Context-aware system for recongnizing daily-activities using smartphone sensors

Abstract

In recent decades, human activity recognition has been the subject of an important amount of research which enabled many applications in different areas, such as time management, healthcare and anomaly detection. Most of those works were based on using multiple special sensors and few address complex activities. In order to solve those issues, we propose a context-aware system based on the combination of ontological reasoning, GPS mining using k-nearest neighbors, and statistical recognition model using cascade neural networks. We first present some complex activity recognition models and discus their limitations. A general architecture of our approach is then presented along with a detailed description of each section of the system. Finally, we will present the results obtained and discus the system?s limitation and ideas that need to be addressed in future work.

1. Introduction

Human Activity recognition (HAR) is commanding wide attention nowadays both in the research domain and in relevant industries due to its applications in Time Management, Rogue Behavior Detection [1] and indoor localization and tracking [2]. A significant amount of research has been done in the structural representation and recognition of human activities, most of which are based on the use of external fixed sensors, or mobile sensors which captures motion.

Surveillance cameras are a typical example of the fixed sensors. In fact, detecting activities from a video sequence has been widely studied [3]. Yet, this approach yields major problems. For instance, the firld of view might affect the recognition process significantly. Also, they are limited to a certain place in which they are fixed, and to a fixed set of motion activities only. Recent works focus on wearable sensors to overcome the fixed sensors limitations [4]. Yet, the type of activities that can be monitored are limited to a set of simple motion activities and not complexes activities like cooking, reading, working...

Embedding motion sensors in smart phones increases the interest in using smart phones for detecting user's activities. Therefor our main approaches make use of the set of sensors embedded in smartphone, as well as other connected objects (computer, smartwatch...) in order to improve the task to activity recognition. The use of smartphones yields new issues in the area of activity recognition. Sensors provided in smartphones are not as accurate as special wearable sensors, therefor requires an efficient system that deals with noisy data. Additionally, due to limited resources, the HAR system is limited by a set of constraints like battery draining and computational complexity.

This work consists of designing a simple, light weight, and accurate system that can learn human activity with minimum user interaction, which can be shared across multiple users regardless of their behaviors.

The system should be able to :

- handle noisy data which are gathered from smartphones sensors and other connected devices.
- distinguish between similar activities or/and activities that occur in the same context.
- handle new unseen labels (activities), and therefor provide a robust and personalized recognition for each user.
- handle complex structured data and process the recognition in real-time.
- take advantage of the limited resources available.

1.1. State of the art

There is an important amount of work regarding activity recognition using sensors, i limit this table to take under consideration only those who deals with complex daily activities such as (reading, cooking, eating) and not only locomotion activities such as (standing, sitting, walking ...)

Accepted. Junior Conference on Data Science and Engineering Paris 2016 (JDSParis16)

since they are proved to be much easier to recognize [5][6], and are not our main interest.

Bao & al [7] presented one of the first papers that dealt with complex activities, yet the set of activities to be recognized are choosed based on the location of multiple sensors in human body to get the best results. For instance, the sensors of arm and wrist help detect if the person is brushing his teeth. The proposed method, while it proved its effectiveness in this controlled scenarios, is not scalable to most of complex activities. Also, the number of special sensors and their placements make it not applicable in most cases, especially smartphone sensors case.

Kerem Altun & al [8] presented a model based on Bayesian Decision Making using different types of sensors signals. However, like the first paper, it deals with controlled activities that cannot be recognized without the set of sensors mounted in specific body areas.

Barbara Bruno & al [9] present a method of recognizing motion activities (sitting, standing, lying and walking) as well as three complex activities (eating, brushing teeth and using the telephone) using only the information gathered from one accelerometer sensor attached to the wrist. Even though the ambition of the work, they proved that using a single sensor to recognize activities can be a hard task especially when recognizing complex activities. Also that the accuracy can be lower when dealing with larger set of activities.

The papers mentioned above [7][8][9] share the same features that are laboratory controlled scenarios which cannot be extended to real life situations, and also that none of them take the temporal properties of an activity and the temporal relationship between activities under consideration. The other downsides are the set of complex activities to be recognized and the sensors used in the data gathering process which are limited.

2. Proposed Approach

In order to address the problem of activity recognition, we propose a context-aware system using an ontological reasoning combined with cascade neural network and GPS mining technic. The statistical recognition model (neural network) is also combined with a symbolic location ontology which allows the system to infer the current activity of the user among the candidate activities. We prove that this is not only useful to refine the model by taking under consideration only the context related activities, but also to deal with the cold start when the user don't have sufficient data to train our system with.

The main intuition of using the ontological reasoning is that the neural network infer the current user's activity by learning on raw data gathered from the set of smartphone sensors (Accelerometer, GPS, light...). But since it might be difficult to recognize the activity among a large set of possible activities, we use the ontological reasoning to refine the model by learning and recognizing only the set of activities of each context, for example: if the user never performs Gardening in the Kitchen, there is no need to consider it as a possible output in this context.

2.1. Data description

The signals received from smartphone sensors and the other connected devices (computer, smartwatch) describe the context that identifies the current user's activity. Which consists of:

Localization features: the Global Positioning System GPS provide an important information which identifies the location in which the user is currently in. This helps recognizing the possible current activity of the user, for example, if the user is currently in the Kitchen it's more probable that his current activity is Cooking rather than say for example Gardening (unless it's specified by the user and learned afterwards). Moreover, the place's name can be easily obtained using Foursquare or Google Places Web Service[10].

Acceleration features: the triaxial accelerometers are used in order to identify the locomotion activities (walking, running, vehicle ...) and trajectory recognition (train X, bus99...). However, those signals cannot be used on their own in identifying complex activities such as eating, studying ...etc. A method proposed by [11] uses multiple accelerometers attached in various boy locations such as arms and legs in the same time and use only those signals to identify complex activities. This method can't differentiate between similar activities like eating and tooth brushing for example. An extension would be to combine the signals of accelerometer and other sensors. He and al [12] show that the best position for the accelerometer is in the pocket in the waist. Where the idea of using a smartphone would be ideal.

Environment features: this type of signals helps describing the current context in terms of time, light, and ambient noise. For example, in an environment with low light and ambient noise is plausible that the actor is sleeping. However, these signals must be combined with other information in order to better describe the environment and distinguish similar activities (like GPS for example).

Other features: other signals can be gathered by the connected devices which helps recognizing current activity. For example, the computer sends the software name or the URL that the user is browsing (Example: if the user is browsing Facebook, his current activity might be Socializing).

2.2. Architecture

The following figure shows the ontological model that describes the representations of activities. As we can see, activities consist of two different sets: locomotion and complex daily activities.

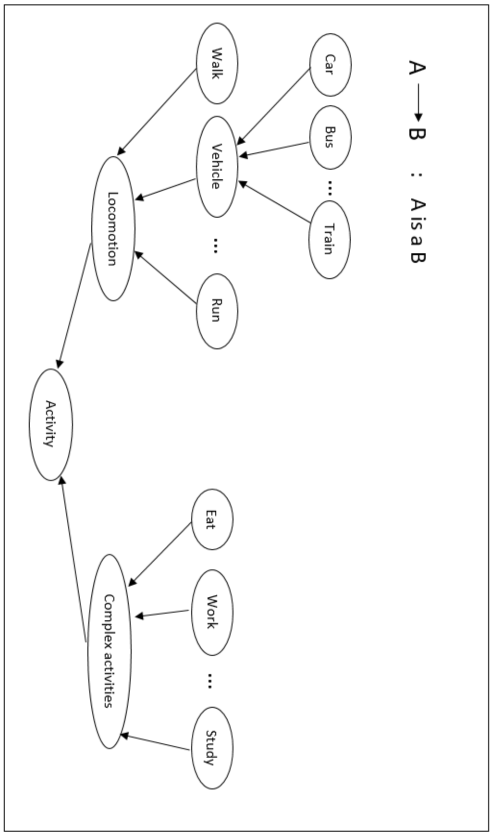


Figure 1. Ontological model

The former represents the set of activities that are recognizable using motion sensors features such as accelerometers, gyroscope, and GPS. The locomotion activities are detected by periodically waking up the device and reading short bursts of sensor data. It only makes use of the mentioned low power sensors in order to keep the power usage to a minimum. For example, it can detect if the user is currently on foot, in a vehicle, on a bicycle or still.

The later represents the set of complex daily human activities such as reading, eating, working, cooking...etc. In order to detect complex activities, a complex recognition model need to be used in order to be able to describe the context that helps recognizing current activity. Therefore, multiple sensors are combined together to generate data, and a non-parametric model is trained over those data.

3. Recognizing locomotion activities

Since this work does not focus on recognizing locomotion activities. The choice was set on using Google Activity Recognition API [13] in order to detect the current locomotion state of the user which are: Walking, Running, Bicycle, Vehicle and not moving.

Google Activity Recognition API is basically an Android interface that allows detecting the current motion stat of the user. This can be very important information since it can be used to detect the user's current activity between candidate activities. Moreover, if the current user motion is of type Vehicle, a trajectory recognition process will be used in order to recognize the type of vehicle the user is using.

This process is done in two steps, first by gathering different signals from the accelerometer, gyroscope and orientation sensors in a defined window of time. These signals are first processed in order to deal with redundant, missing and noisy information. Then, the processed data will be used as an input in Google API in order to predict current motion type.

3.1. Trajectory recognition

As we've seen earlier, the Google API is used only to get an important information about the user's motion state. If the current state is inVehicle, we use a trajectory recognition process in order to detect current trajectory and the type of vehicle the user is in (examples: Metro 99, Train X, Bus 176 ...etc.).

We address the task of trajectory recognition as a map matching problem, where each trajectory is represented by a sequence of GPS points ordered based on the temporal information. The proposed model is based on K Nearest Neighbors algorithm that considers (1) the spatial-temporal information and (2) the marginal velocity of each trajectory.

Many issues are not addressed in most of existing work (exp: [14][15]), such as high volume of trajectory points, high variance of points gathering frequency, missing points (due to network errors) and computational and special complexity optimization. In order to deal with the mentioned issues, we propose a feature mapping function Φ_1 that transforms the data representation from the GPS sequence space into R^{n*m} space. The new space is based on Military Grid Reference System coordinate system [16], where n and m are the height and width of the grid respectively. The new representation is used to process missing data using a linear interpolation, and improve accuracy using dilation (more on next section). Then, another feature mapping function $\Phi_2 : R^{n*m} \mapsto N^d$ will map the sparse grid representation into a simple vectorial representation, which will be used to optimize the recognition process.

3.1.1. Data structure

A trajectory is represented as an sequence of GPS points as follow $T_i = \{p_0, ..., p_k, distance_i, duration_i\}$, each point is defined as $p_k = (Lon_k, Lat_k, accuracy_k, t_k)$ which represent respectively: longitude, latitude, accuracy and timestamp of point k. Two trajectories might have different lengths, and therefore different number of GPS points.

3.1.2. First mapping function

We define a function $\Phi_1 : T_i \mapsto R^{n*m}$ as a feature map that maps the input defined as a trajectory into a new representation space in R^{n*m}.

The new representation is a sparse matrix where n and m are its height and width respectively, it is based on the Military Grid Reference System [16]. MGRS is a projected coordinate system which uses a 2-dimensional Cartesian horizontal position orientation, and can define a grid with square cells having the same defined length.

The feature map is defined as: $\Phi_1(T) = G$(G for Grid) where:

$$Gij = \{^{1:0<x_k-(i*m)<m, 0<y_k-(j*m)<m}_{0:otherwise}$$

The following figure shows the new grid representation based on a trajectory.

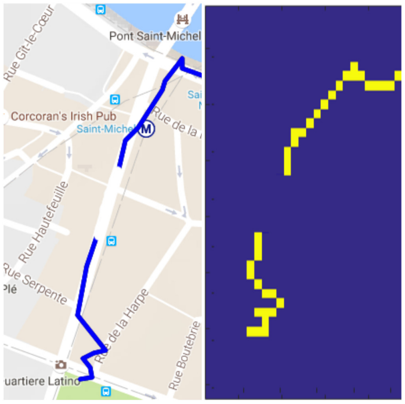


Figure 2. Left: the GPS trajectory, right: the new representation.

3.1.3. Interpolation

As we can see in the figure above, the grid representation cannot be used directly as final representation. The main reason is the gaps that are due to lack of geolocation points, this is a common problem that happens due multiple reasons such as lack of signal, when traveling fast, low recording frequencies and discarding points having low accuracy. In order to refine the new representation, a preprocessing step aiming to fill the gaps is used as follow: at each timestep t_i, we verify if the previously generated point (cell) at t_{i-1} is connected to the current point, if not we use the linear interpolation by defining by the line equation:

$$j = f(i) = \frac{j_2 - j_1}{i_2 - i_1}(i - i_1) + j_1$$

Where i_1 and j_1 are the coordinates of cell t_i and i_2, j_2 are the coordinates of the cell . The following figure shows the result of gap filling by interpolation.

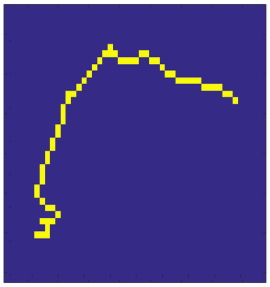


Figure 3. Result after Interpolation.

After the preprocessing step, the new trajectory looks more similar to the original one.

3.1.4. Dilation

One major and frequent inconvenient of using the MGRS representation is that by dividing the map into multiple cells, two close points can be put into two neighbor yet different cells, and therefore considered as two different areas since they're not regrouped in the same one. In order to solve this issue we apply a morphological dilation as follow.

We define a structuring element H (a 3x3 mask that allows defining arbitrary neighborhood structures), where:

$$H = [111; 111; 111]$$

The dilation of the grid G by the structure element H is given by the set operation:

$$G' = G \bigoplus H = \{(p+q) \| p \in G, q \in H\}$$

```
**Algorithm**:
Input: Grid representation G, Structure element H
Output: Grid $G' = G \bigoplus H$
1.init G' to a zero matrix
2.loop over all $q \in H$
2.1.    Compute shifter grid $G_q$
2.2.    Update $G' = G' \vee G_q$
```

The dilation operation is applied to the target trajectory only, the result of dilating a trajectory is shown in the following figure.

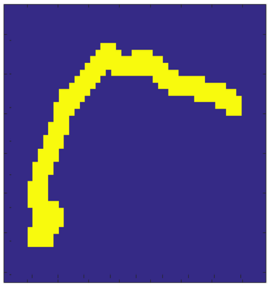


Figure 4. Result after dilation.

3.1.5. Second mapping function

The main utilities of the first transformation is to work with a more simplified representation, and perform different operations in order to refine the saved trajectory. Yet, this new grid representation yields a serious problem which is passing to a higher dimension representation defined by a sparse matrix. This issue needs to be solved since the complexity of KNN model increases with the dimension of data and the similarity measure used.

We define a mapping function $\Phi_2 : R^{n*m} \mapsto N^d$ where $d \ll (n*m)$ that takes as input a grid representation of a trajectory (after performing interpolation and dilation) and maps it into a vectorial representation using the following formula :

$$V_k = \{i*m+j \text{ , for all } G'_{ij} \neq 0\}$$

The compact representation keeps only relevant information and can be used in similarity measurement.

3.1.6. Similarity measurement

Choosing a similarity measurement formula is important to get accurate results. The Euclidian similarity is widely used in sequence comparison. Yet it cannot be applied in this context due to: first, not all trajectories have equal points, and second, the computational complexity of such models are not applicable due to the smartphone limitations. We propose the use of Jaccard index as a similarity, which is defined by:

$$Jaccard(Trajectory_A, Trajectory_B) = \frac{|T_A \cap T_B|}{|T_A \cup T_B|}$$

Instead of using the raw formula, we address the similarity problem as finding how much percentage the current trajectory shares with the target trajectory. Therefore, we use the symmetric version of Jaccard which is defined by:

$$Jaccard(T_A, T_B) = \frac{|\Phi_2(T_A) \cap \Phi_2(T_B)|}{|\Phi_2(T_A)|}$$

By mapping both trajectories using first and second feature maps functions, we calculate the intersection using a hashtable. Therefore, the complexity of the similarity measurement is linear in terms of *l* which represents the length of the shortest trajectory.

4. Recognizing complex activities

The second type of activities are complex daily human activities. Unlike locomotion activities, they are more complex to recognize due to: the amount of similarities between activities, none relevant (or useless) features used to recognize, amount of candidate activities and the limitation of using sensors of the smartphone only.

In order to address this problem, we propose a combination of none-parametric model (Cascade Neural Networks) with a symbolic location ontology. The main intuition behind this combination is that statistical models cannot be used alone to distinguish similar activities among a huge set of activities (about 100) based only on raw data. Hence, the use of the symbolic location ontology helps refine the model by selecting a small set of candidate activities based on the correspondent context, and therefore infer the right activity.

An example would be to distinguish between Eating and tooth brushing, similar activities based on sensors yet different based on the context. Another example would be if the user is in the kitchen, there is no need to add Office activities to the list of candidate activities. The symbolic location ontology is therefore defined as follow:

As we can see, a location can be either an outdoor location or an indoor location. The former represents the set of familiar locations such as work place and home that might include multiple rooms, while the latter represents the set of foreign location and public places such as Libraries, restaurants, Hospitals ... etc.

In order to distinguish between the two types of location, we use GPS signals. However, GPS signals can be used only to determine if the current location of the user is indoor or outdoor, and cannot be used to recognize in which room he is due to the GPS precision level. In order to solve this issue an already implemented module is used to recognize room-level current position based on Wi-Fi fingerprint [17].

The context-aware activity recognition process is done in two different steps using the defined symbolic location. Which are:

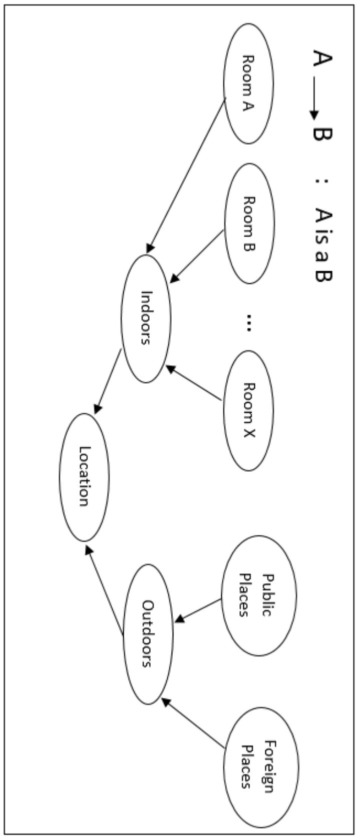


Figure 5. Symbolic location

4.1. Outdoor activities

Depending on the user mobility state: if the user is moving, the previously defined locomotion and trajectory recognition is used in order to predict the user's activity. Otherwise, if the user is immobile (standing still, sitting...) and a foreign place, we predict current activity based on the place name. The name is obtained using Foursquare (an alternative would be using Google Places Web Service[10]).

A natural language processing technic is used in order to predict the most probable activity based on the place type, in which, the place's name is passed to the system as a query, and the goal is to predict the place's type and hence infer the more probable activity of the user. For example, if the place's name is McDonalds, the NLP system will return Restaurant as the type of the place, and therefore we infer the activity eating out.

The models used in the process of information retrieval and text mining are based on a mapping process of a user's query and the set of documents present in a corpus. The search mechanism determines, based on a supposed relevance of documents, those that meet the need of the user. This part of work consists on developing a search engine capable of handling various requests gathered using GPS signals. In order to do so, a technique of text features extraction referred as TF-IDF has been used.

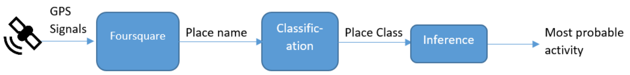


Figure 6. Outdoor activity recognition using text features extraction.

Tf-idf stands for term frequency-inverse document frequency, and the tf-idf weight is a weight often used in information retrieval and text mining. This weight is a statistical measure used to evaluate how important a word is to a document in a collection or corpus. The importance increases proportionally to the number of times a word appears in the document but is offset by the frequency of the word in the corpus. Variations of the tf-idf weighting scheme are often used by search engines as a central tool in scoring and ranking a document's relevance given a user query. [18]

The choice of using this technique comes from the set of its advantages, notably:

- Reduces a corpora of arbitrary length to a sequence of numbers, for each word we maintain tf and idf values.
- Very easy to computer query-document or document-document similarity.
- Can be successfully used for stop-words filtering and query classification.

4.1.1. Indexation

The first step in the process of implementing the search engine, is the generation of the inverted-file. In order to do so, we've first started by generating the corpus that will be used by the engine and that is by assigning to each document a class ,this represents "in our case of work" a class of common public place (examples: restaurants, study place, work place ...). Each of these documents contains the set of key words, context sentences and words in the same lexical field of the correspondent class.

The next step is to collect the set of normalized words in each document. The normalization is done using the snowball tokenizer by keeping only the radical form of words, and that is the form of a word after any prefixes and suffixes are removed, minus the stop-words in order to take only relevant words under consideration. This results in a reduced set of words that represent the vocabulary used in the corpus.

A tree-map is created first by analyzing the set of documents and then by collecting, for each word in the vocabulary, the set of documents in which it appears as well as its frequency in each document.

4.1.2. Vector space mapping

The vector space mapping allows to represent queries and documents in a high-dimensional space. The first step is to generate the weights files of each document using the tf-idf technique. The tf-idf weight is composed by two terms: the first computes the normalized Term Frequency (TF), aka. the number of times a word appears in a document, divided by the total number of words in that document; it measures how frequently a term occurs in a document. Since every document is different in length, it is possible that a term would appear much more times in long documents than shorter ones. Thus, the term frequency is often divided by the document length (aka. the total number of terms in the document) as a way of normalization:

TF(t) = (Number of times term t appears in a document) / (Total number of terms in the document).

The second term is the Inverse Document Frequency (IDF), computed as the logarithm of the number of the documents in the corpus divided by the number of documents where the specific term appears. It measures how important a term is. While computing TF, all terms are considered equally important. However it is known that certain terms, such as ”is”, ”of”, and ”that”, may appear a lot of times but have little importance. Thus we need to weigh down the frequent terms while scale up the rare ones, by computing the following:

IDF(t) = log_e(Total number of documents / Number of documents with term t in it).

Each document is now represented as a real-valued vector of tf-idf weight$\in R^V$ V: vocabulary , so we have $|V|$- dimensional real-valued vector space where terms are axes of this space and documents are vectors of this space.

The use of vector space model with tf-idf weighing formula helps bypassing the three following problems

- Avoid the ”You’re either in or out” Boolean model.
- A term that appears in many documents should not be regarded as more important than one that appears in few documents
- A document with many occurrences of a term should not be regarded as less important than a document with few occurrences of the term

Once the place name is send, it is important to recognize the class in which this name belongs. Although using TF-IDF technique helps representing documents accurately, we need an efficient algorithm that calculates the similarity between thequery and the set of documents with high accuracy.

The idea is to be able to rank documents according to their proximity to the query, and to rank relevant documents higher than non-relevant documents as well.

The use of the Euclidian similarity measurement would not be a good idea since the Euclidian distance is large for vectors of different length which is in our case. Instead, by using cosines similarity measurement, the documents will be ranked according to the angle with the query in decreasing order, since Cosine is a monotonically decreasing function of the angle for the interval$[0, 180]$. In other words, the similarity between *a* query *q* and a document *d* equals to 1 only if $d = q$, and equals 0 if *q* and *d* shares to terms.

In order to improve the search performance, the query is represented as a vector in the high-dimensional space in the same manner of documents representation. That is, by using the same preprocessing technique which include removing stop-words and normalizing terms using the same tokenizer. The terms in the query are then weighted using TF-IDF formula, the first time in the formula ’TF’ is calculated directly from the query, while the second term IDF is loaded from the inverted file previously generated.

For optimization purposes, we first select a candidates classes that contains at least a term of the query, then cosine similarity is calculated between each candidate class and the query using the following formula:

$$Sim(d,q) = Cos\Theta = \frac{\overrightarrow{d}.\overrightarrow{q}}{||\overrightarrow{d}||||\overrightarrow{q}||}$$

A sort by fusion algorithm is used to maintain low computational complexity of the system.

4.2. Indoor activities

We address the complex indoor activity prediction as a supervised recognition problem. We present in the next section data description and the different parameters used by the model.

4.2.1. Data structure

The set of raw data values are represented as signals emitted by sensors embedded in the smartphone and/or other connected objects such as computers. These are time-related, hence the problem of recognition can be defined as follows:

Having a set $T = \{T_0, .., T_{m-1}\}$ where the set of T_i represent Timeslots (time windows) of different sizes. And a set $A = \{A_0, .., A_{n-1}\}$ of possible activities. The goal is to find a function $f : T \mapsto A$ that approximates the real activity performed during T_i.

4.2.2. Model Architecture

Neural networks are widely used due to their adaptation ability to unexpected inputs, and ability to work directly with raw data. In this work, and since we're limited by the smartphone's capacities, deep neural networks or other complex architecture cannot be used because of the back-propagation step. We propose using a cascade correlation architecture also known as Cascade Neural Networks. Which works as follow:

- Start with a minimal network containing only fully connected input and output layers.
- Hidden units will be added one at a time if necessary during learning phase. Once a unit is added, its input side weights are fixed and connected to every input and hidden unit. Hidden units are used to recognize various specific patterns, more units can be added to recognize more complex patterns.

Unlike traditional architectures, cascade correlation architecture can provide three major benefits which are:

- Dynamic architecture: instead of a fixed one, cascade neural network change its size and architecture when training.
- Fast learning: since they don't require back-propagation step, the learning process is much faster.
- Incremental learning: helpful when adding additional information (signal from a new sensor for example) to pre-trained networks.

4.2.3. Model Evaluation

The evaluation of the recognition process depends on the model itself [19]. The model can be in two forms: the first one is called subject-independent in which a single model is created for all users, and then trained over their whole data combined. The evaluation is done in a cross-validation fashion by learning on all users and leaving K users out. However, this method does not take the user's characteristic under consideration which will be an issue for recognizing (for example sport activities have different signal distributions for younger users compared to older users). The second model type is called subject-dependent in which a model is created and tested for each user by using his own data. The evaluation is given as the mean of all precisions obtained for each user. Since out model is local and specified for each user, we have chosen the second type of evaluation.

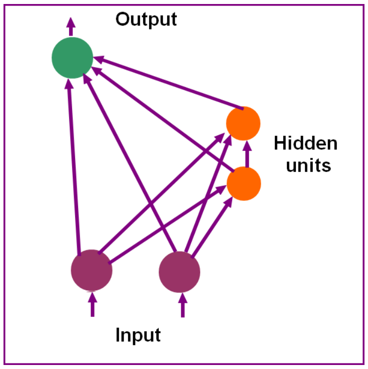


Figure 7. Cascade correlation architecture with two inputs, two hidden units and one output

4.2.4. Parameters Initialization

The initial state of parameters influence significantly the rate of convergence to the optimal solution, and the nature of the solution (local or global). Hence the importance of choosing the right initialization method in the learning process.

An initialization method following a normal distribution is widely used I neural networks. Unlike random initialization, this probability distribution describe theoretically the random factor of the random experiment. The standard normal distribution centered at 0 and reduced to standard deviation equals to 1 is mostly used, which is defined by:

$$N(\mu, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

Using this method will help reducing the convergence time considerably. However, the initialization is fixed independently of the neurons active regions, which can cause the problem of falling into a local and not a global solution.

In order to deal with this issue, another initialization method is used. This is proposed by Nguyen and Widrow [20] which generates initial weight and bias values such that the active regions of neurons in one layer are distributed over the input space.

The main idea of the initialization algorithm is as follow:

- Take a small set of random numbers and consider them as the initial weights.
- The weights are then modified such that the region of interest is divided in a set of small intervals;

- Set the initial weights in the input layer so that each neuron will be assigned its own interval;
- Each hidden neuron has the option to adjust the size of its interval during learning. However, most of these changes will be negligible because of the initialization phase of the algorithm that eliminates most of the possible values.

Using this initialization method helps reaching the global optimal solution in a reasonable time.

4.2.5. OVERFITTING

One major issue we had to deal with in the training process is overfitting. This phenomena happens due to multiple raisons such as the model complexity, data distribution...etc. One way of reducing the model's complexity is to limit the number of candidates units in cascade-correlation networks. However, even though limiting candidate neurons reduce the complexity, it does not favor less complex models. In order to deal with this issue to by adding a second term to the loss function, called regulizer. The evaluation function is then defined as follow:

$$\Sigma_i^n L(f(T_i), A_i) + \lambda R(\Theta)$$

Where the first term measures how far current prediction is from the real activity. While the second term penalizes a complex network. The loss function is defined as the mean squared error MSE:

$$L(f(T_i), A_i) = \Sigma_i^n (f(T_i) - A_i)^2$$

The outputs are represented as a binary vector, which have value$=1$ for the real output activity and 0 for all other activities.

5. Results

5.1. Raw data collection

The method used to collect data is fairly important, Foester and al. [5] showed that the accuracy of activity recognition models dropped from 95% to 66% when using a natural data acquisition process instead of the one controlled in a laboratory. In our case, the obtained results are conducted using data gathered from three different users practicing daily activities in different contexts. The following table presents data distribution :

Data	Phone used	Nbr instances	Nbr activities
1	Nexus 5	112275	$\simeq 101$
2	Nexus 5	118453	$\simeq 101$
3	Honor 7	15326	$\simeq 101$

The complex activities can be grouped into different classes as follow:

Group	Examples of activities
Locomotion	Stand, Walk, Run
Transportation	Bicycle, Vehicle, Train, Bus ...
Daily Activities	Eat, Study, Watch TV, Work ...

5.2. Data processing

The choice of where to process data is important, two cases are present: processing on the server or directly on smartphones. In the former, servers offer an important capacity in terms of computational and special complexity. While in the latter, processing data directly on smartphones helps limit the quantity of data to be send to server. The other interest is working with robust data that are not affected by the transmission noise bias. Nevertheless, the activity recognition on smartphone is a more difficult task since smartphones are limited by the computational and special complexity. Moreover, the signals received by the sensors are prone to noise where the necessity of having a robust data processing methods. In our case of study, we considered processing data and predicting activities directly on smartphones.

5.3. Evaluation

The recognition model is evaluated using in a subject-dependent matter, in which a model is trained and tested for each user independently using his own data. The model's precision is then considered as the mean precision.

We first started by performing a dry run using Decision tree model (DT) and Support vector machines (SVM), the results are then compared to the proposed model in term of precision. In order to improve the models precision, we run experiments using cross validation. The method used in cross validation is called K-fold with K equals to 6 in our case, and that is by dividing data into 6 subsets, we train our models 6 times. Each time, one of the 6 subsets is used as the test data and the 5 other subsets are combined to form the training data. The average error across all 6 runs is returned.

We evaluate the activity recognition results presented in figure 8. The Decision tree results were obtained by using a model having maximum depth set to 30, minimum samples leaf set to 5. While the SVM results were obtained using a polynomial kernel of third order. The set of parameters were chosen based on the results obtained using cross validation.

For Cascade neural network, the activation function used in this experiment is symmetric sigmoid. This function is bipolar since it outputs data in the range [-1, 1], unlike bi-

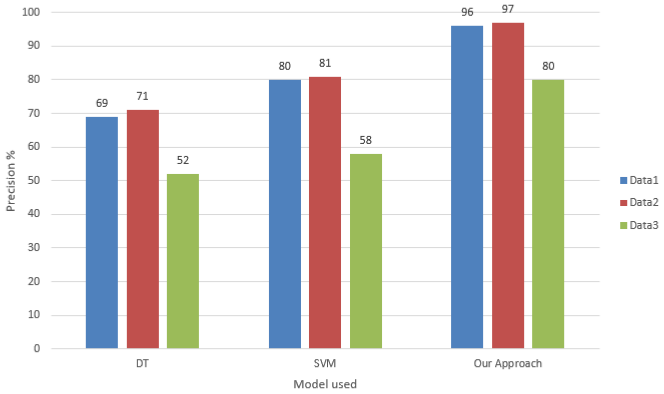
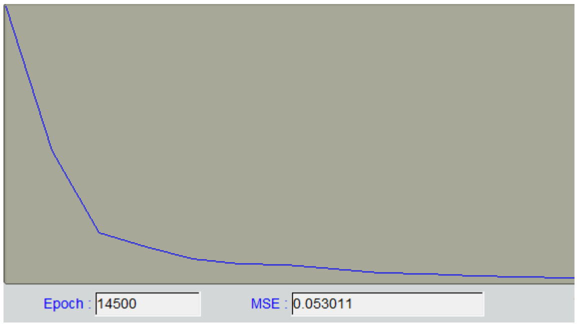


Figure 8. Comparaison between different models for activity recognition using different data sets.

nary functions that outputs in range [0, 1]. This will limit the impact of the vanishing gradient caused by unit values close to 0. The maximum condidate unit to add are fixed to 30 units.



Epoch : 14500 MSE : 0.053011

Figure 9. Evolution of MSE using our model with Data1.

The following figure (9) represents the evolution of the mean square error rate of test data using our model on Data1. In this configuration, the cascade neural network have the same parameters values defined above, with learning rate equals to$\alpha = 0.01$. the value of α is choose using cross validation from a set of values starting from 0.1 and gradually decreased by a factor of $10^{-}1$ until reaching value $10^{-}5$.

The weights initialization is done using Nguyen and Widrow [20] and the regularization is done using norm L2 withλ that is choose empirically. Even though increasing its value will reduce overfitting by reducing the variance of classification, it will also cause adding bias to the estimation. In order to compromise between bias and variance, we used multiple runs with different λ values ranging from 0.1 to 0.7 and the best value choose using cross validation.

6. Conclution and perspectives

In order to solve the problem of complex activity recognition using smartphones, we first reviewed some papers and discuss their limitations. We considered studies that use only wearable sensors and deal with complex activities. Some of those limitations are (1) laboratory controlled (recognizing the activity Eating by using multiple sensors on the arm) and cannot be extended to real life situations, (2) not context-aware systems, hence cannot be used to distinguish between similar activities (models not flexible).

In the interest of unravelling those issues, we proposed a context-aware system based on the combination of ontological reasoning, GPS mining using k-nearest neighbors, and statistical recognition model using cascade neural networks. The results show that by using the context-aware system, the problem of recognizing complex activities becomes simpler and that is by refining the model using ontology reasoning, and that's by considering smaller set of candidate activities based on the context.Then Combining different techniques for different contexts in order to recognize activities.

This system can be further improved by exploiting the important temporal properties of activities, which include the activity duration and sequential information. Flexibility of the model can also be improved by creating a monotonic model that works for each new user instead of the actual user specific model. This will lead to user profiling hence the actual model can be seen as a profile specific model.

Acknowledgements

To each and everyone who have contributed to this humble work.

Bibliography

[1] Rogue Behavior Detection: Identifying Behavioral Anomalies in Human Generated Data. 2014 Numenta and Grok.
[2] Radu, V.; Marina, M.K., HiMLoc: Indoor smartphone localization via activity aware Pedestrian Dead Reckoning with selective crowdsourced WiFi fingerprinting, Indoor Positioning and Indoor Navigation (IPIN), 2013 International Conference on , vol., no., pp.1,10, 28-31 Oct. 2013 doi: 10.1109/IPIN.2013.6817916.
[3] M. S. Ryoo, Interactive Learning of Human Activities Using Active Video Composition, International Workshop on Stochastic Image Grammars (SIG), in Proceedings of International Conference on computer Vision (ICCV), Barcelona, Spain, November 2011.
[4] Oscar D. Lara and Miguel A. Labrador, A Survey on Human Activity Recognition using Wearable Sensors,

IEEE Communications Surveys and Tutorials,2013, 1192-1209

[5]: B. Longstaff, S. Reddy, and D. Estrin, Improving activity classification for health applications on mobile devices using active and semi-supervised learning, in Pervasive Computing Technologies for Healthcare (PervasiveHealth), pp. 17, 2010.

[6] : L. Bao and S. S. Intille, Activity recognition from user-annotated acceleration data, in Pervasive, pp. 117, 2004.

[7] : Bao and S.S. Intille. Activity recognition from user-annotated acceleration data. In Pervasive Computing, 2004

[8] : Kerem Altun and Billur Barshan. Human activity recognition using inertial/magnetic sensor units. In Human Behavior Understanding, 2010.

[9]: Barbara Bruno, Fulvio Mastrogiovanni, Antonio Sgorbissa, Tullio Vernazza, and Renato Zaccaria. Analysis of human behavior recognition algorithms based on acceleration data. In IEEE International Conference on Robotics and Automation, 2013.

[10] ”Google Places API,” http://code.google.com/apis/maps/ documentation/places/.

[11]U. Maurer, A. Smailagic, D. P. Siewiorek, and M. Deisher, ”Activity recognition and monitoring using multiple sensors on different body positions,” in Proc. International Workshop on Wearable and Implantable Body Sensor Networks, (Washington, DC, USA), IEEE Computer Society, 2006.

[12] Z.-Y. He and L.-W. Jin, ”Activity recognition from acceleration data using ar model representation and svm,” in International Conference on Machine Learning and Cybernetics, vol. 4, pp. 2245-2250, 2008.

[13] ”Activity Recognition API”, https://developers.google.com

[14] F. Angiulli and F. Fassetti. Dolphin: An efficient algorithm for mining distancebased outliers in very large datasets. ACM-TKDD, 3(1), 2009.

[15]. Y. Bu, L. Chen, A.W.C. Fu, and D. Liu. Efficient anomaly monitoring over moving object trajectory streams. In Proc. KDD, 2009.

[16] Military grid reference system, https://en.wikipedia.org/wiki/Militarygridreferencesystem

[17] Yan Luo, Orland Hoeber and Yuanzhu ChenEmail author Enhancing Wi-Fi fingerprinting for indoor positioning using human-centric collaborative feedback. Human-centric Computing and Information Sciences2013

[18] Term Frequency-Inverse Document Frequency), http://www.tfidf.com/

[19] E. M. Tapia, S. S. Intille, W. Haskell, K. Larson, J. Wright, A. King, and R. Friedman, ?Real-time recognition of physical activities and their intensities using wireless accelerometers and a heart monitor,? in Proc. International Symposium on Wearable Computers, 2007

[20] Derrick Nguyen and Bernard Widrow. Improving the learning speed by choosing initial values of the adaptive weights. Stanford university CA 94305