An Agent based Platform for Resources Recommendation in Internet of Things

Agostino Forestiero\textsuperscript{a}\textsuperscript{b} and Giuseppe Papuzzo\textsuperscript{a}\textsuperscript{b}

Institute for High Performance Computing and Networking, National Research Council of Italy, Via P. Bucci 8-9C, Rende (CS), Italy

Keywords: Multiagent Systems, Internet of Things, Word Embedding, Recommendation Systems.

Abstract: Internet of Things (IoT) paradigm aims to bridge the gap between physical and cyber world allowing a deeper understanding of users in terms of preferences and behaviors. User devices and services interact and maintain relations which need of effective and efficient selection/suggestion approaches to better meet users’ interests. Recommendation systems provide a set of significant and useful suggestions for users and systems with given characteristics. This paper introduces the design of an agent based platform for building a distributed recommendation system in IoT environment. Internet of Things objects (devices, sensors, services, etc.) are represented with vectors obtained through the Doc2Vec model, a neural word embedding approach able to capture the semantic context representing documents in dense vectors. The vectors are managed by cyber agents that, performing simple and local operations, organize themselves exploiting the vector values. The outcome is the emergence of an organized overlay-network of cyber agents that allows to obtain an efficient recommender system of IoT services. Prelimaries results confirm the validity of the approach.

1 INTRODUCTION

Several objects interconnected with each other achieving a common purpose represent the Internet of Things (IoT) paradigm (Atzori et al., 2010). The number of objects and the amount of data generated by IoT infrastructures are growing hugely and this makes the traditional manage mechanisms inadequate. Intelligent and automated approaches are needed to support decision makers due to the dynamic nature of smart objects, devices and services, involved in the IoT systems. Systems able to perform “things recommendation” is a crucial step to take full advantage of the IoT. Recommendation systems are an important research topic and several works have been proposed both in the industry and academia. These systems allow to individuate a list of useful items for the users in a given context. The usefulness of an item or product or service is generally represented by a “rating”, which indicates how much a given user likes a particular item. The items with a high rating value are presented as recommendations for the user.

Recommendation systems can be categorized as (Balabanović and Shoham, 1997): (i) Collaborative Filtering (CF), an item is recommended to the user according to the past ratings of all users. The approach evaluates the utility of the item $i$ for the user $u$ by estimating the usefulness assigned to item $i$ by the users $v$ who are “similar” to user $u$; (ii) Content-based recommending an item is recommended if it is similar to items that the user has chosen in the past. Information retrieval (IR) techniques address this problem, where the content associated can be handled as a query, and the unrated documents marked with a similarity value to this query. Otherwise, the documents can be converted into word vectors, and then averaged to obtain a prototype vector of each category for a user, as reported in (Lang, 1995); and (iii) Hybrid approaches in which collaborative and content-based approaches are combined. The similarity between two users can be computed using various approaches, but the most popular are correlation (Resnick et al., 1994) and cosine similarity (Breese et al., 1998). Collaborative and content-based approaches use the same cosine measure from information retrieval. But, in content-based recommendation systems measures the similarity between vectors of weights, whereas, in collaborative systems measures the similarity between vectors of the actual ratings specified of the users. Other approaches to the recommendation consist in handling...
of the problem as a classification task. Each pattern represents the content of an item, and a user’s past ratings are used as labels for these patterns. For example, text from fields such as title, author, synopses, reviews, and subject terms are used by (Mooney and Roy, 2000) to recommend books.

Several classification algorithms have been also used to content-based recommend: decision trees, k-nearest neighbor, and neural networks (Pazzani and Bills, 1997). The heterogeneity of possible scenarios, arising from the massive deployment of an enormous amount of smart objects, imposes the use of sophisticated and innovative models and algorithms. We propose an advanced version, enriched with semantic properties, of the agent-based algorithm for building a things recommendation system introduced in (Forestiero, 2017), in which bio-inspired agents work together in order to obtain a common purpose that is the organization of distributed resources. Agents are: extensible, they can be created or modified; stable, when an agent is out of services, other agents share tasks and ensure the continuity of services; and independent, they are running without user intervention. With characteristics as extensibility, stability and autonomy, they can be more adequate in a dynamic system because they automatically adapt to environments change (Selmi et al., 2014).

The algorithm proposed is able to organize things of an IoT environment in order to improve recommendation operations. In particular, each smart object is associated with a “cyber agent”, which represents it in a cyber layer. The cyber layer is a virtual layer in which the cyber agents can “collaborate” among them, in a peer to peer fashion (Forestiero et al., 2008a), in order to obtain a common goal and improving the performances of the system (Forestiero et al., 2008b)(Forestiero et al., 2005). Vectors of real numbers, are exploited to describe IoT objects. The vector can have different meanings, for example: the presence or absence of a given characteristic or it can be the result of a hash function locality preserving so that similar vectors are assigned to things with similar characteristics. In peer to peer systems, indeed, metadata representing the content are often indexed through bit vectors, or keys, which can have different meanings. One is that each bit represents the presence or absence of a given topic (Crespo and Garcia-Molina, 2002) (Platzer and Dustdar, 2005): this method is particularly adapt for contents like documents, because it is possible to identify the different topics existing in the documents. Alternatively, a metadata can be mapped through a hash function into a binary vector. The hash function have to be designed locality preserving (Cai et al., 2004) (Oppenheimer et al., 2005), thus, neighbor vectors are assigned to contents with neighbor/similar characteristics. Similarity measure can be the cosine of the angle or the Euclidean distance between the bit vectors.

In our approach, the Doc2Vec model (Le and Mikolov, 2014), able to represent documents in dense vectors, also capturing the semantic, is exploited to map smart objects. The cyber agents organize themselves based on the similarity of its vector with the wished IoT device/service. The outcome of the algorithm is a logically sorted list of cyber agents based on the similarity with the target IoT device/service, where the distance from the target IoT device/service increase with the distance from the initial position of the list. Thanks to this organized list, the suggestion operations become faster, because we can find similar, probably useful and recommendable vectors (smart objects) in the first positions of the list.

2 SMART RECOMMENDATION ARCHITECTURE

The aim is to design a platform able to provide useful resource suggestions in Internet of Things. To achieve this objective, a semantic multiagent algorithm was designed and implemented. Physical devices, sensors, services, etc. are represented by vectors obtained through the Doc2Vec neural model (Le and Mikolov, 2014) applied to the metadata (text) describing them. Doc2Vec is an unsupervised algorithm to generate vectors starting from sentences/documents based on Word2Vec, a word embedding approach which can generate vectors starting from words. Word2Vec (Le and Mikolov, 2014) is a two layer artificial neural network used to process text to learn relationships between words within a text corpus. Word2Vec takes as its input a large corpus of text and produces a high-dimensional space (typically of several hundred dimensions), with each unique word in the corpus being assigned a corresponding vector in the space. This “word embedding” approach is able to capture multiple different degrees of similarity between words. To create the model of relationships between the words, a particular grouping of text or documents is fed to the Word2Vec process, which is called the training corpus. Word2Vec builds a vocabulary exploiting a corpus and, by training a neural network with three levels, learns the word representations. Word2Vec proposes two kind of models: (i) Continuous Bag of Words (CBOW) that learns the representations by predicting the target word based on its context words; and (ii) Skip-gram, that learns representations by predicting each of the context words
An Agent based Platform for Resources Recommendation in Internet of Things

2.1 Agent based Algorithm

In the infrastructure proposed, each IoT device/service is associated with a cyber agent that represents it in the cyber layer. The cyber agents perform autonomously, in a self-organizing fashion and exploiting only local information, all steps of the algorithm in order to obtain an overlay network useful for recommendation operations. The topology of the emerging overlay depends on the strategy of the algorithm performed by the cyber agents. In this figure, a similarity ordered list of cyber agents emerges. This list allows to select a given number of cyber agents, representing IoT devices/services, that can be suggested to the user and probably useful and relevant because they are very similar to the wished IoT device/service.

Figure 1: Logical layers of the infrastructure. The Overlay Layer depicts the outcome of the algorithm. Based on the target word. So, one has to choose one of the architectures and set values for hyperparameters like embedding size, context size, minimum frequency for a word to be included in the word vocabulary to generate the word embeddings from a large corpus of unlabeled data. The distance/similarity between two IoT devices/services can be computed through the cosine distance/similarity between the vectors representing the description. Given two IoT devices/services, the cosine measure utilized to compute the similarity between them is reported in formula (1). The cosine-based method uses two vectors in n-dimensional space to represent the users u and v, and n will be [I]. The cosine of the angle between two vectors, as reported in formula (1), can be computed to measure the similarity between them, where \( \vec{u} \cdot \vec{v} \) is the dot-product between the vectors \( \vec{u} \) and \( \vec{v} \).

\[
\text{Similarity}(\vec{u}, \vec{v}) = \cos(\vec{u}, \vec{v}) = \frac{\sum_{i=1}^{n} u_i v_i}{\sqrt{\sum_{i=1}^{n} u_i^2} \sqrt{\sum_{i=1}^{n} v_i^2}}
\]  

(1)

Vectors obtained with Doc2Vec library and describing IoT devices/services, are assigned to cyber agents that, in fully distributed and self-organizing modalities, exploiting only local information, organize themselves in order to improve recommendation operations. The outcome of the algorithm is a sorted overlay network of cyber agents, organized on the basis of the similarity with the wished IoT device/service. Figure 1 reports a logical architecture of the algorithm in order to produce an overlay network able to provide useful and relevant information for recommendation operations; and (iii) an overlay layer, in which an overlay network of cyber agents emerges as outcome of the algorithm. The topology of the emerging overlay depends on the strategy of the algorithm performed by the cyber agents. In this figure, a similarity ordered list of cyber agents emerges. This list allows to select a given number of cyber agents, representing IoT devices/services, that can be suggested to the user and probably useful and relevant because they are very similar to the wished IoT device/service.

Figure 1: Logical layers of the infrastructure. The Overlay Layer depicts the outcome of the algorithm.
Algorithm 1: Steps of the algorithm.

```java
while true do
    computeList(Hagents, Lagents);
    if Hagents.length() > 1 then
        [max, submax] = identifyMaxSubmax(Hagents);
        createLink(max, submax);
        remove(Hagents, max);
    end
    else if Lagents.length() > 1 then
        [min, submin] = identifyMinSubmin(Lagents);
        createLink(min, submin);
        remove(Lagents, min);
    end
end
```

ca) which simply delete ca from the neighbors list.

When this starting phase finishes, the last cyber agent contained in the list Lagents, represents the linked cyber agent with the highest vector value among all linked cyber agents with the vector value lower than Vc; while, the Hagents list contains the linked cyber agent with the lowest vector value among all linked cyber agents with vector value higher than Acyber. At a steady situation i.e. after a transition phase, each cyber agent is connected, with virtual or real link, with two cyber agents: the cyber agent having the vector value immediately less similar and the cyber agent with the vector value of immediately more similar of the all cyber agents. It is possible to design a smart recommendation mechanism thanks to the ordering achieved by the algorithm. In fact, the cyber agents organize themselves based on the similarity with the wished IoT device/service and, once the algorithm ends, we can select a given number of IoT devices/service starting from the head of the list. Thanks to the organization, the suggestion provided are very similar and, more probably, useful for the user.

3 EXPERIMENTAL RESULTS

In order to investigate its effectiveness, a Java simulator was implemented in which the characteristics of real scenario are carefully considered. The simulator allows to build a set of IoT devices/services randomly connected to a given mean number of others IoT devices/services (neighbors), in order to create a group of objects connected through a local area network. Each IoT device/services is described thorough metadata: a text file containing a detailed description of the characteristics of the objects. By exploiting the Doc2Vec library, a cyber agent, responsible of an IoT service/device, can obtain a vector representing it. The cyber agents following autonomously the steps of the algorithm organize in order to obtain the overlay useful to improve recommendation operations. Figure 2 shows the mean number of message managed by each cyber agent to obtain the logical sorting, for different value of mean number of linked neighbors.

The experiments were executed for different values of MeanNgh, that is the average number of connections/neighbors of every cyber agent.

It can be noticed how with a limited number of messages, the algorithm reaches a stable situation and an useful overlay is obtained. In Figure 3 the total number of messages exchanged by all cyber agents per step to obtain the sorting. The number of involved cyber agents is set to 5000.

Notice that the number of messages decreases exponentially and the algorithm converges in a finite number of steps. Successively the worst case to obtain the organization, i.e. the maximum possible number of steps needed to each cyber agent to individu-
ate its neighbors, is shown in Figure 4. Even in this case, the experiments were executed for different values of the average number of connections/neighbors of cyber agent, when the number of the involved cyber agents changes. The interesting result is that the maximum number of steps needed to each cyber agent to individuate its neighbors is always very low and, anyway, limited to first application of the algorithm in which no one previous sorting exists.

4 CONCLUSIONS

The design of a multiagent platform for building a recommendation system in Internet of Things boasting semantic features, was presented. This platform relies on an agent based algorithm allowing discovery and recommendation operations faster. IoT objects are represented through vectors obtained by exploiting a word embedding library able to capture the semantic characteristics. Intelligent agents autonomously execute a distributed algorithm allowing to bring out an useful overlay for recommendation operations. Experimental results show as the algorithm, pillar of the platform proposed, enables an effective reorganization of IoT services/devices, allowing very encouraging performance for recommendation operations.

REFERENCES


