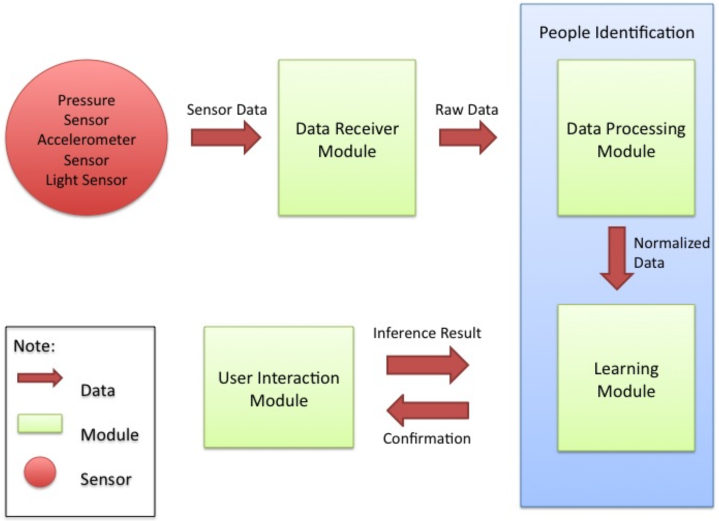

Fig. 4 The Software Diagram

max Regression Model and Gradient Descent Algorithm. The softmax regression model is discussed in subsection below.

3.1 Softmax Regression Model

Softmax Regression is a supervised learning algorithm used for multi-class classification. In Softmax Regression Model, we had a training set $\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots, (x^{(n)}, y^{(n)})\}$ of n labeled examples, where the input features are m dimensional vector $x^{(i)} \in \Re^m$ and the label y can take on k different values, $y^{(i)} \in \{1, 2, \ldots, k\}$. Given a test input x, we want our hypothesis to estimate the probability that $P(y = j|x)$ for each value of $j = 1, 2, \ldots, k$. I.e., we want to estimate the probability of the class label taking on each of the k different possible values. Thus, our hypothesis will output a k dimensional vector (whose elements sum to 1) giving us our k estimated probabilities. Concretely, our hypothesis $h_\theta(x)$ takes the form:

$$h_\theta(x^{(i)}) = \begin{bmatrix} P(y^{(i)} = 1|x^{(i)};\theta) \\ P(y^{(i)} = 2|x^{(i)};\theta) \\ \vdots \\ P(y^{(i)} = k|x^{(i)};\theta) \end{bmatrix} = \frac{1}{\sum_{j=1}^{k} e^{\theta_j^T x^{(i)}}} \begin{bmatrix} e^{\theta_1^T x^{(i)}} \\ e^{\theta_2^T x^{(i)}} \\ \vdots \\ e^{\theta_k^T x^{(i)}} \end{bmatrix}$$

Here, m dimensional vectors $\theta_1, \theta_2, \ldots, \theta_k \in \Re^m$ are the parameters of this model and θ_i^T is the transpose vector of θ_i. Notice that the term $\frac{1}{\sum_{j=1}^{k} e^{\theta_j^T x^{(i)}}}$ normalizes the distribution, so that it sums to one. For convenience, we will also write θ to denote all the parameters of our mode.

$$\theta = \begin{bmatrix} -\theta_1^T- \\ -\theta_2^T- \\ \vdots \\ -\theta_k^T- \end{bmatrix}$$

If we know the h_θ function, we can determine the class label that given input vector x belong to. That is the class label that have maximum estimated probability. But the h_θ function is expressed by the θ parameters, so we have to find all the θ parameters.
The cost function of Softmax Regression Model is shown in the equation below.

$$J(\theta) = -\frac{1}{n}\left[\sum_{i=1}^{n}\sum_{j=1}^{k} 1\{y^{(i)} = j\} \log \frac{e^{\theta_j^T x^{(i)}}}{\sum_{l=1}^{k} e^{\theta_l^T x^{(i)}}}\right]$$

Here, the equation $1\{.\}$ is the indicator equation, so that
$1\{\text{a true statement}\} = 1$ and $1\{\text{a false statement}\} = 0$. In Softmax Regression, we also have:

$$P(y^{(i)} = j|x^{(i)};\theta) = \frac{e^{\theta_j^T x^{(i)}}}{\sum_{l=1}^{k} e^{\theta_l^T x^{(i)}}}$$

Now, the training is mean finding all θ parameters that minimize the cost function. There are several methods to do it such as gradient descent algorithm or limited-memory BFGS algorithm. In this paper, we used gradient descent algorithm that is described below.

3.2 Gradient Descent Algorithm

The gradient descent algorithm is a algorithm used to choosing θ to minimize the cost function $J(\theta)$. It starts with some "initial guess" for θ, and that repeatedly change θ to make $J(\theta)$ smaller, until hopefully converge to a value of θ that minimizes $J(\theta)$. The gradient descent algorithm repeatedly performs the update:

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$

This update is simultaneously performed for all value of j. Here, α is called the learning rate.

To using gradient descent algorithm to minimize the cost function of softmax regression model, we need to compute the partial derivative of cost function $J(\theta)$. It is shown by the equation below.

$$\nabla_{\theta_j} J(\theta) = -\frac{1}{n} \sum_{i=1}^{n} \left[x^{(i)} (1\{y^{(i)} = j\} - P(y^{(i)} = j | x^{(i)}; \theta)) \right]$$

In particular, $\nabla_{\theta_j} J(\theta)$ is itself a m dimensional vector, so that its l-th element is $\frac{\partial}{\partial \theta_{jl}} J(\theta)$, the partial derivative of $J(\theta)$ with respect to the l-th element of θ_j. So we can use it to compute the update value of all parameters in softmax regression model.

But, take a look, if we take each of our parameter vectors θ_j, and subtract fixed vector ψ from it, so that every θ_j is now replaced with $\theta_j - \psi$ (for every $j = 1, 2, \ldots, k$), we have:

$$P(y^{(i)} = j | x^{(i)}; \theta) = \frac{e^{(\theta_j - \psi)^T x^{(i)}}}{\sum_{l=1}^{k} e^{(\theta_l - \psi)^T x^{(i)}}} = \frac{e^{\theta_j^T x^{(i)}} e^{-\psi^T x^{(i)}}}{\sum_{l=1}^{k} e^{\theta_l^T x^{(i)}} e^{-\psi^T x^{(i)}}} = \frac{e^{\theta_j^T x^{(i)}}}{\sum_{l=1}^{k} e^{\theta_l^T x^{(i)}}}$$

It show that if the cost function $J(\theta)$ is minimized by some setting of parameters $(\theta_1, \theta_2, \ldots, \theta_k)$, then it is also minimized by $(\theta_1 - \psi, \theta_2 - \psi, \ldots, \theta_k - \psi)$ for any value of ψ. Thus, the minimizer of $J(\theta)$ is not unique. To fix it, the cost function $J(\theta)$ is modified by adding a weight decay term $\frac{\lambda}{2} \sum_{i=1}^{k} \sum_{j=1}^{m} \theta_{ij}^2$ which penalizes large values of the parameters. Our cost function is now

$$J(\theta) = -\frac{1}{n} \left[\sum_{i=1}^{n} \sum_{j=1}^{k} 1\{y^{(i)} = j\} \log \frac{e^{\theta_j^T x^{(i)}}}{\sum_{l=1}^{k} e^{\theta_l^T x^{(i)}}} \right] + \frac{\lambda}{2} \sum_{i=1}^{k} \sum_{j=1}^{m} \theta_{ij}^2$$

With this weight decay term (for any $\lambda > 0$), the cost function $J(\theta)$ is now strictly convex, and is guaranteed to have a unique minimize solution. Also, the gradient descent algorithm is guaranteed to converge to the global minimum. To apply the gradient descent algorithm, we also need the partial derivative of this new definition of $J(\theta)$. The partial derivate is shown below.

$$\nabla_{\theta_j} J(\theta) = -\frac{1}{n} \sum_{i=1}^{n} \left[x^{(i)} (1\{y^{(i)} = j\} - P(y^{(i)} = j | x^{(i)}; \theta)) \right] + \lambda \theta_j$$

By using gradient descent algorithm with this equation to minimize the cost function $J(\theta)$, we will have a working implementation of softmax regression model.

3.3 Nearest Neighbor Search Algorithm

Nearest neighbor search (NNS), also known as proximity search, similarity search or closest point search, is an optimization problem for finding closest points in metric spaces. The problem is: given a set S of points in a metric space M and a query point $q \in M$, find the closest point in S to q. In many cases, M is taken to be d-dimensional Euclidean space and distance is measured by Euclidean distance. The Euclidean distance between points p and q is the length of the line segment connecting them. In Cartesian coordinates, if $p = (p_1, p_2, \ldots, p_d)$ and $q = (q_1, q_2, \ldots, q_d)$ are two points in d-dimensional Euclidean space, then the distance from p to q, or from q to p is given by:

$$d_{pq} = d_{qp} = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \ldots + (q_d - p_d)^2} = \sqrt{\sum_{i=1}^{d}(q_i - p_i)^2}$$

Now, if $CP(q)$ is the closest point to q in set S, we have:

$$CP(q) = \{p | p \in S; d_{pq} = \min_{i=1}^{n} d_{S_i q}\}$$

3.4 Calculation Process

We have discussed about softmax regression model, gradient descent algorithm and nearest neighbor search generally in subsections above. In this subsection, we describe how we use those algorithm in fact.

We use one accelerometer sensor, one light sensor and four pressure sensor placed on a chair for people recognition, so we have eight values of sensor data.

- Ax: The X-axis accelerometer value
- Ay: The Y-axis accelerometer value
- Az: The Z-axis accelerometer value
- Light: The light sensor value value
- A1: The first pressure sensor value
- A2: The second pressure sensor value
- A3: The third pressure sensor value
- A4: The fourth pressure sensor value

When a user sitting down to the chair, an array of 15 record

$$r = (Ax, Ay, Az, Light, A1, A2, A3, A4)$$

will be created. This array is called RA and it describes the information of one sitting time of a user. In our data training set, there are 10 RA for one user. So if the number of user is k, the number of RA in data training set is $n = 10k$.

We used nearest neighbor search with 8-dimensional Euclidean space for this classification problem. But the Euclidean distance function we used have a litter different to general function. Because these eight sensor data affect to result in different ways, so we modified the Euclidean distance function like this:

$$d_{pq} = \sqrt{\sum_{i=1}^{8}\theta_i(q_i - p_i)^2}$$

The parameters $\theta_1, \theta_2, \ldots, \theta_8$ is the "weight" of each sensor data and we used softmax regression model and gradient descent algorithm to determine them. Our people identify process can be described as following:

When a user sitting in the chair, a RA is created. We take the average of all records of this RA and the average of all records of all RA in data training set and use softmax regression model to compute the parameters used in nearest neighbor search algorithm.The gradient descent algorithm is implemented with learning rate $\alpha = 0.001$ and $\lambda = 0.001$ to minimize the cost function in softmax regression model. Finally, we use nearest neighbor algorithm with determined parameters to classify the new RA to one of k class labeled. The result is the user whom this class labeled stand for. There are always 10 RA for one user in our data training set, but the data training set is dynamic. When a user sitting down to the chair, after the system receive the confirmation from the user, in data training set, oldest RA of this user is replaced by newest RA. By this way, the system can adapt with the change of user's sitting pattern.

4. Evaluation

We evaluated this system in two cases. In the first case, we evaluated with a group of five people and in the other case, we evaluated with a group of ten people. In both case, one person have to sit in a chair twenty times, ten for training and ten for testing. Figure 5 shows the result of first case while figure 6 shows the result of last one. In the case of group of five people, we achieved a accuracy as 90 percentage and 72 percentage in the case of ten people.

In the case of five people, there are three people who is identified with the accuracy as 100%. One people with the accuracy as 90%, it means that there was only one mistake.

With the achievement accuracy as 72% in the case of ten peoples, this method

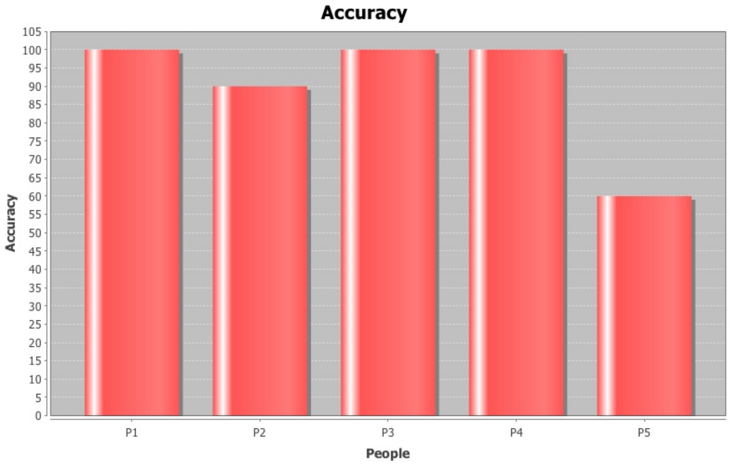

Fig. 5 Accuracy for 5 people case

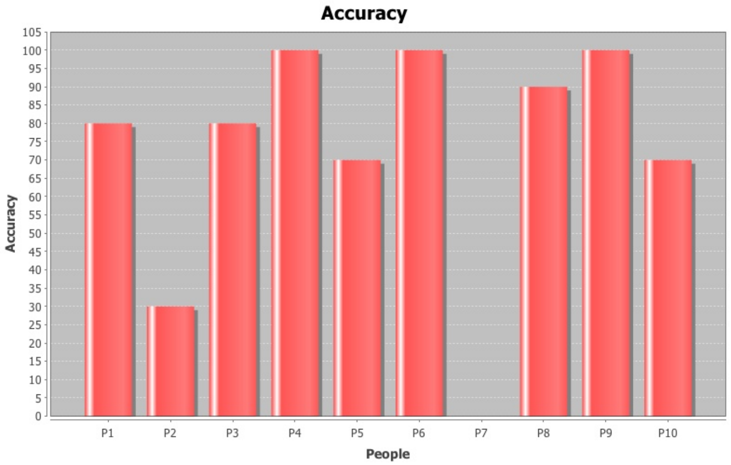

Fig. 6 Accuracy for 10 people case

certainly can be used in a small office or inside a house with small number of users.

5. Related Works

Masafumi Yamada et al. have used 32 pressure sensor placed on a chair to people recognition. They have tested with a group of eight people who are required to sit 20 times, 19 for training and only one for testing. The result is shown in the figure 5. Their system is not recognize user at the time user was sitting down but after few seconds, when the values of sensor get steady as shown in the figure 8. The classifier used is nearest neighbor method. Every testing data are classified to the nearest training data. Used features are classified into four groups to investigate how useful the information of pressure sensors is.

- Feature Set1 (FS1): 32 sensors values (32)
- Feature Set2 (FS2): sum of 32 sensors values (1)
- Feature Set3 (FS3): time difference of FS1 (32)
- Feature Set4 (FS4): normalized sensor values of FS3 (1)

No.	Features	Used part	Average recognition rate(%)	[min, max] of trials
♯ 1	(FS1 + FS2)	Sitting	63	[40, 79]
♯ 2	(FS1 + FS2)	Stable	90	[78, 99]
♯ 3-(a)	(FS1)	Sitting	63	[40, 79]
♯ 3-(b)	(FS4)	Sitting	52	[23, 76]
♯ 3-(c)	(FS2)	Sitting	21	[0, 34]
♯ 4-(a)	(FS1 + FS2 + FS3)	Sitting	86	[77, 99]
♯ 4-(b)	(FS3 + FS4)	Sitting	54	[23, 81]

Fig. 7 Result Table

As we can see, the achieved accuracy in steady part is about 90% but only 56% in the sitting part for average with a group of eight people.

6. Conclusions and Future Works

We have proposed a people identification method based on sitting patterns

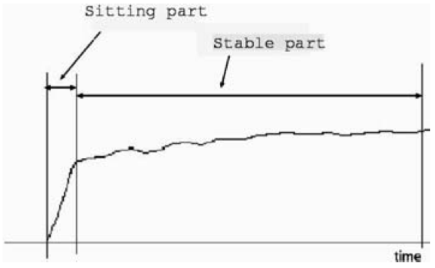

Fig. 8 Behavior of sensor output in time

of user. This method used weak evidence collected by accelerometer sensor, light sensor, and pressure sensor placed on a chair to inference who is sitting on it. We considered this problem as a classification problem and use nearest neighbor search algorithm with "weight" to resolve it. The "weight" used in nearest neighbor search algorithm is determined by softmax regression model while the cost function is minimized by gradient descent algorithm. We also presented the result of experiments which shown that this people identification has the accuracy enough to be used in places which have private properties such as inside of a house or a small office.

How to due with other evidences and what is the best way to place sensor to a chair or something else are the things to be discussed in the future. We also intend to implement a module to recognize the posture of user or the user's mood.

References

1) A.K. Jain, L. Hong, S. Pankanti and R. Bolle
"An Identity Authentication System Using Fingerprints," Proc. IEEE, vol. 85, no. 9, pp. 1,365-1,388, 1997.

2) L. Ma , T. Tan , D. Zhang and Y. Wang
"Personal identification based on iris texture analysis", IEEE Trans. Pattern Anal. Mach. Intell., vol. 25, p.1519 , 2003.

3) R. Hill
Retina Identification, Biometrics: Personal Identification in Networked Society, A.K. Jain, R. Bolle, and S. Pankanti, eds., Kluwer Academic, pp. 123-141, 1999.

4) C. Do and A. Ng.
"Transfer learning for text classification" . In Proceedings of Neural Information Processing Systems (NIPS), 2005

5) Burges, C., Shaked, T., Renshaw, E., Lazier, A., Deeds, M., Hamilton, N., and Hullender, G.
"Learning to Rank using Gradient Descent". In ICML '05 Proceedings of the 22nd international conference on Machine learning.

6) Peter N.Yianilos
"Data structures and algorithms for nearest neighbor search in general metric spaces". SODA '93 Proceedings of the fourth annual ACM-SIAM Symposium on Discrete algorithms

7) Stefan Berchtold, Christian Bohm, Daniel A.Keim, Hans-Peter Kriegel
"A cost model for nearest neighbor search in high-dimensional data space". PODS '97 Proceedings of the sixteenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems.

8) http://www.sunspotworld.com

9) M. Yamada, J. Toyama, M. Kudo
"Person Recognition by Pressure Sensors". Knowledge-Based Intelligent Information and Engineering Systems, Lecture Notes in Computer Science, Vol. 3684, Rajiv Khosla, Robert J. Howlett, Lakhmi C. Jain (Eds.), Springer, 2005, 703-708.