Doctoral Dissertation Defense

Intelligent Data Mining Techniques for Automatic Service Management

Ph.D. Candidate: Qing Wang

Major Professor: Professor S. S. Iyengar

Co-advisor: Professor Shu-Ching Chen

Committee Members: Professor Mark A. Finlayson

Professor Ning Xie

Professor Debra VanderMeer

2018-11-07

Outline

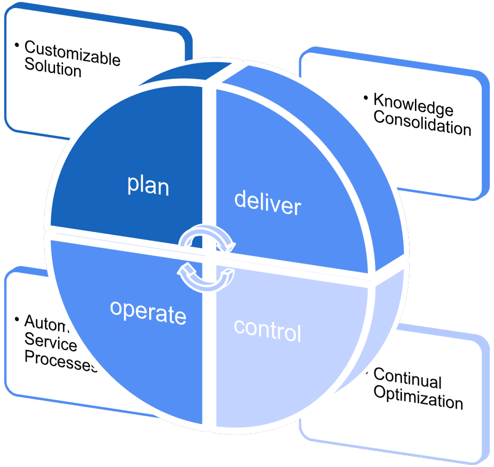
- Introduction
- Research Problems
  - Learn Human Intelligence by Domain Knowledge Base Construction
  - Learn Automation Intelligence by Hierarchical Multi-armed Bandit Model
    - Multi-armed Bandit Problems with Dependent Arms
    - Hierarchical IT Automation Recommendation Modeling
    - Hierarchical Multi-armed Bandit Model
  - Learn Automation Intelligence by Interactive Collaborative Topic Regression Model
    - Interactive Collaborative Filtering Problem
    - Matrix-Factorization based IT Automation Recommendation Modeling
    - Interactive Collaborative Topic Regression Model
- Summary

Introduction

- Today, the success of a business is closely intertwined with its IT performance.

- IT Service Management (ITSM) refers to the all the activities that are performed to **plan**, **deliver**, **operate** and **control** the IT services provided to customers (i.e., business enterprises).

- Traditional ITSM technologies are impossible to handle the challenges introduced by today's growing complexity of IT environment.

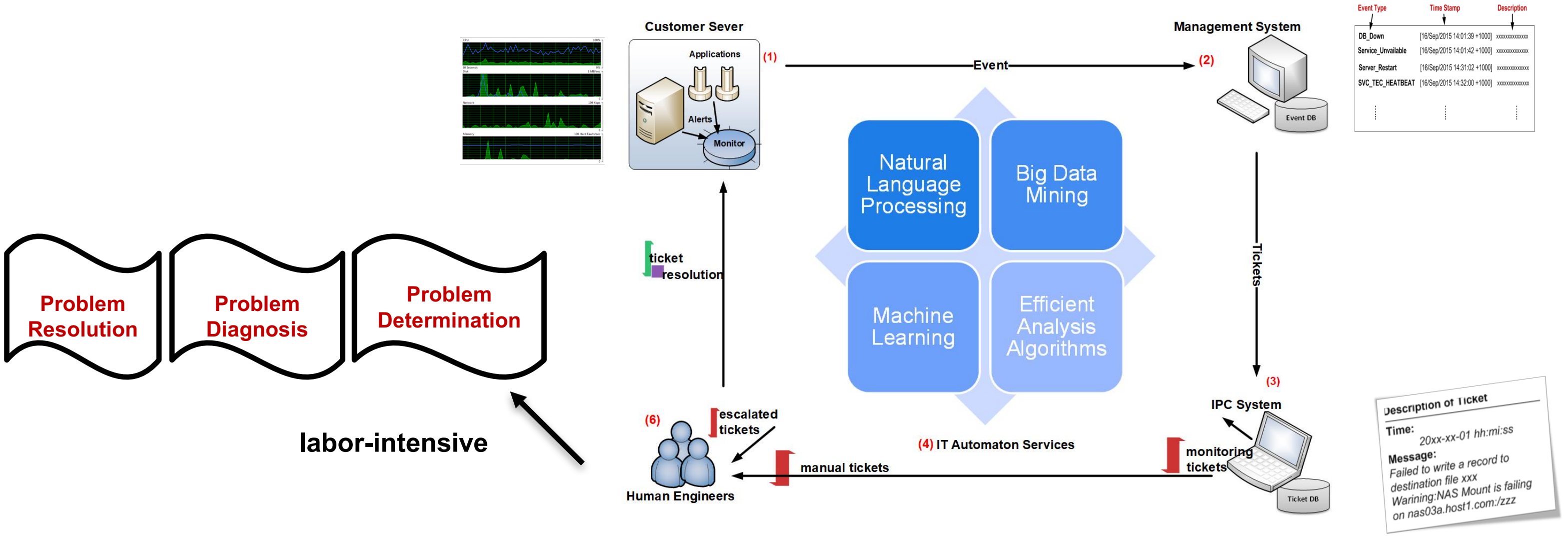


Introduction

- Many ITSM products are booming from different companies. Aiming at providing **higher quality** and **more complex** services, IT service providers are increasingly seeking cognitive techniques to automate or optimize their services.

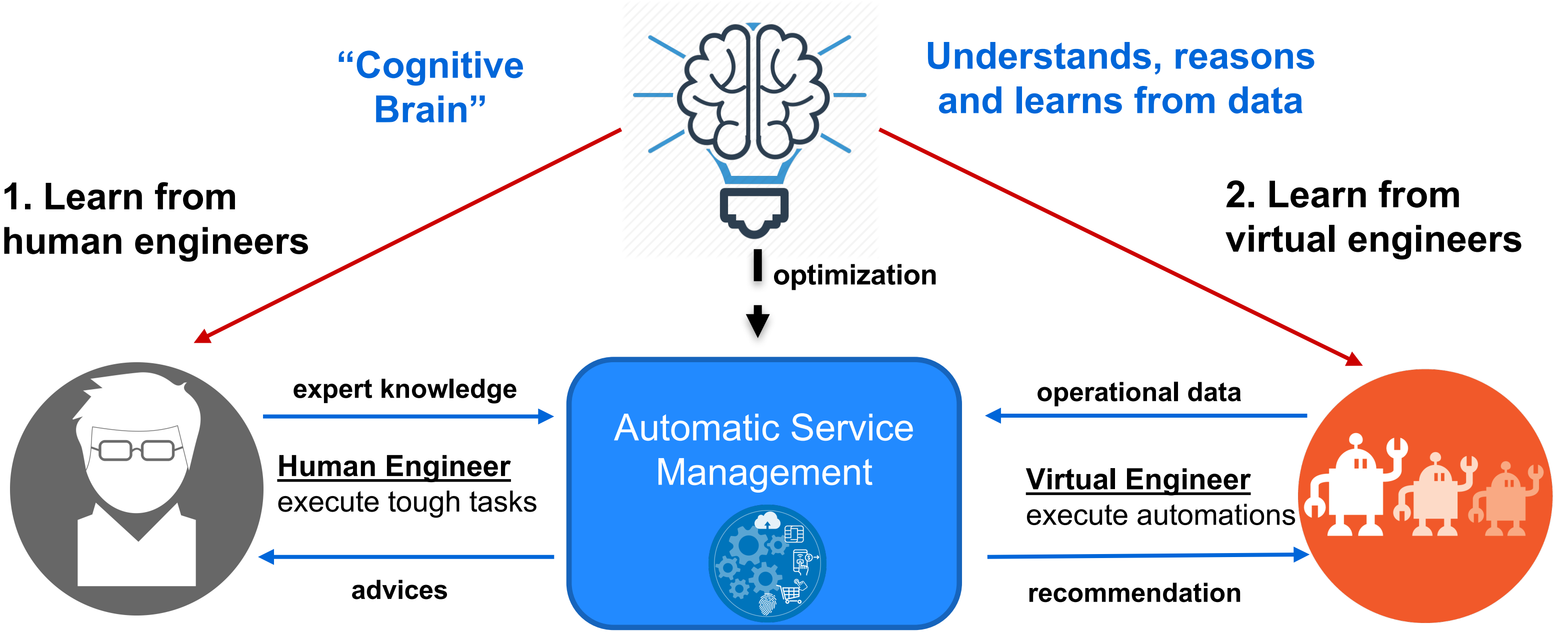
Background

A typical workflow of IT Service Management involves a mix of **human engineers**, **process** and **information technology**.

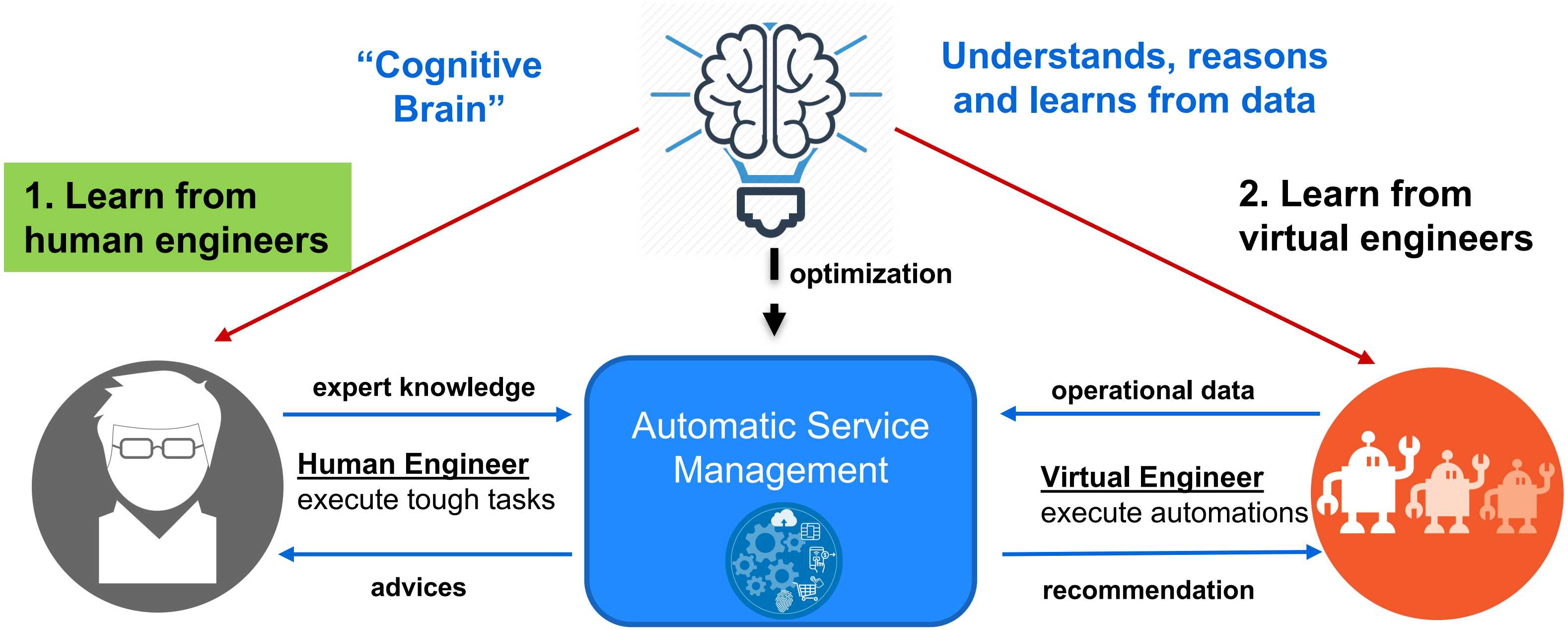


A typical workflow of IT service management.

Overview of Research Problems



Overview of Research Problems



Outline

- Introduction
- Research Problems
    - Learn Human Intelligence by Domain Knowledge Base Construction
    - Learn Automation Intelligence by Hierarchical Multi-armed Bandit Model
        - Multi-armed Bandit Problems with Dependent Arms
        - Hierarchical IT Automation Recommendation Modeling
        - Hierarchical Multi-armed Bandit Model
    - Learn Automation Intelligence by Interactive Collaborative Topic Regression Model
        - Interactive Collaborative Filtering Problem
        - Matrix-Factorization based IT Automation Recommendation Modeling
        - Interactive Collaborative Topic Regression Model
- Summary

Motivation

Structured fields:

often inaccurate or incomplete especially information which is not generated by monitoring systems.

Unstructured text:

written by human engineers in natural language. Potential knowledge includes:

1. What happened? **Problem**
2. What troubleshooting was done? **Activity**
3. What was the resolution? **Action**

STRUCTURED

TICKET IDENTIFIER:	WPPWA544:APPS:LogAdapter:NALAC:STARACTUAT_6600				
NODE	**FAILURECODE**	**ORIGINAL SEVERITY**	**OSTYPE**	**COMPONET**	**CUSTOMER**
WPPWA544	UNKNOWN	4	WIN2K3	APPLICATION	XXXX

UNSTRUCTURED

TICKET SUMMARY:	STARACTUAT_6600 03/01/2014 04:30:28 STARACTUAT_6600 GLACTUA Market=CAAirMiles:Report_ID=MRF600:ReportPeriod From: 2014/02/01 to 2014/02/28:ErrorDesc=For CAAirMiles Actuate is out of balance with STAR BalanceMRF600 & MRF601 Counts. Reconcilation Difference = 2MRF600 & MRF601 Net Fee. Reconcilation Difference = 25MRF600 & MRF601 Gross Fee .Reconcilation Difference = 25

UNSTRUCTURED

RESOLUTION
ProblemSolutionText:***** Updated by GLACTUA ****** Problem Reported : Reconciliation difference Root cause : Reconciliation was run before all reports completed. This is as per the new SLAs. Solution provided : *Reconciliation was re-run after the next set of reports completed.*There was no user impact. Closure code : WRKS_AS_DSIGND RCADescription:***** Updated by GLACTUA ****** Problem Reported : Reconciliation difference Root cause : Reconciliation was run before all reports completed. This is as per the new SLAs. Solution provided : Reconciliation was re-run after the next set of reports completed.There was no user impact. Closure code : WRKS_AS_DSIGND

A ticket in IT service management and its corresponding resolution are given.

Challenges

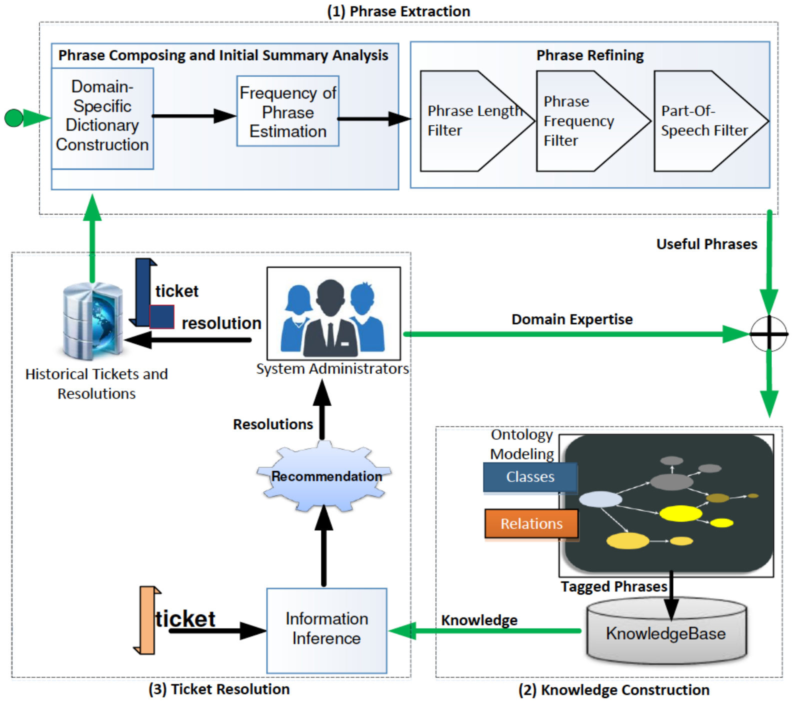
- Challenge 1: Structured fields contribute little information to the problem inference.
- Challenge 2: The ambiguity brought by the free-form text in both ticket summary and resolution poses difficulty in information extraction and problem inference.

Related Work

- Ontology modeling [1]
  - Applied into various domain (e.g., natural language processing [2], recommender systems [3])
  - Automatic ontology generation [4,5,6,7,8]
    - by analysing natural structured text [6, 8]
    - by exploring Wikipedia semi-structured text [7]

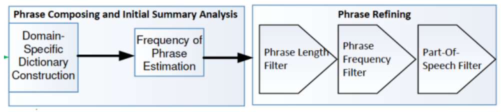
System Overview

- Our proposed integrated framework consists of three stages:
  1. Phrase Extraction Stage
     (a) Phrase Composition and Initial Summary Analysis Component
     (b) Phrase Refining Component
  2. Knowledge Construction Stage
  3. Ticket Resolution Stage



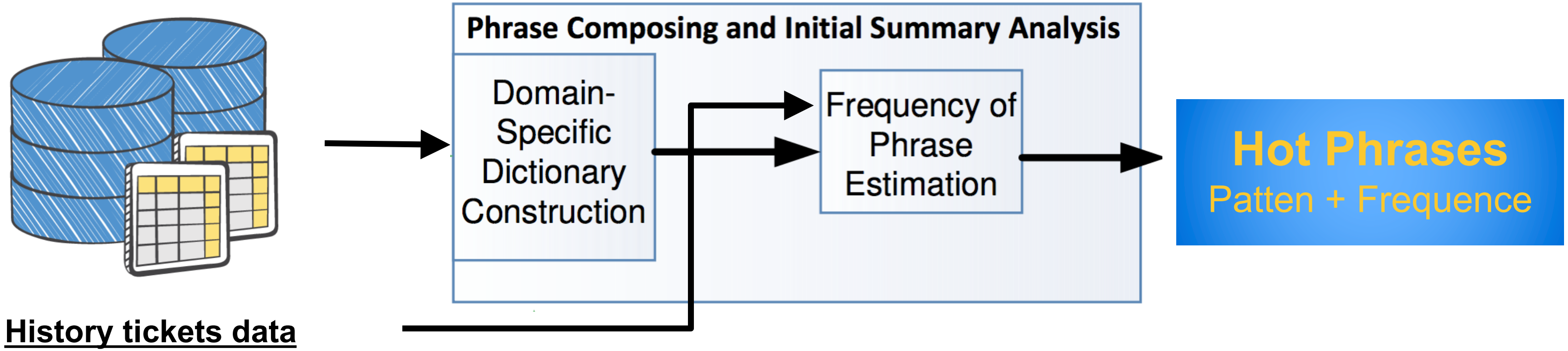
An overview of the integrated framework.

I: Phrase Extraction Stage



- In this stage, our framework finds important domain-specific words and phrases ('kernel').
  - Constructing a domain-specific dictionary
    - Mining the repeated words and phrases from unstructured text field.
    - Refining these repeated phrases by diverse criteria filters (e.g., length, frequency, etc.).

Phrase Composition and Initial Summary Analysis

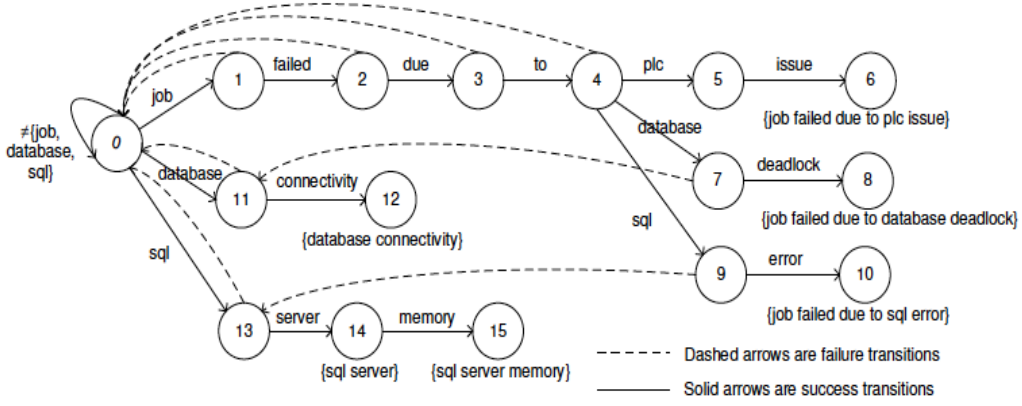


Repeated pattern extraction and frequency estimation.

- Use Stanford NLP Annotator [9] for preprocessing ticket data.
- Build a domain dictionary by using Word-Level Lempel-Ziv-Welch compression algorithm. [10]
- Calculate the frequency of the repeated phrases in tickets data by using Aho-Corasick algorithm. [11]

Frequency of Phrase Estimation

Assume we have a dictionary D composing {
“job failed due to plc issue,”
“job failed due to database deadlock,”
“job failed due to sql error,”
“database connectivity,”
“sql server,”
“sql server memory”
}.

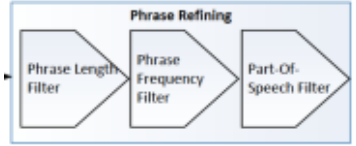


An example of a finite state string pattern matching machine.

Phrases Refining

In this stage, we apply two filters to the extracted repeated phrases allowing the omission of <u>non-informative</u> phrases.

- Phrase Length & Frequency Filters (length > 20 & frequency >= 10)
- Part-Of-Speech Filter



Definition of technical term's schemes

Justeson-Katz Patterns	Penn Treebank Entity Patterns	Examples in Tickets
A N	JJ NN[P|S|PS]*	global merchant
N N	NN[P|S|PS]* NN[P|S|PS]*	database deadlock
A A N	JJ JJ NN[P|S|PS]*	available physical memory
A N N	JJ NN[P|S|PS] NN[P|S|PS]	backup client connection
N A N	NN[P|S|PS] JJ NN[P|S|PS]	load balancing activity
N N N	NN[P|S|PS] NN[P|S|PS] NN[P|S|PS]	socket connectivity error
N P N	NN[P|S|PS] IN NN[P|S|PS]	failures at sfdc
A:Adjective, **N**: Noun, **P**: Preposition		
JJ: Adjective, **NN**: singular Noun, **NNS**: plural Noun, **NNP**: singular proper Noun, **NNPS**: plural proper Noun, **IN**: Preposition		

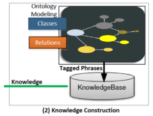
Definition of action term's schemes

Penn Treebank Action Patterns	Examples in Tickets
VB[D|G|N]*	run/check, updated/corrected affecting/circumventing, given/taken
VB: base form Verb, **VBD**: past tense Verb, **VBG**: gerund Verb,**VBN**: past participle Verb,	

Result of Frequency/Length Filter and PoSTag Filter.

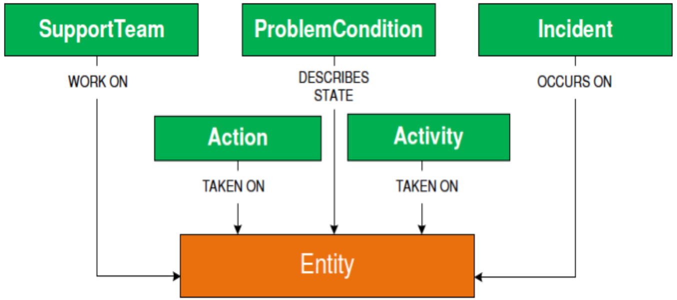
Applied Filter	Left Phrases
Frequency Filter >= 10	1117 items
Length Filter > 20	613 items
PoSTag Filter	323 items

II: Knowledge Construction Stage



In this stage, we first develop an ontology model, and then tag all the phrases of the generated dictionary with the defined classes.

- ➢ Build the ontology model
  - Define classes
  - Define relations
- ➢ Knowledge Archive
  - Manually tag the important phrases in the dictionary with their most relevant defined classes.



Ontology model depicting interactions among classes.

Definition of the Classes in Ontology

Class	Definition	Examples
Entity	Object that can be created/destroyed/replace	memory fault; database deadlock
Action	Requires creating/destroying an entity	restart; rerun; renew
Activity	Requires interacting with an entity	check; update; clean
Incident	State known to not have a problem	false alert; false positive
ProblemCondition	Describe the condition that causes a problem	offline; abended; failed
SupportTeam	Team that works on the problem	application team; databases team

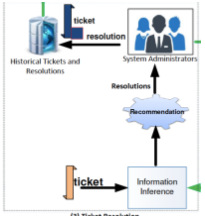
II: Knowledge Construction Stage

➔ Initial Domain Knowledge Base:

Entity	Activity	Action	ProblemCondition	Support Team
automated process	accept	reboot	abended	active direcory team
actual start	accepted	renew	bad data	app team
additional connection	achieved	rerun	deactived	application team
address information	acting	reran	disabled	aqpefds team
afr end	add	reset	dropped	bazaarvoice team
alert	added	restoring	expired	bmc team
alert imr	affecting	retransmit	fails	bsd team
alerts	affects	fixed	failed	Bureau team
alphanumeric values	altered	restart	false alert	business team
amex	aligned	restarted	false positive	bwinfra team
api calls	allocate	renewed	human error	cdm team
application	allocated	fixed	not working	CDM/GLEUDBD team
application code	applied	fixing	offline	cmit team
application impact	assign	recycle	stopped	control m team
atm messages	assigned	recycled	unavailable	convergys team
audit	blocks	recycling	under threshold	csp team
audit log	bring	reopen	wrong	cu team

Class	Number of Tagged Phrases
Entity	628 items
Activity	243 items
Action	24 items
Problem Condition	22 items
SupportTeam	76 items

III: Ticket Resolution Stage



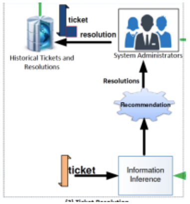
The goal of this stage is to recommend operational phrases for an incoming ticket.

- Information Inference component:
  - **Class Tagger Module** processes incoming ticket in three steps.
    - tokenize the input into sentences;
    - construct a Trie using ontology domain dictionary;
    - find the longest matching phrases of each sentence using the Trie and knowledge base, then map them onto the corresponding ontology classes
  - Define Concept Patterns for Inference: concept patterns based on Problem, Activity and Action concepts:
    - Problem describes an entity in negative condition or state.
    - Activity denotes the diagnostic steps on an entity.
    - Action represents the fixing operation on an entity.

Concept	Pattern	Examples
Problem	Entity preceded/succeeded by ProblemCondition	(jvm) is (down)
Activity	Entity preceded/succeeded by Activity	(check) the (gft record count)
Action	Entity preceded/succeeded by Action	(restart) the (database)

III: Ticket Resolution Stage

Post loading failed due to plc issue. Update the gft after proper validation and processed the job and completed successfully.



(post loading)/(Entity) (failed)/(ProblemCondition) due to (plc issue)/(Entity). (Update)/(Activity) the (gft)/(Entity) after (proper validation)/(Entity) and (processed)/(Activity) the (job)/(Entity) and (completed)/(Action) successfully.

- Problem - {failed: plc issue, post loading}
- Activity - {update: gft, proper validation; process: job}
- Action - {complete: job}

III: Ticket Resolution Stage

The goal of this stage is to recommend operational phrases for an incoming ticket.

- Ontology-based Resolution Recommendation component
    - Ontology model can greatly facilitates our resolution recommendation task by better capturing the similarity between ticket summaries.

Noisy ticket summary examples

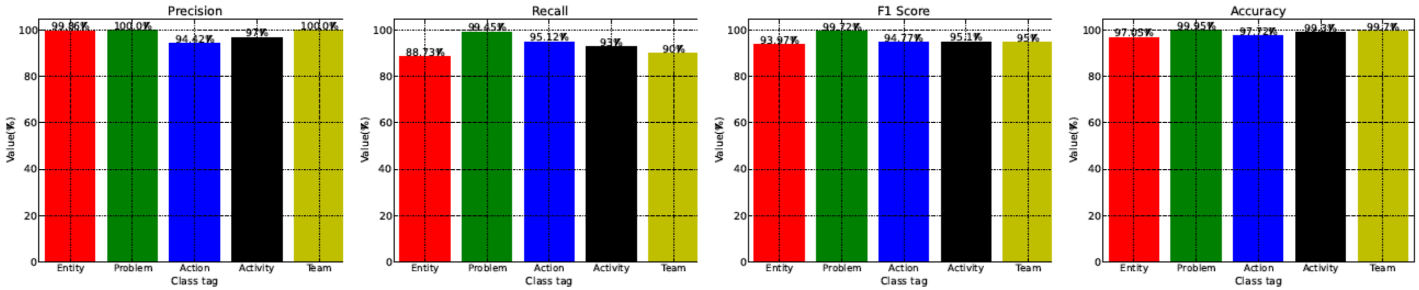
Inside ProcessTransaction. DetermineOutcome failed. Database save failed: Tried an insert, then tried an update
CRPE3I1Server Database save failed on lppwa899 00:19:46 lppwa899 /logs/websphere/wsfpp1lppwa 899CRPE3I1Server/SystemOut.log [3/20/14 0:19:33:371 MST] 0000002b SystemOut 20140320 00:19:33, 371 [WebContainer:30] [STANDARD] [DI_US:01.22] (ng.AEXP_US_ISR_Work_Txn.Action) FATAL lp-pwa899—10.16.4.4—SOAP—AEXP_US_ISR_Roads3_Pkg —AEXPUSISRWork-Inquiry—ProcessInquiry

Experiment

- Dataset
  - Experimental tickets are collected from real production servers of IBM Cloud Monitoring system covers three month time period containing $|D| = 22{,}423$ tickets.
  - Training data: 90% of total tickets
  - Testing data: 10% of total tickets
- Evaluation Metrics
  - Precision, Recall, F1 score and Accuracy.
  - Accuracy = (TP + TN)/(TP + TN + FP + FN)
  - Precision = TP/(TP + FP)   Recall = TP/(TP + FN)
  - F1 score = 2 Precision Recall / (Precision + Recall)

Experiment

- Ground Truth
  - Domain experts manually find and tag all phrases instances into six predefined classes in testing dataset.
- Evaluate our integrated system
  - Class Tagger is applied to testing tickets to produce tagged phrases with predefined classes. Comparing the tagged phrases with ground truth, we obtain the performance.



(a) Precision. (b) Recall. (c) F1-Score. (d) Accuracy.

Evaluation results of our integrated system.

Experiment

- Evaluate Information Inference
  - Usability: we evaluate the average accuracy to be 95.5%, 92.3%, and 86.2% for Problem, Activity, and Action respectively.
  - Readability: we measure the time cost. Domain expert can be quicker to identify the Problem, Activity and Action which output from the Information Inference component from 50 randomly selected tickets.

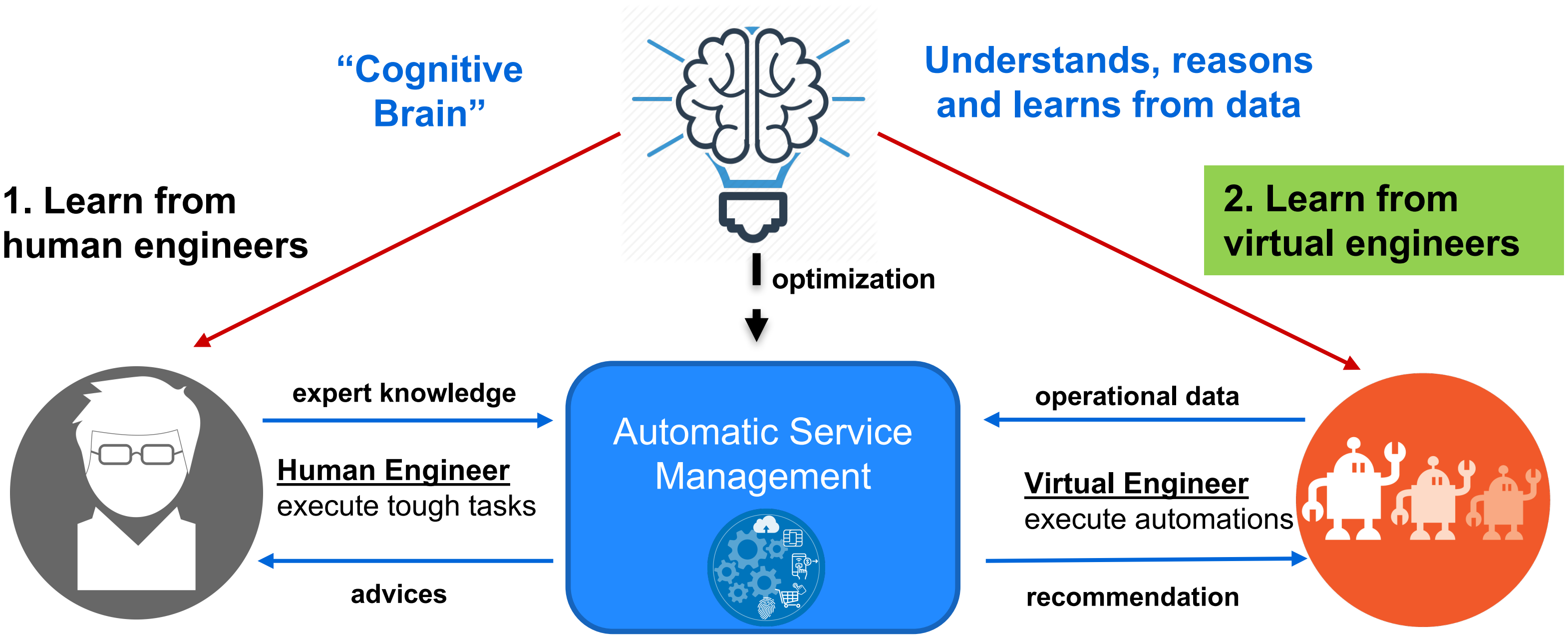
Summary of this section

This work has been published in IEEE SCC 2017 and awarded as **the best student paper**.

Wang, Qing, et al. "**Constructing the knowledge base for cognitive IT service management.**" Services Computing (SCC), 2017 IEEE International Conference on. IEEE, 2017.

The extended work with a deep ranking model is submitted to the journal TSC.

Overview of Research Problems

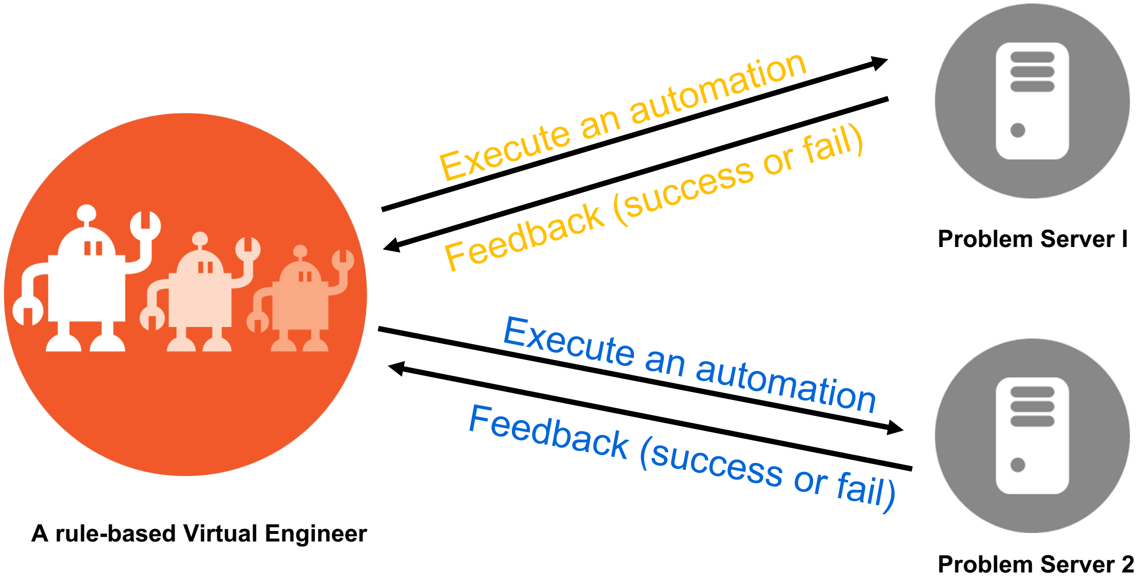


Outline

- Introduction
- Research Problems
  - Learn Human Intelligence by Domain Knowledge Base Construction
  - Learn Automation Intelligence by Hierarchical Multi-armed Bandit Model
    - Multi-armed Bandit Problems with Dependent Arms
    - Hierarchical IT Automation Recommendation Modeling
    - Hierarchical Multi-armed Bandit Model
  - Learn Automation Intelligence by Interactive Collaborative Topic Regression Model
    - Interactive Collaborative Filtering Problem
    - Matrix-Factorization based IT Automation Recommendation Modeling
    - Interactive Collaborative Topic Regression Model
- Summary

IT automation recommendation modeling

IT automation services (ITAS) [12] is introduced into IT service management.



An overview of IT Automation Services

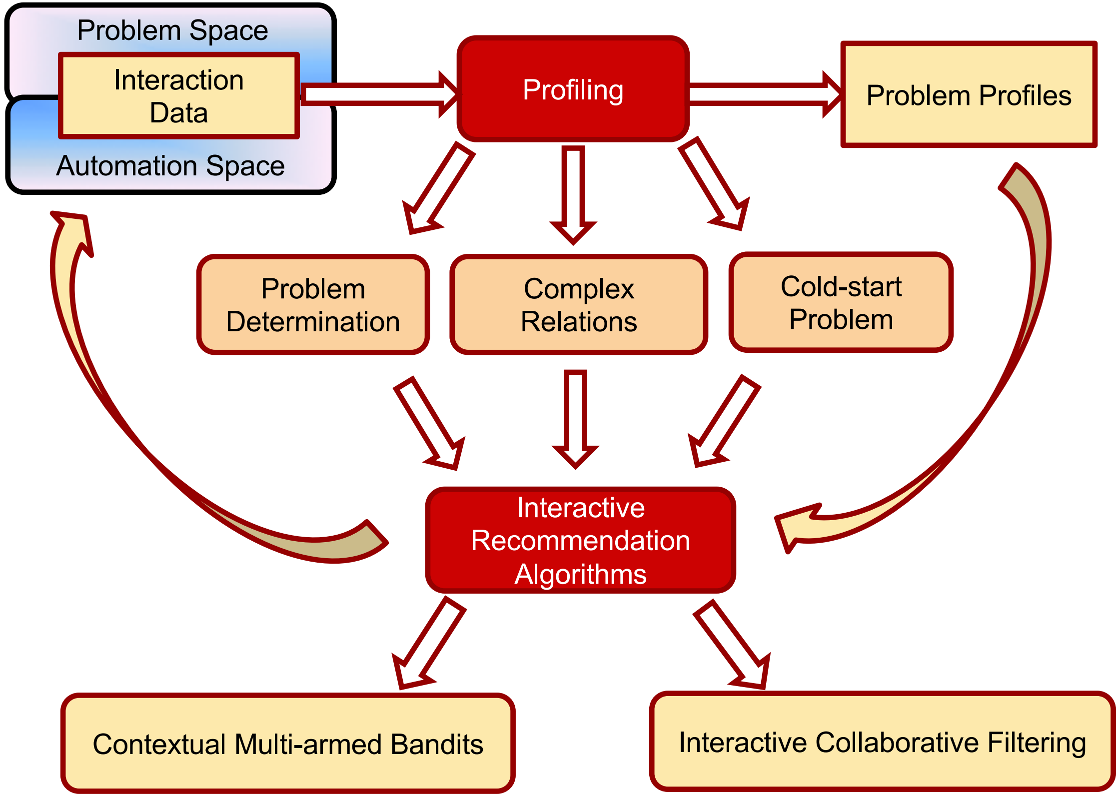
IT Automation Recommendation Modeling

ALERT_KEY	cpc_cpuutil_gntw_win_v3		AUTOMATON_NAME		CPC:WIN:GEN:R:W:System Load Handler		
OPEN_DTTM	CLIENT_ID	HOSTNAME	ORIGINAL SEVERITY	OSTYPE	COMPONET	SUBCOMP OMET	AUTO RESOVLED
2016-04-30 12:43:07	136	LEXSBWS01 VH	2	WIN	WINDOWS	CPU	1
TICKET SUMMARY	CPU Workload High. CPU 1, busy 99% time.		TICKET RESOLUTION		The CPU Utilization was quite reduced, hence closing the ticket.		

feedback

A sample ticket in ITSM with its corresponding automaton.

A General Process of Interactive Recommendation



Challenges

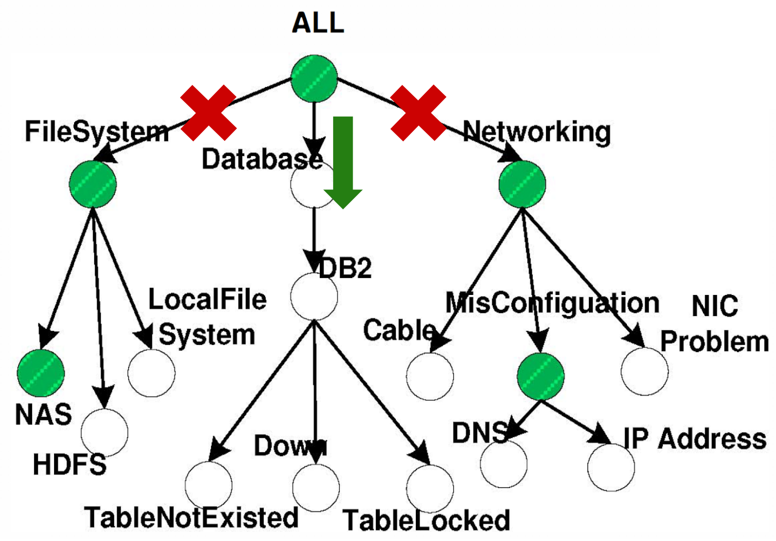
- Challenge 1: How do we appropriately solve the well-known cold-start problem [13] in IT automation services?
- Challenge 2: How do we utilize the interactive feedback to adaptively optimize the recommending strategies of the enterprise automation engine to enable a quick problem determination by IT automation services?

This can be naturally modeled as a contextual multi-armed bandit problem, which has been widely applied into various interactive recommender systems. [14, 15, 16]

Challenges

➢ Challenge 3: How do we efficiently improve the performance of recommendation using the explicit automation hierarchies of IT automation services?

For example, a ticket is generated due to a failure of the DB2 database. The root cause may be database deadlock, high usage or other issues.

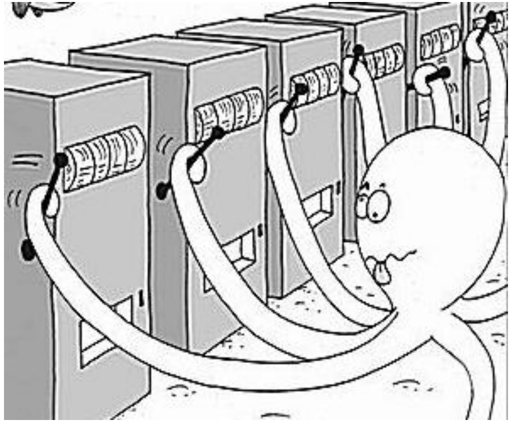


Related Work

- Interactive Recommender Systems [15, 16]
- Multi-armed Bandit Algorithms [14, 20]
  - ε-greedy, UCB [14], Thompson Sampling [20].
  - Used to balance the tradeoff between exploration and exploitation in recommender system.
- Multi-armed Bandit Problems with Dependent Arms [17, 18, 19]
  - Use the taxonomy to explore the dependencies among arms in the context-free bandit setting. [18]
  - Learn the item hierarchy by a small number of user profiles. [19]

Multi-armed Bandit Problem

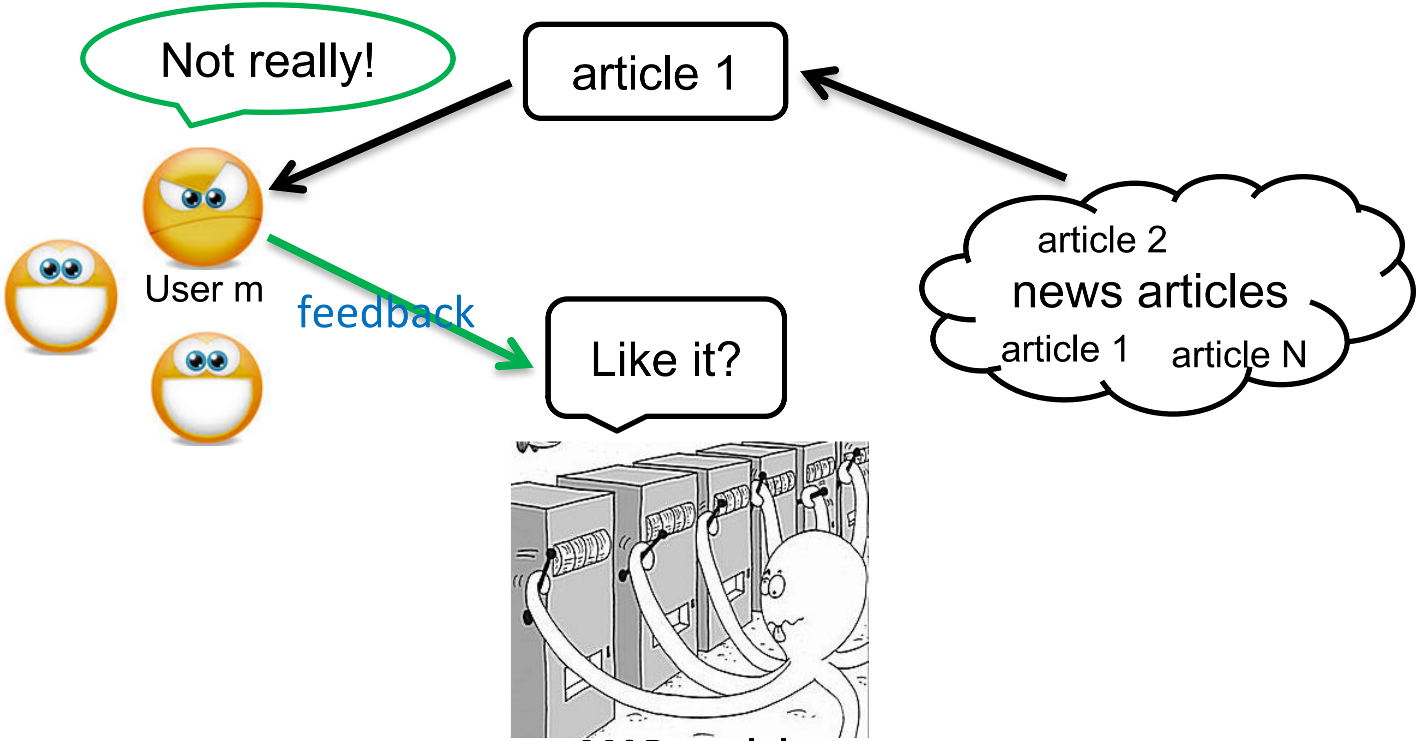
- The MAB problem is a classical paradigm in machine learning in which an online algorithm choses from a set of strategies in a sequence of trials so as to maximize the total payoff of the chosen strategies. [30]



- A gambler walks into a casino
- A row of slot machines providing a random rewards
- A tradeoff between *exploration* and *exploitation.*

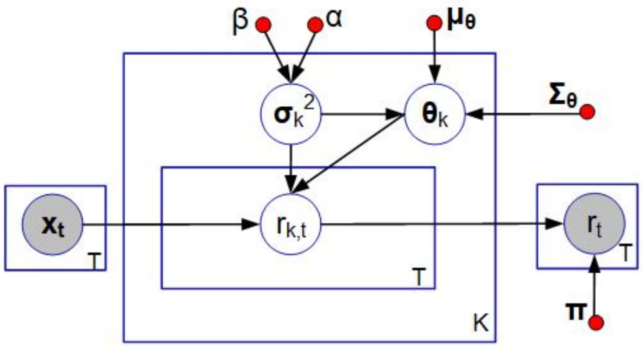
Objective: Maximize the sum of rewards (Money)!

Example: News Recommendation



MAB model

Contextual Multi-armed Bandit Model



A graphic model of conventional contextual MAB.

Table 1: Important Notations

Notation	Description
$a^{(i)}$	the i-th arm.
$\mathcal{A}$	the set of arms, $\mathcal{A} = \{a^{(1)}, ..., a^{(K)}\}$.
$\mathcal{H}$	the hierarchy (taxonomy) defined by domain experts.
$\mathcal{X}$	d -dimensional context feature space.
$\mathbf{x}_t$	the context at time t.
$r_{k,t}$	the reward (payoff) of pulling the arm $a^{(k)}$ at time t.
$\hat{r}_{k,t}$	the predicted reward (payoff) for the arm $a^{(k)}$ at time t.
π	the policy for pulling arm sequentially.
R_π	the cumulative reward of the policy π.
$\mathbf{S}_{\pi,t}$	the sequence of $(\mathbf{x}_i, \pi(\mathbf{x}_i), r_{\pi(\mathbf{x}_t)})$ observed until time $t = 1, ..., T$.
θ_k	the coefficients predicting reward of the arm $a^{(k)}$.
σ_k^2	the reward prediction variance for arm $a^{(k)}$.
α, β	the parameters of the distribution of σ_k^2.
$\mu_\theta, \mathbf{\Sigma}_\theta$	the parameters of the distribution of θ.

The reward $r_{k,t}$ is typically modeled as a linear combination of the feature vector x_t given at time $t = [1, \dots, T]$ as follows:

$$r_{k,t} \sim N(x_t^T \theta_k, \sigma_k^2)$$

The optimal policy π^* is defined as the one with maximum accumulated expected reward after T iterations:

$$\pi^* = arg\max_{\pi} \sum_{t=1}^{T} E_{\theta_{\pi(x_t)}}(x_t^T \theta_{\pi(x_t)} | t)$$

Online IT Automation Recommendation Modeling

In IT automation recommendation modeling,

- let $\boldsymbol{A} = \{\boldsymbol{a}^{(1)}, \ldots, \boldsymbol{a}^{(N)}\}$ denote a set of automations (i.e., scripted resolutions) feasible in IT automation system.
- Every time a ticket is reported, the IT automation engine selects a proper automation according to contextual information (i.e., the ticket symptom) and recommends it as a possible resolution. The contextual information for a reported ticket at time t is represented as a feature vector $\boldsymbol{x_t} \in \boldsymbol{X}$, where $\boldsymbol{X}$ denotes the d-dimensional feature space.
- Every recommended automation $\boldsymbol{a}^{(i)} \in \boldsymbol{A}$ at time t, has a corresponding feedback indicating whether or not the ticket has been successfully resolved.

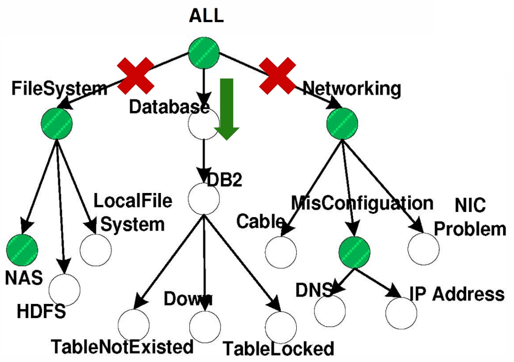
The optimal policy π^* is defined as the one with maximum accumulated expected reward after T iterations,

$$\pi^* = \arg\max_{\pi} E(R_\pi) = \arg\max_{\pi} \sum_{t=1}^{T} E(r_{\mathbf{x}_t, \pi(\mathbf{x}_t)} | t). \qquad (2)$$

Hierarchical IT Automation Recommendation Modeling

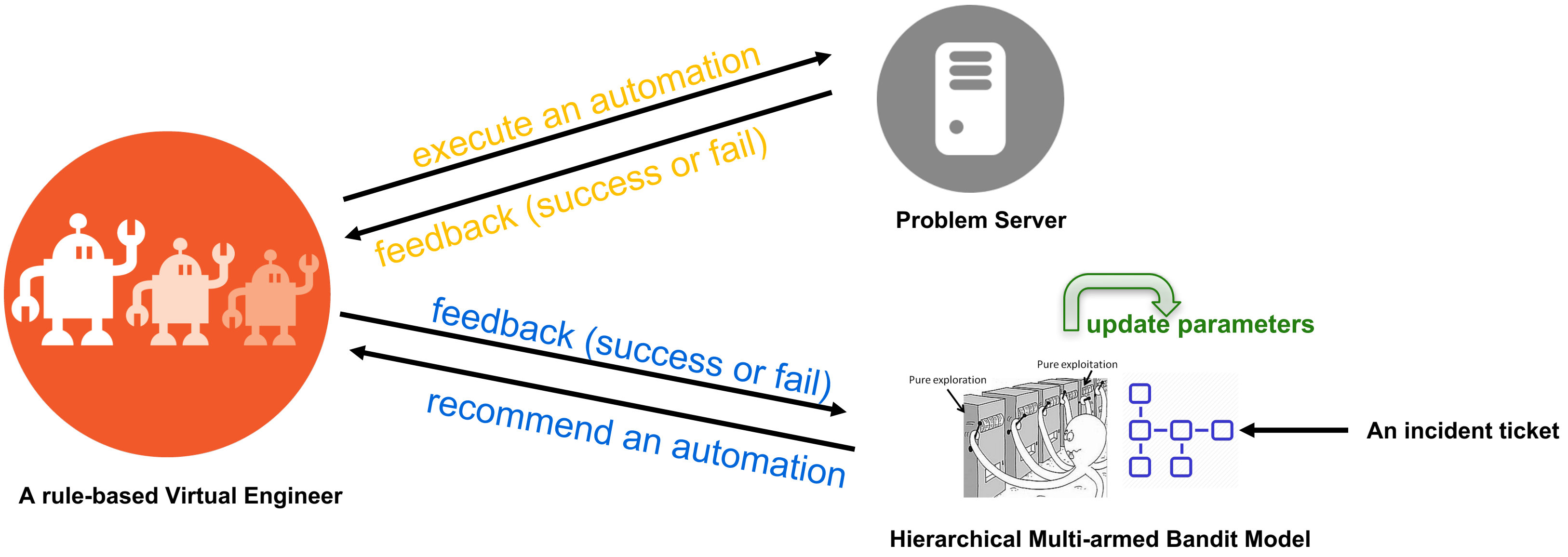
Let $\boldsymbol{H}$ denote the taxonomy. Given a node $\boldsymbol{a}^{(i)} \in \boldsymbol{H}$, $\boldsymbol{pa}(\boldsymbol{a}^{(i)})$ and $\boldsymbol{ch}(\boldsymbol{a}^{(i)})$ are used to represent the parent and children sets, respectively.

- A leaf node of H represents an automation
- Non-leaf node is category or subcategory information.
- Therefore, the multi-armed bandit problem for IT automation recommendation is reduced to selection of a path in H from root to a leaf node, and multiple arms along the path are sequentially selected based on the contextual vector $\boldsymbol{x_t}$ at time t.



$$\pi^* = \arg\max_{\pi} \sum_{t=1}^{T} \sum_{\substack{a^{(i)} \in \pi_{\mathcal{H}}(\mathbf{x}_t|t), \\ ch(a^{(i)}) \neq \emptyset}} E_{\theta_{\pi(\mathbf{x}_t|ch(a^{(i)}))}}(\mathbf{x}_t^T \theta_{\pi(\mathbf{x}_t|ch(a^{(i)}))}|t).$$

Hierarchical IT Automation Recommendation Modeling



Hierarchial Multi-armed Bandit Algorithms

Algorithm 1 The algorithms for HMAB model

procedure MAIN($\mathcal{H}, \pi, \lambda$) ▷ Main entry, π is the policy.
  for $t \leftarrow 1, T$ **do**
    Initialize parameters of $a^{(m)} \in \mathcal{H}$ to α_m, β_m, $\boldsymbol{\Sigma}_{\theta_m} = \mathbf{I}_d$, $\mu_{\theta_m} = \mathbf{0}_{d\times 1}$.
    Get contextual vector $\mathbf{x}_t \in \mathcal{X}$.
    for each path P of $\mathcal{H}$ **do**
      Compute the reward of P using Equation (4.6), by calling EVAL($\mathbf{x}_t, a^{(k)}, \pi$) for each arm $a^{(k)} \in P$.
    end for
    Choose the path P^* with maximum reward.
    Recommend the automation $a^{(*)}$ (leaf node of P^*).
    Receive reward $r_{*,t}$ by pulling arm $a^{(*)}$.
    UPDATE($\mathbf{x}_t, P^*, r_{*,t}, \pi$)
  end for
end procedure

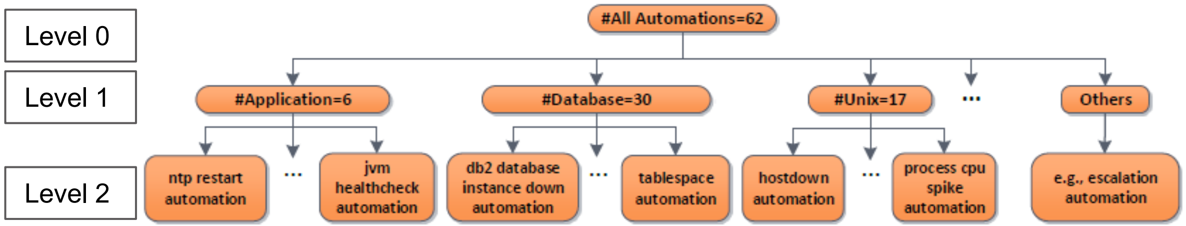
procedure EVAL($\mathbf{x}_t$, $a^{(k)}$, π) ▷ Get a score for $a^{(k)}$, given $\mathbf{x}_t$.
  if π is TS **then**
    Sample $\sigma^2_{k,t}$ according to Equation (4.10).
    Sample $\theta_{k,t}$ according to Equation (4.11).
    return $\hat{r}_{k,t} = \mathbf{x}_t^T \theta_{k,t}$.
  end if
  if π is LinUCB **then**
    return $\hat{r}_{k,t} = \mathbf{x}_t^T \mu_{\theta_{k,t-1}} + \frac{\lambda}{\sigma_{k,t-1}} \sqrt{\mathbf{x}_t^T \boldsymbol{\Sigma}^{-1}_{\theta_{k,t-1}} \mathbf{x}_t}$
  end if
end procedure

procedure UPDATE($\mathbf{x}_t, P, r_t, \pi$) ▷ Update the inference. P is the path in $\mathcal{H}$, r_t is the reward.
  for each arm $a^{(k)} \in P$ **do**
    Update $\alpha_{k,t}$, $\beta_{k,t}$, $\boldsymbol{\Sigma}_{\theta_{k,t}}$, $\mu_{\theta_{k,t}}$ using Equation (4.12).
  end for
end procedure

Hierarchial Multi-armed Bandit Algorithms

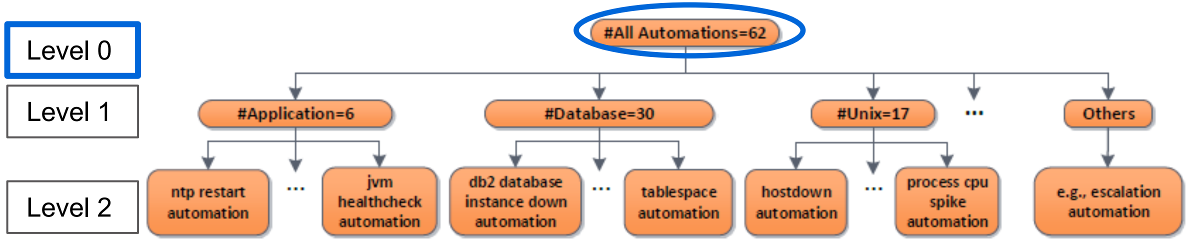
- We formulate it as a contextual bandit problem with dependent arms organized hierarchically, which can match the arm feature spaces from a coarse to fine level.

At time t = [1,…,T]:



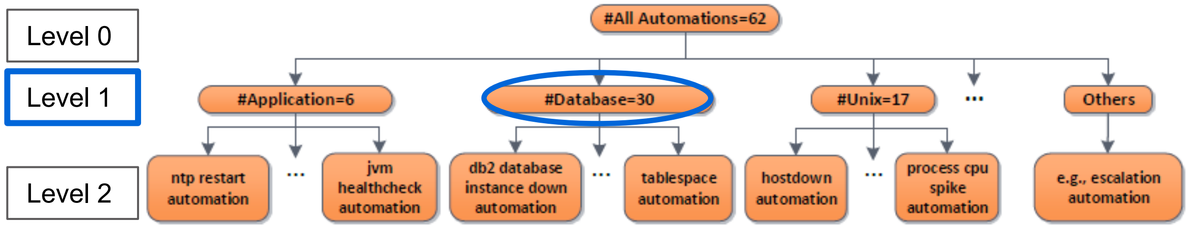
Hierarchial Multi-armed Bandit Algorithms

At time t = [1,…,T]:



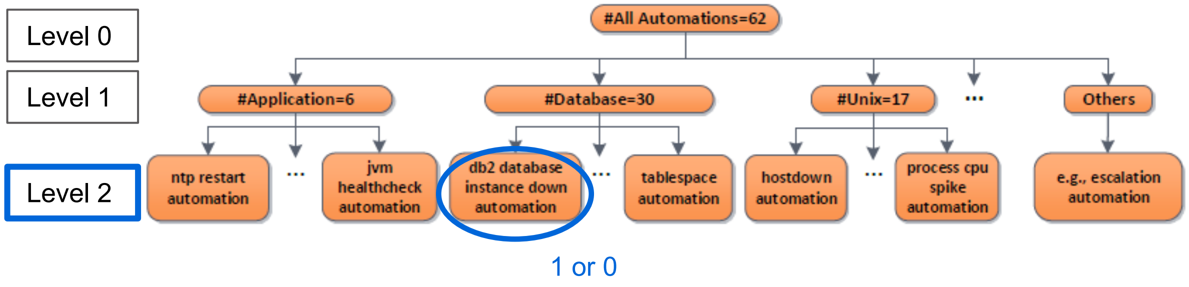
Hierarchial Multi-armed Bandit Algorithms

At time t = [1,…,T]:



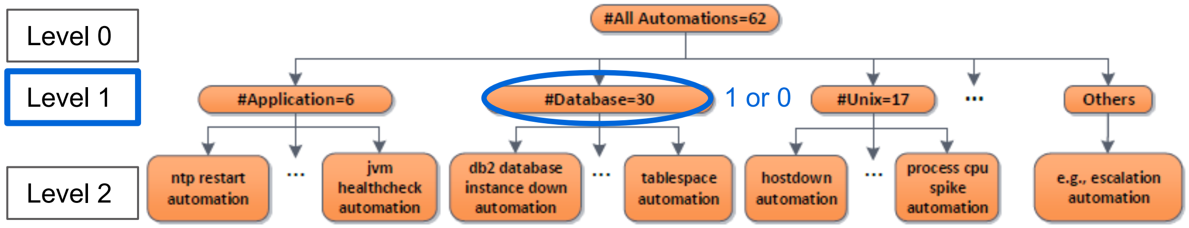
Hierarchial Multi-armed Bandit Algorithms

At time t = [1,…,T]:



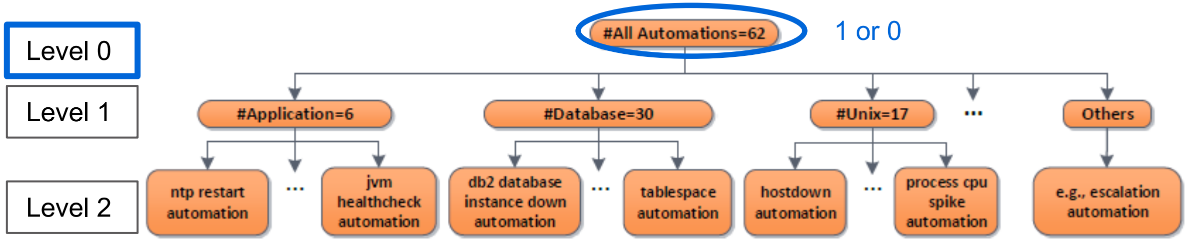
Hierarchial Multi-armed Bandit Algorithms

At time t = [1,…,T]:



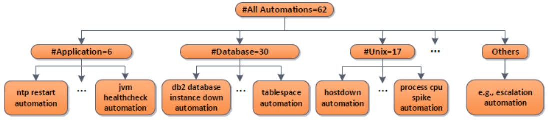
Hierarchial Multi-armed Bandit Algorithms

At time t = [1,…,T]:



Experiment

- Data Set
  - Experimental tickets are collected by IBM Tivoli Monitoring system covering from July 2016 to March 2017 with the size of |D| = 116,429.
  - The dataset contains 1,091 alert keys (e.g., cpusum_xuxc_aix, prccpu_rlzc_std) and 62 automations (e.g., NFS automation, process CPU spike automation) in total.
  - A given three-layer hierarchy H.
- Evaluate Method
  - Replayer method. [21, 29]



Experiment

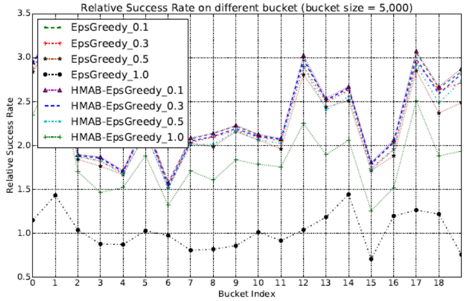


Figure 5: The Relative Success Rate of EpsGreedy and HMAB-EpsGreedy on the dataset is given along each time bucket with diverse parameter settings.

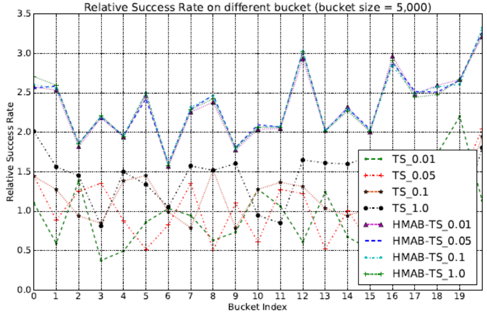


Figure 6: The Relative Success Rate of TS and HMAB-TS on the dataset is given along each time bucket with diverse parameter settings.

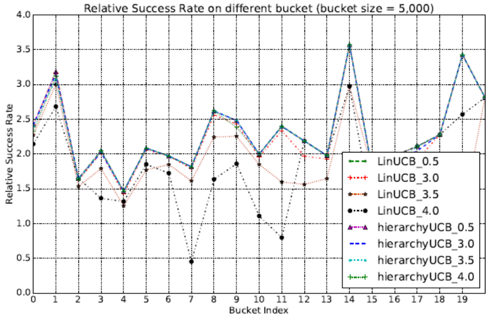


Figure 7: The Relative Success Rate of LinUCB and HMAB-LinUCB on the dataset is given along each time bucket with diverse parameter settings.

Experiment: A Case Study

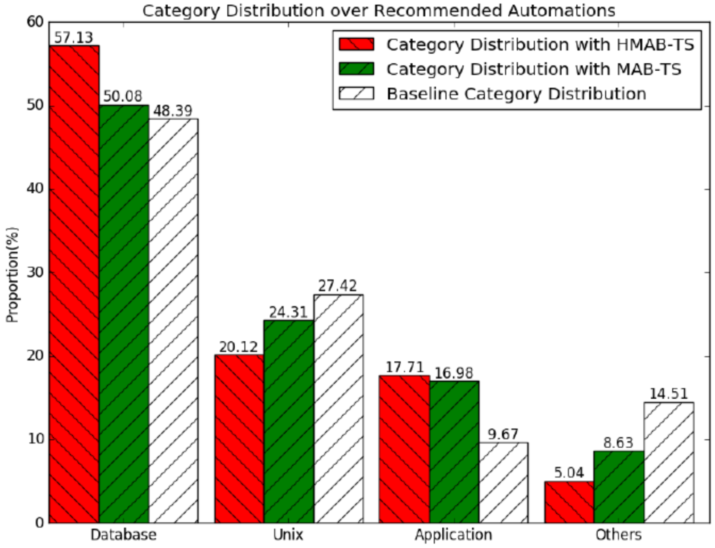


Figure 4.9: The comparison of category distribution on the recommended automations.

ALERT_KEY	ac2_dbinact_grzc_std	**AUTOMATON NAME**	Escalation Handler
TICKET SUMMARY	Database fin91dmo status is inactive.	**TICKET RESOLUTION**	The database is down. It has been restarted, hence closing the ticket.

RECOMMENDED CATEGORY	**(%)**	**RECOMMENDED AUTOMATON**
DATABASE	**57.13**	(1) **database** instance **down** automation; (2) db2 **database inactive** automation; (3) mysql **database offline** automation.
UNIX	20.12	(1) asm space check diskgroup **dbautomation;** (2) **hostdown** automation; (3) certification **expiration** automation.
APPLICATION	17.71	(1) ntp **restart** automation; (2) mq manager **down** automation.
OTHERS	5.04	(1) system load automation; (2) others.

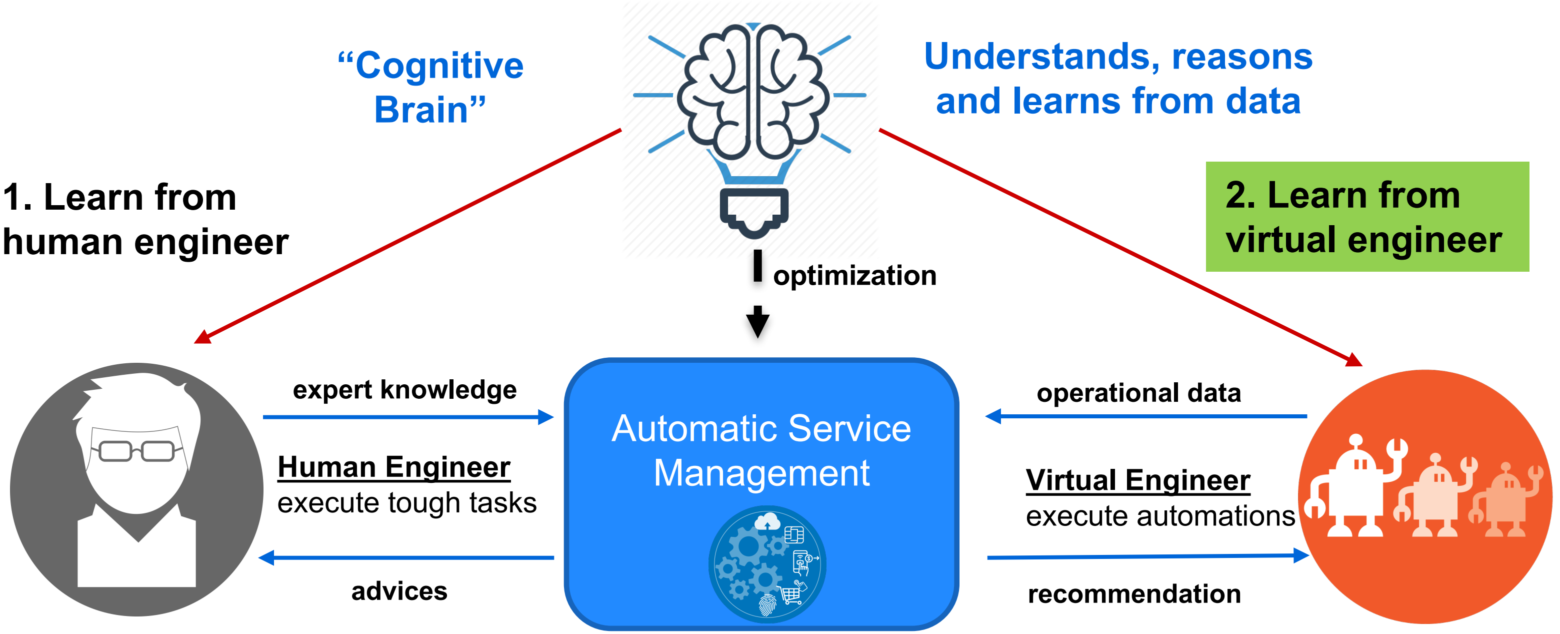
Figure 4.10: The *exploration* by HMAB-TS of a cold-start ticket case.

Summary of this section

This work has been published in SIAM International Conference on Data Mining (SDM) 2018.

Q. Wang, T. Li, S. S. Iyengar, et al. **Online it ticket automation recommendation using hierarchical multi-armed bandit algorithms**. In SDM. SIAM, 2018.

Overview of Research Problems



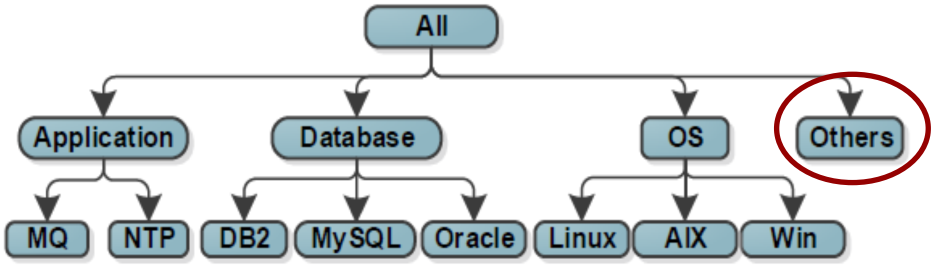
Outline

- Introduction
- Research Problems
  - Learn Human Intelligence by Domain Knowledge Base Construction
  - Learn Automation Intelligence by Hierarchical Multi-armed Bandit Model
    - Multi-armed Bandit Problems with Dependent Arms
    - Hierarchical IT Automation Recommendation Modeling
    - Hierarchical Multi-armed Bandit Model
  - Learn Automation Intelligence by Interactive Collaborative Topic Regression Model
    - Interactive Collaborative Filtering Problem
    - Matrix-Factorization based IT Automation Recommendation Modeling
    - Interactive Collaborative Topic Regression Model
- Summary

Challenges

- Challenge 1: How do we solve the cold-start problem in IT automation services?
- Challenge 2: How do we adaptively recommend a proper automation in IT automation services?

- Challenge 3: How do we effectively recommend a proper automation with no explicit hierarchical information and, in the worse case, with no contextual information of the incident ticket in IT automation services?



Challenges

This can be naturally modeled as an interactive collaborative filtering problem, which has been first introduced in [15].

Ticket Problem 1

ALERT_KEY	xxx_cpusum_xuxc_aix		AUTOMATON_NAME		Process CPU Spike Automation		
OPEN_DTTM	CLIENT_ID	HOSTNAME	ORIGINAL SEVERITY	OSTYPE	COMPONENT	SUBCOMPONENT	AUTO RESOVLED
1456383421000	90	XXX	4	AIX	UNIX	UNKNOWN	1
TICKET SUMMARY	XXX CPU Utilization is very high, workloads affected			TICKET RESOLUTION	Alert in question has recovered, hence closing the ticket.		

Ticket Problem 2

ALERT_KEY	xxx_prccpu_rlzc_std		AUTOMATON_NAME		Process CPU Spike Automation		
OPEN_DTTM	CLIENT_ID	HOSTNAME	ORIGINAL SEVERITY	OSTYPE	COMPONENT	SUBCOMPONENT	AUTO RESOVLED
1454900281000	52	XXX	4	UNKNOWN	LINUX	Process	1
TICKET SUMMARY	XXX Process using high cpu (97.30%) nsrdmpix			TICKET RESOLUTION	Alert in question has recovered, hence closing the ticket.		

Two different ticket problems in IT service management

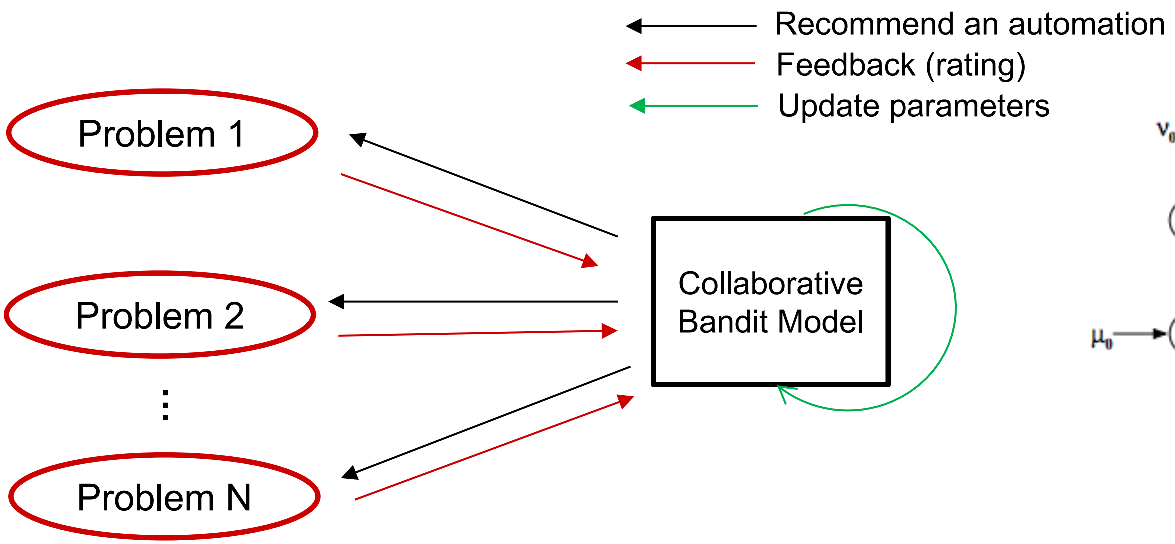
Automaton

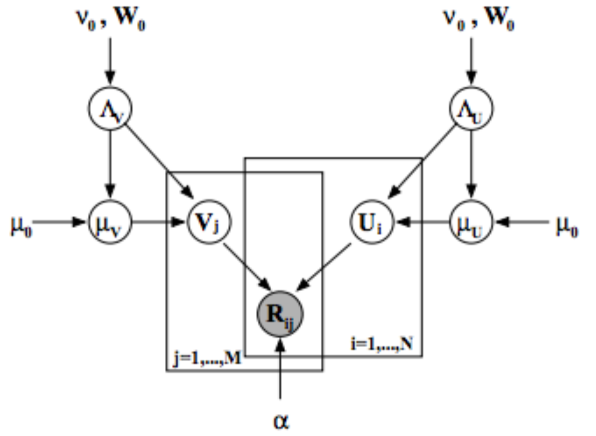
Ticket Problem \ Automaton	1	2	3	4	5
1	✓	✗	✓	?	?
2	✓	✓	?	?	✓
3	✗	?	✓	✗	✗
4	?	✓	?	✗	?
5	✗	?	✓	✓	?

Rating Matrix

Interactive Collaborative Fitlering Problem

No context information can be observed.





Bayesian Probabilistic Matrix Factorization

Matrix-Factorization based IT Automation Recommendation Modeling

There are ***M*** ticket problems and ***N*** automations. The partially observed matrix ***R*** is the preference of the ticket problem for the automation. In the collaborative bandit model, the rating is estimated by a product of ticket problem and automation feature vectors $\boldsymbol{p_m}$ and $\boldsymbol{q_n}$.

$$r_{m,n} \sim N(\, p_m^T q_n,\, \sigma^2)$$

The objective function can be written as follows:

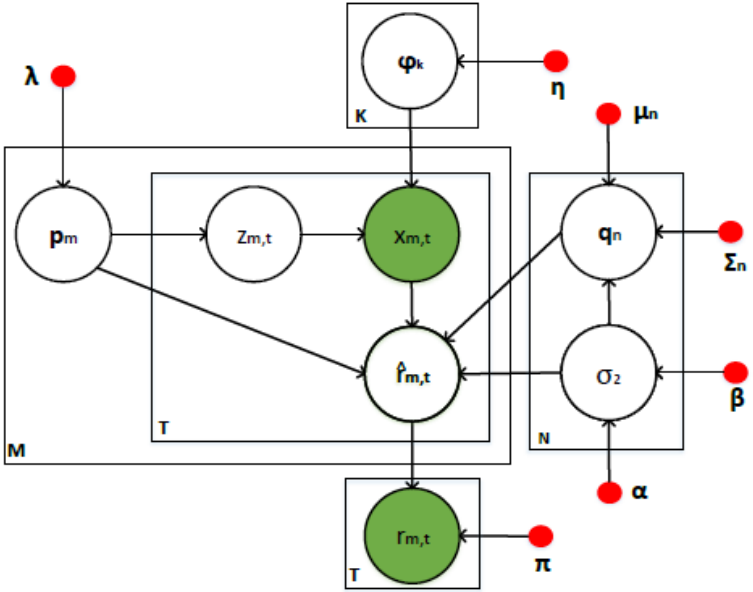
$$\pi^* = \arg\max_{\pi} \sum \mathbb{E}_{\mathbf{p}_m, \mathbf{q}_{\pi(\mathbb{S}(t))}} (\mathbf{p}_m^{\mathsf{T}} \mathbf{q}_{\pi(\mathbb{S}(t))} | t)$$

Where $\mathbb{S}(t) = \{(n(1), r_{m,n(1)}), \ldots, (n(t-1), r_{m,n(t-1)})\}$. $\mathbb{S}(t)$ is available information observed at time t.

Related Work

- Probabilistic Matrix Factorization [27, 28]
- Interactive Collaborative Filtering [15, 24, 25, 26]
    - Study the collaborative filtering in the bandit setting [15, 24]
    - Considering the user-side clustering [25, 26]
- Collaborative topic modeling [22, 23]
    - Integrate topic modeling into an matrix factorization setting

Interactive Collaborative Topic Regression Model



The graphical representation for ICTR model.

$$\mathbf{p}_m|\lambda \sim Dir(\lambda) \qquad p(\sigma_n^2|\alpha,\beta) = \mathcal{IG}(\alpha,\beta)$$

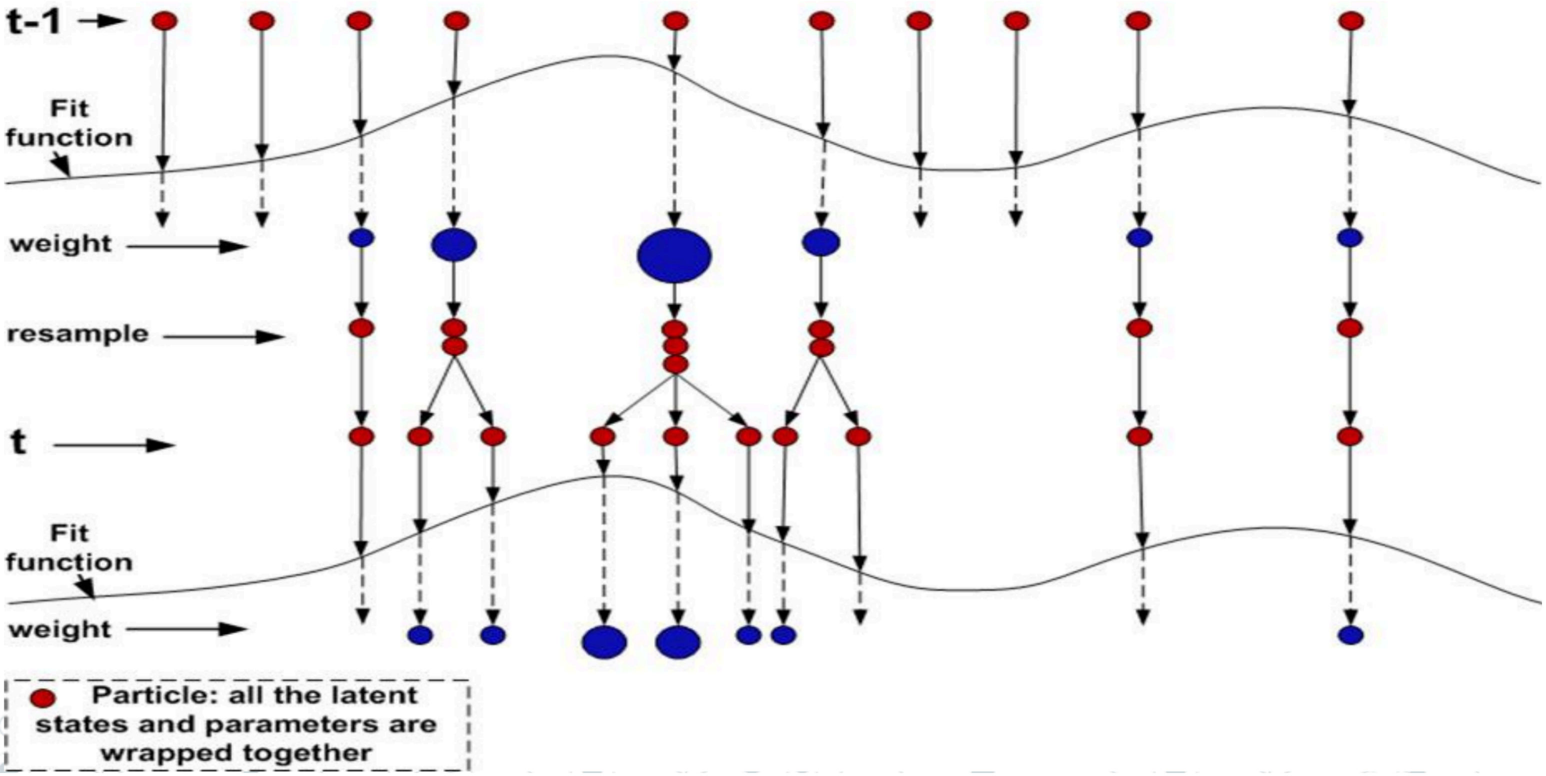
$$\mathbf{q}_n|\mu_\mathbf{q}, \mathbf{\Sigma}_\mathbf{q}, \sigma_n^2 \sim \mathcal{N}(\mu_\mathbf{q}, \sigma_n^2\mathbf{\Sigma}_\mathbf{q}), \quad \varphi_k|\eta \sim Dir(\eta)$$

$$z_{m,t}|\mathbf{p}_m \sim Mult(\mathbf{p}_m), \quad x_{m,t}|\varphi_k \sim Mult(\varphi_k)$$

The predicted rating $\hat{r}_{m,t}$ can be inferred by

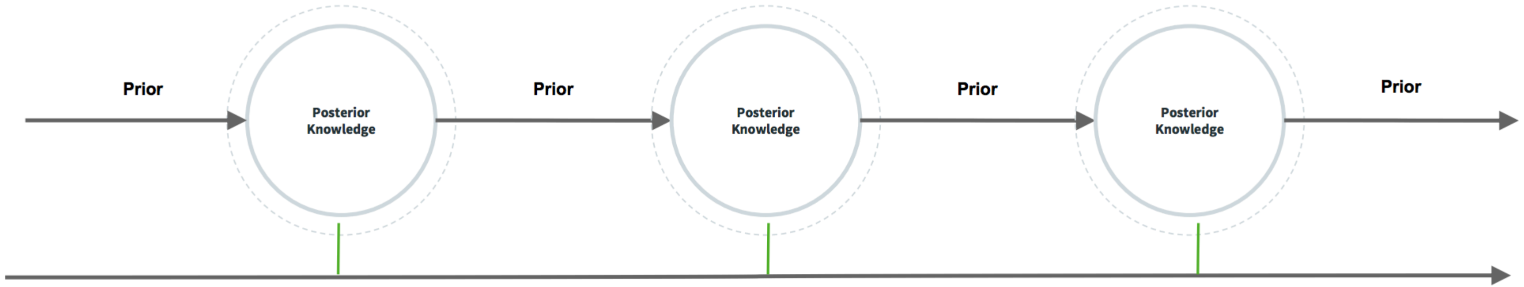
$$\hat{r}_{m,t} \sim \mathcal{N}(\mathbf{p}_m^\intercal \mathbf{q}_n, \sigma_n^2).$$

Online Inference of ICTR Model: Particle Learning



Online Inference of ICTR Model: Particle Learning

Definition 1 (Particle). A particle for predicting the reward $\hat{r}_{m,t}$ is a container that maintains the current status information for both user m and item $x_{m,t}$. The status information comprises of random variables such as $\mathbf{p}_m$, σ_n^2, $\mathbf{\Phi}_k$, $\mathbf{q}_n$, and $z_{m,t}$, as well as the hyper parameters of their corresponding distributions, such as λ, α, β, η, $\mu_{\mathbf{q}}$ and $\mathbf{\Sigma}_{\mathbf{q}}$.



Re-sample Particles with Weights

Let $\mathcal{P}_{m,n(t-1)}$ denote the particle set at time $t-1$ and $\mathcal{P}_{m,n(t-1)}^{(i)}$ be the i^{th} particles given both ticket problem m and automation $n(t-1)$ at time $(t-1)$, where $1 \ll i \ll B$. Each particle has a weight, denoted as $\rho^{(i)}$, where $\sum_{i=1}^{B} \rho^{(i)}$=1. **The fitness of each particle $\mathcal{P}_{m,n(t-1)}^{(i)}$ is defined as the likelihood of the observed data $x_{m,t}$ and $r_{m,t}$.** Therefore,

$$\rho^{(i)} \propto p(x_{m,t}, r_{m,t} | \mathcal{P}_{m,n(t-1)}^{(i)}).$$

As further deriving,

$$\rho^{(i)} \propto \sum_{z_{m,t}=1}^{K} \{\mathcal{N}(\mathbf{r}_{m,t} \,|\, \mathbf{p}_m^{\mathsf{T}} \mathbf{q}_n, \sigma_n^2) E(\mathbf{p}_{m,k}|\lambda)\, E(\varphi_{k,n}|\eta)\}$$

where $E(\mathbf{p}_{m,k}|\lambda) = \frac{\lambda_k}{\sum_{k=1}^{K} \lambda_k}$ and $E(\varphi_{k,n}|\eta) = \frac{\eta_{k,n}}{\sum_{n=1}^{N} \eta_{k,n}}$ represent the conditional expectations of $\mathbf{p}_{m,k}$ and $\varphi_{k,n}$ given the observed reward λ and η of $\mathcal{P}_{m,n(t-1)}^{(i)}$.

Latent State Inference

Provide with new observation $x_{m,t}$ and $r_{m,t}$ at time t, the random state $z_{m,t}$ can be any one of K topics. The posterior distribution of $z_{m,t}$ is shown as follows, where $\theta \in \mathcal{R}^K$:

$$z_{m,t} | x_{m,t}, r_{m,t}, \mathcal{P}_{m,n(t-1)}^{(i)} \sim Mult(\theta),$$

θ can be computed by

$$\theta \propto E(p_{m,k} \,| r_{m,t}, \lambda) \cdot E(\Psi_{k,n} \,| r_{m,t}, \lambda)$$

$$E(\mathbf{p}_{m,k} | r_{m,t}, \lambda) = \frac{\mathcal{I}(z_{m,t} = k) r_{m,t} + \lambda_k}{\sum_{k=1}^{K} [\mathcal{I}(z_{m,t} = k) r_{m,t} + \lambda_k]},$$

$$E(\mathbf{\Phi}_{k,n} | r_{m,t}, \eta) = \frac{\mathcal{I}(x_{m,t} = n) r_{m,t} + \eta_{k,n}}{\sum_{n=1}^{N} [\mathcal{I}(x_{m,t} = n) r_{m,t} + \eta_{k,n}]}.$$

Where $\mathcal{I}(\cdot)$ is an indicator function, returns 1 when the input Boolean expression is true and otherwise return 0.

Parameter Statistics Inference

Assume $\mu'_{\mathbf{q}}$, $\boldsymbol{\Sigma}'_{\mathbf{q}}$, α', β', λ', and η' are the sufficient statistics at time t, which are updated on the sufficient statistics $\mu_{\mathbf{q}}$, $\boldsymbol{\Sigma}_{\mathbf{q}}$, α, β, λ, η at t-1, and new observation data $x_{m,t}$ and $r_{m,t}$ at time t as follows.

$$
\begin{aligned}
\boldsymbol{\Sigma}'_{\mathbf{q}_n} &= (\boldsymbol{\Sigma}^{-1}_{\mathbf{q}_n} + \mathbf{p}_m \mathbf{p}_m^{\intercal})^{-1} \\
\mu'_{\mathbf{q}_n} &= \boldsymbol{\Sigma}'_{\mathbf{q}_n}(\boldsymbol{\Sigma}^{-1}_{\mathbf{q}_n}\mu_{\mathbf{q}_n} + \mathbf{p}_m r_{m,t}) \\
\alpha' &= \alpha + \frac{1}{2} \\
\beta' &= \beta + \frac{1}{2}(\mu^{\intercal}_{\mathbf{q}_n}\boldsymbol{\Sigma}^{-1}_{\mathbf{q}_n}\mu_{\mathbf{q}_n} + r^{\intercal}_{m,t} r_{m,t} - \mu'^{\intercal}_{\mathbf{q}_n}\boldsymbol{\Sigma}'^{-1}_{\mathbf{q}_n}\mu'_{\mathbf{q}_n}) \\
\lambda'_k &= \mathcal{I}(z_{m,t} = k) r_{m,t} + \lambda_k \\
\eta'_{k,n} &= \mathcal{I}(x_{m,t} = n) r_{m,t} + \eta_{k,n}
\end{aligned}
$$

At time t, the sampling process for the parameter random variables $\mathbf{q}_n$, σ^2_n, $\mathbf{p}_m$, $\boldsymbol{\Phi}_k$ is summarized as below:

$$
\begin{aligned}
&\sigma^2_n \sim \mathcal{IG}(\alpha', \beta'), \\
&\mathbf{q}_n | \sigma^2_n \sim \mathcal{N}(\mu'_{\mathbf{q}_n}, \sigma^2_n \boldsymbol{\Sigma}'_{\mathbf{q}_n}), \\
&\mathbf{p}_m \sim Dir(\lambda'), \\
&\boldsymbol{\Phi}_k \sim Dir(\eta').
\end{aligned}
$$

Integrate with Policies: Thompson sampling

Without new observation $x_{m,t}$ and $r_{m,t}$, the particle re-sampling, latent state inference and parameter statistics inference for time t, therefore, we utilize the latent vectors p_m and q_n sampled from their posterior distributions at time t-1 predicting the reward for each arm.

In our model, each item has B independent particles. Based on Thompson sampling, the policy select an arm $\mathrm{n(t)}$ using the following equation:

$$n(t) = \arg\max_{n} \left(\bar{r}_{m,n}\right),$$

Where $\bar{r}_{m,n}$ denotes the average reward:

$$\bar{r}_{m,n} = \frac{1}{B} \sum_{i=1}^{B} \mathbf{p}_m^{(i)\intercal} \mathbf{q}_n^{(i)}.$$

Integrate with Policies: UCB

According to UCB policy, it select an arm $\mathrm{n(t)}$ based on the upper bound of the predicted reward. Assuming that

$$r_{m,t}^{(i)} \sim \mathcal{N}(\mathbf{p}_m^{(i)\intercal}\mathbf{q}_n^{(i)}, \sigma^{(i)2})$$

$$\bar{r}_{m,n} = \frac{1}{B}\sum_{i=1}^{B} r_{m,t}^{(i)}$$

the UCB is developed by the mean and variance of predicted reward.

$$n(t) = \arg\max_n \left(\bar{r}_{m,n} + \gamma\sqrt{\nu}\right),$$

where $\gamma \gg 0$ is a predefined threshold, and the variance is

$$\nu = \frac{1}{B}\sum_i^B \sigma^{(i)2}$$

ICTR Algorithms

Algorithm 2 The algorithms for ICTR model

procedure MAIN(B) ▷ Main entry.
  Initialize B particles, i.e., $\mathcal{P}^{(1)}_{m,n(0)} \dots \mathcal{P}^{(B)}_{m,n(0)}$.
  for $t \leftarrow 1, T$ **do**
    User m arrives for item recommendation.
    $n(t) = \arg\max_{n=1,N}$ EVAL(m, n) by Equation (5.24) or Equation (5.25).
    Receive $r_{m,t}$ by rating item $n(t)$.
    UPDATE$(m,\ n(t),\ r_{m,t})$.
  end for
end procedure

procedure EVAL(m, n) ▷ Get a rating score for item n, given user m.
  for $i \leftarrow 1, B$ **do** ▷ Iterate on each particle.
    Get the user latent vector $\mathbf{p}^{(i)}_m$.
    Get the item latent vector $\mathbf{q}^{(i)}_n$.
    Predict i^{th} reward $r^{(i)}_{m,t}$.
  end for
  Compute the average reward as the final reward $r_{m,t}$.
  return the score.
end procedure

procedure UPDATE(m, $n(t)$, $r_{m,t}$) ▷ Update the inference.
  for $i \leftarrow 1, B$ **do** ▷ Compute weights for each particle.
    Compute weight $\rho^{(i)}$ of particle $\mathcal{P}^{(i)}_{m,n(t)}$ by Equation (5.17).
  end for
  Re-sample $\mathcal{P}'_{m,n(t)}$ from $\mathcal{P}_{m,n(t)}$ according to the weights $\rho^{(i)}$s.
  for $i \leftarrow 1, B$ **do** ▷ Update statistics for each particle.
    Update the sufficient statistics for $z_{m,t}$ by Equation (5.21).
    Sample $z_{m,t}$ according to Equation (5.20).
    Update the statistics for $\mathbf{q}_n$, σ^2_n, $\mathbf{p}_m$, $\mathbf{\Phi}_k$ by Equation (5.22).
    Sample $\mathbf{q}_n$, σ^2_n, $\mathbf{p}_m$, $\mathbf{\Phi}_k$ by Equation (5.23).
  end for
end procedure

Experiment

- Data Set

Data Set	IBM Global IT Ticket	Data Set	Yahoo News	MovieLens (10M)
# ticket problems	1,091	#users	226,710	71,567
# automations	62	#items	652	10,681
# ratings	116,429	#ratings	280,410,150	10,000,054

- Evaluate Method
  - Replayer method. [21]

Experiment

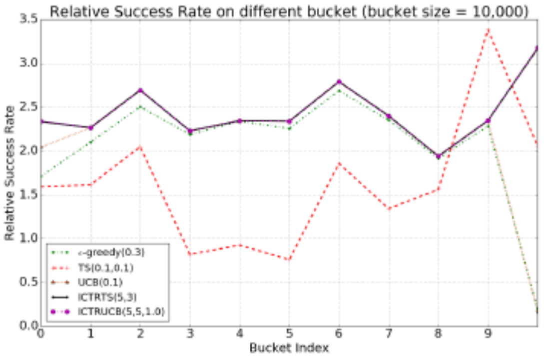


Fig. 1: The average RSR of IT ticket data is given along each time bucket. All algorithms shown here are configured with their best parameter settings.

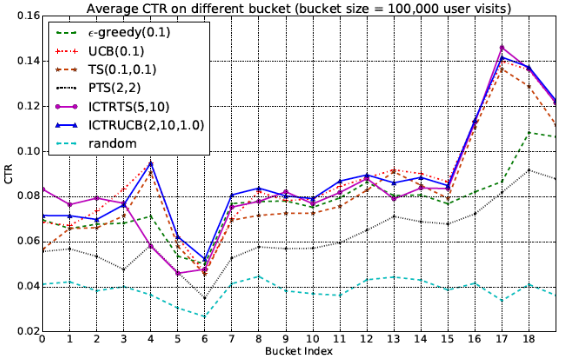


Fig. 2: The average CTR of Yahoo! Today News data is given along each time bucket. All algorithms shown here are configured with their best parameter settings.

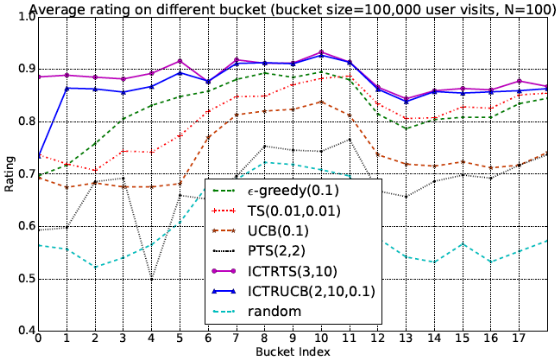


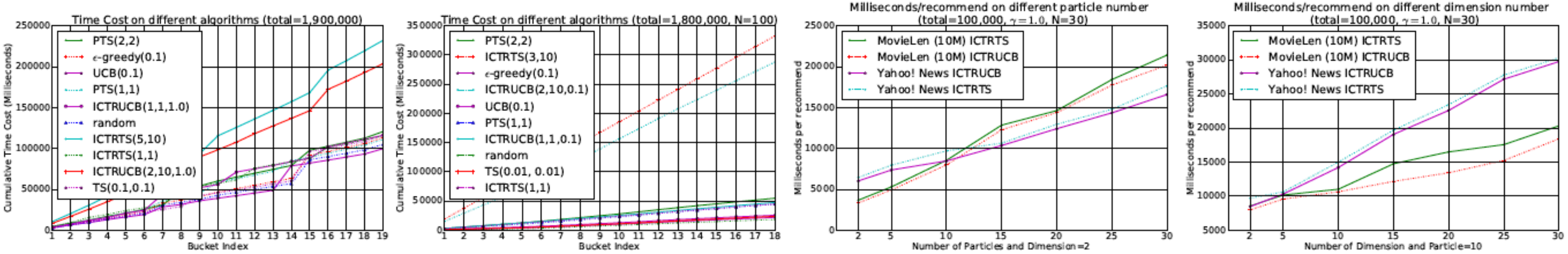
Fig. 3: The average rating of MovieLens (10M) data is given along each time bucket. All algorithms shown here are configured with their best parameter settings.

Experiment

TABLE 4: Average CTR/rating on two real world datasets.

Algorithm	Yahoo! Today News				MovieLens (10M)			
	mean	std	min	max	mean	std	min	max
ϵ-greedy(0.01)	0.06916	0.00312	0.06476	0.07166	0.70205	0.06340	0.60752	0.78934
ϵ-greedy(0.1)	**0.07566**	0.00079	0.07509	0.07678	**0.82038**	0.01437	0.79435	0.83551
ϵ-greedy(0.3)	0.07006	0.00261	0.06776	0.07372	0.80447	0.01516	0.77982	0.82458
ϵ-greedy(1.0)	0.03913	0.00051	0.03842	0.03961	0.60337	0.00380	0.59854	0.60823
UCB(0.01)	0.05240	0.00942	0.04146	0.06975	0.62133	0.10001	0.45296	0.73369
UCB(0.1)	**0.08515**	0.00021	0.08478	0.08544	**0.73537**	0.07110	0.66198	0.85632
UCB(0.5)	0.05815	0.00059	0.05710	0.05893	0.71478	0.00294	0.63623	0.64298
UCB(1.0)	0.04895	0.00036	0.04831	0.04932	0.63909	0.00278	0.60324	0.61296
TS(0.01,0.01)	0.07853	0.00058	0.07759	0.07921	**0.83585**	0.00397	0.82927	0.84177
TS(0.1,0.1)	**0.07941**	0.00040	0.07869	0.07988	0.83267	0.00625	0.82242	0.84001
TS(0.5,0.5)	0.07914	0.00106	0.07747	0.08041	0.82988	0.00833	0.81887	0.84114
TS(1.0,1.0)	0.07937	0.00079	0.07788	0.08044	0.83493	0.00798	0.82383	0.84477
PTS(2,2)	**0.06069**	0.00575	0.05075	0.06470	**0.70484**	0.03062	0.64792	0.74610
PTS(2,10)	0.05699	0.00410	0.05130	0.06208	0.65046	0.01124	0.63586	0.66977
PTS(5,10)	0.05778	0.00275	0.05589	0.06251	0.63777	0.00811	0.62971	0.65181
PTS(5,20)	0.05726	0.00438	0.05096	0.06321	0.62289	0.00714	0.61250	0.63567
PTS(10,20)	0.05490	0.00271	0.05179	0.05839	0.61819	0.01044	0.60662	0.63818
ICTRTS(2,5)	0.06888	0.00483	0.06369	0.07671	0.70386	0.15772	0.48652	0.85596
ICTRTS(2,10)	0.06712	0.01873	0.03731	0.08487	0.56643	0.10242	0.42974	0.67630
ICTRTS(3,10)	0.06953	0.00783	0.05857	0.07804	**0.88512**	0.00052	0.88438	0.88553
ICTRTS(5,10)	**0.08321**	0.08236	0.08492	0.06292	0.55748	0.14168	0.38715	0.73404
ICTRTS(7,10)	0.05066	0.00885	0.04229	0.06423	0.517826	0.07120	0.42297	0.59454
ICTRTS(7,20)	0.04925	0.00223	0.04672	0.05285	0.61414	0.12186	0.44685	0.73365
ICTRUCB(2,10,0.01)	0.06673	0.01233	0.04588	0.08112	0.44650	0.06689	0.38678	0.53991
ICTRUCB(2,10,1.0)	**0.08597**	0.00056	0.08521	0.08675	**0.86411**	0.01528	0.85059	0.88547
ICTRUCB(3,10,0.05)	0.07250	0.00426	0.06799	0.07694	0.54757	0.13265	0.43665	0.73407
ICTRUCB(3,10,1.0)	0.08196	0.00296	0.07766	0.08530	0.57805	0.08716	0.46453	0.67641
ICTRUCB(5,10,0.01)	0.07009	0.00722	0.06411	0.08244	0.62282	0.02572	0.59322	0.65594
ICTRUCB(5,10,1.0)	0.08329	0.00140	0.08098	0.08481	0.80038	0.24095	0.29625	0.88554

Experiment



(a) Cumulative time cost of Yahoo! Today News is given along each time bucket.

(b) Cumulative time cost of MovieLens (10M) is given along each time bucket.

(c) Time cost is given with different number of particles.

(d) Time cost is given with different number of latent feature vector dimensions.

Fig. 4: Time cost comparison on both two datasets.

A Case Study on Ticket Data

- ICTR model can fully learns the latent feature vector of each automation.
- We are trying to categorize an automation named "process missing" using Euclidean distance.

UNCATEGORIZED AUTOMATION	process missing	
CATEGORIZED AUTOMATION	**CATEGORY**	**EUCLIDEAN DISTANCE**
(1) db2 percent db connection executing is to high automation	DATABASE	1.086
(1) process cpu spike automation	UNIX	1.014
(2) swap automation		0.858
(1) windows service automation	**WINDOWS***	**0.565***

An example of categorizing an automation.

A Case Study on MovieLens (10M)

Table 5.5: Movie topic distribution of MovieLens (10M).

Topic Cluster I			Topic Cluster II		
MovieId	MovieName	MovieType	MovieId	MovieName	MovieType
32	12 Monkeys	Sci-Fi,Thriller	344	Pet Detective	Comedy
50	Usual Suspects	Crime,Mystery,Thriller	588	Aladdin	Children,Animation,Comedy
590	Dances with wolves	Adventure,Drama,Western	595	Beauty and the Beast	Animation,Children,Musical
592	Batman	Action,Crime,Sci-Fi,Thriller	2857	Yellow Submarine	Adventure,Animation,Comedy,Musical

Summary of this section

This work has been published by IEEE TKDE 2018 and IEEE Big Data 2018.

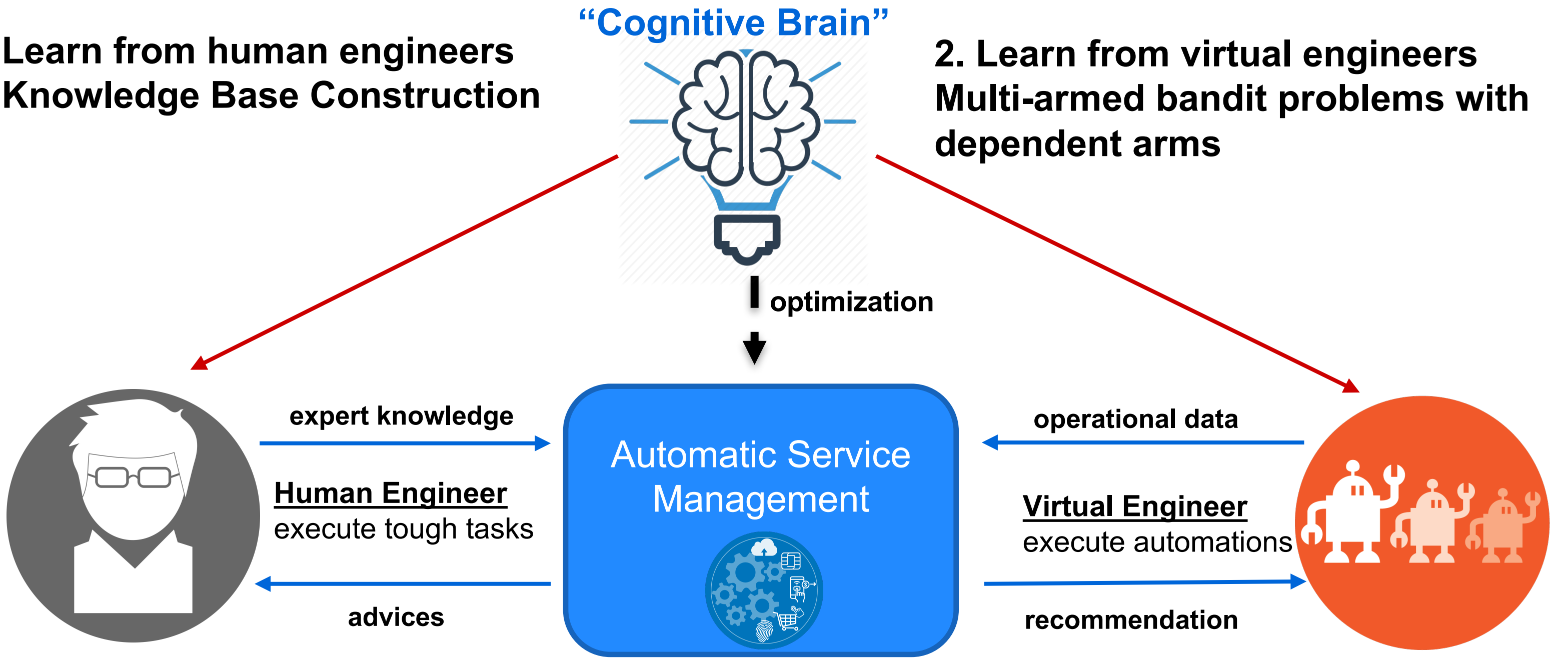
1. **Wang, Qing**, et al. "**Online interactive collaborative filtering using multi-armed bandit with dependent arms**." *IEEE Transactions on Knowledge and Data Engineering* (2018).

2. **Wang, Qing**, et al, "**AISTAR: An Intelligent Integrated System for Online IT Ticket Automation Recommendation**", In Proceedings of the 6th annual IEEE International Conference on Big Data (IEEE Big Data 2018), Seattle, WA, USA 2018.

Summary

1. **Learn from human engineers**
   Knowledge Base Construction

2. **Learn from virtual engineers**
   Multi-armed bandit problems with dependent arms



References

1. N. F. Noy, D. L. McGuinness et al., “Ontology development 101: A guide to creating your first ontology,” 2001.
2. K. Mahesh, S. Nirenburg et al., “A situated ontology for practical nlp,” in IJCAI, vol. 19. Citeseer, 1995, p. 21.
3. W. IJntema, F. Goossen, F. Frasincar, and F. Hogenboom, “Ontology-based news recommendation,” in EDBT/ICDT. ACM, 2010, p. 16.
4. Bedini, Ivan, and Benjamin Nguyen. "Automatic ontology generation: State of the art." PRiSM Laboratory Technical Report. University of Versailles (2007).
5. Ding, Ying, and Schubert Foo. "Ontology research and development. Part 1-a review of ontology generation." Journal of information science 28.2 (2002): 123-136.
6. M. Dahab, A. Hassan, and A. Rafea. “Textontoex: Automatic ontology construction from natural english text.” Expert Systems with Applications, 34(2):1474{1480, 2008.
7. O. Deshpande, D. S Lamba, M. Tourn, et al. “Building, maintaining, and using knowledge bases: a report from the trenches.” In SIGMOD, pages 1209-1220. ACM, 2013.
8. C. Lee, Y. Kao, et al. “Automated ontology construction for unstructured text documents.” Data & Knowledge Engineering, 60(3):547-566, 2007.
9. https://stanfordnlp.github.io/CoreNLP/annotators.html
10. Welch, T. Technique for high-performance data compression. Computer 17, 6(1984), 8-19.
11. Commentz-Walter, Beate. "A string matching algorithm fast on the average. "International Colloquium on Automata, Languages, and Programming. Springer Berlin Heidelberg, 1979.
12. IBM Enterprise IT Automation Services. www.redbooks.ibm.com/redpapers/pdfs/redp5363.pdf.
13. S. Chang, J. Zhou, P. Chubak, J. Hu, and T. S. Huang. A space alignment method for cold-start tv show recommendations. In Proceedings of the 24th International Conference on Artificial Intelligence, 2015.
14. P. Auer. Using confidence bounds for exploitation-exploration trade-offs. Journal of Machine Learning Research, 3(Nov):397-422, 2002.
15. X. Zhao, W. Zhang, and J. Wang. Interactive collaborative filtering. In CIKM, pages 1411-1420. 2013.
16. C. Zeng, Q. Wang, S. Mokhtari, and T. Li. Online context-aware recommendation with time varying multi-armed bandit. In SIGKDD, pages 2025-2034. 2016.

References

17. S. Pandey, D. Agarwal, and V. Chakrabarti, D.and Josifovski. Bandits for taxonomies: A model-based approach. In SDM, pages 216-227. SIAM, 2007.
18. S. Pandey, D. Chakrabarti, and D. Agarwal. Multi-armed bandit problems with dependent arms. In ICML, pages 721-728. ACM, 2007.
19. Y. Yue, A. Hong, and C. Guestrin. Hierarchical exploration for accelerating contextual bandits. arXiv preprint arXiv:1206.6454, 2012.
20. O. Chapelle and L. Li. An empirical evaluation of Thompson sampling. In NIPS, pages 2249-2257, 2011.
21. L. Li, W. Chu, J. Langford, T. Moon, and X. Wang. An unbiased offline evaluation of contextual bandit algorithms with generalized linear models. JMLR, 26:19-36, 2012.
22. C. Wang and D. M Blei. 2011. Collaborative topic modeling for recommending scientific articles. In SIGKDD. ACM, 448–456.
23. K. Wang, W. Zhao, H. Peng, and X. Wang. 2017. Bayesian Probabilistic Multi-Topic Matrix Factorization for Rating Prediction. (2017).
24. J. Kawale, H. H Bui, B. Kveton, L. Tran-Thanh, and S. Chawla. 2015. Efficient Thompson Sampling for Online Matrix-Factorization Recommendation. In NIPS.
25. Q.Wu, H.Wang, Q. Gu, and H.Wang. 2016. Contextual bandits in a collaborative environment. In SIGIR. ACM, 529–538.
26. L. Zhou and E. Brunskill. 2016. Latent contextual bandits and their application to personalized recommendations for new users. preprint arXiv:1604.06743 (2016).
27. R. Salakhutdinov and A. Mnih. 2007. Probabilistic Matrix Factorization. In NIPS.
28. R. Salakhutdinov and A. Mnih. 2008. Bayesian probabilistic matrix factorization using Markov chain Monte Carlo. In ICML. ACM, 880–887.
29. Qing Wang, Chunqiu Zeng, Wubai Zhou, Tao Li, S. S. Iyengar, Larisa Shwartz, Genady Ya. Grabarnik, “Online Interactive Collaborative Filtering Using Multi-armed Bandit with Dependent Arms”, In the IEEE Transactions on Knowledge and Data Engineering (TKDE).
30. http://research.microsoft.com/en-us/projects/bandits/

Related Publications

1. **Qing Wang**, Chunqiu Zeng, S. S. Iyengar, Tao Li, Larisa Shwartz, Genady Ya. Graharnik, “AISTAR: An Intelligent Integrated System for Online IT Ticket Automation Recommendation”, In Proceeding of the 6th annual IEEE International Conference on Big Data (IEEE Big Data 2018), Seattle, Washington, 2018.
2. **Qing Wang**, Wubai Zhou, Chunqiu Zeng, Tao Li, Larisa Shwartz, Genady Ya. Graharnik, “A Knowledge-Based Deep Ranking Model for Cognitive IT Service Management”, In the IEEE Transactions on Service Computing (TSC) (submitted).
3. **Qing Wang**, Chunqiu Zeng, Wubai Zhou, Tao Li, S. S. Iyengar, Larisa Shwartz, Genady Ya. Graharnik, “Online Interactive Collaborative Filtering Using Multi-armed Bandit with Dependent Arms”, In the IEEE Transactions on Knowledge and Data Engineering (TKDE).
4. **Qing Wang**, Tao Li, S. S. Iyengar, Larisa Shwartz, Genady Ya. Graharnik, “Online IT automation recommendation Using Hierarchical Multi-armed Bandit Algorithms", SIAM International Conference on Data Mining (SDM 2018), San Diego, California, USA, 2018.
5. **Qing Wang**, Wubai Zhou, Chunqiu Zeng, Tao Li, Larisa Shwartz, Genady Ya. Graharnik, “Constructing the Knowledge Base for Cognitive IT Service Management”, In Proceeding of the 14th IEEE International Conference on Services Computing (IEEE SCC 2017), Honolulu, Hawaii, USA, 2017. **[Best Student Paper Award]**

Other Publications

1. Wubai Zhou, Wei Xue, Ramesh Baral, **Qing Wang**, Chunqiu Zeng, Tao Li, Jian Xu, Zhen Liu, Larisa Shwartz, Genady Ya. Graharnik, “STAR: A System for Ticket Analysis and Resolution”, In Proceeding of the 23rd annual ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2017), Halifax, Nova Scotia, Canada, 2017.
2. Wei Xue, Wubai Zhou, Tao Li, **Qing Wang**, “MTNA: A Neural Multi-Task Model for Aspect Category Classification and Aspect Term Extraction on Restaurant Reviews”, In Proceeding of the 8th International Joint Conference on Natural Language Processing (IJCNLP 2017), Taipei, Taiwan, 2017.
3. Chunqiu Zeng, **Qing Wang**, Shekoofeh Mokhtari, Tao Li, “Online Context-Aware Recommendation with Time Varying Multi-Armed Bandit”, In Proceeding of the 22nd annual ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2016), San Francisco, USA, 2016.
4. Tao Li, Wubai Zhou, Chunqiu Zeng, **Qing Wang**, Qifeng Zhou, Dingding Wang, Yue Huang, Jia Xu, Wentao Wang, Minjing Zhang, Steve Luis, Shu-Ching Chen and Naphtali Rishe, “DI-DAP: An Efficient Disaster Information Delivery and Analysis Platform in Disaster Management” (CIKM 2016), Indianapolis, USA, 2016.
5. Chunqiu Zeng, **Qing Wang**, Wentao Wang, Tao Li, Larisa Shwartz, “Online Inference for Time varyingTemporal Dependency Discovery form Time Series”, (IEEE Big Data 2016), Washington D.C., USA.
6. Tao Li, Chunqiu Zeng, Wubai Zhou, Wei Xue, Yue Huang, Zheng Liu, Qifeng Zhou, Bin Xia, **Qing Wang**, Wentao Wang, Xiaolong Zhu, “FIU-Miner (a fast, integrated, and user-friendly system for data mining) and its applications”, Knowledge and Information Systems, 2016.

Acknowledgement

- **Advisor:** Dr. S. S. Iyengar and Dr. Tao Li (RIP)
- **Co-Advisor:** Dr. Shu-Ching Chen
- **Committee Members:**

  Dr. Mark A. Finlayson

  Dr. Ning Xie

  Dr. Debra VanderMeer
- **Mentors:**

  Dr. Larisa Shwartz (@IBM), Dr. Genday Ya. Grabarnik (@St. John's University)
- **Department Staff:**

  Dr. Jason Liu, Olga Carbonell, Ariana Taglioretti, Vanessa Cornwall, Lian Zhang, Eric S. Johnson, Luis Rivera, etc.
- **Thank every colleague from KDRG.**

Q & A