Split-Merge Algorithm and Gaussian Mixture Models for AAL

GuoQing Yin and Dietmar Bruckner

Institute of Computer Technology
Vienna University of Technology, Austria, Europe
{yin, bruckner}@ict.tuwien.ac.at

***Abstract*- Analyzing time series sensor data and build statistical model in real time has to overcome two problems at least: the data count increase with time and the distribution of the data is dynamically. To deal with this kind of problems Gaussian mixture model and split-merge algorithm provide useful way.**

In an AAL project we handle the time series sensor data from a medical box contactor and a meal entrance contactor. Using Gaussian mixture model and split-merge algorithm to analyze the sensor data gathered for about one and a half months and built the statistical model.

Keywords: Gaussian Mixture Models, Split-Merge Algorithm, Real Time Analysis.

I. INTRODUCTION

There are many papers about Gaussian mixture model (GMM) and split-merge algorithm: for speaker identification [1] propose a self-splitting Gaussian mixture learning (SGML) algorithm for Gaussian mixture modeling, [2] presents a split and merge EM algorithm to overcome the local maximum problem in Gaussian mixture density estimation. In [3] the authors developed a new methodology for fully Bayesian mixture analysis, using reversible jump Markov chain Monte Carlo methods to jumping the parameter subspaces and the different numbers of components in the mixture, while [4] propose a new kind of dynamic merge-or-split learning (DMOSL) algorithm to deal with the selection of number of Gaussians in the mixture, [5] describe an EM algorithm for nonparametric maximum likelihood (ML) estimation with variance component structure, [6] introduces a greedy algorithm for learning Gaussian mixture model, using combination of global and local search, [7] introduced a split-and-merge operation in order to alleviate the problem of local convergence of the usual EM algorithm.

a. AAL Background and ATTEND Project

Ambient Assisted Living (AAL) is a new search field, it focus on enhance the life quality of the elderly and prolong the independent living in the elderly own home with the help from model technology. But because of the elderly have their own problems, such as action obstacles, memory disorder … how can the elderly people use the model technology system?

Within the scope of the project ATTEND (AdapTive scenario recogniTion for Emergency and Need Detection) a system will be developed that increases the time frame of independent living of elderly persons in their used living environment. The system comprises an intelligent, adaptive network of sensors, which are to be installed in the living environment of the user in order to thoroughly observe his behavior. An important aspect is that the sensors shall work independent and in a preferably invisible fashion.

ATTEND learns about normal behavior of the user. In case of unusual behavior an alarm plan can be worked out (e. g. enquiring the user, calling a neighbor, calling an external organization). The system is intended to increase comfort, security and social inclusion of the customer and ideally also help with the early detection of upcoming medical problems. In case of an emergency the system can contact primary and secondary users (family, neighbor, care giver) via external interfaces.

An important point in the development will be the requirement of minimal installation and maintenance effort. In later stages of development the system should act like a butler in the background and start acting – depending on how good the butler is – in various situations on its own.

In this paper we use Gaussian mixture model and split-merge algorithm to analyze the sensor data, for example the data from medical box and the data from meal entry contactor. The generally daily models about the sensors will be build and according the model if some unusual behaviors happened, the system will send aware signal to user or alarm signal to neighbor or caregiver.

For example a user takes tablets from a medical box at similar time points every day. A contact sensor installed at the door of medical box. If the door opened or closed a signal will be send to the controller. According the gathered data for some days, for example one and a half months, the system learns the model that when the medical box will be opened and closed, that means when the user take tablets. If one day the user forgets to take tablets at some time points, the system will sent aware signal to the user. The same situation is for the contact sensor installed at the meal entry. Every day the meal will be send to the user through a meal entry. A contact sensor gets data every time when the meal sends into the room or the tableware send out of the entry. A model will be build according the gathered data for a time interval, for example one and a half months. If one day there is no meal send to the user at some time points, the system will send aware to the user, neighbor or caregiver.

b. Basic Parameters

In the medical box there are contact sensor installed, if the door opened the sensor send value "0" to the controller, if the door closed sensor send value "1" to the controller. We gathered and analyzed the time points (t) that the door opened and closed in all one and a half months. There should be a time points set T = ($t_{1,}$ $t_{2,}$ $t_{3\ ...}$ t_n). Because the user takes tablets every day several times at some time points, for example in the morning, before lunch or after lunch, and in the evening just before go to bed, there should be some time points distribution. The distribution is Gaussian mixture model. The same situation happened with the sensor data from meal entrance.

Through analyse the time points gathered about one and a half months, a generally model of the sensor data will be build, which means, for the medical box sensor is when the user will take tablets, and for the meal entrance contact sensor is when the meal will be send to the user or the tableware send out of the entry. In fact this is a cluster analyse problem. Here the gathered time points composed a cluster. From Gaussian mixture models and split-merge algorithm we can get useful and affordable results.

II. Gaussian Mixture Models And Merge-split Algorithm

This section deals with the mathematical background of the algorithms.

a. Gaussian Mixture Model

A standard Gaussian function is defined as

$$\varphi_{\mu,\sigma^2}\,(x) = \frac{1}{\sigma\sqrt{2\pi}}\ \mathrm{e}^{-\frac{(x-\mu)^2}{2\sigma^2}} \qquad (1)$$

Here μ is the expected value (mean value of the clustering time points), σ is the standard deviation of each time point cluster, x is the time point value (t_n).

Before we use split-merge algorithm for clustering the time points set, we have to define the range of the parameters: $0 \leq \mu \leq 24$ (because there are only 24 hours one day, so the time point interval is between 0 and 24), $0.5 \leq \sigma \leq 2$ (the initial value can be changed according the real situation) , $0 \leq t_n \leq 24$. The number of initial components $s = 3$, that means there are sets $\{\mu_{1,}\ \mu_{2,}\ \mu_{3;}\ \sigma_{1,}\ \sigma_{2,}\ \sigma_{3;} P_{1,} P_{2,}\ P_3\ \}$. Here P is the percent value of each time point component and $\sum P_{(1,2,3)} = 1$ and these prior parameters are random variables. Threshold value for split, merge and delete components: $\mu_{\ threshold,}$ $\sigma_{\ threshod,}$ $\sigma_{\ threshod2,}$ $P_{threshold.}$ The maximum number of time points for adjusting the learning rate is M and current angle count is M'. Each new time point that gets into the set T is $T_{r.}$ (r ≥ 1). The index s is the component index within the mixture model.

With these parameters and definition we can begin to cluster the data set with split-merge algorithm.

b. Split-Merge Algorithm

- Compute and then normalize posteriors

$$\mathrm{P_s(T_r) = P_s}{*}\varphi_{\mu,\sigma^2}\ \mathrm{(T_r);\ P_s(T_r) = P_s(T_r)\ /\ \sum\ P_s(T_r)} \qquad (2)$$

- Compute new means

$$\mu_s = \mathrm{(1–P_s(T_r))}{*}\mu_s\mathrm{+P_s(T_r)}{*}\mathrm{(M'}{*}\mu_s\mathrm{+T_r)/(M'+1)} \qquad (3)$$

- Compute new variances

$$\sigma_s\mathrm{= ((1–P_s(T_r))}{*}\sigma_s\mathrm{+P_s(T_r)}{*}\mathrm{(M'}{*}\sigma_s\mathrm{+}|\mu_s\mathrm{-T_r}|\mathrm{)/(M'+1)} \qquad (4)$$

- Compute new priors

$$\mathrm{P_s = (T'\ *\ P_s + P_s(T_r))\ /\ (M' + 1)} \qquad (5)$$

- Keep the learning rate and adaptability

$$\text{If M' ≥ M, M' = M} \qquad (6)$$

- After some initial iterations, start checking if it is necessary to split components: If $\sigma_s > \sigma_{threshod,}$ then create new component (index S) from old component (index s)

$$\mu_S = \mu_s + \sigma_s\ /\ 2;\ \ \mu_s = \mu_s - \sigma_s\ /\ 2 \qquad (7)$$

$$\sigma_S = \sigma_s = \sigma_s\ /\ 2 \qquad (8)$$

$$\mathrm{P_S = P_s = P_s\ /\ 2} \qquad (9)$$

- If necessary, merge components (s' and s'')

If ($|\mu_{s'}$ - $\mu_{s''}\ |$ < $\mu_{\ threshold}$ and $|\ \sigma_{s'}$ - $\sigma_{s''}\ |$ < $\sigma_{threshold2}$) then merge component s'' into s' and delete component s''. Here $\sigma_{threshold2}$ <= $\sigma_{threshod.}$

$$\mu_{s'} = (\mu_{s'} * \mathrm{P}_{s'} + \mu_{s''} * \mathrm{P}_{s''}\ )\ /\ (\mathrm{P}_{s'} + \mathrm{P}_{s''}\ ) \qquad (10)$$

$$\sigma_{s'} = \max\ (\sigma_{s',}\ \sigma_{s''}) \qquad (11)$$

$$P_{s'} = P_{s'} + P_{s''} \qquad (12)$$

- If any component's prior decreased too much so that P(s') < $\mathrm{P}_{threshold}$ then delete the component and adjust the other priors P(s):

$$P(s) = P(s) / \sum P(s) \qquad (13)$$

- Repeat with all new values.

III. Result And Conclusion

a. Result

The first example is about the original sensor data from medicine box about one and a half month. The sensor data showed in Fig.1.

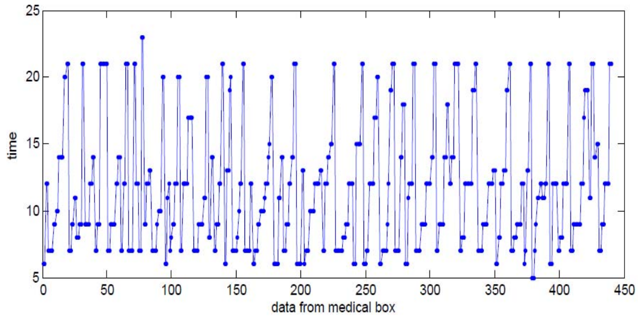

Fig. 1. The data from the medical box.

X axis of Fig. 1 is count of the data in all the one and a half months, there are about 440 count in the data set T. Y axis of Fig. 1 is the time points of each data. The interval is from 0 to 24. From Fig. 1 we can see that the user takes many tablets every day, but not very regularly. Some days the user only opens and closes the medical box at 2 different time points, but on other days the user opens and closes the medical box at 6 different time points. There are perhaps some wrong data, that means the user opens and closes the medical box but not for tablets. For this kind of data we can not distigwish them with the right data just from the contact sensor. It is difficult to filt them. But we can use them as normal data, because the "wrong data" distrubuted in all the data sets as a "background noise". On the other hand the count of the wrong data should not be too much. The influence of this kind of data to the learning result can be ignored.

Further more if the user opens and closes the medical box regulerlly every day but not for tablets, it must be a behavior of the user. So the system should learns the time points that the behavior happened.

Fig.2 showed the learned result (the green line) with the histogram together, from Fig. 2 we can see that all the important time points group be found through the learning.

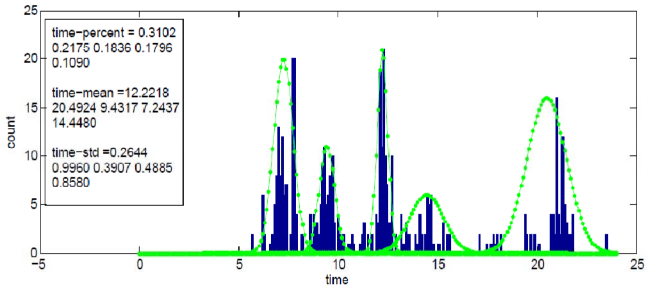

Fig. 2. To compare the learn result with the histogram

X axis of Fig.2 is the time axis of one day, the Y axis is the count of how many time points gathered in each time interval. From the Fig.2 we can see that there are mainly 5 time points that the user takes tablets: in the morning about 7.24, 9.43, at noon 12.22, 14.44, and at evening 20.49. The probability that at these time points the user takes tablets are 17.96%, 18.36%, 31.02%, 10.9%, and 21.75%. Because the data gathered from one and a half months, the user has not taken tablets every day at the exactly same time points, and there perhaps some wrong data there, so there are standard deviations for each time points: 0.49, 0.39, 0.26, 0.86, and 1.

Another important point from Fig.2 is that, if the user does some behaviour regularly at some time points the standard deviations will be smaller. It indicated a more precise learning result. For example in Fig. 2 the user takes tablets at noon focus on 12.22, so the standard deviation of this time point is 0.26, on the other hand at evening the user takes tablets not so focus on one time points, so there has a standard deviation value 1.

Furthermore if we analyze the learning result we can get such a conclusion: if the user has regularly daily behaviours, the learning result will be more precise, on the other hand if a user has a random life style, the learning result will be more inaccurate. In extreme situation will lead to a wrong learning result. Because the user them self is the "trainer".

Fortunately most of the elderly has relatively stable life style, for example: when get up, when shower, when take breakfast, when has lunch, when has a rest…… when gone to bed. The regularly life style is the basic of a useful learning result. If the stable life style changed, there should be something happened or there will be something happen. Use this idea we can predict the hidden health problems of the user earlier.

The second example is from the meal entrance contactor, the data showed in Fig. 3.

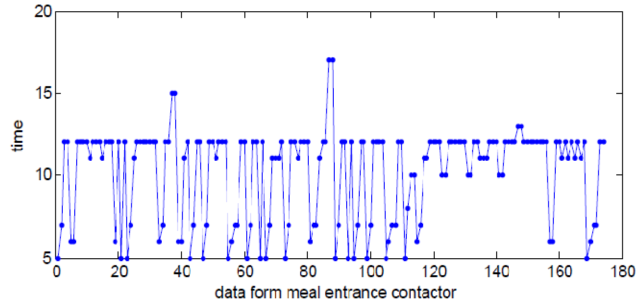

Fig. 3. The data from the meal entrance contactor

Fig. 3 showed the data from the meal entrance contactor. There are about 174 data points gathered about one and a half months in the data set. From Fig. 3 we can see that there are regulerlly meal send time point, it is about at 12 o'clock. In the evening there are without meal send to the user.

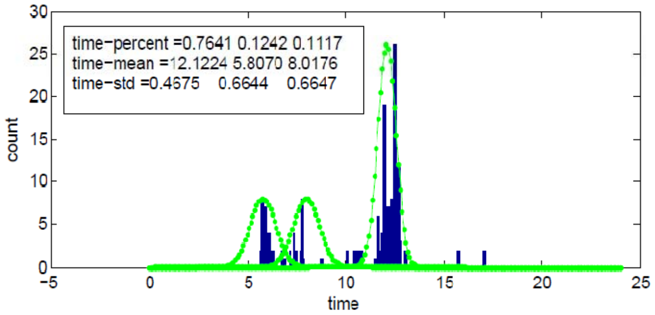

Fig. 4. To compare the learn result with the histogram

Fig. 4 is the learning result from the data set. There are mainly 3 time points: in the morning 5.8, 8, and at noon 12.12. The probabilities that at these time points the user get meal are 12.42%, 11.17%, and 76.41%. The standard deviations for each time points are 0.66, 0.66, and 0.46.

From the learning result we see that the elderly has own life style, in the morning get 2 times meal and at noon once, but in the evening take without dinner. If the life style changed or at the time points (in the standard deviation interval) there are no meal send to the user, the system should send signal to neighbour or caregiver.

b. Conclusion

From above result we can say that Gaussian Mixture Model and the split-merge algorithm is a powerful tool for unsupervised learning and very useful in practical.

For the application of AAL the stable life style of the user is the basic of a useful learning result.

IV. Outlook

In the future different sensor data and different algorithm will be tried to analyze the behaviors of the user. A more robust learning algorithm should be developed. Furthermore the life style changing caused by hidden health problem will be searched too.

References

[1] Shih-Sian Cheng, Hsin-Min Wang, Hsin-Chia Fu (2004). "A Model-Selection-Based Self-Splitting Gaussian Mixture Learing with Application to Speaker Identification", in EURASIP Journal on Applied Signal Processing 17, 2626-2639.

[2] N. Ueda, R. Nakano, Z. Ghahramani, and G. Hinton, "SMEM algorithm for mixture models," *Neural Computation*, vol. 12, no. 9, pp. 2109–2128, 2000.

[3] SYLVIA RICHARDSON, PETER J. GREEN (1997), "On Bayesian Analysis of Mixtures with an Unknown Number of Components", in J. R. Statist. Soc. B 59, No. 4, *pp*. 731-792.

[4] Jinwen Ma and Qicai He (2005). "A Dynamic Merge-or-Split Learning Algorithm on Gaussian Mixture for Automated Model Selection", Department of Information Science, School of Mathematical Science and LMAM, Peking University, Beijing, 100871, China.

[5] Aitkin, M. (1999). "A General Maximum Likelihood Analysis of Variance Components in Generalized Linear Models", In Biometrics 55, 117-128, March 1999.

[6] N. Vlassis and A. Likas, "A Greedy EM Algorithm for Gaussian Mixture Learning," *Neural Processing Letters*, 15: 77-87, 2002.

[7] Zhihua Zhang, Chibiao Chen, Jian Sun, Kap Luk Chan (2003). "EM algorithms for Gaussian mixture with split-and-merge operation" in Pattern Recognition 36 1973-1983.