Analyzing Expert Behaviors in Collaborative Networks

Huan Sun[1], Mudhakar Serivatsa[2], Shulong Tan[1], Yang Li[1], Lance Kaplan[3], Shu Tao[2], and Xifeng Yan[1]

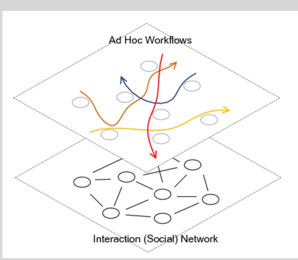
[1]University of California, Santa Barbara

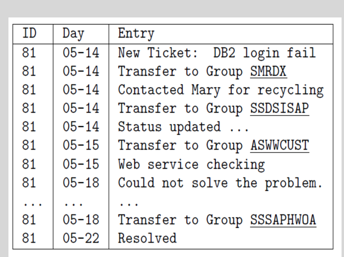
[2]IBM T. J. Watson, [3]U.S. Army Research Lab

Introduction

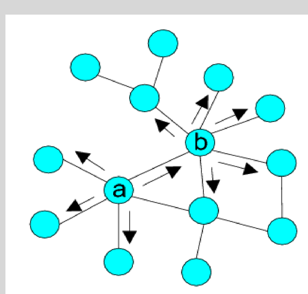
Collaborative Networks

A special type of social networks, where members collaborate with each other to complete specific tasks.

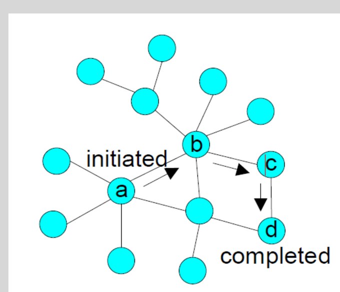




Differences from general social networks



Social nets



Collaborative nets

Core Problems

- **How do experts make routing decisions?**
- **Who have made inefficient routing decisions?**
- **How to optimize the routing performance through targeted training?**
- **Can the completion time of a task be predicted so that one can act early for difficult tasks?**

Code Online

www.cs.ucsb.edu/~huansun/behavemodel.htm

Observations

Observation 1

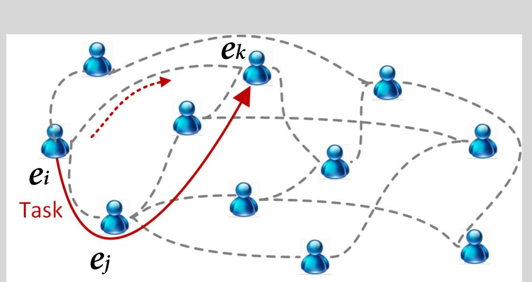
Tasks with similar content, but different routing sequences e.g., two problem tickets in IBM IT service department

Task ID	Task Content	Routing Sequence
492	Need password reset for kasperj on machine "pathfinder"". Route to NUS_N_DCRCHAIX	12 → 505 → 1914 → 1915 → 1916 → 247
494	Need password reset for jhallacy on machine ""pathfinder"". Route to NUS_N_DCRCHAIX	12 → 13 → 86

Routing decision is not deterministic, given a certain task.

Observation 2

An expert might not directly send a task to a resolver.



Is it because he does not understand the task very well, thus randomly routing it? Or he believes the other expert has a better chance to solve it, or a better chance to find the right expert to solve it?

Observation 3

An expert tends to transfer a task to some expert whose expertise is *neither* too close *nor* too far from his own. That is, not necessarily the final resolver!

Observations (Cont'd)

Observation 3

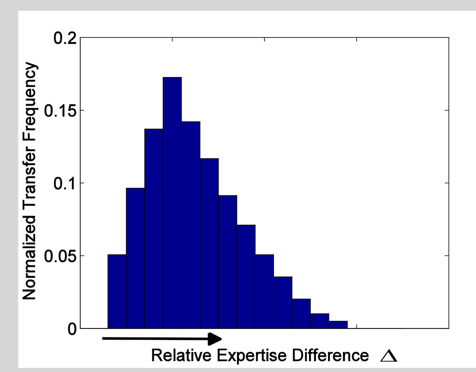


Figure plots the distribution of the relative expertise difference between a task sender and receiver in the training set.

$$\Delta(e_i,e_j)=\|e_i-e_j\|/\|e_i\|$$

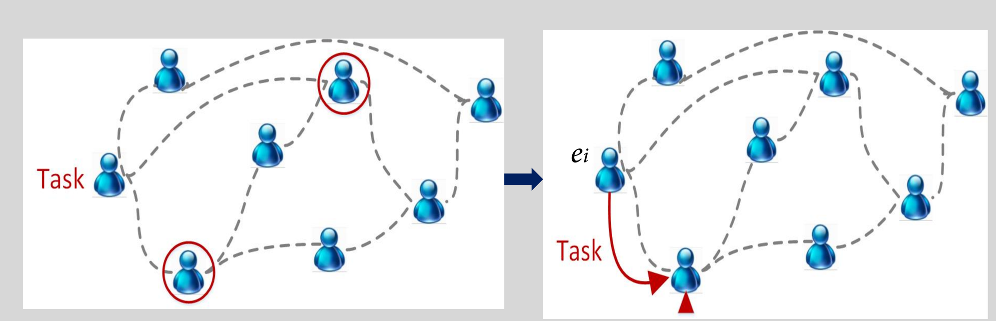
The log-normal density shows the general trend of an expert sending a task to another, given their expertise.

Modeling Expert Decision Logic

A Two-phase Assumption

When an expert transfers a task,

Phase 1 Phase 2



Candidate pool establishment Specific expert selection

Phase 1: The establishment of candidate pool C.

Two Routing Strategies	
Task-Neutral Routing (TNR)	**Task-Specific Routing (TSR)**
C: All the neighbors (all the experts he has contacted)	C: experts in one 's neighborhood, but estimated capable of solving the task

How to decide C in TSR?

Logistic Model

Estimate expert knowledge and capability based on the tasks he has dealt with before.

$$P(e_i,t)=\frac{1}{1+\exp(-(W_t t+W_2 e_i+b))}$$

Where e_i denotes the expert's expertise; t is the description of a task; W_1 , W_2 , b are model parameters. Train sets are formulated based on a set of historical tasks and their routing sequences.

Phase 2: How to select an expert from C ?

- Uniform Random (UR)
- Volume-biased (load-based) Random (VR)

$$P(e_i \xrightarrow{t} e_j \mid \mathrm{VR})=\mathrm{I}(e_j\in C)\frac{v_{ij}}{\sum_{e_k\in C} v_{ik}}$$

v_{ij} : the number of tasks transferred from e_i to e_j

- Expertise Difference (EX)

f_{ij} : the general trend of expert e_i sending a task to e_j, given their expertise.

Modeling Expert Decision Logic (Cont'd)

$$P(e_i \xrightarrow{t} e_j \mid \mathrm{EX})=\mathrm{I}(e_j\in C)\frac{f_{ij}}{\sum_{e_k\in C} f_{ik}}$$

f_{ij} is estimated based on the log-normal distribution observed previously.

$$f_{ij}\propto\frac{1}{\Delta(e_i,e_j)}e^{-\frac{[\ln\Delta(e_i,e_j)-\mu]^2}{2\sigma^2}}$$

Generative Model

Overall, six routing patterns:

$$\mathrm{TNR}^{ur},\mathrm{TNR}^{vr},\mathrm{TNR}^{ex},\mathrm{TSR}^{ur},\mathrm{TSR}^{vr},\mathrm{TSR}^{ex}$$

Decision generation process:

For each expert e_i to transfer tasks,

-Draw the mixture weights of 6 routing patterns:

$$\theta_i\sim\mathrm{Dirichlet}(\alpha)$$

(reflecting e_i 's preferences over adopting different routing patterns)

-For each task t to be transferred by expert e_i,

* Draw a pattern label: $Z_{i,t}\sim\mathrm{Mult}(\theta_i)$

* Draw an expert from the candidate pool to receive t.

The likelihood of the transfer relationships is formulated:

$$\mathcal{L}=P(e_i\xrightarrow{t}r_{i,t},\forall t\in\mathcal{T}_i,\forall e_i\in\mathcal{E}\mid\alpha,\mu,\sigma^2)$$

Solve model parameters by $\arg\max_{\alpha,\mu,\sigma^2}\log\mathcal{L}$ through variational EM.

Task Completion Time Prediction

- Why: To take early actions for those difficult tasks
- How: Given a task and its initial expert, simulate a routing sequence of L experts to process the task.

$$\widehat{CT}=\sum_{m=1}^{L} m\prod_{n=1}^{m-1}[1-P(r_n,t)]P(r_m,t)$$

Where m and n are the indices of experts in the sequence.

Experiments

Datasets

•Three datasets: Real-world problem ticket data collected from a problem ticketing system, in an IBM IT service department.

•75% training, 25% testing.

Datasets	Description	# of tasks	# of experts	% of tasks with CT		
				=2	=3	>=4
DB2	Database usage	26,740	55	44.2	34.3	21.5
WebSphere	Enterprise software	65,786	234	39.0	36.2	24.8
AIX	Operation system	120,780	404	40.0	39.4	20.6

Evaluation Measures

1. Routing sequence likelihood (LL) on testing datasets
2. Mean absolute error (MAE)

$$\mathrm{MAE}=\frac{1}{|\text{Test Set}|}\sum_{t\in\text{Test Set}}|\widehat{CT}_t-CT_t|$$

3. Step loss measure (SL)

$$\mathrm{SL}=\frac{1}{|\text{Test Set}|}\sum_{t\in\text{Test Set}}\mathrm{I}(|\widehat{CT}_t-CT_t|\rhd 1)$$

Experiments (Cont'd)

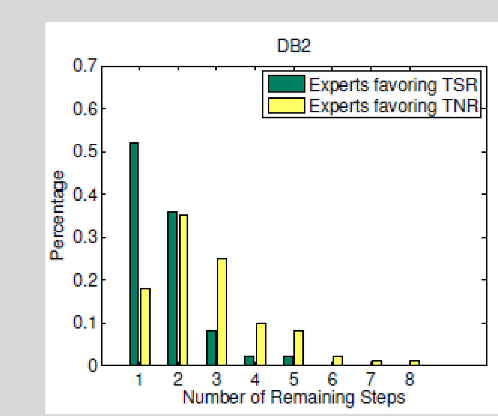
Alternative Algorithms

1. Regression: Support Vector Regression (SVR) [2], Bayesian Regression [3]
2. Generative models: Miao et al. [1]

Model Accuracy

DB2			
Models	Step Loss (%)	MAE	LL($\times10^4$)
TNR	4.11	0.30	-0.28
TSR	4.56	0.29	-0.25
TNR+TSR	**1.77**	**0.08**	**-0.07**
TNR+TSR−EX	3.05	0.14	-0.10
Miao *et al.* [12]	9.89	0.68	-0.61
SVR	14.78	0.80	N/A
Bayesian regression	13.77	0.84	N/A

Routing Efficiency



Experts favoring TNR	Weight on TNR patterns > Weight on TSR patterns.
Experts favoring TSR	Weight on TNR patterns < Weight on TSR patterns

Efficiency evaluation

Check the number of remaining experts needed to resolve a task, after an expert favoring TNR or TSR routes it.

Optimizing Collaborations

Which expert should be trained first to adopt TSR? How much efficiency gain can we expect?

- Random: random selection
- Frequent transferor: select the expert who transfers the most tasks
- Least efficient: select the least efficient expert

Methods	Efficiency Improvement (%)
Random	0.27
Frequent Transferrer	0.91
Least Efficient	1.21
Recommendation with Our Model	**2.75**

Conclusion

•We analyzed the decision making and cognitive process of an expert during task routing in collaborative networks, through a generative model.

•Our analytical model accurately predicts a task's completion time in the current collaborative network, with more than 75% improvements under three different quality measures.

•We have also shown that our model provides guidance on optimizing the performance of collaborative networks.

References

[1] Miao et al. "Generative Models for Ticket Resolution in Expert Networks", *KDD*, 2010

[2] Cristianini et al., "An introduction to support vector machines and other kernel-based learning methods ", *Cambridge university press*, 2000

[3]Mitchell et al., "Machine learning", *Burr Ridge, IL: McGraw Hill, 45*, 1997.