Available online at www.sciencedirect.com
ScienceDirect
Procedia Computer Science 52 (2015) 310 – 317
Procedia
Computer Science

The 6th International Conference on Ambient Systems, Networks and Technologies
(ANT 2015)

Using accuracy analysis to find the best classifier for Intelligent Personal Assistants

Ricardo Ponciano[a], Sebastião Pais[a], João Casal[a]

[a] TIMWE I&D, Parkurbis, Covilhã 6200-824, Portugal

Abstract

An *Intelligent Personal Assistant* (IPA) is an agent that has the purpose of helping the user with his daily tasks. This paper is focused on IPAs for Internet of Things (IoT) environments. In this sense, a good IPA has the capability of surveying his user behaviour and suggest tasks or make decisions with the intention of simplifying the user interaction with his surroundings. With this in mind, this paper focuses on studying the accuracy of various classifiers, with the objective of finding the one that suits better the needs of an IPA for IoT. The aim is to test each algorithm with a dataset of events, that relate to past behaviours of the user, and find if there is an opportunity to notify the user that he/she may want to take an action or create an automation based on the learned behaviour.

Peer-review under responsibility of the Conference Program Chairs
Keywords: Intelligent Personal Assistant; Internet of Things; WEKA; Accuracy; Precision; Recall; Naive Bayes; J48; Decision Tree;

1. Introduction

In a world more and more dominated by technology, a natural process of change potentiates the emergence of systems that aid on, and even automate, tasks of a given user. Therefore, today, when our daily lives are more hasty than ever before, it is natural the need of technology aided management of every type of tasks, even on real world environments.

A solution that is emerging on the last few years proposes the use of intelligent assistants, best known as Intelligent Personal Assistants (*IPA*). These systems can study the user's behaviour on a long term, and aid on given tasks, using the knowledge learned from that study. For example, if the user wakes up every day at 8AM, the system can learn to set the alarm every day to 8AM.

This way, it is necessary to create a system that can perceive what the user defines as a task, and analyse the environment to get every type of information that can lead the user to take that behaviour. That can be achieved with

* Corresponding author. Tel.: +351-962-312-240
E-mail address: joao.casal@timwe.com

Peer-review under responsibility of the Conference Program Chairs
doi:10.1016/j.procs.2015.05.090

feedback of the interaction between the user and the IPA, or through his direct (or indirect) interaction with certain objects that the IPA can access to get that information.

So, if the user turns on the light every day when he wakes up (in our example, at 8AM), and at the same time there is a sensor that gives the information that the room is dark, the system needs to be capable of discriminate that the user woke up because it was 8AM, and that he turned on the light because it was dark. A good system can (and needs to) discriminate the right information that led to the task.

Our solution, within the *AMBRO* project (section 7), to this problem is an IPA that interacts with several objects present in the user's surroundings. This system allows the user to perform his daily tasks while it gathers information from the environment that it integrates. Our system architecture and its main features are described in[2 3].

However, for our system to work as envisioned, it is necessary to implement and integrate an intelligent control algorithm that can achieve the necessary requirements. So, before we can do this, we need to decide what will make the algorithm accomplish this task. With this is mind, *WEKA toolbox* (ver. 3.6.11)[4] was used to perform tests using the various classifiers available in order to ascertain which one could achieve the best performance. Moreover, each classifier was tested with various subsets of the main *dataset* to see the effects of its size on the performance results.

2. Related Work

Some work has been previously done on this subject, to test the performance of some classifiers tested in this paper, but in different situations. Langley et al[1] studies the Bayes classifier on a monotone conjunctive target concept. In[5], Langley et al examine previous works on the Naive Bayesian classifier and review its limitations. Bui et al[6] investigate the performance of three approaches (Naive Bayes, Decision Tree, SVM) on models for spatial prediction of landslide hazards in Vietnam. Daniela Xhemali et al[7] test three types of classifiers (Naive Bayes, Neural Networks and Decision Tree) for automatic analysis and classification of attribute data from training course web pages. Hashemi et al[8] also tested Naive Bayes and J48 for text classification, while Kotsiantis et al[9] reviews the performance of various classification techniques (Naive Bayes, Neural Networks, Decision Trees, kNN, SVM and Rule-learners). The same way,[10] compare the results of classification and combining techniques (C4.5, NB, 3NN, Ripper, SMO, BP).

On the field of IPAs, Maes[12] studies the interaction between an user and the IPA. Atzori et al[13] analyses the paradigms of the Internet of Things. In[14], Miorandi et al reflects on the applicability of IoT technologies.

3. Data Corpus

3.1. Data Collection

An important step on this study, is the collection of the right data that can be used on the process. The ideal collection would be composed of information from real users during a large span of time, where those users would be using the system on a normal daily routine, and where the IPA could gather real data, that could be used on the training and testing phase. The problem with this approach is a question of logistics: it would be impractical to wait months to gather enough data that could be used for the intentions of this study. Besides that, the same experiment should be performed multiple times to generate different *datasets* (although on this paper, we only focus on the results of one *dataset* and of several partitions of it).

To overcome this issue, a behaviour simulator was created so that in the presence of a set of rules and tasks, that could define the behaviour of a real user, it will create a set of events related to that behaviour. In this context, we understand an "event" as the *feedback* gathered at the moment that a task was performed by the user. Using the example previously described, when performing the task of waking up every day at 8AM, the user is creating an event that contains every information that the IPA can gather on the user interaction, as well as environmental data recovered by the objects being used (being that event associated with the "Wake up at 8AM" task).

3.1.1. Events and their structure

As stated before, an event is the fulfilment of a task by the user, and can be thought as a behaviour of that user. As can be seen in table 2, each event consists of a set of attributes (*feedback* gathered by the IPA on the various