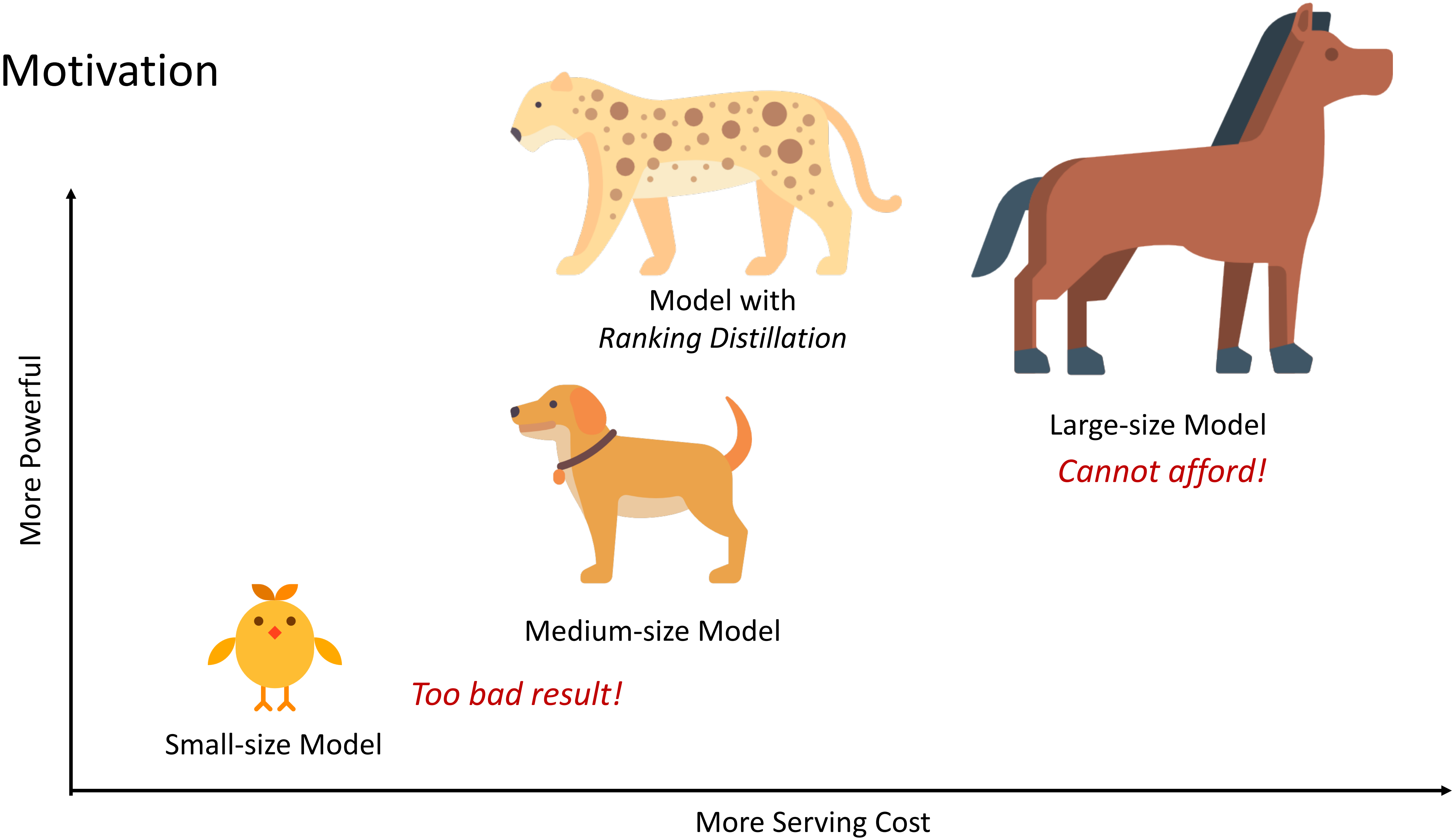
Ranking Distillation: Learning Compact Ranking Models With High Performance for Recommender System

Jiaxi Tang and Ke Wang

Simon Fraser University

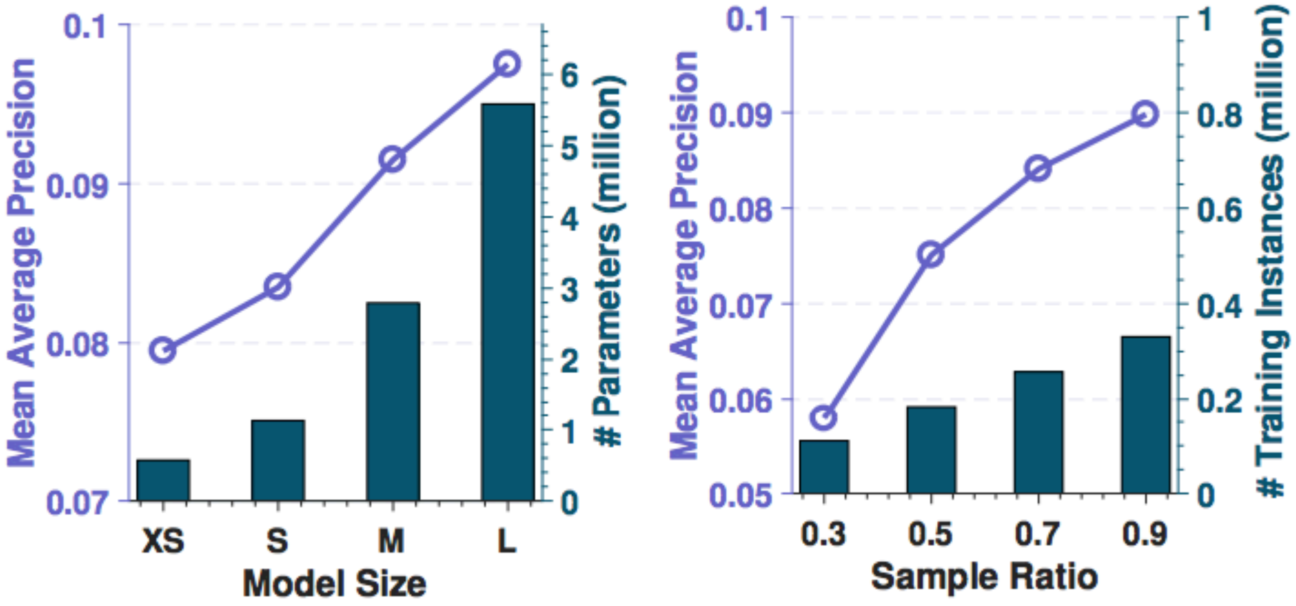
jiaxit@sfu.ca, wangk@cs.sfu.ca

Motivation



Model with
Ranking Distillation
Large-size Model
Cannot afford!
More Powerful
Medium-size Model
Too bad result!
Small-size Model
More Serving Cost

Motivation



(a) MAP *vs.* model size **(b) MAP *vs.* training instances**

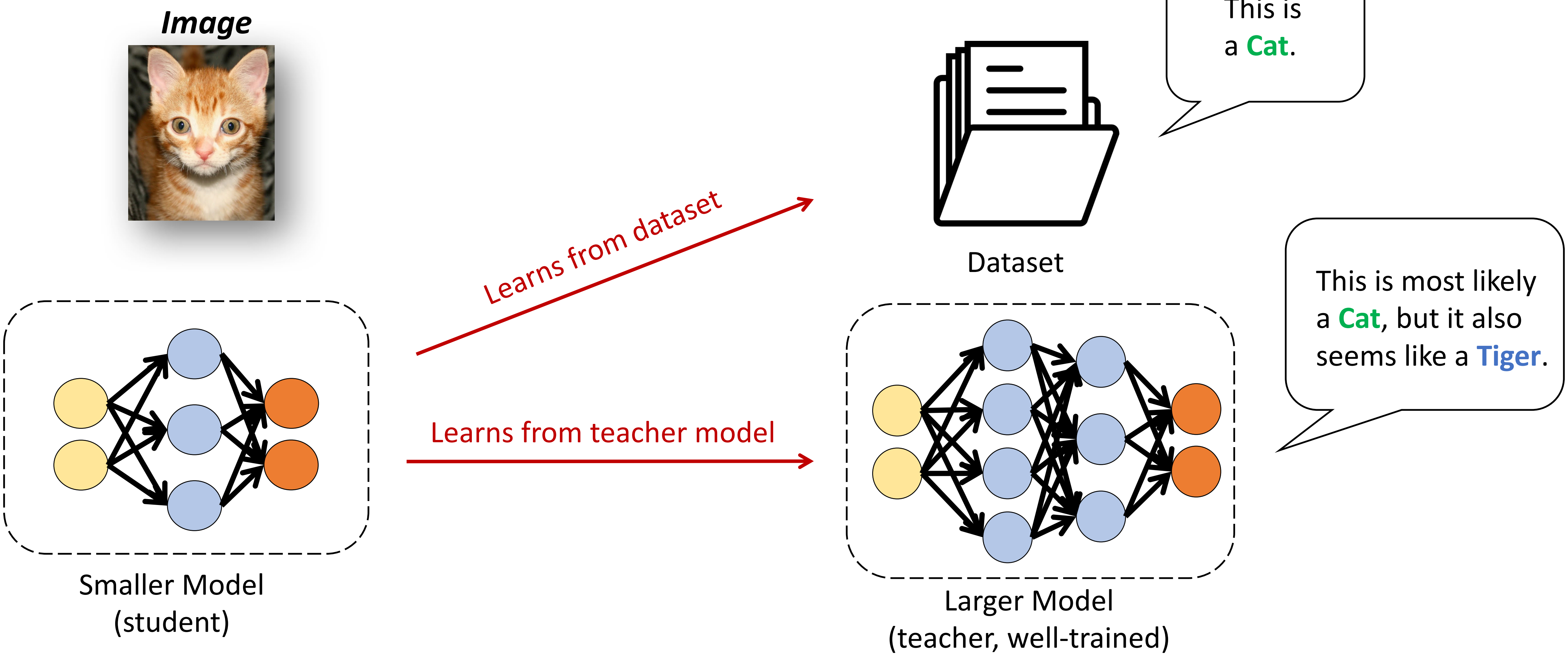
(a). Lager model size -> Better performance

Hard to serve the model for inefficiency!

(b). More training instances -> Better performance

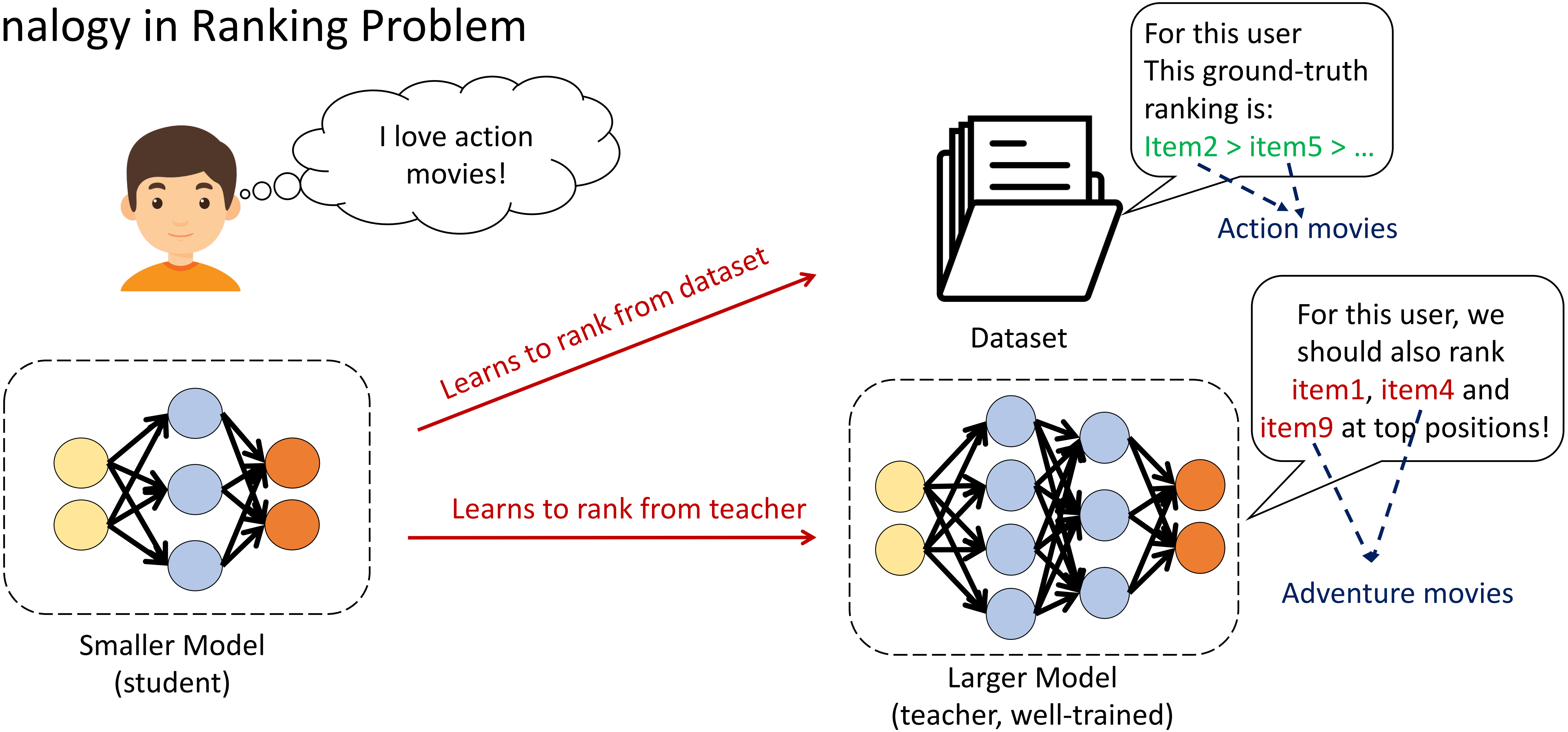
Not always available!

What is *Knowledge Distillation (KD)*?



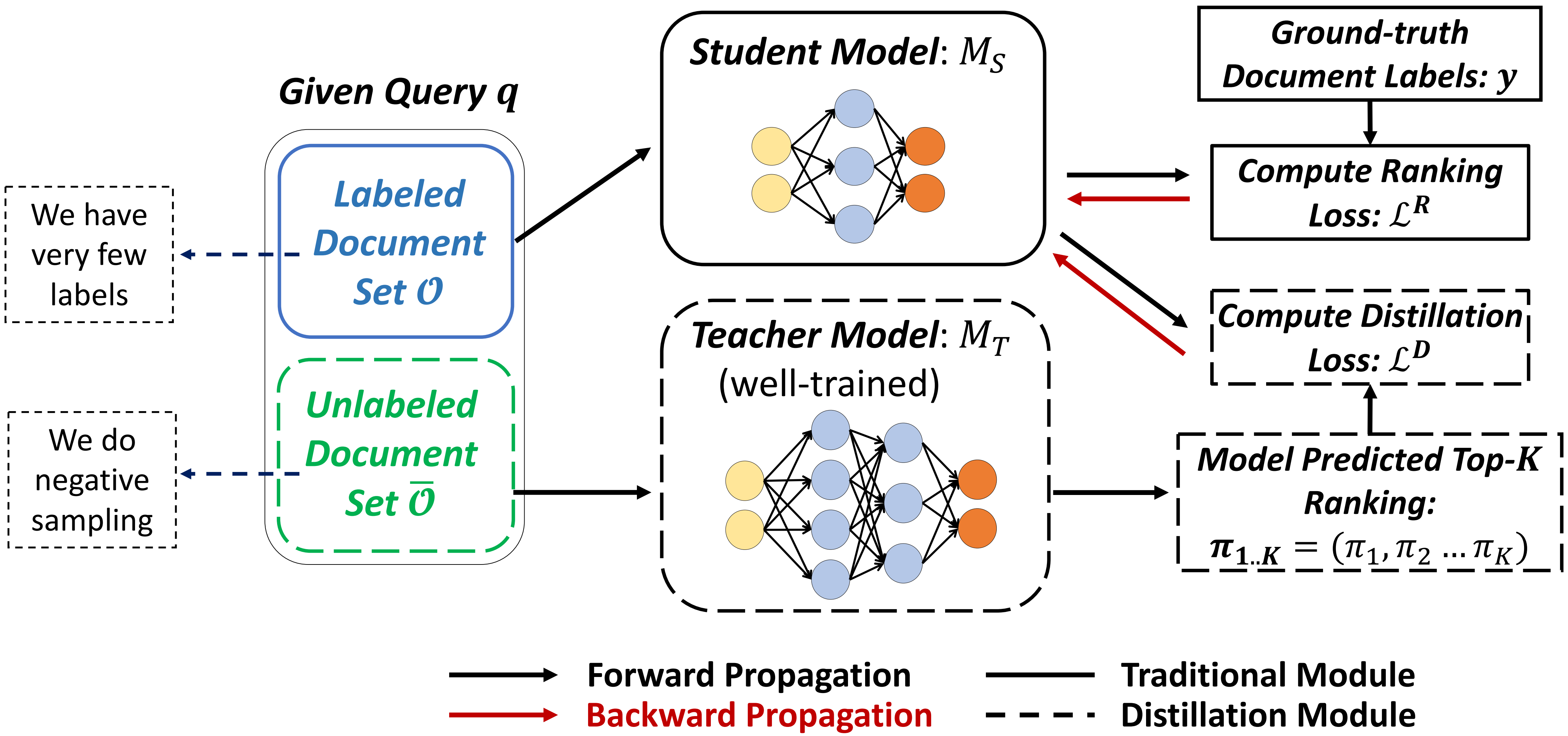
Makes the student model more robust, generalizable and thus perform better.

Analogy in Ranking Problem



Ranking Distillation

In recommendation: Query -> User Profile, Document -> Item



Ranking Distillation

Loss for a single query:

$$\mathcal{L}(\theta_S) = \underbrace{(1-\alpha)\mathcal{L}^R(\boldsymbol{y}, \hat{\boldsymbol{y}})}_{\textit{Ranking loss}} + \underbrace{\alpha\mathcal{L}^D(\boldsymbol{\pi}_{1..K}, \hat{\boldsymbol{y}})}_{\textit{Distillation loss}}.$$

Ranking loss:

$$\mathcal{L}^R(\boldsymbol{y}, \hat{\boldsymbol{y}}) = -\Big(\sum_{d \in \boldsymbol{y}_{d+}} \log(P(rel = 1|\hat{y}_d)) + \sum_{d \in \boldsymbol{y}_{d-}} \log(1 - P(rel = 1|\hat{y}_d))\Big),$$

point-wise

$$\mathcal{L}^R(\boldsymbol{y}, \hat{\boldsymbol{y}}) = -\sum_{d_i, d_j \in C} \log(P(d_i \succ d_j|\hat{y}_i, \hat{y}_j)),$$

pair-wise

Distillation loss:

$$\mathcal{L}^D(\boldsymbol{\pi}_{1..K}, \hat{\boldsymbol{y}}) = -\sum_{r=1}^{K} w_r \cdot \log(P(rel = 1|\hat{y}_{\pi_r})) = -\sum_{r=1}^{K} w_r \cdot \log(\sigma(\hat{y}_{\pi_r})),$$

weighted point-wise [1]

Pros: simple, only consider positive documents.
Cons: non-differentiable.

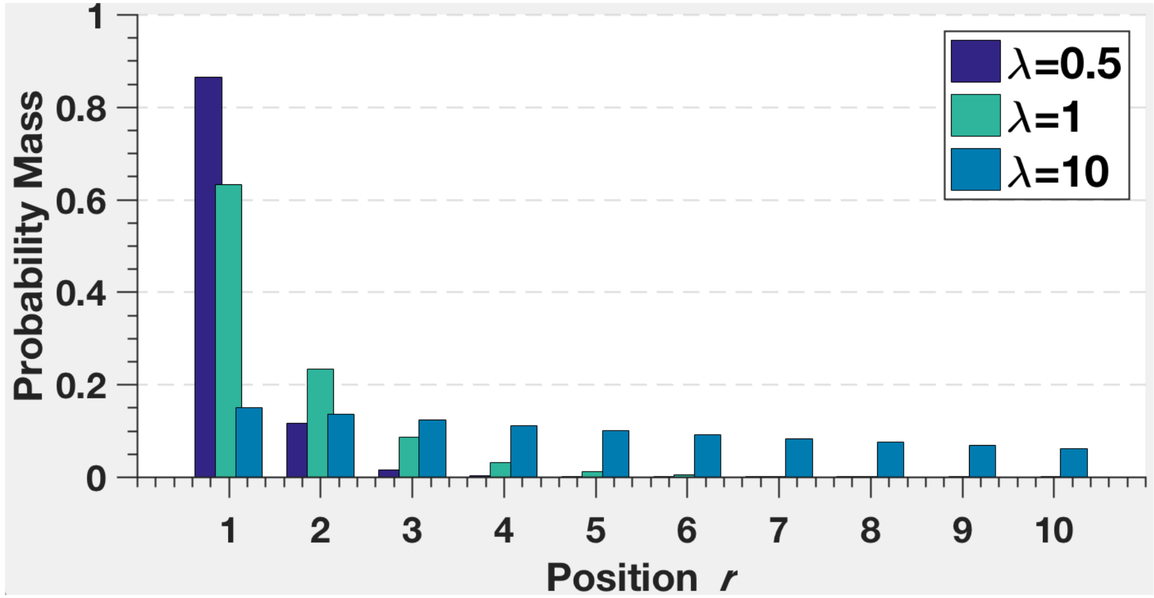
[1] Learning with noisy labels. Nagarajan Natarajan, Inderjit S Dhillon, Pradeep K Ravikumar, and Ambuj Tewari. NIPS 2013.

Weighting by position importance w^a

Assumption: The teacher predicted unlabeled documents at top positions are more correlated to the query and are more likely to the positive ground-truth documents.

An empirical weight following a exponentially decayed function[1]:

$$w_r^a \propto e^{-r/\lambda} \quad \text{and} \quad \lambda \in \mathbb{R}^+.$$



[1] Improving pairwise learning for item recommendation from implicit feedback. Stefen Rendle and Christoph Freudenthaler. 2014. WSDM 2014.

Weighting by ranking discrepancy w^b

Assumption: During the training process, we should have a dynamic weight to upweight the erroneous parts in distillation loss, and downweight the parts that already learned perfectly.

Example:

	Teacher's rank	Student's rank
π_1	1	1
π_2	2	5
π_3	3	156

$$\mathcal{L}^D = w_1^b * \log(\hat{y}_{\pi_1}) + w_2^b * \log(\hat{y}_{\pi_2}) + w_3^b * \log(\hat{y}_{\pi_3}) \dashrightarrow w_3^b >> w_2^b > w_1^b$$

How do we know student's rank without computing relevance scores for all items?

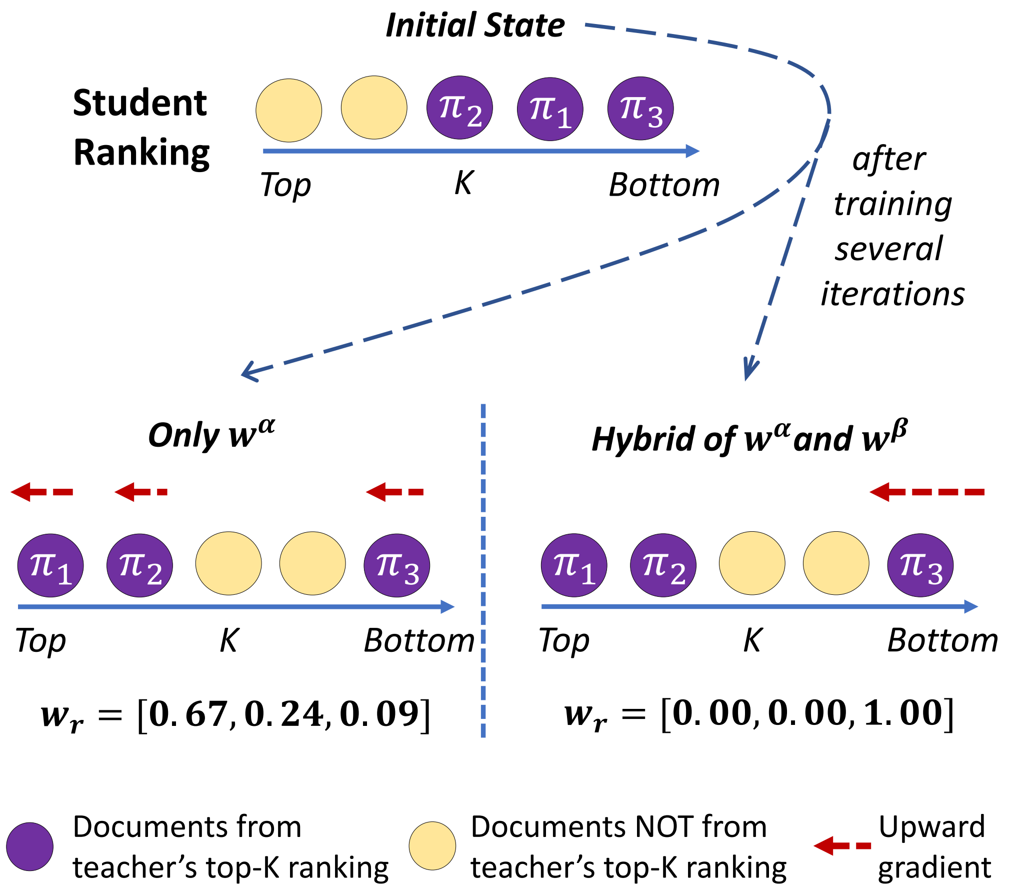
To get a documents approximated rank in a list of N documents, we can randomly sample ϵ << N documents in this list, and:

Estimated rank = $\lfloor n \times (N - 1)/\epsilon \rfloor + 1$

where n is the number of documents whose scores are greater than the given documents score.

$$w_3^b = \tanh(\mu \times (156 - 3))$$

Ranking Distillation by both weights



1. Choose a proper *K, e.g. K=3*
2. Using w^{a} during the first few iterations
3. Using hybrid weights then.

$$w_i = w_i^\alpha \cdot w_i^\beta$$

normalize (optional)

$$w_i \propto w_i^\alpha \cdot w_i^\beta$$

Experimental results

- Task: Sequential Recommendation,
  query -> user & her/his sequence
  document -> item
- Datasets: *Gowalla* & *Foursquare*
- Base Model: *Fossil*[1] & *Caser*[2]
- Baselines:
  - Model-T: Teacher model
  - Model-S: Student model
  - Model-RD: Student model trained with *ranking distillation*

Datasets	#users	#items	avg. actions per user
Gowalla	13.1k	14.0k	40.74
Foursquare	10.1k	23.4k	30.16

Datasets	Model	Time (CPU)	Time (GPU)	#Params	Ratio
Gowalla	*Fossil-T*	9.32s	3.72s	1.48M	100%
	Fossil-RD	4.99s	2.11s	0.64M	43.2%
	Caser-T	38.58s	4.52s	5.58M	100%
	Caser-RD	18.63s	2.99s	2.79M	50.0%
Foursquare	*Fossil-T*	6.35s	2.47s	1.01M	100%
	Fossil-RD	3.86s	2.01s	0.54M	53.5%
	Caser-T	23.89s	2.95s	4.06M	100%
	Caser-RD	11.65s	1.96s	1.64M	40.4%

[1] Fusing similarity models with markov chains for sparse sequential recommendation. Ruining He and Julian McAuley, 2017, ICDM
[2] Personalized Top-N Sequential Recommendation via Convolutional Sequence Embedding. Jiaxi Tang and Ke Wang, 2018, WSDM

Experimental results

Gowalla

Model	Prec@3	Prec@5	Prec@10	nDCG@3	nDCG@5	nDCG@10	MAP
Fossil-T	0.1299	0.1062	0.0791	0.1429	0.1270	0.1140	0.0866
Fossil-RD	0.1355	0.1096	0.0808	0.1490	0.1314	0.1172	0.0874
Fossil-S	0.1217	0.0995	0.0739	0.1335	0.1185	0.1060	0.0792
Caser-T	0.1408	0.1149	0.0856	0.1546	0.1376	0.1251	0.0958
Caser-RD	0.1458	0.1183	0.0878	0.1603	0.1423	0.1283	0.0969
Caser-S	0.1333	0.1094	0.0818	0.1456	0.1304	0.1188	0.0919

Foursquare

Model	Prec@3	Prec@5	Prec@10	nDCG@3	nDCG@5	nDCG@10	MAP
Fossil-T	0.0859	0.0630	0.0420	0.1182	0.1085	0.1011	0.0891
Fossil-RD	0.0877	0.0648	0.0430	0.1203	0.1102	0.1023	0.0901
Fossil-S	0.0766	0.0556	0.0355	0.1079	0.0985	0.0911	0.0780
Caser-T	0.0860	0.0650	0.0438	0.1182	0.1105	0.1041	0.0941
Caser-RD	0.0923	0.0671	0.0444	0.1261	0.1155	0.1076	0.0952
Caser-S	0.0830	0.0621	0.0413	0.1134	0.1051	0.0986	0.0874

Summarize:
Models trained with RD have similar performance with their teachers.

Tried but failed

1. Using the Top-*K* documents from teacher model as positive documents, and using the Bottom-*K* documents as negative documents. Then apply point-wise, pair-wise, list-wise distillation loss.
    - Possible reason: negative documents can be anywhere except Top-K, so we don’t need to care too much about them.

2. Using the pair-wise distillation loss within teacher’s Top-*K* documents, to make the partial order as correct as possible.
    - Possible reason: the gradient contains both up-wards gradient and down-ward gradient, which cause issues in trainability.

-- It is ‘good’ that teacher’s ranking and student’s ranking at top positions are not perfectly matched.
e.g. teacher’s ranking: d1 > d2 > d3 > …, student’s ranking: d2 > d3> d1 > …

Summarization

1. We use the Top-K unlabeled documents from teacher model's ranking as positive documents, and use a smaller student to learn to rank these documents at higher positions.

2. We propose two different weighting schemes to boost the training process.

3. The proposed 'Ranking Distillation' can be regarded as:
   - A knowledge transfering method
   - A semi-supervised method
   - A data-augmentation method

Q&A