Directed Graph Analysis: Algorithms and Applications

Jiankai Sun
Advisor: Srinivasan Parthasarathy

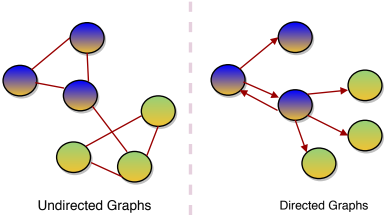
The Ohio State University

October 15, 2018

The Ohio State University

Introduction to Directed Graph Analysis

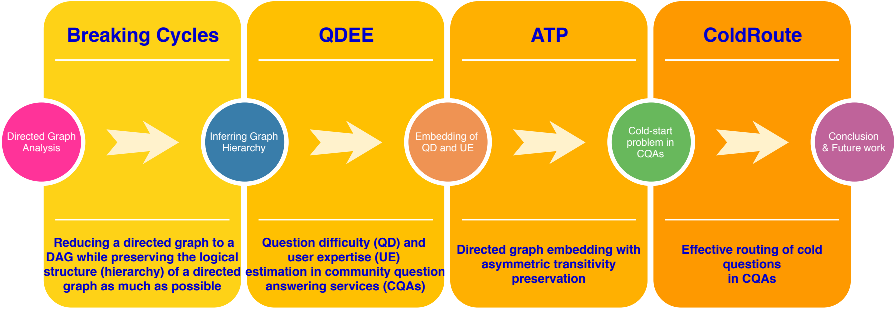
- A directed graph is a graph, i.e. a set of objects (named as nodes or vertices) that are connected together, where all the edges have an association direction with them
- Examples: WWW, Twitter, Citation network, Dependencies in software modules, Taxonomy graphs in knowledge base, etc
- Directionality $\Longrightarrow$ Asymmetrical relationships (matrices) pose challenges
  - spectral analysis is more complex
  - can not be easily extended from undirected graphs



Proposal Statement

- Focus on two properties: graph hierarchy and asymmetric transitivity
- inferring and leveraging graph hierarchy
  - Alg.: breaking cycles while preserving the logical hierarchy of the directed graph as much as possible
  - App.: question difficulty and user expertise estimation in CQAs
  - App.: recommending people to unfollow some accounts from their following list in Twitter
- preserving and leveraging asymmetric transitivity
  - Alg.: representing nodes in a low-dimensional vector space in which asymmetric transitivity can be captured
  - App.: routing newly posted questions to suitable users in CQAs

Work flow of Directed Graph Analysis: Algorithms and Applications



Breaking Cycles in Noisy Hierarchies

Jiankai Sun [1]
Deepak Ajwani [2] Patrick Nicholson [2] Alessandra Sala [2]
Srinivasan Parthasarathy[1]

[1]The Ohio State University

[2]Bell Labs, Nokia, Ireland

WebSci'17, June 26 -28, 2017

NOKIA Bell Labs

The Ohio State University

Outline

1. Motivation
2. Related Work
3. Our Framework: Breaking Cycles via Graph Hierarchies
4. Experiments
5. Conclusion

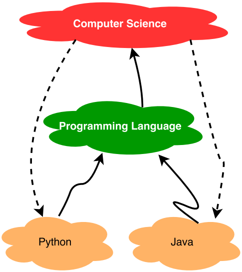
NOKIA Bell Labs

The Ohio State University

[0] Pic from Web

Motivation

- Ontological knowledge bases such as Wikipedia categories, created in crowd-sourced way, cause errors (cycles)
- Taxonomy graphs that capture "has a" or "is a" relationships should be acyclic
- Breaking Cycles to get a Directed Acyclic Graph (DAG) can benefit other applications such as job/dataflow scheduling



NOKIA Bell Labs

THE OHIO STATE UNIVERSITY

Related Work

- Simple Heuristic Based on BFS or DFS
  - DFS: un-deterministic
  - BFS: remove more edges, even non-cycle edges
- Minimum Feedback Arc Set
  - NP-Hard
  - Cannot preserve graph hierarchy
- Domain-specific Algorithms

NOKIA Bell Labs

[0] Pic from Web

THE OHIO STATE UNIVERSITY

Graph Hierarchy Based Framework

Goal: break cycles from a directed graph, while preserving the underlying hierarchy of the relationships as much as possible

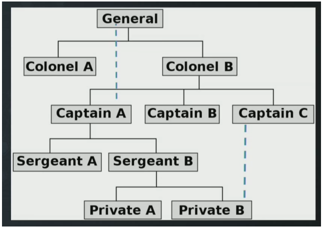
1. Inferring graph hierarchy
   - TrueSkill
   - SocialAgony
2. Proposing strategies to select violation edges as candidates for removal based on graph hierarchy
   - Forward
   - Backward
   - Greedy

NOKIA Bell Labs

The Ohio State University

Finding a ranking function to infer graph hierarchy

- f assigns a ranking score to each node in the graph
- A higher ranking score indicates the corresponding node is higher up (or more general) in the hierarchy
- Edges violate the hierarchy (edges from a higher/general group to a lower/specific group) are potential edges for removal



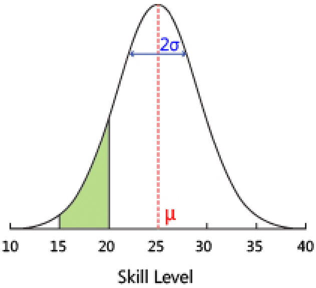
NOKIA Bell Labs

THE OHIO STATE UNIVERSITY

[0] Figure: http://bit.ly/2sZTH3c

Inferring Graph Hierarchy by TrueSkill

- TrueSkill ranking system is a skill based ranking system to rank Xbox players, developed by Microsoft Research
- Each player has two numbers
  - μ: average skill of the player
  - σ : degree of uncertainty in the player's skill



NOKIA Bell Labs

The Ohio State University

View it as a competition graph

- a directed graph $G = (V, E) \Rightarrow$ a multi-player tournament with $|V|$ players and $|E|$ competitions
- an edge $(u, v) \in E \Rightarrow u$ loses the game between u and v

Updates of skill levels given an edge (u, v)

- If player v has a higher skill level than u, then the outcome of edge (u, v) is expected $\Rightarrow$ small updates in skill level μ and σ.
- If player u has a higher skill level than v, then the outcome of edge (u, v) is unexpected $\Rightarrow$ large updates in skill level μ and σ.

NOKIA Bell Labs

The Ohio State University

Inferring Graph Hierarchy by TrueSkill

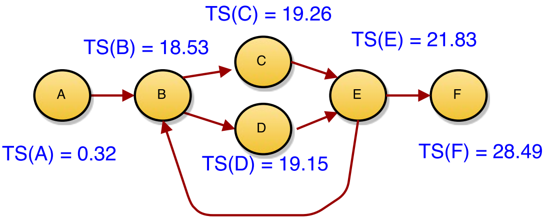


Figure: TrueSkill Compuation Demo

- A node v's ranking score in the graph hierarchy: $f_{ts}(v) = \mu_v - 3\sigma_v$
- As far as we know, **graph hierarchy inference as a competition problem** has not been researched yet

Inferring Graph Hierarchy by Social Agony

- In social networks such as Twitter, people are not likely to follow people who are lower in the hierarchy
- Agony can be caused when people follow other people who are lower in the hierarchy
- Social agony proposed by Gupte et al. assumes the existence of an edge indicates a rank recommendation
  - An edge $u \Rightarrow v$ indicates a recommendation of v from u
  - If there is no reverse edge from v to u, it could indicate that v is higher up in the hierarchy than u

NOKIA Bell Labs

[0]Figure: http://bit.ly/2r7afHV

THE OHIO STATE UNIVERSITY

Computation of Graph Agony

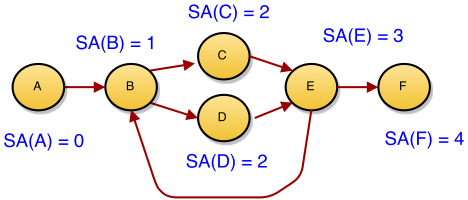
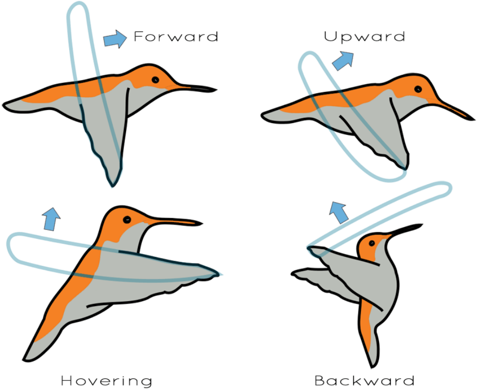


Figure: SocialAgony Compuation Demo

- Gupte et al., Tatti et al. proposed efficient algorithms to find a ranking r to minimize the agony of the graph
- A node v's ranking score in the graph hierarchy inferred by social agony: $f_{agony}(v) = r(v)$

We provide 3 solutions to select violation edges

- Forward
- Backward
- Greedy



NOKIA Bell Labs

THE OHIO STATE UNIVERSITY

[0] Figure: `http://bit.ly/2sCJNrf`

Forward to select edges to remove and break cycles

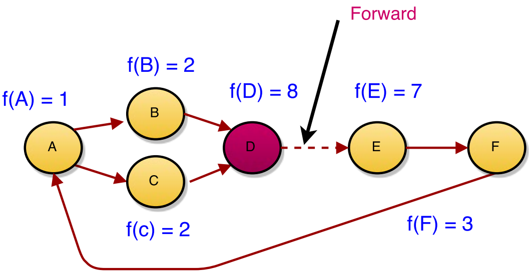


Figure: Strategy Forward to select violation edges

- *Forward*: Select the node which has the *highest* ranking score in the SCC and then remove its all *out* edges.

Backward to select edges to remove and break cycles

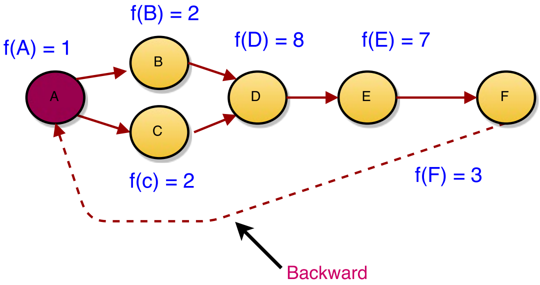


Figure: Strategy Forward to select violation edges

- *Backward*: Select the node which has the *lowest* ranking score in the SCC and then remove its all *in* edges.

Greedy to select edges to remove and break cycles

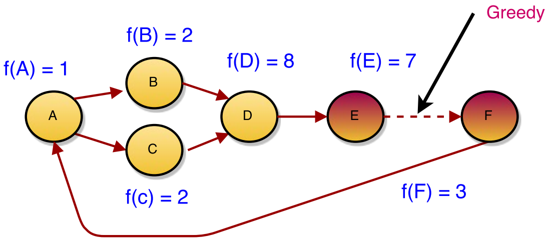


Figure: Strategy Forward to select violation edges

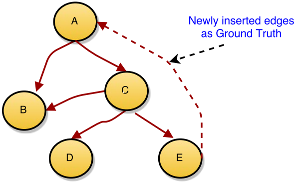
- *Greedy*: Select the edge which violates the hierarchy the *most* to remove.

Combine Them Together

- Two ways to infer graph hierarchy: TrueSkill and SocialAgony
- Three solutions to select edges: *Forward, Backward, Greedy*
- $\Rightarrow$ Six strategies to break cycles
  - TS_G, TS_B, TS_F
  - SA_G, SA_B, SA_F
- Assembled together: *H_Voting* selects the edge with the highest voting score for removal
  - voting score for an edge e: $\sum_m (I_m(e))$
  - $m \in \{TS\_G, TS\_F, TS\_B, SA\_G, SA\_F, SA\_B\}$
  - if edge e is removed by method m, $I_m(e) = 1$, otherwise $I_m(e) = 0$
  - remove the edge with the highest voting score first

Experimental Setup

- Few large real taxonomy graphs have ground truth (edges are labeled as errors)
- Introduce cycles (randomly) to real and synthetic DAGs
    - insert edges that violate the partial order

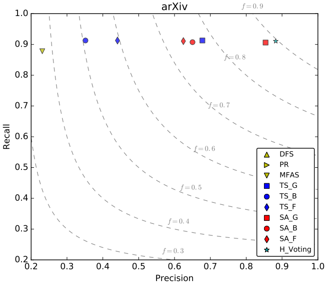


- Evaluation Measures: precision, recall, and f-measure

NOKIA Bell Labs

The Ohio State University

Performance

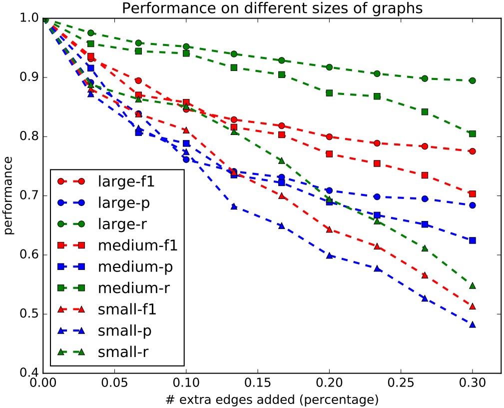


- Results on more datasets showing comparable results are available in our paper

NOKIA Bell Labs

THE OHIO STATE UNIVERSITY

Sensitivity to Number of Noisy Edges



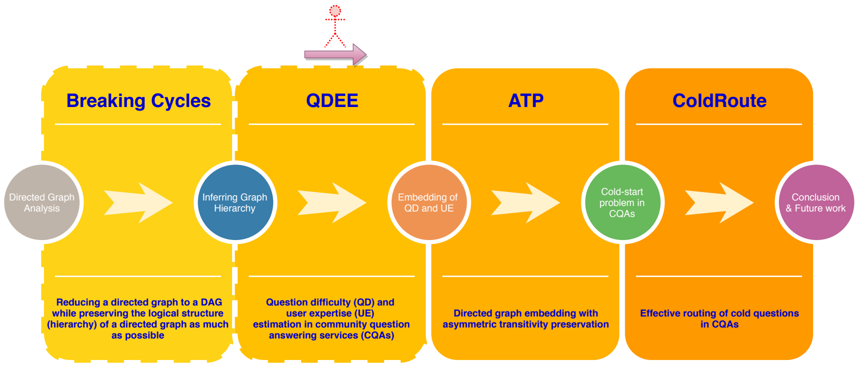
Conclusion & Future Work

- Main Contribution
  - our approach addresses the problem of breaking cycles while preserving the graph hierarchy
  - we are the first researchers to infer graph hierarchy by viewing it as a competition problem
  - we propose several strategies and an ensemble approach to identify edges that should be removed
- Future Work
  - propose a model-based approach to predict which edge should be removed
- **Code is available on GitHub** [1]

NOKIA Bell Labs

The Ohio State University

[1] `https://goo.gl/491v7q`


Breaking Cycles
QDEE
ATP
ColdRoute
Directed Graph Analysis
Inferring Graph Hierarchy
Embedding of QD and UE
Cold-start problem in CQAs
Conclusion & Future work
Reducing a directed graph to a DAG while preserving the logical structure (hierarchy) of a directed graph as much as possible
Question difficulty (QD) and user expertise (UE) estimation in community question answering services (CQAs)
Directed graph embedding with asymmetric transitivity preservation
Effective routing of cold questions in CQAs

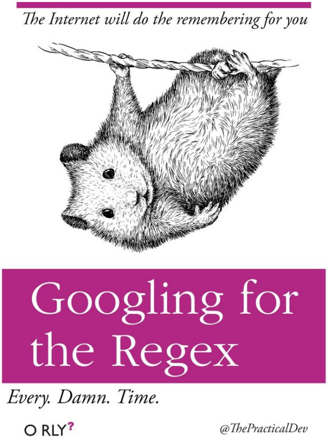
QDEE: Question Difficulty and Expertise Estimation in Community Question Answering Services

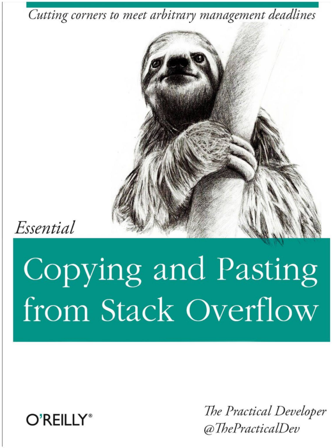
Jiankai Sun
Sobhan Moosavi   Rajiv Ramnath   Srinivasan Parthasarathy

The Ohio State University

ICWSM, June 25 - 28, 2018

Object-Oriented-Programming (OOP) vs StackOverflow-Oriented-Programming (SOP)



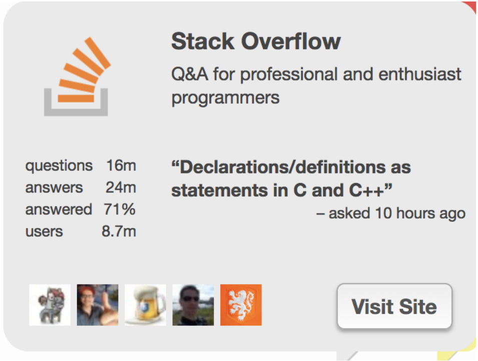


[0] Left pic: `https://goo.gl/5vKuaR`

[0] Right pic: `https://goo.gl/XAG4DP`

Motivation

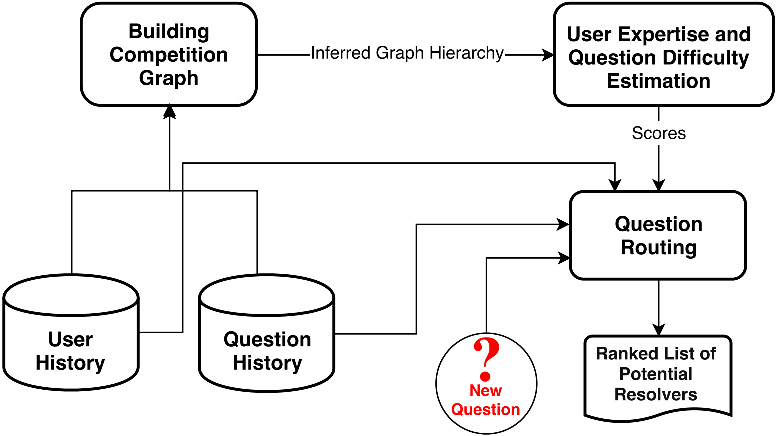
- What if you could not find answers for your questions in Stack Overflow? 4.8m unanswered



0 Pic: https://goo.gl/3VEsV9

QDEE Framework

Goal: Routing questions to users with matching expertise based on question difficulty level



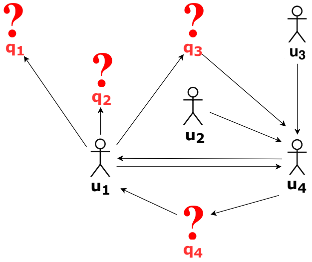
How to Build the Competition Graph [Liu et al. SIGIR'11]

Heterogeneous Nodes Type

- question
- users have two roles: asker and answerer

Construction of Edges

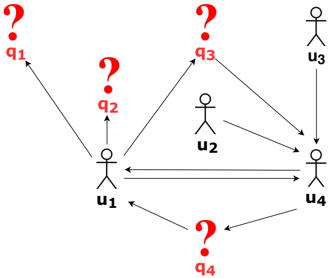
- asker ⇒ question : (u_1, q_3)
- question ⇒ best answerer : (q_3, u_4)
- asker ⇒ best answerer : (u_1, u_4)
- *non-best answerers* ⇒ best answerer : (u_2, u_4), and (u_3, u_4).



Data Sparseness Problem and EGA

Data Sparseness Problem

- Each question has only one in-edge (from asker) and one out-edge (to the best answerer)
- May not provide enough information to achieve an accurate estimation



Expertise Gain Assumption (EGA)

Users typically gain expertise across multiple interactions with the CQA and tend to ask more difficult questions within the same domain over time

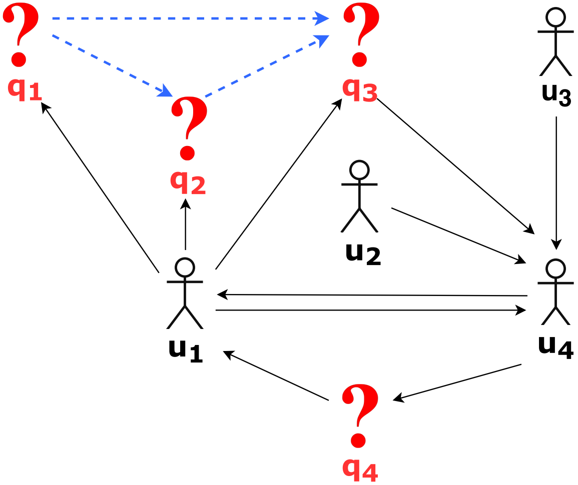
Illustration of EGA

The difficulty levels of questions are increasing over time

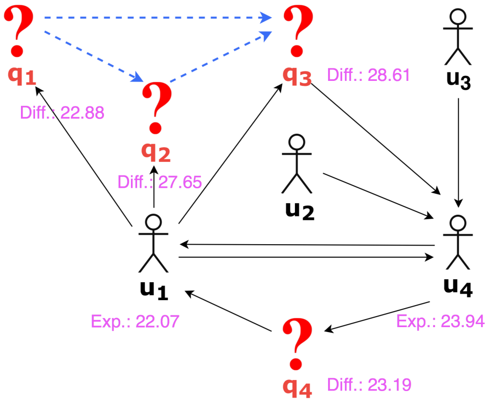
Table: Questions asked by a Stack Overflow user in Python.

Questions in Python	Question-Date
q_a: use basic build-in function *sum* on a *list*	July 2013
q_b: changing a list element in multiple lists	Sept. 2013
q_c: *list comprehension* and *generator*	Oct. 2013
q_d: copying 2-D Python list of arbitrary length	Feb. 2014
q_e: using *regular expressions* in *Python*	Nov. 2014

Competition Graph with EGA



How to estimate question difficulty and user expertise



- Two ways: TrueSkill [Herbrich et al. NIPS'07, Liu et al. SIGIR'11] and Social Agony [Gupte et al. WWW'11, Tatti ICDM'15]
- The Biggest Advantage: Language Agnostic

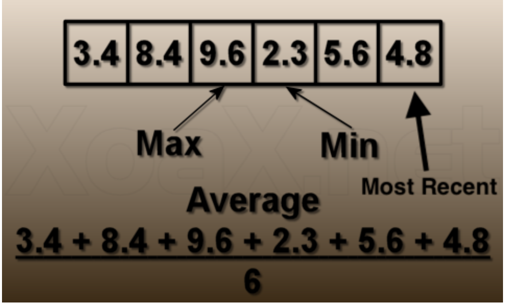
Cold-Start Estimation and Routing

- Cold-Start: newly posted questions with no answers (unseen nodes during the training)?
- A common problem in recommender systems [Cheng et al. WWW'17, Wang et al. TIST'14, CIKM'12, Sun et al. CIKM'12]

[0] Pic: https://goo.gl/aVh8v5

Cold-Start Difficulty Estimation

- Given a cold-start (q^*) and well-resolved questions $q_1, q_2, ..., q_k$ asked by the same user within the same domain
- q^*'s difficulty level : maximum difficulty level of $q_1, ..., q_k$ $---$*Max*
- Max$(q_1, ..., q_k)$ $! = q_k$ $---$*Most Recent*



0 Pic: `https://goo.gl/5SzVCJ`

Cold-Start Routing: Identifying a Set of Potential Answerers to Route Newly Posted Questions

- Language agnostic
  - **Q:** Users who have answered questions with similar difficulty as q^*
- Language conscious
  - **T:** Users who have answered questions that are textually similar to q^*
  - Potential Answerers to Route **QT**: Combination of **Q** and **T**

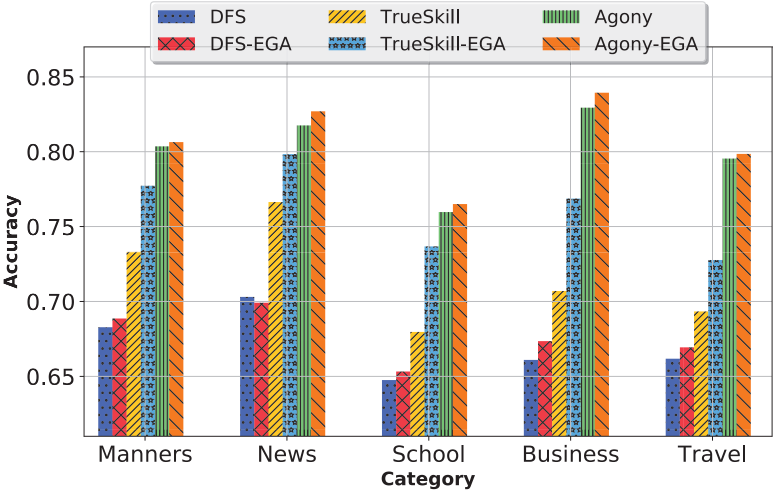
0 Pic: `https://goo.gl/QiDGwM`

Experimental Setup

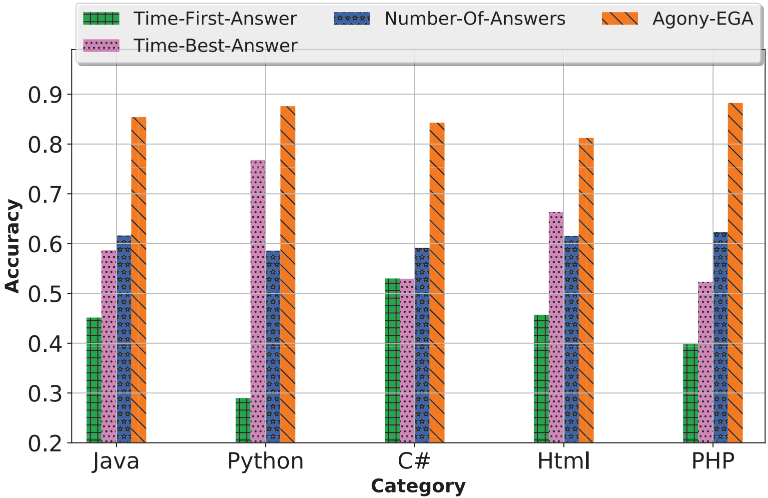
- Datasets: Yahoo! Answers (Japanese) and Stack Overflow (English)
- Ground Truth: Question's *coin* (or *bounty*), given to the user who provided the best answer, as indicator of question difficulty level
- Accuracy Metric:

$$Accuracy = \frac{\text{\# correctly predicted question pairs}}{\text{\# valid question pairs}}$$

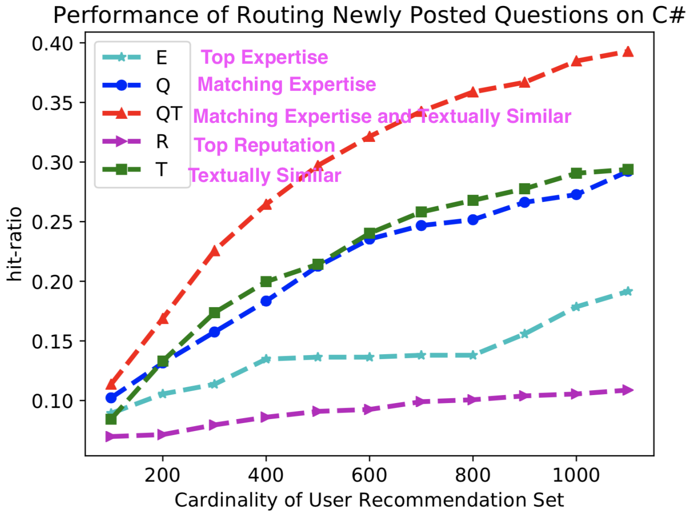
Yahoo! Answers: Advantages of Leveraging EGA



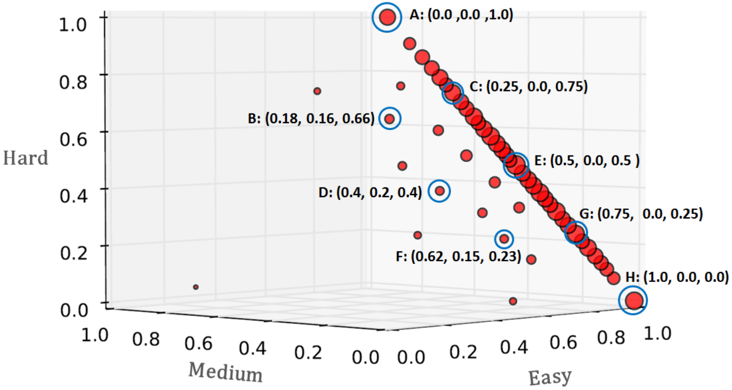
Stack Overflow: Social Agony vs State-of-the-art



Different Approaches for Cold-Start Routing



Patterns of User Answering Questions



Conclusion

- Proposed Expertise Gain Assumption (EGA) to solve the data sparseness problem in CQAs
- Leveraged graph hierarchy (social agony) to estimate question difficulty and user expertise
- Proposed approaches to route cold questions to users with matching expertise
- Supported by NSF grants CCF-1645599 and IIS-1550302 and a grant from the Ohio Supercomputer Center (PAS0166)
- Code is available on GitHub: `https://github.com/zhenv5/QDEE`

Q & A

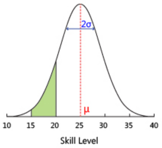
THE OHIO STATE UNIVERSITY

Backup Slides

THE OHIO STATE UNIVERSITY

Inferring Graph Hierarchy by TrueSkill [Herbrich et al. NIPS'07, Liu et al. SIGIR'11]

- A skill based ranking system to rank Xbox players, developed by Microsoft Research
- Each player has two numbers
  - μ: average skill of the player
  - σ: degree of uncertainty in the player's skill
- a directed graph $G = (V, E) \Rightarrow$ a multi-player tournament with $|V|$ players and $|E|$ competitions
- an edge $(u, v) \in E \Rightarrow u$ loses the game between u and v
- a node v's ranking score in the graph hierarchy: $f_{ts}(v) = \mu_v - 3\sigma_v$



Inferring Graph Hierarchy by Social Agony [Gupte et al. WWW'11, Tatti ICDM'15]

- In social networks such as Twitter, people are not likely to follow people who are lower in the hierarchy
- Agony can be caused when people follow other people who are lower in the hierarchy
- Defined by the severity of their violation, agony to u caused by edge (u, v) is equal to $max(r(u) - r(v) + 1, 0)$
- The agony in the network given a ranking r
  - sum of agony on each edge
- Goal: find a ranking r that minimize the total agony in the graph

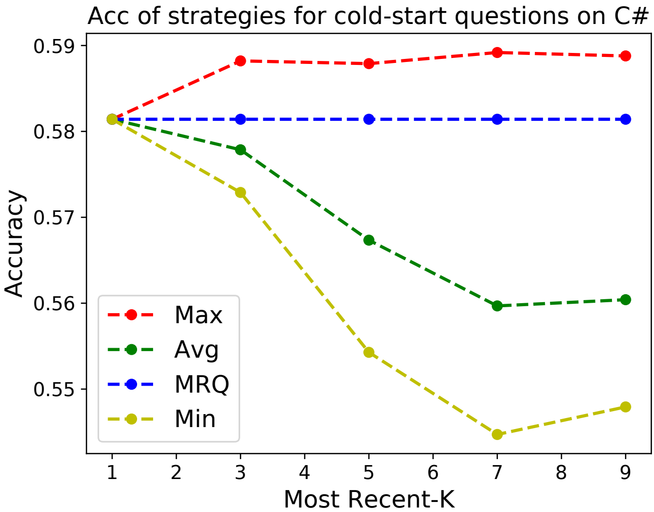
Cold-Start Difficulty Estimation

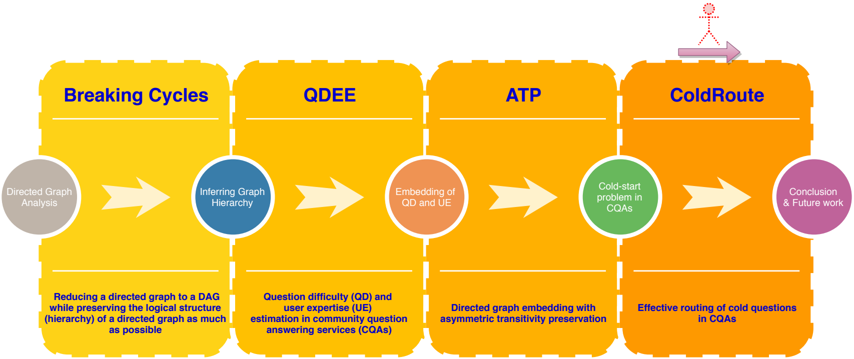
- what if the cold-start question q^* was asked by a new registered user who has no asking history?
- KNN: The difficulty score of q^* is predicted as the averaged difficulty scores of its nearest neighbors ($d_{knn}(q^*)$). [Wang et al. EMNLP'14]

Combine Them Together

$$d(q^*) = \alpha \cdot d_{knn}(q^*) + (1 - \alpha) \cdot d_{ega}(q^*)$$

Strategy Selection for Cold-Start Estimation




Breaking Cycles
QDEE
ATP
ColdRoute
Directed Graph Analysis
Inferring Graph Hierarchy
Embedding of QD and UE
Cold-start problem in CQAs
Conclusion & Future work
Reducing a directed graph to a DAG while preserving the logical structure (hierarchy) of a directed graph as much as possible
Question difficulty (QD) and user expertise (UE) estimation in community question answering services (CQAs)
Directed graph embedding with asymmetric transitivity preservation
Effective routing of cold questions in CQAs

ColdRoute: Effective Routing of Cold Questions in Stack Exchange Sites

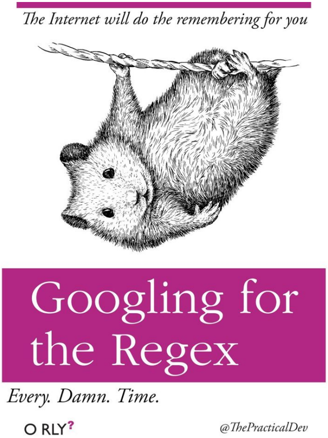
Jiankai Sun [1]
Abhinav Vishnu [2] Aniket Chakrabarti [3] Charles Siegel [2]
Srinivasan Parthasarathy [1]

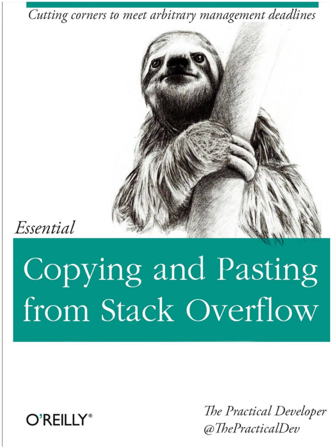
[1]The Ohio State University

[2]Pacific Northwest National Laboratory [3]Microsoft

ECML PKDD, Sep 2018

Object-Oriented-Programming vs StackOverflow-Oriented-Programming



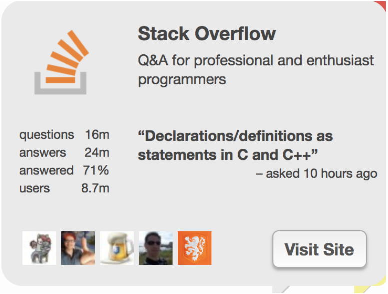


0 Left pic: `https://goo.gl/5vKuaR`

0 Right pic: `https://goo.gl/XAG4DP`

Motivation

- What if you could not find answers for your questions in Stack Overflow? 4.8m unanswered
- Cold questions: newly posted questions without answer (cold questions) asked by new registered (cold askers) or existing askers
- Cold-Start Problem: Find the right experts to answer cold questions



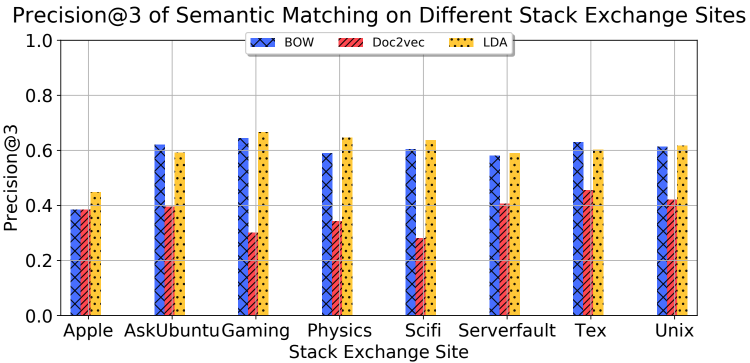
[0] Pic: https://goo.gl/3VEsV9

Related Work: Semantic Matching Models

- Leverage textual information to route cold questions
  - question → (answer with the highest semantic similarity) → best answerer
  - cold question → user with the highest semantic similarity
- How to model textual information?
  - BOW : bag of words [Zhou et al. 2012; 2013; Figueroa and Neumann 2013;]
  - LDA: Latent Dirichlet allocation, Topic Modeling [Guo et al. 2008; Ji et al. 2012]
  - Doc2Vec: Distributed Representations of Sentences and Documents [Le and Mikolov 2014; Dong et al. 2015]

Finding best answers by semantic matching

- $Precision@3$ computes the average number of times that the best answer is ranked top-3 by a specific method



Challenge I:
Find the indicator of the best answerer

Up Vote

3

Down Vote

1. MLP is sensitive to feature scaling. Have you normalzied your data?
2. Modify your network structure: add more hidden layers and change number of perceptrons of each layer
3. change activation function to sigmod/tanh/relu etc.

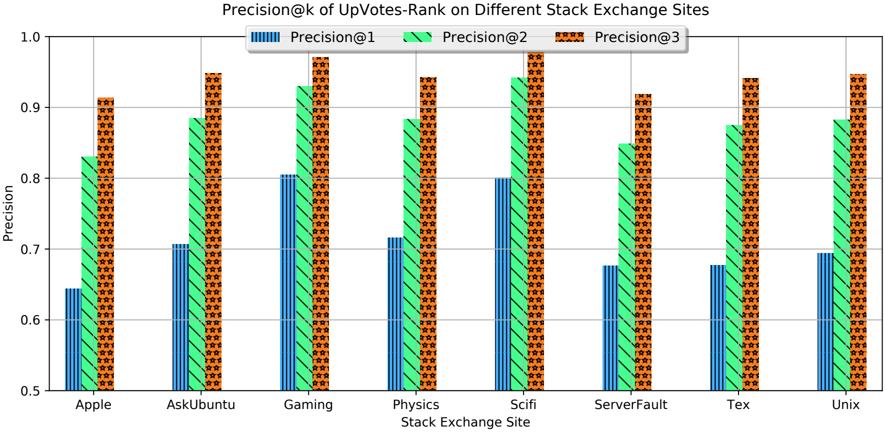
share improve this answer

answered Jun 5 '17 at 18:07
zhenv5
103 ● 4

4. Change learning rate: `learning_rate`, `learning_rate_init`. 5. Toggle `early_stopping` – ijoseph Apr 10 at 22:13

How about voting score (up-votes - down-votes)?

UpVotes-Rank: select the answerer with the highest voting score as the best answerer



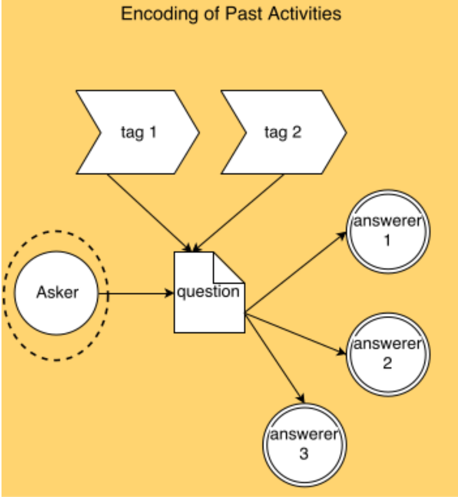
Model it as A Regression Problem

- How to identify the best answerer for a newly posted question?
    - Model the cold-start problem as a regression problem?

- Predict voting score for each question-answerer pair
- Select the user who has the highest voting score as the best answerer

Challenge II:

How to encode past activities as feature vectors?



Encode Past Activities by One-hot Encoding

	Feature Vector $\vec{X}$																	Target $\vec{y}$
$\vec{x}^{(1)}$	0	0	1	$\cdots$	1	0	0	$\cdots$	1	0	0	$\cdots$	0	1	0	$\cdots$	4	$y^{(1)}$
$\vec{x}^{(2)}$	0	0	1	$\cdots$	0	1	0	$\cdots$	1	0	0	$\cdots$	0	1	0	$\cdots$	3	$y^{(2)}$
$\vec{x}^{(3)}$	0	0	1	$\cdots$	0	0	1	$\cdots$	1	0	0	$\cdots$	0	1	0	$\cdots$	2	$y^{(3)}$
$\vec{x}^{(4)}$	0	1	0	$\cdots$	0	0	1	$\cdots$	0	0	1	$\cdots$	0	0	1	$\cdots$	5	$y^{(4)}$
$\vec{x}^{(5)}$	0	1	0	$\cdots$	0	1	0	$\cdots$	0	0	1	$\cdots$	0	0	1	$\cdots$	6	$y^{(5)}$
$\vec{x}^{(6)}$	1	0	0	$\cdots$	1	0	0	$\cdots$	0	1	0	$\cdots$	1	1	1	$\cdots$	2	$y^{(6)}$
$\vec{x}^{(7)}$	1	0	0	$\cdots$	0	0	1	$\cdots$	0	1	0	$\cdots$	1	1	1	$\cdots$	4	$y^{(7)}$
	q_1	q_2	q_3	$\cdots$	u_1	u_2	u_3	$\cdots$	a_1	a_2	a_3	$\cdots$	t_1	t_2	t_3	$\cdots$		
	Question				Answerer				Asker				Question Tags					

Challenge III:
Feature vector is very sparse

- Each feature vector $\vec{x}^{(i)}$ has only $(3 + ||\vec{t_i}||_1)$ ones.
- $||\vec{t_i}||_1$ represents question q_i's number of tags (number of ones in the vector $\vec{t_i}$).
- Average number of tags per question in our experiments is 2.5

Advantages of Factorization Machine [Rendle et al. 2010, 2012]

- Regression model: $\hat{\vec{y}}(\vec{x}) = w_0 + \sum_{i=1}^{p} w_i x_i + \sum_{i=1}^{p} \sum_{j=i+1}^{p} x_i x_j < \vec{v}_i, \vec{v}_j >$
- Can handle sparse settings very well in comparison with other regressors such as linear regression
- Gives us the flexibility to explore the different features' relative importance in cold question routing (feature selection)
    - asker
    - tags
    - textual descriptions: question head and question body

A Toy Example

- Given a cold question q_4 asked by a new asker a_4 with tags $t = \{t_1, t_2, t_3\}$, Predict voting score of u_3?

- Linear Regression

$$\hat{y}(\vec{x}) = \boxed{w_0 + w_{q_4} + w_{u_3} + w_{a_4} + \sum_{i=1}^{3} w_{t_i}} + \sum_{i \in \boldsymbol{S}} \sum_{j \in \boldsymbol{S}, i<j} \langle \vec{v}_i, \vec{v}_j \rangle$$

- where $w_0 \in \mathbb{R}, \vec{w} \in \mathbb{R}^p, \vec{V} \in \mathbb{R}^{p \times k}$

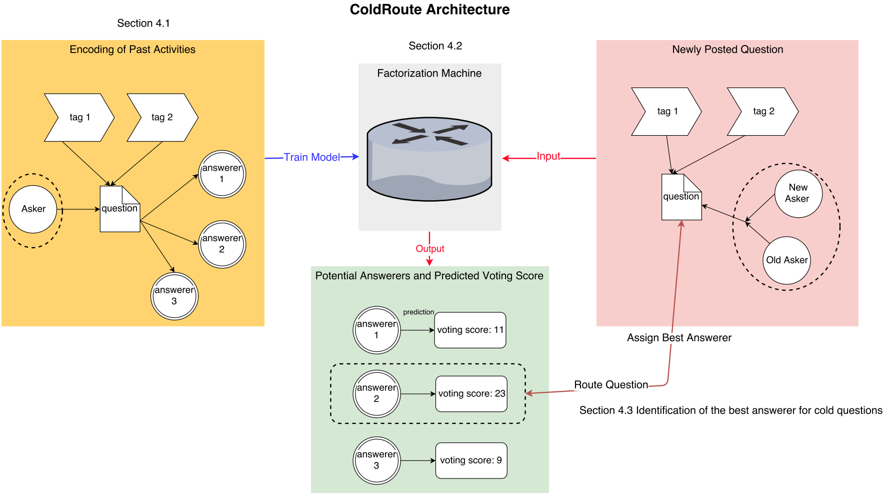
A Toy Example

- Given a cold question q_4 asked by a new asker a_4 with tags $t = \{t_1, t_2, t_3\}$, Predict voting score of u_3?
- Linear Regression

$$\hat{y}(\vec{x}) = \boxed{w_0 + w_{q_4} + w_{u_3} + w_{a_4} + \sum_{i=1}^{3} w_{t_i}} + \sum_{i \in \boldsymbol{S}} \sum_{j \in \boldsymbol{S}, i<j} \langle \vec{v}_i, \vec{v}_j \rangle$$

- Interactions among question, asker, answerer, and tags $\boldsymbol{S} = \{q_4, u_3, a_4, t_1, t_2, t_3\}$
- where $w_0 \in \mathbb{R}, \vec{w} \in \mathbb{R}^p, \vec{V} \in \mathbb{R}^{p \times k}$

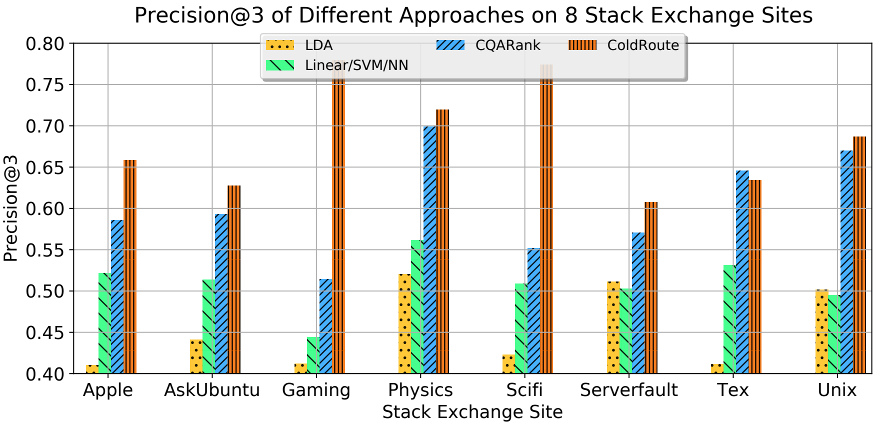
Putting it all together: Architecture of ColdRoute



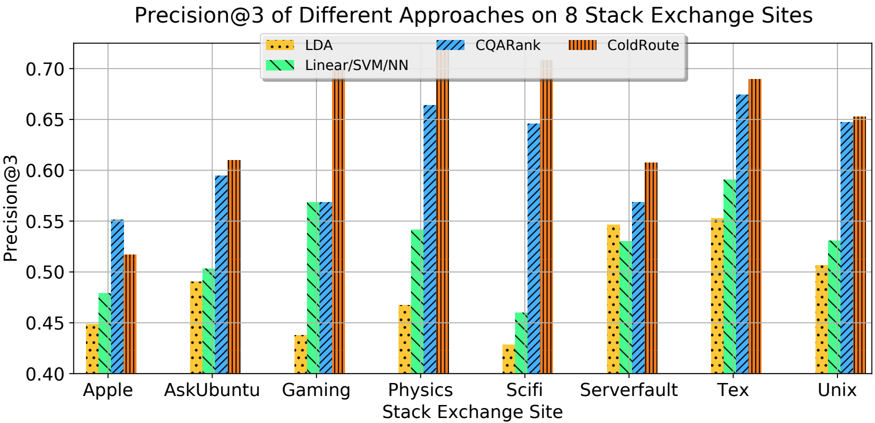
Experiments

- Datasets: **8** Stack Exchange Sites (Apple, Physics, Gaming, etc)
- Evaluation metrics: Mean Reciprocal Rank (MRR), Precision@k, and Accuracy
- Two different kinds of cold questions:
  - asked by new registered users
  - asked by existing users (have asked questions before)

Routing Cold Questions Asked by Existing Askers (ColdRoute vs state-of-the-arts)



Routing Cold Questions Asked by New Askers (ColdRoute vs state-of-the-arts)



Feature Selection for Routing Cold Questions Asked by New Askers (Evaluated by $Precision@3$)

	Apple	Ask.	Gaming	Physics	Scifi
BOW	0.3840	0.4357	0.3000	0.3799	0.2484
Doc2Vec	0.3840	0.4096	0.3563	0.3493	0.2547
LDA	0.4487	0.4902	0.4375	0.4672	0.4286
ColdRoute-HB	0.5133	0.4989	0.4625	0.5109	0.4534
ColdRoute-B	0.4829	0.5139	0.4563	0.5284	0.3975
ColdRoute-H	0.4829	0.5468	0.5063	0.5633	0.4907
ColdRoute-T	0.5171	0.6100	0.7000	0.7249	0.7081

Conclusion and Future Work

Conclusion

- Propose **ColdRoute** for tacking cold questions routing in CQAs
- Overcome **3** challenges
  - Use voting score as the indicator of the best answerer
  - Encode users' past activities by one-hot encoding
  - Address sparse settings by Factorization Machine (**FM**)
- Simple modes are good like **FM** which are proven to be better than simple DL algorithms
- Code: `https://github.com/zhenv5/ColdRoute`

Future work

- Improve over the state-of-the-art models a lot, Still have improvement space
- Combine the power of deep neural network and factorization machines

Acknowledgments

- **Acknowledgments:** This work is supported by NSF grants CCF-1645599, IIS-1550302, and CNS-1513120, and a grant from the Ohio Supercomputer Center (PAS0166). All content represents the opinion of the authors, which is not necessarily shared or endorsed by their sponsors.

Q & A

THE OHIO STATE UNIVERSITY

Appendix

FM as a regressor

Consider a 2-way FM ($d = 2$) as an example:

$$\hat{\vec{y}}(\vec{x}) = w_0 + \sum_{i=1}^{p} w_i x_i + \sum_{i=1}^{p} \sum_{j=i+1}^{p} x_i x_j < \vec{v}_i, \vec{v}_j > \quad (1)$$

where the model parameters that have to be estimated are:

$$w_0 \in \mathbb{R}, \vec{w} \in \mathbb{R}^p, \vec{V} \in \mathbb{R}^{p \times k} \quad (2)$$

And $< \cdot, \cdot >$ is the dot product of two vectors of size k:

$$< \vec{v}_i, \vec{v}_j > = \sum_{f=1}^{k} v_{i,f} v_{j,f} \quad (3)$$

where a row $\vec{v}_i \in \vec{V}$ describes the i-th variable with $k \in \mathbb{N}_0^+$ factors. k represents the dimensionality of the factorization.

Gradient descent to update parameters

$$\frac{\partial}{\partial \theta}\hat{\vec{y}}(\vec{x}) = \begin{cases} 1, & \text{if } \theta \text{ is } w_0 \\ x_i, & \text{if } \theta \text{ is } w_i \\ x_i \sum_{j=1}^{p} v_{j,f} x_j - v_{i,f} x_i^2, & \text{if } \theta \text{ is } v_{i,f} \end{cases} \tag{4}$$

Evaluation Metric

MRR.

The MRR measure is given by

$$MRR = \frac{1}{|Q|} \sum_{q \in Q} \frac{1}{r^q_{best}} \tag{5}$$

where r^q_{best} is the position of question q's best answerer in the predicted ranking list. It's worth mentioning that MRR is equivalent to Mean Average Precision (MAP) since the number of correct elements (the best answerer) in the predicted ranking list is just 1.

Evaluation Metric

Precision@k.

The $Precision@k$ is applied to measure the average number of times that the best answerer is ranked on top-k by a certain algorithm.

$$Precision@k = \frac{\{q \in Q | r^q_{best} <= k\}}{|Q|} \tag{6}$$

Accuracy. The Accuracy is used to measure the ranking quality of the best answerer, given by

$$Accuracy = \frac{1}{|Q|} \sum_{q \in Q} \frac{|R^q| - r^q_{best}}{|R^q| - 1} \tag{7}$$

Where $Accuracy = 1$ (best) means that the best answerer returned by an algorithm always ranks on top while $Accuracy = 0$ means the opposite.

Statistics of Stack Exchange Sites (Ask., Ser. are short for AskUbuntu and Serverfault respectively)

	Apple	Ask.	Gaming	Physics	Scifi	Ser.	Tex	Unix
# Questions	80,466	257,173	75,696	93,529	38,026	238,764	129,182	111,505
# Answers	119,878	337,198	130,294	137,258	78,652	398,470	169,354	171,016
# Unique Users	65,851	189,955	51,192	41,115	26,673	130,951	48,049	65,279
# Questions having Best Answers	29,765	85,843	45,798	38,094	21,740	117,275	76,862	53,856
# Unique Tags	1,048	3,020	4,437	876	2,349	3,514	1,525	2,438
Avg # Tags per Question	2.824	2.6982	1.2823	2.9634	2.1967	2.882	2.2752	2.7868
# Askers	40,206	137,171	25,153	31,415	12,413	93,739	42,819	45,773
# Asker (asked only 1 question) (%)	76.74%	75.88%	74.23%	63.26%	74.71%	64.04%	62.55%	68.48%
Avg # Questions per Asker	1.9758	1.8557	2.9689	2.8849	3.0031	2.4411	2.9851	2.4022

References

- [Submission] **J. Sun**, B. Bandyopadhyay, A. Bashizade, J. Liang, S. Parthasarathy, and P. Sadayappan "ATP: Asymmetric Transitivity Preserving Directed Graph Embedding,"
- [Submission] **J. Sun**, A. Vishnu, A. Chakrabarti, C. Siegel, and S. Parthasarathy, "ColdRoute: Effective Routing of Cold Questions in Stack Exchange Sites,"
- [Sun. et al. ICWSM'18] **J. Sun**, S. Moosavi, R. Ramnath, and S. Parthasarathy, "QDEE: Question Difficulty and Expertise Estimation in Community Question Answer Systems," *The 12th International AAAI Conference on Web and Social Media (**ICWSM**)*, 2018
- [Liang. et al. SDM'18] J. Liang, P. Jacobs, **J. Sun**, and S. Parthasarathy, "Semi-supervised embedding in Attributed Networks with Outliers," *SIAM International Conference on Data Mining (**SDM**)*, 2018
- [Sun. et al. WebSci'17] **J. Sun**, D. Ajwani, P. Nicholson, A. Sala, and S. Parthasarathy, "Breaking Cycles in Noisy Hierarchies," *The 9-th International ACM Web Science Conference (**ACM WebSci**)*, 2018
- [Cheng et al. WWW'17] P. Cheng, S. Wang, J. Ma, **J. Sun**, and Hui Xiong, "Learning to Recommend Accurate and Diverse Items," *The 26th International Conference on World Wide Web (**WWW**)*, 2017
- [Wang et al. TIST'14] S. Wang, **J. Sun**, B. Gao and J. Ma, "VSRank: A Novel Framework for Ranking-based Collaborative Filtering," in *ACM Transactions on Intelligent Systems and Technology (**ACM TIST**)*, 2014
- [Sun et al. CIKM'12] **J. Sun**, S. Wang, B. Gao and J. Ma, "Learning to rank for hybrid recommendation," *The 21-st ACM International Conference on Information and Knowledge Management (**CIKM**)*, 2012
- [Wang et al. CIKM'12] S. Wang, **J. Sun**, B. Gao and J. Ma, "Adapting vector space model to ranking-based collaborative filtering," *The 21-st ACM International Conference on Information and Knowledge Management (**CIKM**)*, 2012

References 2

- [Herbrich et al. NIPS'07] TrueSkill^{TM}: A Bayesian Skill Rating System
- [Gupte et al. WWW'11] Finding Hierarchy in Directed Online Social Networks
- [Tatti ICDM'15] Hierarchies in Directed Networks
- [Liu et al. SIGIR'11] Competition-based User Expertise Score Estimation
- [Wang et al. EMNLP'14] A Regularized Competition Model for Question Difficulty Estimation in Community Question Answering Services
- [He et al. SIGIR'17] Neural factorization machines for sparse predictive analytics
- [Guo et al. arXiv'17] DeepFM: A Factorization-Machine based Neural Network for CTR Prediction
- [Zhao et al. DASFAA'15] Cold-Start Expert Finding in Community Question Answering via Graph Regularization