ACCELEROMETER-BASED ACTIVITY RECOGNITION ON A MOBILE PHONE USING CEPSTRAL FEATURES AND QUANTIZED GMMS

Jussi Leppänen and Antti Eronen

Nokia Research Center, Tampere, Finland

ABSTRACT

The use of cepstral coefficients derived from a filter bank with logarithmically spaced band center frequencies and Gaussian mixture models (GMMs) with quantized parameters (qGMMs) are proposed for accelerometer-based activity recognition of mobile phone users. The use of a filter bank with logarithmically spaced band center frequencies is shown to yield better results than the use of a filter bank with linear spacing between band center frequencies. GMMs and qGMMs are shown to achieve similar recognition accuracies. However, the computation time using qGMMs is shown to be either at the same level or faster when compared to GMMs, depending on model complexity. Using the proposed approach, we achieve an accuracy of 72.6% and 91.3% on two recognition tasks with seven and five activities, respectively.

Index Terms— Physical activity recognition, Gaussian mixture model with quantized parameters, mobile phone

1. INTRODUCTION

Various sensors such as the microphone, accelerometer, magnetometer and gyroscope are nowadays common in mobile phones. This creates a possibility to develop algorithms which process the sensor data to produce, for example, inferences on the user activity. Previously, the availability of some of the sensors was limited to only the higher end smart phones. Currently, sensors are becoming common in the cheaper category phones as well. Mobile phone based activity recognition applications are thus becoming available for a larger audience.

The lower end phones are characterized by limited amounts of memory and computing capacity making the development of low-complexity and scalable algorithms for sensor data processing essential. In addition, many of the use cases for mobile sensing, such as automatic status updates to social networking services ([1]), should be running continuously in the background with minimal power consumption.

In this paper, we present our system for accelerometer-based user activity recognition running on mobile phones. The paper is organized as follows. In section 2, we describe earlier work relevant to this paper. In section 3, we describe the data that was used to train and test the system. Sections 4 and 5 describe the cepstral features and Gaussian mixture models (GMMs) with quantized parameters (qGMMs) used in the system, respectively. In section 6, we present experiments which investigate the accuracy and computational complexity of the system. Finally, we present our conclusions in section 7.

2. RELATED WORK

Over the past years, many different approaches to physical activity recognition using accelerometer data have been proposed. For the features, a common choice is either time domain features or a combination of time and frequency domain features [1]-[7]. Various classifiers, such as k-nearest neighbor (k-NN), support vector machines (SVM), logistic regression, decision trees and GMMs have been used in combination with the time and frequency domain features [1]-[7].

The use of cepstral features has not been very common for accelerometer-based activity recognition. Some studies on the subject do, however, exist. In [8], cepstral coefficients obtained using linear filter bank spacing were used for classifying between activities such as lying and walking. A classification accuracy of 78% was reached using a GMM classifier. Cepstral features based on a filter bank with three hand-picked bands were used for user gait classification in [9]. These features were shown to outperform cepstral features that were derived from linear prediction coefficients. The authors reported a classification accuracy of over 93% using GMMs with 8 mixture components per model. In [10], cepstral coefficients calculated directly based on the logarithm of Fast Fourier Transform (FFT) magnitude along with several other features were used for recognition of physical activities based on several sensors including the microphone, accelerometer and a high frequency light sensor. An overall accuracy between 80% and 84% was reported in classifying between eight everyday activities.

Accelerometer-based activity recognition specific to mobile phones has been studied, for example in [6] and [7]. In [6], time-based features, such as mean, standard deviation and average absolute difference are calculated from accelerometer data recorded using mobile phones. These are then used together with decision tree, logistic regression, and multi-layer perceptron (MLP) classifiers. An accuracy of 91.7% is reported when the MLP classifier is used on a six-class classification task. In [7], the Jigsaw continuous sensing engine is presented. Among other sensing tasks, the system performs accelerometer-based activity recognition using time and frequency domain features and a decision tree classifier. Accuracies of around 95% are reported when classifying between five different activities (walking, running, stationary, vehicle and cycling).

In this paper, we propose the use of cepstral features derived using a logarithmically scaled filter bank for accelerometer-based activity recognition on a mobile phone. In addition, for the same purpose, we also propose the use of GMMs with quantized parameters for classification. To the authors' best knowledge, the above two methods have not been published previously for accelerometer-based activity recognition.

Table 1. Mapping of user-provided annotations to activities and the number of 1-minute recordings per activity.

User-provided annotation	Activity	Number of recordings
Breakfast, Lying, Sitting, Sleep, Standing, Still	Idle/still	14185
Walking	Walking	1393
Running	Running	46
Skiing	Skiing	162
Bicycling	Cycling	943
Car, Subway train, Taxi, Train	Vehicle	1067
Cleaning, Cooking, Skating	Other	23

3. DATA COLLECTION

The data used for the experiments presented in this paper was collected with various Symbian S60 and Symbian^3 mobile phones. Several users carried the phones while doing their everyday activities. The phones were running a data collection application which would prompt the user at set intervals to annotate what they were doing. After the user inputted the annotation, accelerometer data was captured for approximately one minute. Each annotation comprised the user's current activity (standing, walking, running, etc.) and the location of the phone (pocket, hand, handbag, etc.). The users were instructed to carry on doing what they were doing prior to annotation for the duration of the recording. The default interval of the recordings was 20 minutes, but this could be changed by the user.

The accelerometer data consisted of 3-axis accelerometer readings recorded at an approximately 34 Hz sampling rate, 8-bit resolution and a ±2g data range. The sampling was not constant and varied slightly due to phone processor load etc. The accelerometer data from six different users was selected for the experiments presented here. Table 1 shows the amount of data collected by these users and how the user-provided annotations were mapped to the activities we are considering in this paper. The total amount of recorded data entries is 17819 which is equivalent to approximately 300 hours of accelerometer data. In addition to the data set described above, we used, for model training purposes, approximately 44 hours of accelerometer data recorded in a non-periodic manner.

4. FEATURE EXTRACTION

The features used in our system are cepstral coefficients calculated using a filter bank with logarithmically spaced band center frequencies. Looking at the frequency content in accelerometer data for different activities, it seems that the most useful information for differentiating between activities is in the shape of the spectrum rather than in the exact locations of peaks in the spectrum. The frequency content of a few seconds of accelerometer data from two persons walking is shown in Fig. 1. Both spectra have a similar shape, but the spectrum peak locations are not aligned in frequency. The spectral peaks correspond to the step rate and its integer multiples, and naturally walking should be classified as walking independent of the step rate. Cepstral coefficients calculated from the output of a mel-spaced filter bank are known to capture the rough spectral shape for audio signals while ignoring the pitch information and thus seem suitable to perform the analogous task for accelerometer signals.

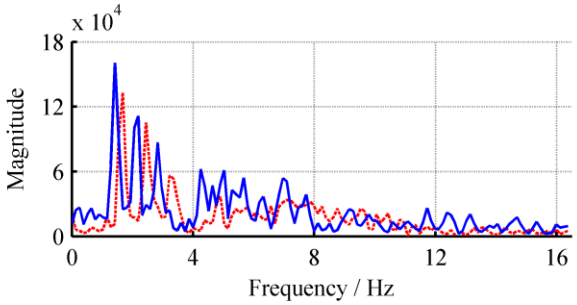

Fig. 1. FFT magnitude calculated from accelerometer recordings of walking activity from two persons.

An additional benefit of using cepstral coefficients and GMMs for activity recognition is that we can use the same classifier for accelerometer-based activity recognition as we use for audio-based environment recognition [11]. We are able run the same code, but with different parameters, for both tasks. Some details on the feature extraction used in our system is described next.

Since the orientation of the accelerometer during system usage cannot be assumed constant, feature extraction is done on the magnitude of the 3-axis accelerometer data. The feature extraction process follows closely the calculation of the well-known mel-frequency cepstral coefficients (MFCCs) [12].

The main difference to MFCCs is the scaling used in the filter bank. The filter bank used in the MFCC calculation is based on the mel-scale which happens to be basically linear in the range of frequencies that are encountered in the accelerometer data used here. We use the following scaling function, which has the same form as the mel-scale:

$$Q(f) = 10log_{10}\left(1+\frac{f}{3}\right), \quad (2)$$

where f is the linear frequency in Hertz. The scaling allows us to have higher resolution at the lower frequencies, which appear to be more informative for activity recognition purposes.

We use 18 triangular filters that are centered at frequencies such that the corresponding scaled frequencies, determined by (2), are linearly spaced. We then perform a DCT on the logarithm of the filter bank magnitudes and use the first 12 components (including the 0^{th}) of the DCT output as the final feature vector. Having 18 filters in the filter bank and using 12 cepstral coefficients was found to work well in our experiments. The choice is not strict but there is a small range of values that perform more or less the same.

5. CLASSIFICATION

In our system, the activities are modeled with qGMMs. Hidden Markov models with quantized parameters where first proposed in [13] where they were shown to reduce memory requirements while maintaining the accuracy of continuous density HMMs in speech recognition. QGMMs are created from continuous density GMMs with diagonal covariance matrices by applying a scalar quantization on the mean and variance parameters. The quantization is done separately for the mean and variance parameters. If certain conditions hold, the quantization allows for faster probability calculation during recognition compared to continuous density models. In addition to quantizing the model

parameters, we employ feature vector quantization to further increase the computation speed [14].

The probability calculation for an N-by-1 observation vector x using continuous density GMMs with diagonal covariance matrices is done using the following formula:

$$\log b(x) = \log \sum_{k=1}^{K} \exp\left\{ \log\left(w_k \frac{1}{\prod_{i=1}^{N} \sqrt{2\pi\sigma_{ki}^2}} \right) - \sum_{i=1}^{N} \frac{(x_i - \mu_{ki})^2}{2\sigma_{ki}^2} \right\}, \quad (3)$$

where K is the number of densities in the Gaussian mixture, μ_{ki} and σ_{ki} are the mean and standard deviation of the ith feature vector component of the kth density, respectively, and N is the feature vector dimension. For each density, there are two parts, a constant and the Mahalanobis distance to the feature vector x. When the means μ_{ki}, standard deviations σ_{ki} and feature vector components x_i are quantized, the Mahalanobis distance can take a discrete set of values. This means that we can pre-compute and store these values in a Mahalanobis distance table T, whose elements are defined as:

$$T_{f,m,s} = \frac{(x_f - \mu_m)^2}{2\Sigma_s}, \quad (4)$$

where μ_m is the mth mean quantization value, Σ_s is the sth variance quantization value and x_f is the fth feature quantization value. In this case, the total number of values in the table is $F \cdot M \cdot S$, where F, M and S are the number of quantization levels for the feature, mean and variance values, respectively. To be able to use the same quantization for all feature vector components, the feature values are normalized to zero mean and unity variance using the global mean and variance estimates obtained from the training data.

During probability calculation, for every feature vector, we first quantize the feature vector components, and then perform a table lookup from T for each feature vector component and sum them up. For a single Gaussian mixture, the log likelihood is thus calculated as follows:

$$\log b(x) = \log \sum_{k=1}^{K} \exp\left\{ C_k - \sum_{i=1}^{N} T_{f_i, m_{ki}, s_{ki}} \right\}, \quad (5)$$

where C_k is the (pre-computed) first part of (3) and m_{ki} and s_{ki} are the mean and variance quantization levels for ith mean and variance component of the kth density and f_i is the feature quantization level of the ith feature vector component of feature vector x. When inputting a sequence of feature vectors, the log likelihoods are calculated for each and summed together. The class corresponding to the GMM yielding the largest log likelihood is chosen as the classification result.

The activity models are trained by first using maximum-likelihood (ML) training to create a GMM for each activity. ML training was performed using the HTK hidden Markov model toolkit [15]. After the ML training the model parameters are quantized by applying a Lloyd-Max quantizer on the mean and variance parameters separately [16], [17]. 5- and 3-bit quantization is used for the mean and variance parameters, respectively. These values were chosen because they perform well and a mean and variance value pair can be stored in an 8-bit byte. The same quantizer that is used for the means is also used as the feature quantizer.

6. EXPERIMENTS

In this section, we describe several experiments on the front- and back-end parameters. We present how the parameters affect the recognition accuracy as well as the computational complexity of the system. The system was tested on two classification tasks. The first, Task A, comprises of classifying between seven different activities; 'idle/still', 'walking', 'running', 'cycling', 'skiing', 'vehicle' and 'other'. The second task, Task B, is the same as Task A, but with the 'other' and 'skiing' classes removed.

The experiments were carried out using leave-one-user-out cross validation. At each iteration, we left out all the data from a single user and trained the activity models using the remaining data. The system was then tested on the data of the left out user. As we had six people contributing to the test data collection, this was repeated six times and the accuracies were calculated from the results. We report accuracies using two numbers; weighted accuracy and class average accuracy. The weighted accuracy is simply the percentage of correct classifications of all classifications. This number tells us how often the classifier is correct if it would be run periodically on a mobile phone of a user. However, the weighted average accuracy is biased towards the recognition rate of the activities that are more frequently found in the data. For this reason, we also present the average of the activity specific accuracies.

6.1. Recognition accuracy

The proposed system is tested on a series of experiments studying the effect of different parameters on the activity recognition accuracy. The front-end of the proposed system outputs a vector of 12 cepstral coefficients from a frame of 120 samples (~4 seconds). A 50% frame overlap was used. The tests were done on 30 second clips of accelerometer data. This results in 15 feature vectors that are fed into the qGMM classifier. Classification was done using a qGMM classifier with a mixture of 16 quantized Gaussian densities for each activity. The proposed system achieved a class average accuracy of 72.6% on Task A. The activity specific classification results are shown in the first column of Table 2. In the experiments below, unless otherwise mentioned, the accuracies are reported for Task A and the parameters of the system are fixed to that of the proposed system.

First, we tested how the filter bank with logarithmically spaced band center frequencies performs versus a linear filter bank, such as the one which was used in [8]. The results, which are presented in Table 3, show that using a logarithmic filter bank outperforms the linear filter bank.

Next, a test was run to understand how the quantization of the model parameters affects the performance of the system. Table 4 shows the result of this test. The labels used for the qGMMs are of the form NmMvLf, where N, M and L are the number of bits used for the mean, variance and feature quantizers, respectively. From the table we see that using enough quantization levels, qGMMs perform at the same level as GMMs. The memory requirements for storing the model parameters for the different quantizer setups are also shown in Table 4. The differences in model parameter memory requirements are due to the Mahalanobis distance tables that are stored along with the models. The sizes are relatively small for all setups, and should not be problem when used in systems running on modern smart phones. If we were to run the system on an environment where memory and computational resources are heavily constrained, the memory requirements of the model parameters may be an issue. From Table 4 we see that if we use only a single Gaussian density per model, the memory requirements are equal for the most heavily quantized models and the continuous density models. In this case, the continuous density models outperform the quantized models.

Table 2. Recognition accuracies for the proposed system, a k-NN based system and the Jigsaw engine. The accuracies for the Jigsaw engine are the ones reported in [7], and are obtained with a different training and testing data than the other systems.

	Task A		Task B	
Activity	proposed	k-NN	proposed	Jigsaw[7]
Idle/still	94.3	92.9	94.4	95.19
Walking	76.7	40.7	90.2	96.81
Running	95.7	97.8	95.7	98.01
Cycling	68.9	66.9	92.9	92.05
Vehicle	82.3	61.2	83.3	87.47
Skiing	55.6	35.2	-	-
Other	34.8	17.4	-	-
Average	72.6	58.9	91.3	93.9

Table 3. Weighted and class average recognition accuracies for linear and logarithmic filter banks on Task A.

Filter bank	W Acc. / %	CA Acc. / %
Linear	87.1	59.1
Logarithmic	90.5	72.6

6.2. Computational complexity

Our activity recognizer is fully implemented using fixed-point arithmetic. This ensures fast computation on mobile phones where a floating-point unit is not available. We tested the system on a Nokia E7 mobile phone and computed the time required for feature extraction and classification for 30 seconds of accelerometer data. The computation time was calculated by first collecting 30 seconds of accelerometer data and then timing the execution of 500 back-to-back feature extractions and classifications. Table 5 shows the computation times for qGMMs and GMMS with several different numbers of Gaussians per model. From the table we see that for 16 Gaussians per model there is a large difference in the probability calculation times in favor of qGMMs. When we only have a single Gaussian per model, the computation time is approximately the same. What can also be observed is that the classification accuracies when using 16 Gaussians or a single Gaussian are very close to each other. This would imply that the use of qGMMs does not provide any benefit over GMMs.

6.3. Experimental results in context

In this section we present how our activity recognition compares to earlier work done in the field. The activity recognition task used to test the Jigsaw continuous sensing engine, presented in [7], comprises the same activities we have in Task B. The results for the Jigsaw engine are presented in the last column of Table 2. The Jigsaw engine achieves higher recognition accuracies as our system, but the two sets of accuracies are, of course, not directly comparable as the training and testing data sets are different.

It was reported that, for one classification, the Jigsaw system processes 4 seconds of accelerometer data in 5ms on a Nokia N95 mobile phone and slightly less on an iPhone [7]. Our system goes through 30 seconds of accelerometer data at the same sampling rate in approximately 2.2ms on a Nokia E7 mobile phone. Both the Nokia N95 and Nokia E7 have an ARM 11 CPU, but the E7 runs at the clock speed 680MHz vs. the 330MHz of the N95. Taking this into account, we can approximate that processing of 30 seconds of accelerometer data would take roughly 4.5ms for our system on the N95, which is less than it takes for the Jigsaw to process 4 seconds of accelerometer data.

Table 4. Recognition accuracies and model set sizes for GMMs and qGMMs with various quantization setups.

Quantization	No. Gauss.	Accuracies W / CA Acc.	Model set size / kB
none	16	89.3 / 72.0	6.4
5m3v5f	16	90.5 / 72.6	18.9
4m2v4f	16	87.0 / 70.3	4.5
3m1v3f	16	81.5 / 70.2	2.6
None	1	89.9 / 71.4	0.7
5m3v5f	1	90.0 / 72.2	17.0
4m2v4f	1	90.4 / 70.7	2.6
3m1v3f	1	88.6 / 67.8	0.7

Table 5. Recognition accuracies and computation times for a selection of classifier parameter sets.

Models	No. Gauss.	Accuracies W / CA Acc.	Computation time Probs.	Total
GMMs	16	89.3 / 72.0	1.61ms	3.21ms
qGMMs	16	90.5 / 72.6	0.77ms	2.37ms
GMMs	4	91.0 / 72.9	0.55ms	2.15ms
qGMMs	4	90.5 / 73.4	0.42ms	2.02ms
GMMs	1	89.9 / 71.4	0.29ms	1.89ms
qGMMs	1	90.0 / 72.2	0.32ms	1.92ms

An interesting observation was made when running this test. When running experiments for Task A, increasing the number of Gaussians per model from one was not found to increase the recognition accuracy (see Table 5). However, for Task B, our system achieved higher recognition rates when using 16 Gaussians per model than when using only a single Gaussian for each model (91.3% vs. 87.6% class average accuracy). In this case, the use of qGMMs instead of GMMs is justified as the computation time for the probability calculation time is decreased from 1.1ms to 0.6ms.

Finally, we tested our system versus an easily implementable approach using time-based features and a k-NN classifier. We calculated the mean, variance, mean crossing rate and maximum absolute difference for each frame of accelerometer data to be used as the features. The best accuracies with this system was achieved using a frame length of 16 samples, frame skip of 8 samples, 5 nearest neighbors and 250 exemplars per activity. The accuracies for this system are shown in the second column of Table 2.

7. CONCLUSIONS

The experiments presented in this paper show that activity recognition using cepstral coefficients derived from a logarithmically scaled filter bank and qGMMs can be run efficiently on a mobile phone. It was shown that for cepstral coefficient calculation, the use of a filter bank with logarithmically rather than linearly spaced band center frequencies results in higher accuracies. In addition, qGMMs were found to give the same performance as GMMs. Computation time was shown to be equal or faster for qGMMs when compared to GMMs, depending on the recognition task and the complexity of the models. The memory requirements were shown to be higher for qGMMs than GMMs, especially when very simple, single Gaussian models were used.

8. REFERENCES

[1] E. Miluzzo, N. Lane, K. Fodor, R. Peterson, H. Lu, M. Musolesi, S. Eisenman, X. Zheng, and A. Campbell. “Sensing meets mobile social networks: The design,

implementation and evaluation of the CenceMe application," *In Proceedings of SenSys08*, ACM New York, NY, USA, 2008, pp. 337–350.

[2] N. Ravi, N. Dandekar, P. Mysore, and M. Littman, "Activity Recognition from Accelerometer Data," *in Proc. Seventeenth Conference on Innovative Applications of Artificial Intelligence*, Pittsburgh, PA, USA, 2005, pp. 1541-1546.

[3] T. Huynh and B. Schiele, "Analyzing features for activity recognition," *in Proc. Conf. Smart Objects and Ambient Intelligence*, Grenoble, France, 2005, pp. 159–164.

[4] J. Pansiot, D. Stoyanov, D. McIlwraith, B. Lo, and G.Z. Yang. "Ambient and Wearable Sensor Fusion for Activity Recognition in Healthcare Monitoring Systems." *In Proc. BSN 07*, 2007, pp. 208-212.

[5] N. C. Krishnan and S. Panchanathan, "Analysis of low resolution accelerometer data for continuous human activity recognition," *in Proc. Int. Conf. on Acoustics, Speech and Signal Processing,* Las Vegas, NV, USA, 2008, pp. 3337-3340.

[6] J.R. Kwapisz, G. M. Weiss, and S.A. Moore. "Activity recognition using cell phone accelerometers," *in Proc. Fourth International Workshop on Knowledge Discovery from Sensor Data*, 2010, pp. 10-18.

[7] H. Lu, J. Yang, Z. Liu, N. D. Lane, T. Choudhury, and A. T. Campbell, "The Jigsaw continuous sensing engine for mobile phone applications," *in Proc. 8th ACM Conference on Embedded Networked Sensor Systems*, New York, NY, USA, 2010, pp.71-84.

[8] L. Ming, V. Rozgić, G. Thatte, L. Sangwon, A. Emken, M. Annavaram, U. Mitra, D. Spruijt-Metz and S. Narayanan, "Multimodal Physical Activity Recognition by Fusing Temporal and Cepstral Information," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol.18, no.4, Aug. 2010.

[9] R.K. Ibrahim, E. Ambikairaajah, B.G. Celler and N.H. Lovell, "Linear predictive modeling of gait patterns", *in Proc. Int. Conf. on Acoustics, Speech and Signal Processing*, Taipei, Taiwan, April 2009, pp. 425-428.

[10] J. Lester, T. Choudhury, G. Borriello, "A Practical Approach to Recognizing Physical Activities", *in Proc. PERVASIVE 2006, LNCS 3968*, 2006, pp. 1-16.

[11] A. Eronen, V. Peltonen, J. Tuomi, A. Klapuri, S. Fagerlund, T. Sorsa, J. Huopaniemi, "Audio-based context recognition", *IEEE Trans. on Audio, Speech, and Language Proc*., Vol. 14, No. 1, pp. 321-329, Jan. 2006.

[12] S.B. Davis, and P. Mermelstein (1980), "Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 28(4), pp. 357–366.

[13] M. Vasilache, "Speech Recognition Using HMMs with Quantized Parameters," *in Proc. Int. Conf. on Spoken Language Processing*, Beijing, China, 2000, vol.1, pp. 441-443.

[14] M. Vasilache, J. Iso-Sipilä and O. Viikki, "On a Practical Design of a Low Complexity Speech Recognition Engine", *in Proc. Int. Conf. on Acoustics, Speech and Signal Processing*, Montreal, Quebec, Canada, 2004, vol. 5, pp. 113-116.

[15] S. Young, D. Ollason, V. Valtchev, and P. Woodland, *The HTK Book (for HTK Version 3.4)*, Cambridge University Engineering Department, 2006.

[16] J. Max, "Quantizing for Minimum Distortion," *IRE Transactions on Information Theory*, vol. 6, Mar. 1960.

[17] S. P. Lloyd, "Least Squares Quantization in PCM," *IEEE Transactions on Information Theory*, vol. 28, Mar. 1982.