The following publication Y. Chen, J. Zhang, L. Xu, M. Guo and J. Cao, "Modeling Latent Relation to Boost Things Categorization Service," in IEEE Transactions on Services Computing, vol. 13, no. 5, pp. 915-929, 1 Sept.-Oct. 2020 is available at https://doi.org/10.1109/TSC.2017.2715159.

Modeling Latent Relation to Boost Things Categorization Service

Yuanyi Chen, Jingyu Zhang, Liting Xu, Minyi Guo, *senior member, IEEE,* and Jiannong Cao, *Fellow, IEEE,*

Abstract—While it is well understood that the Internet of things (IoT) offers the capability of integrating the physical world and the cyber world, it also presents many significant challenges with numerous heterogeneous things connected and interacted, such as how to efficiently annotate things with semantic labels (i.e., things categorization) for searching and recommendation. Traditional ways for things categorization are not effective due to several characteristics (e.g., thing's text profiles are usually short and noise, things are heterogeneous in terms of functionality and attributes) of IoT. In this paper, we develop a novel things categorization technique to automatically predict semantic labels for a given thing. Our proposed approach formulates things categorization as a multi-label classification problem and learns a binary support vector machine classifier for each label to support multi-label classification. We extract two types of features to train classification model: 1) explicit feature from thing's profiles and spatial-temporal pattern; 2) implicit feature from thing's latent relation strength. We utilize a latent variable model to uncover thing's latent relation strength from their interaction behaviours. We conduct a comprehensive experimental study based on three real datasets, and the results show fusing thing's latent relation strength can significantly boost things categorization.

Index Terms—Things Categorization, Multi-label classification, Interaction behaviours, Latent variable model, Internet of things

1 Introduction

Recent years have witnessed numerous physical things (e.g, mobile phones, wallets and key-chains) embedded with sensing, communication and computing capabilities are being inter-connected to form an Internet of things, which is mainly attributed to the rapid advances in identification technologies and micro sensors, such as radio frequency identification (RFID), self-powered sensors, and nano technology sensors. Physical things embedded with smart sensors are seamlessly integrated into the information network, people can query and change their state and associated information over the Internet. Interconnection of physical things providing the ability to share information across platforms through a unified framework, developing a common operating picture for enabling innovative applications, such as supply chain management, smart healthcare and intelligent transportation. Meanwhile, the IoT also presents a few significant challenges with increasing heterogeneous things participate in sensing and communicating, such as how to efficiently annotate these heterogeneous things with semantic labels for browsing, searching and recommendation. Traditional ways (for a review see Section 2) for things categorization will suffer serious challenges due to unique characteristics in IoT:

- Text-based categorization methods [33], [36], [37] cannot achieve satisfactory performance as the text profiles of things are usually short and noise in IoT. Additionally, labels are usually expensive and unlabelled things are abundant in IoT.
- Semantic-based categorization methods [1], [6], [7], [8], [23] are not effective as they require time-consuming preparation of prior knowledge, such as manually defining the descriptions of things and their corresponding concepts under a uniform format like Resource Description Framework (RDF).
- Link-based categorization methods [19], [22], [30] are infeasible in IoT since the connection of things in IoT are usually implicit, unlike people has observable links in social network or web-pages are linked by universal resource locator (URL) in Internet.

Fortunately, the interaction behaviours of things can be easily recorded and obtained using ubiquitous sensing technologies, such as RFID and sensor readings. These interaction behaviours, which embedded with rich spatial-temporal information and implicitly imply the regularities of users, provide us a new approach to uncover the latent connection of things. Things are discrete without explicit connection in IoT, but human and things will interact in daily activities, and these interactions can provide rich information (e.g., activity, location and time) for uncovering thing's implicit connection. Our proposed approach can derive the latent relation strength among things from their interaction behaviour and further form a relation graph of things, where their implicit connections can be revealed. This kind of relation analysis can boost many valuable services in IoT, such as:

Things clustering, which aims to cluster heterogeneous things into different groups according to a predefined proximity measure. The key of things clustering is designing proximity metric to measure the similarity of things. However, traditional ways based on text features or thing's attributes are not effective as the unique characteristics in IoT (e.g., the text descriptions of things are usually incompleted, things are heterogeneous in term of different attributes thus cannot be represented in a uniform space).

- *Yuanyi Chen is the correspondence author and with the Department of Computer Science and Technology, Shanghai Jiao Tong University, Shanghai, China. Email: cyyxz@sjtu.edu.cn.*
- *Jingyu Zhang, Liting Xu and Minyi Guo are with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China.*
  E-mail: zhangzhang@sjtu.edu.cn, {xu-lt,guo-my}@cs.sjtu.edu.cn
- *Jiannong Cao is with the Department of Computing, Hong Kong Polytechnic University, Hong Kong, China. E-mail: csjcao@comp.polyu.edu.hk.*

The relation analysis of things can enhance the performance of things clustering in terms of things tend to interact with other things with similar characteristics according to [40]. Therefore, things clustering can be solved by some graph-based algorithms (e.g., community detection algorithm [38]) based on thing's relation graph.

Things Categorization, which aims to automatically predict the labels for a given thing. Millions of things connected and interacted in the IoT will result in serious challenges for things management and network scaling, while effectively things categorization is an essential step to cope with these challenges. For instance, in education scenario, things categorization enables learners to get rich data and further improve their knowledge [7]. As mentioned earlier, text-based models [33], [37] or link-based models [22], [30] are not effective for things categorization in the IoT.

Fortunately, things interactions are not completely random as human daily activities usually follow a regular pattern [34], such as people usually cook in the kitchen and eat breakfast at 7:00 am-9:00 am. Therefore, things relation analysis based on their interactions can boost things categorization since different things used by the same person at the same location or time may be similar (e.g., having the same label).

Context-aware Activity Recognition, which aims to recognize human activities (e.g., eating, cooking and toileting) from sensor readings. Existing approaches usually use the sensor values of things as input to train a probabilistic model to find the most likely sequence of activities (e.g., Hidden Markov Model [32] and support vector machine [25]). However, these methods are inefficient when human activities are performed in a complex situation (i.e., interleaved or concurrent) [12].

Complex human activities can be defined as a task that several things interact at specific location at a certain timestamp (see in Table 2), and then can be modeled as a 3-dimensional tensor: $Activity \in R^{Things \times Location \times Time}$. After modeling the sensor readings with 3-dimensional tensor by deriving thing's latent relation strength from their interaction behaviours, activity recognition can be solved by finding a matching scheme to measure the similarity of two tensors.

In this paper, we present a novel things categorization technique for the IoT to automatically predict the labels for a given thing. We formulate things categorization as a multi-label classification problem and learn a binary SVM classifier for each label in the label space to support multi-label classification. We extract two types of features to train classification model: 1) explicit feature from thing's profiles and spatial-temporal pattern; 2) implicit feature from similar things in terms of thing's latent relation strength. The principle underlying our approach for modeling thing's latent relation strength is the homophily theory in thing's interaction [39], [40], which suggests the stronger the relation the higher likelihood that a certain type of interaction will take place between a pair of things. In this way, we consider that the latent relation strength directly impacts the interaction frequency of a pair of things, and further model the latent relation strength as a hidden cause of their interaction frequency.

The remainder of the paper is organized as follows: Section 2 surveys related work about things categorization in IoT. Section 3 describes the proposed approach for modelling the latent relation strength of things in detail. Section 4 demonstrates how to utilize the learnt latent relation strength to boost things categorization. Section 5 reports and discusses the experimental results. Finally, we present our conclusion and future work in section 6.

2 Related work

In this section, we survey related works about things categorization and discuss how these works differ from our study.

Text-based Categorization. IoT things usually have short and noisy text descriptions, such as thing's name, manufacturer and instruction manual. Thus text-based methods can be used to label things, which firstly extract text-based features (e.g., term frequency and information gain) and then perform categorization with classifiers. To overcome some limitations of traditional text features, the work [33] proposed a novel feature selection method based on term frequency and T-test for text categorization, and [37] utilized the compactness of the appearances of the word and the position of the first appearance of the word to construct distributional features for text categorization.

Since assigning labels to large samples is costly and time-consuming, the work [36] proposed a web-assisted text categorization framework, which firstly extracted important keywords from the available labelled documents to form the queries, then utilized search engines to retrieve relevant documents for semi-supervised categorization. Unfortunately, this approach is impractical for things categorization in IoT as there is few information about physical things in Internet nowadays.

Semantic-based Categorization. A few studies ([1], [6], [7], [8], [23]) have been proposed to label things using semantic web technologies. The idea behind semantic-based categorization is that firstly define a metadata model to describe all the cyber-physical characteristics (e.g., geophysical, functional and non-functional) of a thing, then use ontology language description logic to label physical things in terms of different dimensions (e.g.,spatial, temporal and thematic).

To enrich the description of thing's characteristics for educational purpose, the work [7] exploited shared vocabularies for categorization by three steps: 1) defining a data model for representing Point-of-Interest; 2) mapping the relational database to the data model; 3) generating RDF data and enriching with links to related data. The work [8] proposed an IoT semantic categorization framework, which representing the model data as linked data and associating with the existing data on the Web (e.g., Linked Open Data). The work [6] proposed a hierarchical context model based on ontology to label things and their contextual relationships. The work [23] further utilized Time-of-Arrival for thing's geospatial categorization in IoT. The drawbacks of semantic-based categorization methods include: 1) The time-consuming preparation of prior knowledge, such as defining the descriptions of things and their corresponding concepts under a uniform format like RDF; 2) Most of semantic-based categorization methods are based on problem-solving principle, which defines the ontology related to a certain task (e.g., home energy [1] and education purpose [7]) or activities thus are lack of scalability.

Link-based Categorization. IoT things are implicitly connected in a network or graph by some attributes (e.g., location, owner and manufacturer), thus link-based methods [19], [22], [30], [31] can leverage these connections to improve categorization performance. For example, the work [35] labelled things by modelling things as web tables with headers and cell values. More exactly, this categorization process includes three steps: 1) querying the background knowledge base sources to generate initial ranked lists of candidate assignments for schemas, content values and relations between schemas; 2) using a probabilistic graphical model to capture the correlation between schemas, content values and schema relations to make class, entity and relation assignments; 3) producing linked data triples after the mapping is complete and performing things categorization using link-based methods [19].

More recently, increasing studies [2], [4], [17] aimed at giving social-like capabilities to the things in IoT, namely social internet of things (SIoT) based on the notion of social relationships among things. In SIoT [2], things are able to interact with other things in an autonomous way with respect to the owners, and can easily crawl the IoT made of billions of things to discover services and information in a trust-oriented way. For example, SIoT [2] described four kinds of relationships for things in IoT: co-location relationship, co-work relationship, co-owner and social relationship. Lilliput [4] further extended the SIoT by integrating things as well as online social networks, which is an ontology-based platform by fusing online social networks and things as a social graph. Socialite [17] utilized semantic model for the SIoT, which includes device types, capabilities, users, relationships and rules leveraging such models. Therefore, many link-based methods [22], [30] can be utilized to label things by modeling heterogeneous things and their relationships with a graph. Unfortunately, link-based methods are ineffective for things categorization in IoT due to 1) acquiring a sufficient number of labelled things to enable accurate learning for link-based categorization usually are expensive or impractical; 2) SIoT may ignore some implicit factors that may influence categorization performance (such as usefulness and availability), for instance, Microwave and Toaster may have different manufacture or owner, but both they are kitchen appliances and can heat foods.

To our best knowledge, only several studies [39], [40] focus on boosting things categorization by exploring regularities in the interactions between human and things. These approaches discovered the latent relation strength of things by mining three dimensional information of the interaction behaviours: user, temporarily and spatiality. However, we find these approaches mentioned above fail to model thing's latent relation strength and their interaction behaviours by analyzing three real datasets, for example, we observe that the interaction probability of two things and their history interaction frequency follows a roughly power law distribution. Additionally, these studies infer thing's relation strength without considering their attributes profiles (e.g., the manufacturer, type and capability).

Our proposed approach differs from the above-mentioned works in the following three aspects: 1) we regard things categorization as a multi-label classification problem and learn a binary SVM classifier for each label in the label space to perform things categorization; 2) we extract two types of features to train SVM classifier, one is explicit features from thing's text profiles and spatial-temporal pattern, another is implicit features from thing's latent relation strength; 3) we derive thing's latent relation strength by jointly considering thing's profile similarities and interaction behaviours with a latent variable model. Recently, latent variable model has been widely used in a few studies on text mining. For instance, latent semantic analysis (LSA) [18] supposed that there is an underlying semantic structure in text and the relationship between terms and documents can be derived in this semantic space. Several studies [20], [42] based on LSA are proposed to deal with short text classification. Probabilistic latent semantic analysis (pLSA) [14] extended LSA by explicitly defining latent topic of a document as the latent variable during a random process, which is widely used in text summarization [26] and image annotation [43]. Latent Dirichlet Allocation (LDA) [3] further extended pLSA by adding priors (Dirichlet Distribution) to the document collection, which occupies an important position in many fields of text mining (such as text classification [5] and review-based sentiment analysis [24]).

TABLE 1: Notations used in the paper

SYMBOL	DESCRIPTION
O,T,Loc,U	the set of things, timestamps, locations, labels
N,Q,F,H	the number of things, timestamps, locations, interactions
A_i	the attribute set of thing o_i
a_i^j	the value of the j-th attribute of o_i
$Y^{(ij)}$	the interaction set between thing o_i and o_j
$X^{(ij)}$	the variables to capture the tendency of interactions
$z^{(ij)}$	A similarity vector based on thing's attributes
$I^{(ij)}$	the latent relation strength between o_i and o_j
Ω_i	the k-neighbour set of o_i in terms of relation strength
w	A K-dimensional similarity vector to be estimated
σ^2	the variance of Gaussian distribution
$\alpha_l,\beta_l,\theta_l$	the parameters of power law distribution
$G=\{V,E,W\}$	thing's top-k relation graph, $V=V_s\cup V_r$
V_s,V_r	the labelled things set and unlabelled things set
M	the transition matrix of random walk with restart
F_{Latent}	implicit features from thing's relation graph
$F_{Cluster}$	implicit features by clustering thing's interactions
F_{text}	text-based features from thing's text descriptions
F_S	spatial features from thing's spatial pattern
F_T	temporal features from thing's temporal pattern

3 Modeling the Latent Relation of Things from Interaction Behaviour

In this section, we first present the problem statement of modeling thing's latent relation strength from their interaction behaviours. Then detail the proposed approach, a latent variable model to derive thing's latent relation strength.

3.1 Problem Statement

For ease of the following presentation, we first define the key data structures and notations used in the proposed approach. Table 1 lists the relevant notations used in this paper.

Definition 1. (**Thing**) A Thing o_i in IoT, denote by $<ID_i, A_i>$, where ID_i is the identifier of o_i and $A_i=\{a_i^1,...,a_i^j,...,a_i^{|A_i|}\}$ is the attributes set of o_i (e.g., type, color and manufacturer). a_i^j is the value of the j-th attribute of o_i.

As identified by [13] and [16], things in IoT are sensing and actuating physical devices that providing the ability to share information across platforms through a unified framework. Thus things has the following three characteristics: 1) physical devices. Thing's attributes are directly related to the physical characteristics of devices [9]; 2) embedded-in sensors, which are utilized to provide sensing, computing and communication ability; 3) unique identity. For example, the associated IP address can be utilized as thing's identifier. Things concept can be explained by the following example.

Example 1: Considering a thing o named smart oven [1] (as shown in Figure 1), which is a physical oven but embedded in several sensors (e.g., laser scanner for reading bar codes on the lids

1. https://www.wired.com/2016/03/tovalas-smart-oven-wants-replace-microwave/

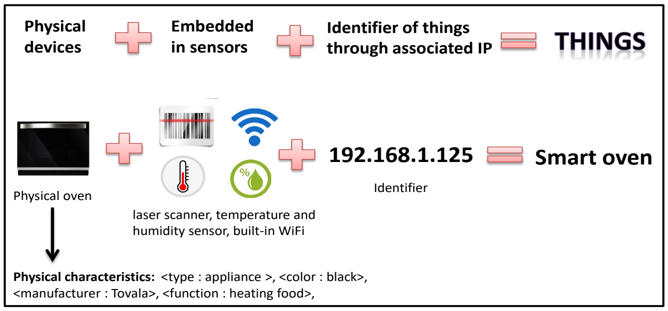


Fig. 1: An illustrative example of thing and its attribute set

TABLE 2: An example of things interaction behaviour

Daily activity	Preparing breakfast,3/5/2016, 08:13:12, 08:24:18
Things	Freezer, Microwave, Sink faucet - hot, Plate, Pan
Starting Time	08:15:38, 08:17:21, 08:19:35, 08:22:14, 08:20:25
Ending Time	08:21:24, 08:23:19, 08:20:11, 08:23:38, 08:20:46

of compatible meals, temperature and humidity sensor for detecting whether the heat and humidity inside the oven is optimal, and built-in WiFi for downloading new and newly perfected recipes from the cloud). For an oven, its physical characteristics consist of $< type : appliance >, < color : black >, < manufacturer : Tovala >, < function : heatingfood >$, thus we can obtain its attributes set: $A_o = \{appliance, black, Tovala, heatingfood\}$. According to thing's definition, we denote thing o as $< 192.168.1.125, A_o >$ by using IP address as thing's identifier.

Definition 2. (**Interaction Behaviour**) Interaction behaviour among things happens when people use things in daily activities (e.g., Preparing breakfast, Dish washing and Brushing teeth). Let $O = \{o_1, o_2, ..., o_N\}, T = \{ts_1, ts_2, ..., ts_Q\}$ and $Loc = \{loc_1, ..., loc_F\}$ denote the set of things, timestamps and locations, respectively. An interaction between o_i and o_j, denote by $y \in Y^{(ij)} = \{y_1^{(ij)}, y_2^{(ij)}, ..., y_H^{(ij)}\} = \{< o_i, o_j, ts, loc > | o_i, o_j \in O \bigwedge ts \in T \bigwedge loc \in Loc\}$, indicates that a user used o_i and o_j in location loc at timestamp ts. To extract the timestamp of thing's interaction behaviours, we divide a day into 24 hourly slots. To this end, we generate the total number of hashed time slots is 24, denote as TS=$\{ts_1, ts_2, ..., ts_{24}\}$. For instance, if two things interact at 1:32 pm, 3/15/2016, the time slot of this interaction is ts_{14}.

In our experiment, we utilize a context-aware experience sampling tool (CEST) and state-change sensors to collect thing's interaction behaviours, i.e., the state-change sensors recorded data about the movement of things and the participants used CEST to record information about their activities. During the study, each participant was given a PDA to run CEST tool. The participant utilizes the CEST to select the activity what he/she is doing, and records the start and end time of this activity. Figure 2 shows an example of the type of data that was collected by the state-change sensors and CEST. As shown in Figure 2a, the starting and ending time of things are automatically recording by their embedded-in state-change sensors. Then, we can obtain the participated things of an activity by observing the sensor activations during the activity duration (as shown in Figure 2b). As shown in Table 2, we generate the daily activity (Preparing breakfast) involved five things, and consider there is an interaction behaviour between each pair of things during this activity.

Definition 3. (**Latent Relation Strength**) The latent relation strength between o_i and o_j denote by $I^{(ij)}$,which is determined by i) the attribute similarity of o_i and o_j; and ii) the interaction behaviours between o_i and o_j. In other words, for the larger latent relation strength $I^{(ij)}$, two things (o_i and o_j) are required to be more similar in term of either attributes or are more likely to interact with each other.

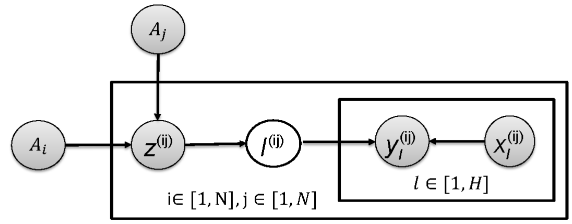


Fig. 3: Graphical model of learning thing's latent relation strength

With the aforementioned definitions, the problem of modeling thing's latent relation strength can be formally stated as follows:

Given a set of things: $O = \{< ID_1, A_1 >, < ID_2, A_2 >, ..., < ID_N, A_N >\}$; and their history interaction behaviours $Y = \{y_1, y_2, ..., y_H\}$, the problem of modeling latent relation strength aims to discover the implicit connection of things by exploiting their attributes and observable interaction behaviours.

3.2 Approach

In this section, we first describe the modeling part of the proposed approach, a latent variable model to infer thing's latent relation strength from their interaction behaviours, and then present its inference process.

3.2.1 Model Description.

Previous studies [39], [40] have shown that homophily is ubiquitous in IoT, which suggests the likelihood that a certain type of interaction will take place between a pair of things relate positively to their latent relation strength. In this way, we model thing's latent relation strength as the hidden cause for their interaction behaviours. Such interactions could be, for example, preparing breakfast, eating lunch and brushing teeth. We further consider thing's latent relation strength as a hidden effect of thing's profile similarities. The profiles similarity are caused by thing's attributes, such as the manufacturer, the functionality and the geographic locations that they belong to, etc.

Formally, let $Y^{(ij)} = \{y_1^{(ij)}, y_2^{(ij)}, ..., y_H^{(ij)}\}$ denote the interaction behaviours between o_i and o_j, $I^{(ij)}$ denote the latent relation strength between o_i and o_j. Then, we utilize a graphical model to represent the influence caused by the profiles similarity to $I^{(ij)}$, as well as the influence of $I^{(ij)}$ on interaction behaviours, as shown in Figure 3. In this figure, the gray-colored nodes depict observed variables (i.e., $z^{(ij)}, Y^{(ij)}$ and $\{x_1^{(ij)}, x_2^{(ij)}, ..., x_H^{(ij)}\}$), which are all visible in the training phase.

The detailed description of variables in this figure is explained as follows:

- $z^{(ij)}$ denotes the profiles similarity of o_i, o_j, which is a K-dimensional vector calculated based on the attributes set of o_i and o_j (i.e., A_i and A_j).
- $I^{(ij)}$ is the latent relation strength between o_i and o_j, which is a hidden factor for thing's interaction behaviours and influenced by their profiles similarity.

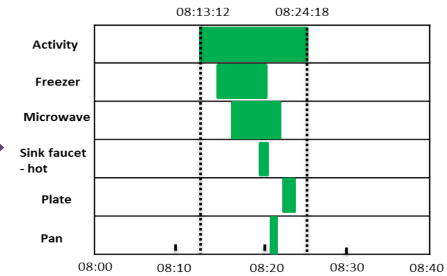
Activity Information from CEST Tool

Activity	Day	Starting Time	Ending Time	Location
Preparing breakfast	3/5/2016	08:13:12	08:24:18	Kitchen

Sensor Activations Information from Built-in Sensors

Sensor ID	Day	Starting Time	Ending Time	Location	Thing
36	3/5/2016	08:15:38	08:21:24	Kitchen	Freezer
26	3/5/2016	08:17:21	08:23:19	Kitchen	Microwave
28	3/5/2016	08:19:35	08:20:11	Kitchen	Sink faucet - hot
131	3/5/2016	08:22:14	08:23:38	Kitchen	Plate
19	3/5/2016	08:20:25	08:20:46	Kitchen	Pan
	……	……			

(a)



(b)

Fig. 2: An illustrative example of collecting thing's interaction behaviours

- $X^{(ij)} = \{x_1^{(ij)}, x_2^{(ij)}, ..., x_H^{(ij)}\}$ are auxiliary variables that we introduce to increase the accuracy of the model. Such variables capture auxiliary causes of the interactions which are independent of the latent relation strength. For example, the total number of interactions that a thing participated in represents its intrinsic tendency to interact, which can moderate the effect of latent relation strength on interaction behaviours.

Our model represents the relationships among these variables by modeling the conditional dependencies (as shown in Figure 3), so the joint distribution decomposes as follows:

$$P(I^{(ij)}, Y^{(ij)}|A_i, A_j) = P(I^{(ij)}|A_i, A_j) \prod_{l=1}^{H} P(y_l^{(ij)}|I^{(ij)}, x_l^{(ij)}) \qquad (1)$$

Given the attributes of o_i and o_j, we model the conditional probabilities $P(I^{(ij)}|A_i, A_j)$ using the widely-used Gaussian distribution:

$$P(I^{(ij)}|A_i, A_j) = (w^T z^{(ij)}, \sigma^2) \qquad (2)$$

where w is a $K-$dimensional weight vector to be estimated and σ^2 is the variance of Gaussian model, $z^{(ij)}$ is the profiles similarity based on A_i and A_j.

For modeling the dependency between $Y^{(ij)}$ and $I^{(ij)}, X^{(ij)}$, we analyze the characteristics of thing's interaction behaviours using three real world datasets: 1) Our dataset is collected by 13 participants during six months, which consists of 32,716 interaction records from 196 things ; 2) MIT S1, S2. The two datasets are published by the AI group in MIT [28], which consists of 503 interaction records from 146 things in total. More details of these datasets are shown in Table 3. Figure 4 shows the likelihood of two things may interact as a function of their historical interaction frequency. As shown in Figure 4, we observe there exists a positive correlation between the likelihood of two things may interact and their historical interaction frequency, indicating a clustering phenomenon in thing's interaction behaviours. This phenomenon may be intuitively explained by the following tendencies: 1) things with similar attributes (e.g., provided similar services and located in the same geographic location) tend to interact; 2) things with the same label (e.g., cooking tools and office supplies) tend to interact. As mentioned earlier, we consider thing's latent relation strength is determined by their attribute similarity and interaction behaviours We believe that this clustering phenomenon in thing's interaction behaviours can be exploited for uncovering thing's latent relation strength. Thus, in the following, we study and model thing's latent relation strength and their interaction frequency.

Based on Figure 4, we intuitively think the distribution follows a roughly power-law form. Even though the right part of the figure increases linearly (i.e., increases exponentially in regular scale) and thus fits power-law distribution very well, the left part may sometimes deviate irregularly (i.e., the probability is high at some points). A reasonable explanation is that the likelihood of two things may interact cannot judge from few interactions. Generally speaking, the fact that two things with more historical interactions tend to interact is confirmed in our data analysis. Moreover, the linear portion of the plot in Figure 4 covers the majority (90%) of the interaction behaviours In this way, we model the dependency between $Y^{(ij)}$ and $I^{(ij)}, X^{(ij)}$ with a power-law distribution:

$$P(y_l^{(ij)}|I^{(ij)}, x_l^{(ij)}) = (\alpha_l I^{(ij)} + \beta_l x_l^{(ij)})^{\theta_l} \qquad (3)$$

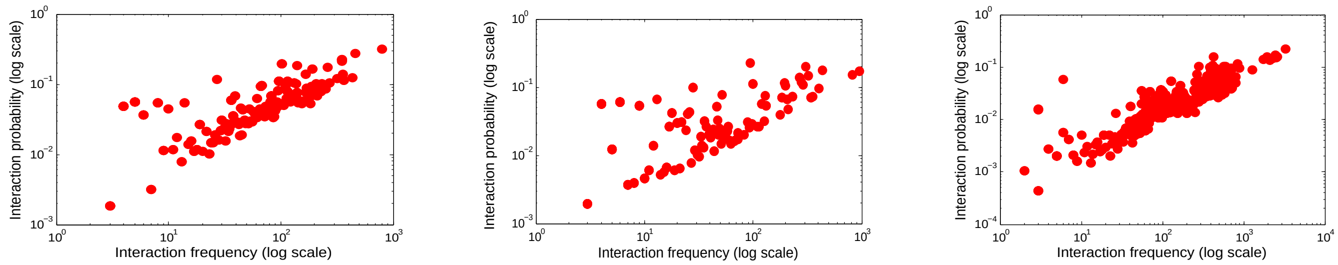
where α_l, β_l and θ_l are parameters of power law distribution to be estimated, $l = 1, 2, ..., H$.

We further add L_2 regularizes on these hyper parameters (e.g., α_l, β_l, θ_l) to avoid over-fitting, which can be regarded as Gaussian prior:

$$\begin{aligned} &P(\alpha_l, \beta_l) \propto e^{-(\lambda_1/2)(\alpha_l^2+\beta_l^2)}, l = 1, ..., H \\ &P(\theta_l) \propto e^{-(\lambda_2/2)(\theta_l)^2}, l = 1, ..., H \\ &P(w) \propto e^{-(\lambda_3/2)(w^T w)} \end{aligned} \qquad (4)$$

The dataset are represented as a set of thing pairs: $\Phi = O \times O$, denoted as $D = \{(i_1, j_1), ..., (i_N, j_N)\}$. During training phase, the variables $z^{(ij)}, y_l^{(ij)}$ and $x_l^{(ij)}$ are all visible, $(i, j) \subseteq \Phi$. According to Equation 1, given all the observed variables, the joint probability is shown as:

$$\begin{aligned} &\prod_{l=1}^{H} P(\Phi|w, \alpha_l, \beta_l, \theta_l) P(w, \alpha_l, \beta_l, \theta_l) = \\ &\prod_{(i,j)\in D} P(I^{(ij)}|z^{(ij)}, w) P(w) \prod_{l=1}^{H} P(D|I^{(ij)}, x_l^{(ij)}, \alpha_l, \beta_l, \theta_l) P(\alpha_l, \beta_l, \theta_l) \\ &\propto \prod_{(i,j)\in D} \left(e^{-(1/2\delta^2)(w^T z^{(ij)} - I^{(ij)})^2} \prod_{l=1}^{H} (\alpha_l I^{(ij)} + \beta_l x_l^{(ij)})^{\theta_l} \right) \\ &e^{-(\lambda_3/2) w^T w} \prod_{l=1}^{H} e^{-(\lambda_2/2)(\theta_l)^2} e^{-(\lambda_1/2)(\alpha_l^2+\beta_l^2)} \end{aligned} \qquad (5)$$



(a) MIT S1 (b) MIT S2 (c) Our dataset

Fig. 4: Fraction of interaction probability as a function of thing's historical interaction frequency

3.2.2 Model Inference.

We estimate the unknown model parameters $\Sigma = \{w, \alpha_l, \beta_l, \theta_l\}$ by maximizing the likelihood function as shown in Equation 5. As for the hyper parameters $\sigma^2, \lambda_1, \lambda_2, \lambda_3$, for simplicity, we take a fixed value ($\sigma^2 = 0.5$, $\lambda_1 = \lambda_2 = \lambda_3 = 0.01$) in experiment. Applying a logarithmic transformation to both sides of Equation 5, we obtain the following expression:

$$\begin{aligned} L((i,j) \in D, w, \alpha_l, \beta_l, \theta_l) &= \sum_{(i,j)\in D} -\frac{1}{2\sigma^2}(w^T z^{(ij)} - I^{(ij)})^2 \\ &+ \sum_{(i,j)\in D}\sum_{l=1}^{H} \theta_l log(\alpha_l I^{(ij)} + \beta_l x_l^{(ij)}) - \frac{\lambda_3}{2}(w^T w) \\ &- \sum_{l=1}^{H}\frac{\lambda_2}{2}\theta_l^2 - \sum_{l=1}^{H}\frac{\lambda_1}{2}(\alpha_l^2 + \beta_l^2) \end{aligned} \tag{6}$$

Note the function L (see in Equation 6) is concave, then we optimize the parameters $\alpha_l, \beta_l, \theta_l$ and variable $I^{(ij)}$ with a stochastic gradient descent algorithm. We use Netwton-Raphson algorithm to update these parameters in each iteration:

$$I^{(ij)new} = I^{(ij)old} - \frac{\partial L}{\partial I^{(ij)}} / \frac{\partial^2 L}{\partial (I^{(ij)})^2} \tag{7}$$

$$\alpha_l^{new} = \alpha_l^{old} - \frac{\partial L}{\partial \alpha_l} / \frac{\partial^2 L}{\partial (\alpha_l)^2} \tag{8}$$

$$\beta_l^{new} = \beta_l^{old} - \frac{\partial L}{\partial \beta_l} / \frac{\partial^2 L}{\partial (\beta_l)^2} \tag{9}$$

$$\theta_l^{new} = \theta_l^{old} - \frac{\partial L}{\partial \theta_l} / \frac{\partial^2 L}{\partial (\theta_l)^2} \tag{10}$$

Where the coordinate-wise gradients and the second order derivatives can be found in Appendix A (the appendix file is included in the supplemental file).

As for w, the coordinate-wise gradient is as following:

$$\frac{\partial L}{\partial w} = -\frac{1}{\sigma^2}\sum_{(i,j)\in D} z^{(ij)}(w^T z^{(ij)} - I^{(ij)}) - \lambda_3 w \tag{11}$$

The root of $\partial L / \partial w = 0$ can be solved by ridge regression [11]:

$$w = (\lambda_3 \sigma^2 I + Z^T Z)^{-1} Z^T C \tag{12}$$

where $Z = [z^{(i_1 j_1)}, z^{(i_2 j_2)}, ..., z^{(i_N j_N)}]^T, C = [I^{(i_1 j_1)}, I^{(i_2 j_2)}, ..., I^{(i_N j_N)}]^T$, I is the identity matrix.

Algorithm 1 shows the procedure for optimizing these parameters, we optimize model parameters $\Sigma = \{w, \alpha_l, \beta_l, \theta_l\}$ using Newton-Raphson until converged.

Algorithm 1 The algorithm for optimizing parameters

Require: Data samples $D = \{(i_1, j_1), ..., (i_N, j_N)\}$.
Ensure: Model parameters $\Sigma = \{w, \alpha_l, \beta_l, \theta_l | l = 1, 2, ..., H\}$.

```
while not converged do
for each Newton-Raphson step do
    ***Step1: Estimate latent relation strength***
    for (i, j) ∈ D do
        Update I^(ij) according to Equation 7.
    end for
    ***Step2: Estimate parameters α_l, β_l, θ_l***
    for l = 1, 2, ..., H do
        Update α_l, β_l, θ_l according to Equation 8, 9, 10
    end for
end for
Update w according to Equation 12.
endwhile
return Σ = {w, α_l, β_l, θ_l | l = 1, 2, ..., H}.
```

4 Boosting Things Categorization Using Latent Relation

Modelling thing's latent relation strength can facilitate a few valuable services (e.g., things categorization, recommendation and searching) in IoT. Due to space constraints, we briefly introduce an important application: things categorization, which aims to automatically predict appropriate semantic labels that a given thing. A thing may be associated with multiple labels in IoT. For example, a microwave associated with a label *cooking* may also be tagged with *appliance*, and a television may label with *entertainment* and *appliance*. Therefore, things categorization can be formulated as a multi-label classification problem. In this study, we propose a method for things categorization by learning a binary SVM classifier for each label to support multi-label classification. To train the classification model, we extract two kinds of features for each thing: 1) implicit features from similar things in terms of the learnt latent relation strength, which is derived by building a top-k relation graph where similar things are linked by virtual edges; 2) explicit features of things, such as text features (e.g., Term Frequency or Term Frequency Inverse Document Frequency) and spatial-temporal pattern.

4.1 Implicit Features from the Learnt Latent Relation Strength

We extract the implicit features among things in order to formulate descriptive features of a given thing from its similar things. To capture the implicit features from similar things, we first construct a top-k relation graph of things based on the learnt relation strength. In thing's relation graph, things are linked by their latent relation strength, which is derived from their interaction behaviours. Then, we perform random walk with restart (RWR) [10] to derive the relation strength between each pair of things. The goal of RWR is to predict the label probability for a given thing by exploring the latent relation strength with similar things, and using the label probability as implicit feature for classification.

(1) Construct top-k relation graph of things (RGT). The idea of extracting implicit features is to infer descriptive features of a given thing from its neighbour and labelled things, since only few things are labelled in IoT. In this way, we construct a top-k relation graph of things by connecting things together.

Formally, let $G =< V,E,W >$ denote the top-k relation graph among things , where $V = V_s \cup V_r$ (V_s is the labelled vertex set and V_r is the unlabelled vertex set) is the set of nodes, E is the set of edges in G. For $o_i, o_j \in V_s$, we define $W_{ij} = 1$ if o_i and o_j have the same class label, 0 otherwise. If at least one of o_i, o_j is unlabelled, W_{ij} is defined as:

$$W_{ij} = \begin{cases} exp(-1/\eta I^{(ij)}), & \text{if } o_j \in \Omega_i \text{ or } o_i \in \Omega_j \\ 0, & \text{otherwise} \end{cases} \tag{13}$$

where Ω_i denotes the k nearest neighbour set of o_i in term of relation strength and Ω_j is also similar, η is a weight coefficient.

(2) Perform RWR on RGT. Let $\Omega_i = \{o_{j1}, o_{j2}, ..., o_{jk}\}$ be the set of k nearest neighbours which is connected with o_i in RGT, τ_i denotes a tag μ associated with thing o_i ($\mu \in U$). Let $P(\tau_i = \mu|\Omega_i)(\mu \in U)$ denote the probability that o_i may associate with label μ, we initialize $P(\tau_i = \mu|\Omega_i) = 1/|U|$ for unlabelled things, and $P(\tau_i = \mu|\Omega_i) = 1$ for labelled things if thing o_i has label μ or 0 otherwise. Then, we utilized RWR to find $P(\tau_i = \mu|\Omega_i)(\mu \in U)$ for each thing.

Without lost of generality, we assume the random walker starts from an unlabelled things o_i on graph G. Then, the random walker iteratively transmits to other nodes which have edges with o_i, with the probability that is proportional to the edge weight between them. At each step, o_i also has a restarting probability λ to return itself. We can obtain the steady-state probability of o_i by visiting other vertexes until the RWR process is converged. The RWR process can be formulated as Equation 14:

$$P_{t+1}(\tau_i = \mu|\Omega_i) = (1-\lambda)MP_t(\Omega_i) + \lambda P_t(\tau_i = \mu|\Omega_i) \tag{14}$$

where $P_t(\tau_i = \mu|\Omega_i)$ represents the estimation probability in step t, $P_t(\Omega_i)$ denote the estimation probabilities of all nearest neighbours of o_i at step t, denote by $P_t(\Omega_i) = [P_t(\tau_{j1} = \mu|\Omega_{j1}), P_t(\tau_{j2} = \mu|\Omega_{j2}), ..., P_t(\tau_{jk} = \mu|\Omega_{jk})]$. M is the transition matrix, which is obtained based on weight matrix W by row normalization, as shown in Equation 15.

$$M = W\Psi^{-1} \tag{15}$$

where $W = [w(i, j1), w(i, j2), ..., w(i, jk)]^T$, Ψ is the usual normalizer and defined as $\Psi = \sum_{js \in \Omega_i} W(i, js)$.

The label probability estimation for each label on a thing o_i can be obtained when the RWR process is converged, which are regarded as implicit features for SVM training (F_{Latent}). The process on how to extract implicit features can be explained by the following example.

Example 2: As shown in Figure 5, suppose we have 5 things (three labelled things:{A,B,C} and two unlabelled things:{D,E}) and 4 labels (l1,l2,l3 and l4). After constructing top-2 relation graph of things based on their latent relation strength, we extract thing's implicit features by performing RWR on the relation strength according to Equation 14 until converged. Finally, we obtain the label probability estimation for each possible label on a thing and regard the label probability estimation as implicit features. For instance, the implicit features of thing A are a 4-dimensional vector $[0.33, 0.51, 0.3, 0.27]$, while the implicit features of thing D is: $[0.29, 0.43, 0.36, 0.23]$.

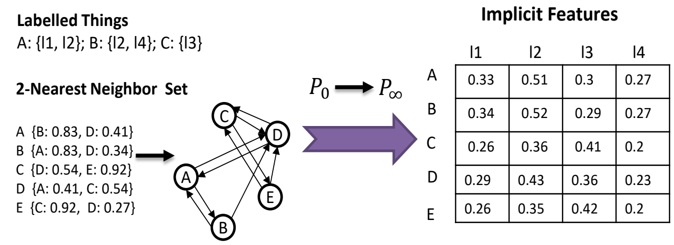


Fig. 5: An example of extracting implicit features from top-k relation graph

Let $|U|$ denotes the number of labels, $|V|$ denotes the total number of things, and $|E|$ denotes the total number of edges in the relation graph. It takes $O(|U| * |V|)$ time to initialize the label probabilities for all things. Then at each iteration, we need to process each edge twice to update the label probabilities, once for each thing at each end of the edge. We also need $O(|U| * |E|)$ time to learn from the initial label probabilities, so the time complexity of each iteration is $O(|U| * (|E| + |V|))$. Therefore, the total time complexity for extracting implicit features is $O(t * |U| * (|E| + |V|))$, where t is the maximum number of iteration needed to reach the steady state. We will experimentally demonstrate that this algorithm converges in a few iterations. And since the number of labels $|U|$ is constant, the computational complexity is generally linear in the number of edges and nodes in the relation graph.

4.2 Explicit Features from Text and Spatial-temporal Pattern

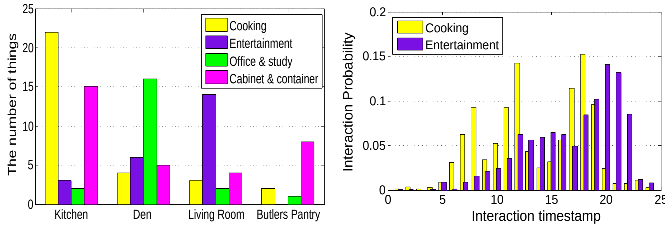
Things usually have some short and noisy text profiles (e.g., thing's color, type and manufacturer), which can be used to extract text-based features for multi-label classification. On the other hand, thing's interactions imply some spatial-temporal patterns as human daily activities usually follow a regular temporal pattern [34], for instance, people usually eat dinner at 5:00 pm-7:00 pm, which means the interacted things tend to be *cooking* tools during the time slot. We extract three explicit features from thing's text profiles and spatial-temporal patterns for training classification model.

4.2.1 *Text-based Feature*

We utilize the well-known Term Frequency Inverse Document Frequency (TF/IDF) to extract keywords from things' text descriptions [29], and the weight of keywords are regarded as text-based features (F_{text}).

4.2.2 *Spatial Feature*

To show the spatial pattern of thing's interaction behaviours in IoT, we aggregate the number of things associated with different labels at a specific location using a real-world dataset that consists of 196 things, more details of this dataset are shown in Table



(a) Spatial pattern (b) Temporal pattern

Fig. 6: Spatial-temporal pattern of things associated with different labels

3. As shown in Figure 6a, things with different labels have very different spatial pattern corresponding to different locations. For instance, things associated with *Cooking* and *Cabinet & container* are mainly located in *Kitchen*, while things associated with *Office & study* are mainly located in *Den*. Therefore, we consider the spatial pattern to be a very useful feature for things categorization. Formally, we define spatial pattern as a F-dimensional vector: $F_S(i) = [SF^i(loc_1), SF^i(loc_2), ..., SF^i(loc_F)]$. Note that, $SF^i(loc_k)$ is computed as

$$SF^i(loc_k) = \frac{N^i(loc_k)}{\sum_{j=1}^{F} N^i(loc_j)} \tag{16}$$

where $N^i(loc_k)$ is the number of interactions involved thing o_i in location loc_k, $\sum_{j=1}^{F} N^i(loc_j)$ is the total number of interactions involved thing o_i, and F is the number of locations.

4.2.3 Temporal Feature

Figure 6b reports the hourly distribution of things associated with different labels at different timestamps using a real-world dataset collected over half years (see in Table 3). From this figure, we can observe two very different temporal patterns corresponding to two kinds of labels (i.e., *Cooking* and *Entertainment*). For instance, the interaction of things associated with label *Cooking* have clearly three peak periods, corresponding to breakfast, lunch and dinner time, respectively. On the contrary, for things associated with label *Entertainment*, the interaction has one peak period (from 6:00 pm to 10:00 pm). Therefore, the temporal pattern of thing's interaction is discriminative feature for distinguishing different labels, such as *Cooking* and *Entertainment*. Formally, we define temporal pattern as a T-dimensional vector: $F_T(i) = [TF^i(t_1), TF^i(t_2), ..., TF^i(t_T)]$. $TF^i(t_k)$ is computed as

$$TF^i(t_k) = \frac{N^i(t_k)}{\sum_{j=1}^{T} N^i(t_j)} \tag{17}$$

where $N^i(t_k)$ is the number of interactions involved thing o_i at timestamp t_k, $\sum_{j=1}^{T} N^i(t_j)$ is the total number of interactions involved thing o_i, and T is the number of timestamps.

4.3 Things Categorization by fusing Explicit Features and Implicit Features

In our approach, we first formulate things categorization as a multi-label classification problem and then decompose into several distinct single-label binary classification problems. For instance, as shown in Figure 7, *Microwave* labelled *Cooking* is positive sample for a classifier for *Cooking* but negative sample for a classifier for *Storage*, while *Garbage bag* labelled *Storage* is positive sample for a classifier for *Storage* but negative sample for a classifier for *House Appliances*. To perform things categorization, our approach outputs the aggregation of the labels positively predicted by all the independent binary classifiers.

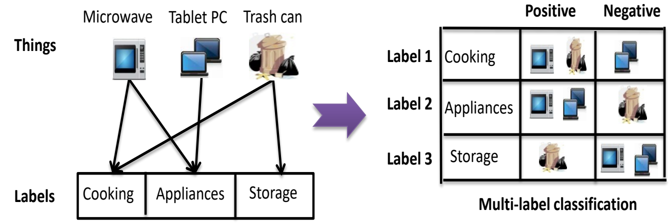


Fig. 7: An example of formulating things categorization to a multi-label classification problem

TABLE 3: Detailed information of the three datasets

	MIT S1	MIT S2	Dataset 3
# of things	76	70	196
# of daily activities	33	35	93
# of interaction records	295	208	32716
# of participants	1	1	13
# of collecting period (days)	14	14	180

After extracting four kinds of features: 1) Implicit features (F_{Latent}) from thing's relation graph; 2) Text-based features (F_{text}) from thing's text descriptions; 3) Spatial features (F_S) from thing's spatial pattern when interacting; 4) Temporal features(F_T) from thing's temporal pattern when interacting, we combine the features ($F_{Latent} + F_{text} + F_S + F_T$) together as inputs for training a set of binary SVM classifier for things categorization.

5 Experiment Evaluation

In this section, we first describe the experiment settings including data sets, baseline methods and evaluation metric. Then, we report and discuss the experimental results.

5.1 Datasets.

Three real world datasets about things interaction in IoT: two public datasets from MIT [28] AI group and one collected dataset (Dataset 3) in our experiment, are used for experimental evaluation. More details of these datasets are reported in Table 3.

– **MIT Dataset.** The first two datasets (MIT S1 and MIT S2) are published by the AI group in MIT, which collected two subject's daily activities during two weeks. The first subject was a 30-year-old woman who spent free time at home, and the second was an 80-year-old woman who spent most of her time at home. Both subjects lived alone in one-bedroom apartments, 77 and 84 sensors are installed in everyday things (e.g., *Microwave, Refrigerators and Stoves*) of the two subject's apartment, respectively. Each data collection board was marked on a plan-view of the environment for collecting data after these sensors were installed, and the location (e.g *Bathroom*) and type (e.g *Toaster*) of each thing associated sensor was prior known. For recording the subject's activities information, a context-aware experience sampling tool was used for

Ambient Sensors	Activities	Tagged things
Force sensor or Pressure sensor	Sleeping, sitting, napping	Beds, couches
Contact sensors	Opening and closing of the doors, cupboards	Door frames, shower cupboards
Proximity sensors	Detecting close distance objects	Chairs, closets and taps
Infrared Receiver Sensor	Watching TV	Near the TV

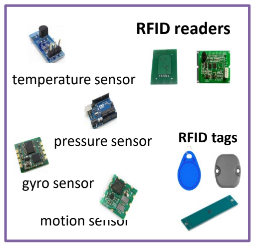


Fig. 8: Part of sensors and devices used in our experiment

labelling activities. Finally, 76 and 70 things that have participated in interactions are used to conduct experiments.

– **Our Dataset.** Our experiment environment includes one workspace (e.g., office, laboratory and meeting room) and two smart houses (e.g., bedroom, living room and kitchen). In our experiment, there are 196 things are tagged with RFID and various sensors (e.g., motion, pressure and temperature sensors, as shown in Figure 8) for collecting interaction behaviours. For generating thing's interaction behaviours, three types of information need to be recorded: 1) Activity information. To obtain the activity information, each participant utilized a context-aware experience sampling tool to mark and record their activities when interacting with things; 2) Temporal information. To map the interacting time to the corresponding timestamps, we split one day into 24 timestamps with one hour as an interval as mentioned earlier; 2) Spatial information. For static things (e.g., cabinet, toaster and door), the spatial information is prior known. For mobile things (e.g., RFID-tagged remote control and coffee cup), we utilized a fingerprint-based positioning algorithm to estimate the unknown location [44]. Thirteen participants participated in the data collection phase during six months, and more than 32,000 interaction behaviours of things are recorded in the experiment.

We manually labelled these things with different semantic labels as the ground-truth data for performance evaluation. Note that some things are labelled with multiple labels (e,g, *Microwave* is labelled with both *Cooking* and *House Appliances*, *Television* is labelled with both *Entertainment* and *House Appliances*), thus a thing may belong to multiple categories. Finally, we manually labelled these things with 798 different labels, more details information can be found in Appendix B (the appendix file is included in the supplemental file) due to space limitation.

5.2 Baseline Methods.

We extract five kinds of feature for a thing to train a set of binary SVM classifiers (the whole set of features are shown in Table 4), and aggregate of the labels positively predicted by all the independent binary classifiers as things categorization result. Among the five features, $F_{Cluster}$ has not been discussed before, which means deriving thing's implicit features by clustering thing's interactions directly instead of using the proposed graphical modeling approach. The feature extraction process of $F_{Cluster}$ is similar to F_{Latent}, which firstly constructs top-k relation graph of things based on thing's interactions and then performs RWR on the relation graph to extract thing's implicit features.

Based on these features, we evaluated 8 methods for things categorization as listed in Table 6. Among the 8 methods, TE needs to extract TF/IDF features from thing's text profiles. Thing's text profiles have not been discussed before, we detail it in the following. For MIT datasets, we utilize the activity description that things participated in as thing's text profiles to extract text-based features. For instance , a *Closet* participate in three kinds of activities: $\{$*Preparing a snack,Doing laundry,Bathing*$\}$, then we combine the descriptions of the three activities (as shown in Table 5) as its text description, i.e., "Domestic work,Preparing a snack,Clean house,Personal needs,Personal hygiene". For things of our collected dataset, we utilize the text description from E-commerce company (e.g., ebay [2] and Taobao [3]) as thing's text profiles to extract text-based features.

TABLE 4: Features for things categorization

Features	Description
F_{text}	The text-based feature using TF/IDF (Section 4.2.1)
F_S	The spatial pattern of interaction behaviours(Section 4.2.2)
F_T	The temporal pattern of interaction behaviours(Section 4.2.3)
$F_{Cluster}$	The label probabilities for labels by clustering interactions
F_{Latent}	The label probabilities for labels on a thing (Section 4.1)

TABLE 5: An example of text description for MIT datasets

Activity	Description
Preparing a snack	Domestic work,Preparing a snack
Doing laundry	Domestic work,Clean house
Bathing	Personal needs,Personal hygiene

5.3 Evaluation Metrics.

We use two widely used metrics (Hamming Loss and F-measure) for multi-label classification to evaluate the performance, which are defined as:

– **Hamming Loss**, which is used to evaluate how many times a thing is misclassified, i.e., a label not associated with a thing is predicted or a label associated with the thing is not predicted. Formally, the Hamming Loss is defined as:

$$HammingLoss = \frac{1}{|D_{te}|} \sum_{i \in D_{te}} \frac{HD(v_i, \overline{v_i})}{|L|} \tag{18}$$

where $|D_{te}|$ is the number of test samples, $|L|$ is the number of labels, v_i and $\overline{v_i}$ are the ground truth and prediction vectors for testing thing o_i. $HD(v_i, \overline{v_i})$ is the hamming distance between v_i and $\overline{v_i}$.

– **F-measure**, which is a particular kind of average between precision and recall that has been widely used in many

2. http://www.ebay.com/
3. http://www.taobao.com/

TABLE 6: Methods for comparison

Method	Description
TE	Using F_{text} to train SVM classifier
S	Using F_S to train SVM classifier
T	Using F_T to train SVM classifier
EF	Combination of F_{text}, F_S and F_T to train SVM classifier
CL	Using $F_{Cluster}$ to train SVM classifier
IF	Using F_{Latent} to train SVM classifier
CL+EF	Combination of F_{text}, F_S, F_T and $F_{Cluster}$ to train SVM classifier
IF+EF	Combination of F_{text}, F_S, F_T and F_{Latent} to train SVM classifier

TABLE 7: The profile similarity features of things

Feature	Description
$z_1^{(ij)}$	1 if o_i and o_j have the same manufacturer, 0 otherwise
$z_2^{(ij)}$	1 if o_i and o_j are owned by the same user, 0 otherwise
$z_3^{(ij)}$	1 if o_i and o_j are located in the same place, 0 otherwise
$z_4^{(ij)}$	1 if o_i and o_j have the same functionality, 0 otherwise
$z_5^{(ij)}$	1 if o_i and o_j have the same color, 0 otherwise

prediction problems including binary classification, multi-label classification and structured output prediction. Let v_i and $\overline{v_i}$ denote the ground truth and prediction vectors for testing thing o_i, the F-measure is defined as:

$$F-measure = \frac{1}{|D_{te}|}\sum_{i\in D_{te}}\frac{2\times|v_i\bigcap\overline{v_i}|}{v_i+\overline{v_i}} \tag{19}$$

Due to the small size of dataset, we perform five-fold cross-validation and also report the corresponding standard deviation as error bar for each case. Firstly, each dataset was randomly split into 5 equal groups (N=5). Secondly, trains the model on 4 groups of data, and records the error for the excluded data. This process is repeated 10 times, each time records the performance (Hamming Loss and F-measure) for the excluded data set. Finally, this whole procedure is repeated 10 times with different random splits of the data to produce the final results. We report the mean of the performance and standard deviation produced with the 50 (5x10) sets of test data as the ultimate experiment results.

5.4 Parameter Setting.

For our dataset, we utilize five attributes to capture the profiles similarity of each pair of things (o_i, o_j), which is defined as: $z^{(ij)} = [z_1^{(ij)},...,z_5^{(ij)}]^T$ and the meaning of the five features are reported in Table 7. For MIT datasets, we utilize two attributes to capture the profiles similarity for each pair of things (o_i, o_j): $z^{(ij)} = [z_2^{(ij)}, z_3^{(ij)}]^T$, since the other three types of information are not provided.

5.5 Experiment Results.

We conduct two groups of experiments and report their results. The first group is to perform parameter turning for models (IF and CL) using implicit features. The second group is to compare the effectiveness and efficiency of models using explicit features and implicit features for things categorization, respectively.

5.5.1 Impact of Model Parameters

Tuning algorithm parameters, such as the parameter λ of RWR process and the number of neighbours (k) for constructing top-k relation graph, are critical to the performance of methods (IF and CL) using implicit feature. We tune λ and k on the three datasets, and plot the Hamming Loss and F-measure with different values in Figure 9.

Set the number of neighbours for constructing relation graph equals to 10, we test the performance of IF by varying λ, and present the results in Figure 9a and Figure 9b. As mentioned earlier, for each dataset, we perform 5-fold cross-validation and repeat 10 times for each cross-validation, and report the corresponding standard deviation as error bar for each case. From the figure, it is observed that best Hamming Loss and F-measure are reached when λ = 0.7. We further observe both the Hamming Loss and F-measure slightly increase with the increasing of λ from 0.1 to 0.7, and then decrease when λ is greater than 0.7. The reason is that the convergence of RWR is determined by λ, i.e., the greater λ results in the faster convergence, and further bring better performance. But a larger λ will cause a high probability to back to the target thing, thus reducing the number of neighbours with high latent relation strength and further decreasing the performance. As shown in Figure 9e and Figure 9f, similar results are also observed in turning λ for CL (for example, the mean of F-measure and Hamming Loss achieve the best results (43.74% and 38.16%, respectively) when $\lambda = 0.7$ for MIT S1.).

Set $\lambda = 0.7$, Figure 9c Figure 9d report the performance of IF with different number of neighbours (k), where k is in the range [5,10,...,40], because there are 342 things in total and a greater value of k is usually ignored when constructing top-k relation graph. From the two Figures, we observe the best Hamming Loss and F-measure are reached when k = 10 and 15 on MIT datasets and Dataset 3, respectively. The is because that our dataset has much more things than MIT datasets. However, the performance decreases with increasing k, since a greater k will bring in some noisy neighbours thus may decrease the performance. Similar results are also observed in turning k for CL, for example, the mean of F-measure and Hamming Loss achieve the best results (50.88% and 30.52%, respectively) when $k = 15$ for Dataset 3.

5.5.2 Explicit Features vs Implicit Features

In this part, we compare the effectiveness and efficiency of models using explicit features and implicit features for things categorization. We evaluate the categorization effectiveness from two aspects: 1) the performance of explicit Features (TE, S, T and EF), implicit Features (CL and IF) and their hybrid (EF+CL and EF+IF) with fixed mark-off rate. Here the mark-off rate means the ratio of unlabeled things. In this case, we perform 5-fold cross-validation and repeat 10 times for each cross-validation, and report the mean and the corresponding standard deviation as error bar; 2) the performance of five methods (EF, CL, IF, EF+CL and EF+IF) with different mark-off rate. In this case, we randomly removed the category labels of a certain percentage (named testing things with mark-off rate) from each category of the ground-truth dataset. The methods are used to recover the category labels for those testing things. For each case, we report the average performance and corresponding standard deviation as error bars by repeating the experiments 10 times.

Performance Comparison. We compare the performance of 8 methods (TE, S, T, EF, CL, IF, EF+CL, EF+IF) on the three datasets, as shown in Table 8.

For methods (TE, S, T and EF) using explicit features, we can observe from Table 8 that: 1) For methods using explicit features, EF that combines the spatial, temporal and text information always outperforms the baseline methods (TE, S, T), which merely utilizes one type of feature. For instance, EF outperforms TE by 22.09% and 11.73% on MIT S1 and MIT S2 in terms of F-measure, respectively. This result suggests that, fusing spatial, temporal and text feature is beneficial for improving the performance of things categorization; 2) For Dataset 3, TE achieves much worse performance than S and T in terms of both Hamming Loss and F-measure. The reason is that Dataset 3 utilizes the description from E-commerce company as thing's text description to extract text-based features, which usually are short and noisy. For example, a text description from E-commerce site for cabinet

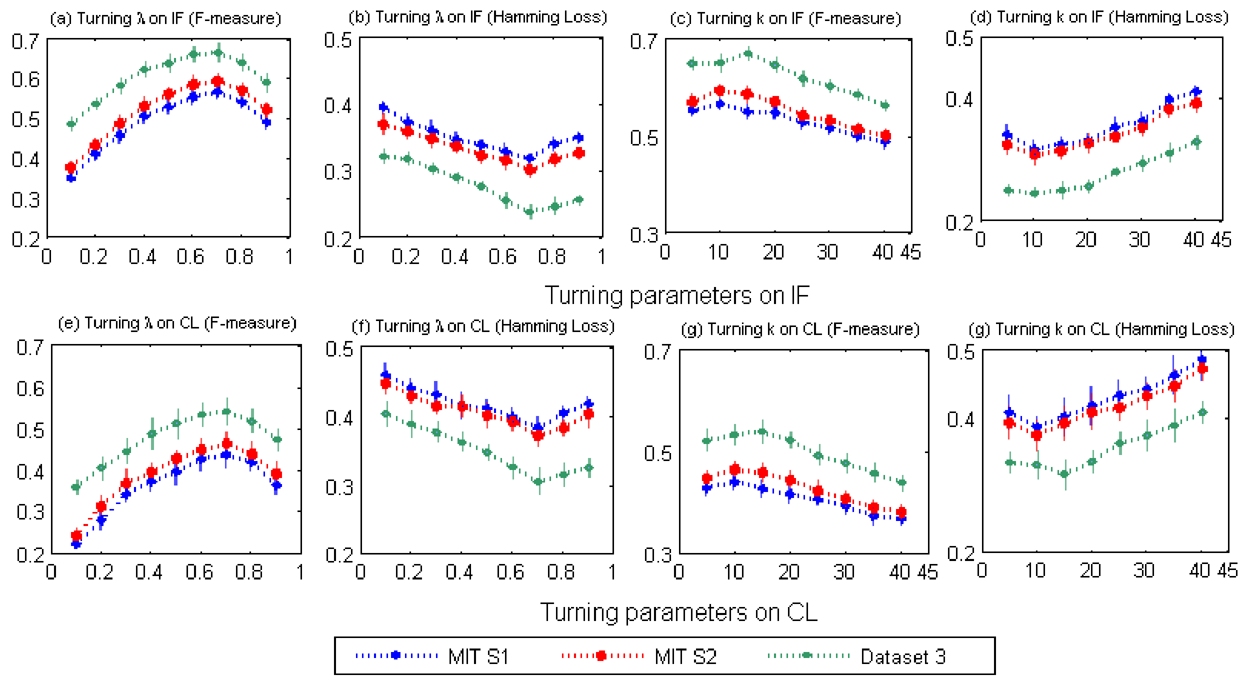


Fig. 9: Impact of parameters (λ and k) for IF and CL (mean plus standard error bars)

TABLE 8: Performance (%) comparison with different baselines (F1=F-measure, HL=Hamming Loss)

Method	MIT S1		MIT S2		Dataset 3	
	F1 (%)	HL(%)	F1 (%)	HL(%)	F1 (%)	HL(%)
TE	47.31 ± 2.84	39.42 ± 2.47	59.05 ± 2.38	45.71 ± 1.94	30.17 ± 2.15	43.31 ± 2.42
S	35.02 ± 2.71	46.31 ± 1.87	50.08 ± 2.62	40.17 ± 2.36	41.59 ± 3.12	37.94 ± 2.86
T	39.42 ± 2.67	40.15 ± 2.89	57.16 ± 3.12	36.31 ± 2.79	50.62 ± 3.23	36.83 ± 2.89
EF	69.38 ± 2.47	26.89 ± 2.06	70.78 ± 2.95	24.28 ± 1.89	66.14 ± 1.83	22.11 ± 2.37
CL	43.74 ± 3.45	38.16 ± 2.79	46.38 ± 3.77	37.15 ± 2.93	53.88 ± 3.25	30.52 ± 3.76
IF	56.43 ± 2.95	31.85 ± 1.85	59.22 ± 2.46	30.13 ± 2.18	64.37 ± 2.79	23.93 ± 2.12
EF+CL	72.69 ± 2.74	24.17 ± 2.18	73.17 ± 3.35	20.71 ± 2.76	73.58 ± 2.53	20.19 ± 2.49
EF+IF	77.15 ± 2.36	20.89 ± 2.14	79.92 ± 3.27	16.58 ± 2.17	83.51 ± 2.78	14.13 ± 2.86

is "the home cabinet with simplicity of modern style, two / three door", then the keywords extracted from this text are {cabinet, door}. Therefore, the cabinet is likely to be misclassified as door. The results suggest that text-based features based on the well-known TF/IDF feature for things categorization are not effective in IoT, since the text descriptions of things are usually short and noisy.

For methods using implicit features (CL and IF), we observe from Table 8 that IF outperforms CL significantly in terms of both F-measure and Hamming Loss, showing the advantages of using graphical model to mine thing's latent relation strength and derive implicit features. For instance, the F-measure of IF is about 56.43% on MIT S1, 59.22% on MIT S2 and 64.37% on Dataset 3, the performance is improved by 12.69% (MIT S1), 12.84% (MIT S2) and 10.49% (Dataset 3) compare with CL respectively. The reasons for better precision are: 1) CL extracts implicit features from thing's relation graph by clustering thing's interactions directly, which are powerless to capture information from things without interactions for deriving categorization features. On the contrary, our proposed graphical model can be applied in two ways. First, if both things attributes set and their interaction behaviours are known, we can estimate the latent relation strength based on both things attributes set and their interaction behaviours. Second, when the interaction behaviours are unobserved, we can estimate thing's latent relation from their attributes similarity. This in fact demonstrates a strength of our proposed graphical model: the lower part of the model is generative so that the overall model will not suffer much from missing interaction behaviours during training. Once the model is learned, for new data the latent variables can be inferred using only the upper level of variables in the model; 2) our proposed graphical model introduces a few auxiliary variables that capture auxiliary causes of thing's interactions, which can moderate the effect of latent relation strength on interaction behaviours thus increase the accuracy of the model. CF intuitively clusters thing's interactions for predicting labels, particularly the betweenness centrality is strongly biased towards nodes with high degree, or nodes that are central in large local groups of nodes [15]. Moreover, the random walk may terminate with a high likelihood when reaches an unlabeled nodes with numerous interactions during extracting implicit features [27].

For methods using both explicit features and implicit features (EF+CL and EF+IF), we can observe from Table 8: 1) the methods using hybrid features outperform merely using explicit feature (EF) or implicit features (CL or IF) significantly in terms of both

Hamming loss and F-measure. For example, EF+IF outperforms EF by around 7.97% on MIT S1, 9.14% on MIT S2, and around 17.37% on Dataset 3 in terms of F-measure. This result shows the unified method (EF+IF) is superior to the state-of-art method in terms of categorization effectiveness, which suggests that the learnt latent relation strength from thing's interactions behaviours can significantly boost things categorization; 2) the performance improvement of our dataset is much higher than the other two datasets for both EF+CL and EF+IF. This is because our dataset consists of much more interaction behaviours than MIT datasets, which can be utilized to learn thing's latent relation strength better.

Performance with different Mark-off Rates. We investigate the impact of different mark-off rates to the performance of EF, CL, IF, EF+CL and EF+IF. As shown in Figure 10, the performance of all methods with different feature sets degrade to some extent as the mark-off rate increases. Nevertheless, EF+CL and EF+IF show better performance consistently than CL and IF over all mark-off rates as they include both explicit features and implicit features. For example, the F-measure of EF+IF on our dataset is 78.44% when the mark-off rate is 40%, while 54.02% of EF with the same mark-off rate. This clearly demonstrates the effectiveness of hybrid features by combining both implicit feature and explicit feature. We also observe the unified method (EF+CL and EF+IF) can achieve considerable performance even when the mark-off rates are relatively high, while explicit features (EF) perform poorly with few labeled things to train model. For example, the F-measure of IF+EF on MIT S2 drops 5.6% when the mark-off rate increases from 40% to 60%, while 18.3% of EF with the same condition. This is because explicit features require either enough text profiles or obvious spatial-temporal pattern to extract features for training model, while hybrid features can utilize implicit feature extracted from thing's relation graph to boost things categorization even with few labeled samples.

Efficiency of Extracting Implicit Features. The time complexity of the proposed things categorization model consists of three parts: 1) the first part is the graphical model for inferring the model parameters. Since this part can be done in offline phase, the learned parameter values can be applied to estimate the latent relation strength for a new pair of things in constant time; 2) the second part is extracting implicit features. We have proved in Section 4.1 that the time complexity of this part is generally linear in the number of edges and nodes in the relation graph. We will show the feature extraction process will converge fast in the following experiments; 3) the third part is multi-label classification by SVM model, which is scalable to big datasets as suggested by a few studies [21], [41]. Therefore, the proposed things categorization model is scalable to large dataset.

We use $\varepsilon = 10^{-5}$ as the termination condition of iterations. The iteration numbers when implicit feature extraction process terminates are plotted in Figure 11a by varying the training sample sizes. From this figure, we observe the iteration numbers when implicit feature extraction process terminates are less than 400 for the three datasets, which shows the feature extraction process is scalability to large dataset. We further report the run time with different training sample sizes by setting the iteration number as 500 in Figure 11b. From this figure, we can observe time complexity is generally linear in the ratio of unlabeled nodes as expected.

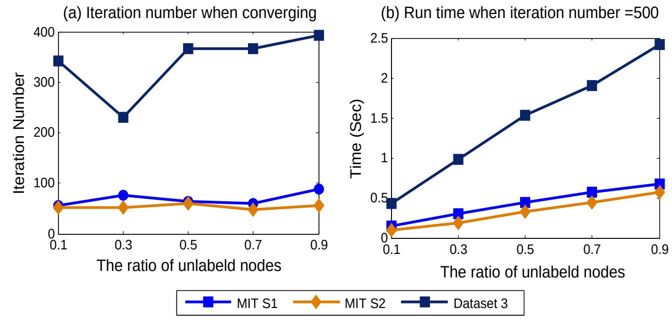


Fig. 11: The efficiency of extracting implicit features: (a) the comparison of iteration numbers when converging; (b) the comparison of run time

6 Conclusion

In this paper, we investigate things categorization problem, which aims to automatically associate things with semantic tags in IoT. Things categorization is a crucial pre-requisite for a few valuable services in IoT, such as things browsing, searching and recommendation. We propose a novel things categorization algorithm which learns a binary SVM classifier for each type of label. For training SVM classifier, we extract two kinds of features: explicit features and implicit features. More exactly, we extract three types of explicit features: text feature from thing's text profiles, spatial feature from thing's location distribution and temporal feature from the hourly distribution of thing's interaction. For extracting the implicit feature, we firstly construct a relation graph based on the learnt latent relation strength from thing's interaction behaviours, then exploit thing's relatedness to generate implicit feature. Finally, we conduct a comprehensive experimental study based on three real datasets . Experimental results show that this proposed approach significantly outperforms state-of-art methods based on explicit features, showing the superiority of our approach and also supporting the assumption that the latent relation strength among things can boost things categorization.

As future work, we plan to facilitate more valuable services in IoT based on the learnt latent relation strength of things from their interaction behaviours, such as, things searching, clustering and service discovery.

Acknowledge

This work is sponsored by the National Basic Research 973 Program of China (No. 2015CB352403), the National Natural Science Foundation of China (NSFC) (61261160502, 61272099), the Program for National Natural Science Foundation of China / Research Grants Council (NSFC/RGC)(612191030), the Program for Changjiang Scholars and Innovative Research Team in University (IRT1158, PCSIRT), the Scientific Innovation Act of STCSM (13511504200), and EU FP7 CLIMBER project (PIRSES-GA-2012-318939).

References

[1] I. Atanasov, A. Nikolov, E. Pencheva, R. Dimova, and M. Ivanov. An approach to data annotation for internet of things. *International Journal of Information Technology and Web Engineering (IJITWE)*, 10(4):1–19, 2015.

[2] L. Atzori, A. Iera, and G. Morabito. From" smart objects" to" social objects": The next evolutionary step of the internet of things. *IEEE Communications Magazine*, 52(1):97–105, 2014.

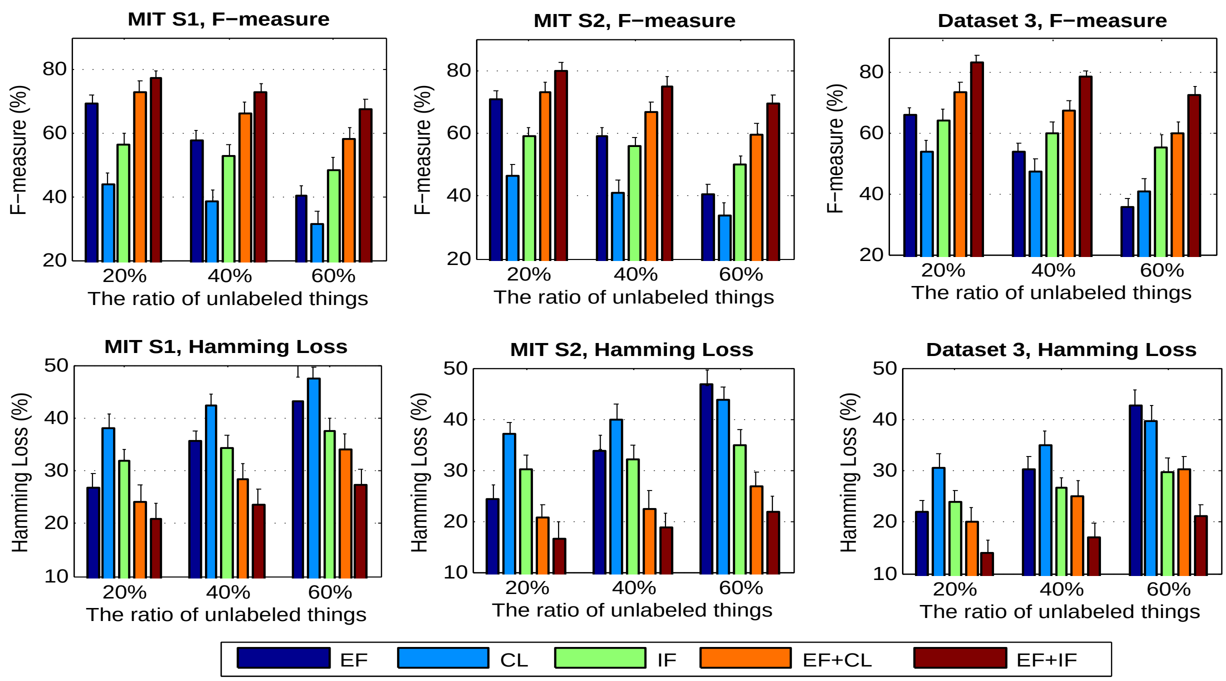


Fig. 10: Impact of mark-off rate on the three datasets

[3] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

[4] J. Byun, S. H. Kim, and D. Kim. Lilliput: Ontology-based platform for iot social networks. In *Services Computing (SCC), 2014 IEEE International Conference on*, pages 139–146, 2014.

[5] X. Chen, Y. Xia, P. Jin, and J. A. Carroll. Dataless text classification with descriptive LDA. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA.*, pages 2224–2231.

[6] Y. Chen, J. Zhou, and M. Guo. A context-aware search system for internet of things based on hierarchical context model. *Telecommunication Systems*, 62(1):77–91, 2016.

[7] M. Coccoli and I. Torre. Interaction with objects and objects annotation in the semantic web of things. In *DMS*, pages 383–390, 2014.

[8] S. De, P. Barnaghi, M. Bauer, and S. Meissner. Service modelling for the internet of things. In *Computer Science and Information Systems (FedCSIS), 2011 Federated Conference on*, pages 949–955. IEEE, 2011.

[9] F. C. Delicato, P. F. Pires, and T. Batista. *Resource Management for Internet of Things*. Springer, 2017.

[10] F. Fouss, A. Pirotte, J.-M. Renders, and M. Saerens. Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation. *IEEE Transactions on knowledge and data engineering*, 19(3):355–369, 2007.

[11] J. Friedman, T. Hastie, and R. Tibshirani. *The elements of statistical learning*, volume 1. Springer series in statistics Springer, Berlin, 2001.

[12] T. Gu, Z. Wu, X. Tao, H. K. Pung, and J. Lu. epsicar: An emerging patterns based approach to sequential, interleaved and concurrent activity recognition. In *Seventh Annual IEEE International Conference on Pervasive Computing and Communications, PerCom 2009, 9-13 March 2009, Galveston, TX, USA*, pages 1–9, 2009.

[13] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami. Internet of things (iot): A vision, architectural elements, and future directions. *Future Generation Computer Systems*, 29(7):1645–1660, 2013.

[14] T. Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 50–57. ACM, 1999.

[15] I. Hulpus, C. Hayes, M. Karnstedt, and D. Greene. Unsupervised graph-based topic labelling using dbpedia. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pages 465–474. ACM, 2013.

[16] S. D. T. Kelly, N. K. Suryadevara, and S. C. Mukhopadhyay. Towards the implementation of iot for environmental condition monitoring in homes. *IEEE Sensors Journal*, 13(10):3846–3853, 2013.

[17] J. E. Kim, A. Maron, and D. Mosse. Socialite: A flexible framework for social internet of things. In *2015 16th IEEE International Conference on Mobile Data Management*, volume 1, pages 94–103. IEEE, 2015.

[18] T. K. Landauer. *Latent semantic analysis*. Wiley Online Library, 2006.

[19] G. Limaye, S. Sarawagi, and S. Chakrabarti. Annotating and searching web tables using entities, types and relationships. *Proceedings of the VLDB Endowment*, 3(1-2):1338–1347, 2010.

[20] T. Liu, Z. Chen, B. Zhang, W.-y. Ma, and G. Wu. Improving text classification using local latent semantic indexing. In *ICDM. Fourth IEEE International Conference on*, pages 162–169. IEEE, 2004.

[21] T.-Y. Liu, Y. Yang, H. Wan, H.-J. Zeng, Z. Chen, and W.-Y. Ma. Support vector machines classification with a very large-scale taxonomy. *ACM SIGKDD Explorations Newsletter*, 7(1):36–43, 2005.

[22] L. K. McDowell. Relational active learning for link-based classification. In *Data Science and Advanced Analytics (DSAA), 2015. 36678 2015. IEEE International Conference on*, pages 1–10. IEEE, 2015.

[23] I. Nevat, G. W. Peters, K. Avnit, F. Septier, and L. Clavier. Location of things: Geospatial tagging for iot using time-of-arrival. *IEEE Transactions on Signal and Information Processing over Networks*, 2(2):174–185, 2016.

[24] A. Onan, S. Korukoglu, and H. Bulut. Lda-based topic modelling in text sentiment classification: An empirical analysis. *Int. J. Comput. Linguistics Appl.*, 7(1):101–119, 2016.

[25] H. Qian, Y. Mao, W. Xiang, and Z. Wang. Recognition of human activities using svm multi-class classifier. *Pattern Recognition Letters*, 31(2):100–111, 2010.

[26] C. Shen, T. Li, and C. H. Ding. Integrating clustering and multi-document summarization by bi-mixture probabilistic latent semantic analysis (plsa) with sentence bases. In *AAAI*, 2011.

[27] P. Talukdar and K. Crammer. New regularized algorithms for transductive learning. *Machine Learning and Knowledge Discovery in Databases*, pages 442–457, 2009.

[28] E. M. Tapia, S. S. Intille, and K. Larson. Activity recognition in the home using simple and ubiquitous sensors. In *International Conference on Pervasive Computing*, pages 158–175, 2004.

[29] S. Tata and J. M. Patel. Estimating the selectivity of tf-idf based cosine similarity predicates. *ACM Sigmod Record*, 36(2):7–12, 2007.

[30] Y. Tian, Q. Yang, T. Huang, C. X. Ling, and W. Gao. Learning contextual dependency network models for link-based classification. *IEEE transactions on knowledge and data engineering*, 18(11):1482–1496, 2006.

[31] T. Tietz, J. Waitelonis, J. Jäger, and H. Sack. refer: a linked data based text annotation and recommender system for wordpress. In *Proceedings*

of the ISWC 2016 Posters & Demonstrations Track, Kobe, Japan, October 19, 2016., 2016.

[32] T. Van Kasteren, A. Noulas, G. Englebienne, and B. Kröse. Accurate activity recognition in a home setting. In *Proceedings of the 10th international conference on Ubiquitous computing*, pages 1–9. ACM, 2008.

[33] D. Wang, H. Zhang, R. Liu, and W. Lv. Feature selection based on term frequency and t-test for text categorization. *CoRR*, abs/1305.0638, 2013.

[34] Y. Wang, N. J. Yuan, D. Lian, L. Xu, X. Xie, E. Chen, and Y. Rui. Regularity and conformity: location prediction using heterogeneous mobility data. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1275–1284. ACM, 2015.

[35] Z. Wu, Y. Xu, C. Zhang, Y. Yang, and Y. Ji. Towards semantic web of things: From manual to semi-automatic semantic annotation on web of things. In *International Conference on Big Data Computing and Communications*, pages 295–308. Springer, 2016.

[36] Z. Xu, R. Jin, K. Huang, M. R. Lyu, and I. King. Semi-supervised text categorization by active search. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 1517–1518. ACM, 2008.

[37] X.-B. Xue and Z.-H. Zhou. Distributional features for text categorization. *IEEE Transactions on Knowledge and Data Engineering*, 21(3):428–442, 2009.

[38] L. Yang, X. Cao, D. Jin, X. Wang, and D. Meng. A unified semi-supervised community detection framework using latent space graph regularization. *IEEE transactions on cybernetics*, 45(11):2585–2598, 2015.

[39] L. Yao and Q. Z. Sheng. Correlation discovery in web of things. In *Proceedings of the 22nd International Conference on World Wide Web*, pages 215–216, 2013.

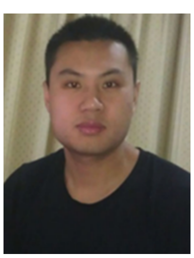
[40] L. Yao, Q. Z. Sheng, B. J. Gao, A. H. Ngu, and X. Li. A model for discovering correlations of ubiquitous things. In *2013 IEEE 13th International Conference on Data Mining*, pages 1253–1258, 2013.

[41] H. Yu, J. Yang, and J. Han. Classifying large data sets using svms with hierarchical clusters. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 306–315. ACM, 2003.

[42] S. Zelikovitz and F. Marquez. Transductive learning for short-text classification problems using latent semantic indexing. *International Journal of Pattern Recognition and Artificial Intelligence*, 19(02):143–163, 2005.

[43] R. Zhang, L. Zhang, X.-J. Wang, and L. Guan. Multi-feature plsa for combining visual features in image annotation. In *Proceedings of the 19th ACM international conference on Multimedia*, pages 1513–1516. ACM, 2011.

[44] Z. Zheng, Y. Chen, T. He, F. Li, and D. Chen. Weight-rss: a calibration-free and robust method for wlan-based indoor positioning. *International Journal of Distributed Sensor Networks*, 2015:55, 2015.

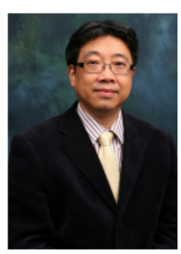


Yuanyi Chen received his BSc degree from Sichuan University in 2010, the master's degree from Zhejiang University in 2013. From September 2014 to September 2015, he was a jointly-supervised PhD candidate at Hong Kong Polytechnic University. He is currently a PhD candidate at Department of Computer Science and Engineering, Shanghai Jiao Tong University. His research topic includes the Internet of Things and mobile computing.

Jingyu Zhang received the BSc degree from Hunan Normal University in 2008 and the ME degree from Chongqing Jiaotong University in 2011. He is currently a PhD candidate at the Department of Computer Science and Engineering, Shanghai Jiao Tong University. His research interests include mobile computing, computer architecture, cache optimization, and adaptive video streaming.

Liting XU received the BSc degree and Master degree from Jiangsu University, China, in 2005 and 2007, respectively, both in computer science and applications. During 20082011, she was a Software R&D Engineer at TERAOKA Weigh-System Pte. Ltd, Singapore. From 2011 to 2012, she was with ASM Technology Singapore Pte Ltd, as a Software R&D Engineer. After that, she had been a Software Modelling Engineer in Continental Automotive Singapore Pte Ltd. In May. 2013, she joined Shanghai Jiao Tong University, China, as Research Engineer. Her present research interests include computer supported cooperative work (CSCW), applications of distributed computing and web visualization on the Grid.

Minyi Guo received the B.S. and M.E. degrees in Computer Science from Nanjing University, China in 1982 and 1986, respectively. From 1986 to 1994, he had been an assistant professor of the Department of Computer Science at Nanjing University. He received the Ph.D.degree in information science from University of Tsukuba, Japan in 1998. He is currently a chair professor and the head of the Department of Computer Science and Engineering at Shanghai Jiao Tong University. His research interests include parallel computing, distributed computing and mobile computing. He has more than 300 publications in major journals and international conferences in these areas. He is on the editorial board of the journals IEEE Transactions on Parallel and Distributed Systems and IEEE Transactions on Computers. He is a senior member of the IEEE, a member of the ACM, IEICE IPSJ, and CCF.



Jiannong Cao received the BSc degree from Nanjing University, China, in 1982, and the MSc and PhD degrees from Washington State University, USA, in 1986 and 1990, all in computer science. He is currently a chair professor and the head of the Department of Computing at Hong Kong Polytechnic University. His research interests include parallel and distributed computing, computer networks, mobile and pervasive computing, fault tolerance, and middle-ware. He co-authored four books, coedited nine books, and published more than 300 technical papers in major international journals and conference proceedings. He has directed and participated in numerous research and development projects and, as a principal investigator, obtained over HK$25 million grants. He is the chair in Technical Committee on Distributed Computing, IEEE Computer Society. He has served as an associate editor and a member of editorial boards of many international journals, and a chair and a member of organizing/program committees for many international conferences. He is a fellow of the IEEE, a member of the ACM, and a senior member of China Computer Federation.