A Generic Top-N Recommendation Framework For Trading-Off Accuracy, Novelty, and Coverage

Zainab Zolaktaf, Reza Babanezhad, Rachel Pottinger

Department of Computer Science

University of British Columbia

Motivation

- Top-N recommendation
  - Recommend to each user a set of N items from a large collection of items
  - Used in Netflix, Amazon, IMDB, etc.
- Problem
  - Tend to recommend things users are already aware of
  - E.g., Suggests “Star Wars: The Force Awakens” to users who have seen “Star Wars: Rogue One”

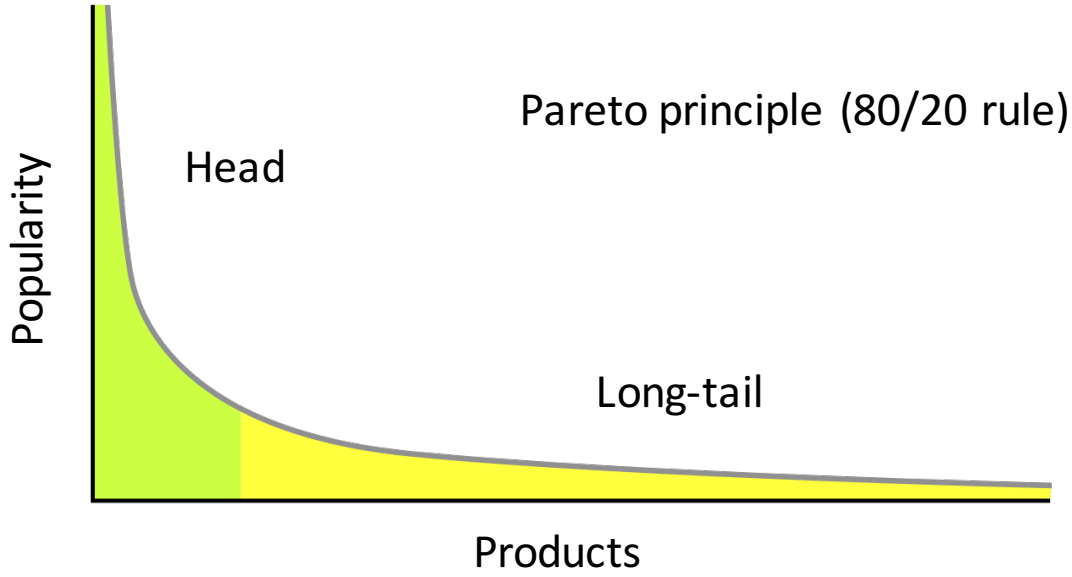
Motivation

- Many recommendation systems
  - Analyze interaction data
    - e.g., ratings on movies
  - Focus on accurately predicting user preference history
- Interaction data often suffers from popularity bias and sparsity
  - Have to recommend popular items to maintain performance accuracy
  - Rich get richer effect
- Accuracy alone is not leading to effective suggestions?

	5	4	5	2	3	5
	4	5	4			3
	1					
	5	4				
	1			2		

Recommendation system effectiveness?

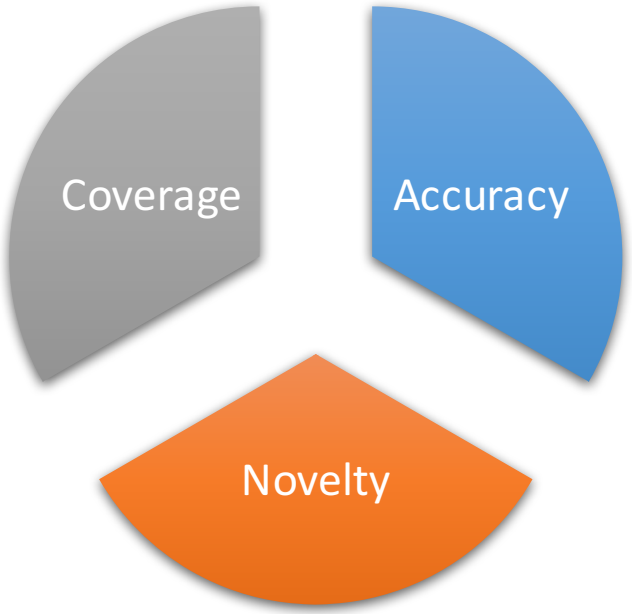
- Consumer
  - Accuracy
  - Novelty
  - …
- Providers of items
  - Keep consumers happy
  - Item-space coverage
    - Generates revenue
  - …
- Less focus on popular items



- Long-tail items
  - Generate the lower 20% of the observations
  - Empirically validated: Correspond to almost 85% of the items in several datasets

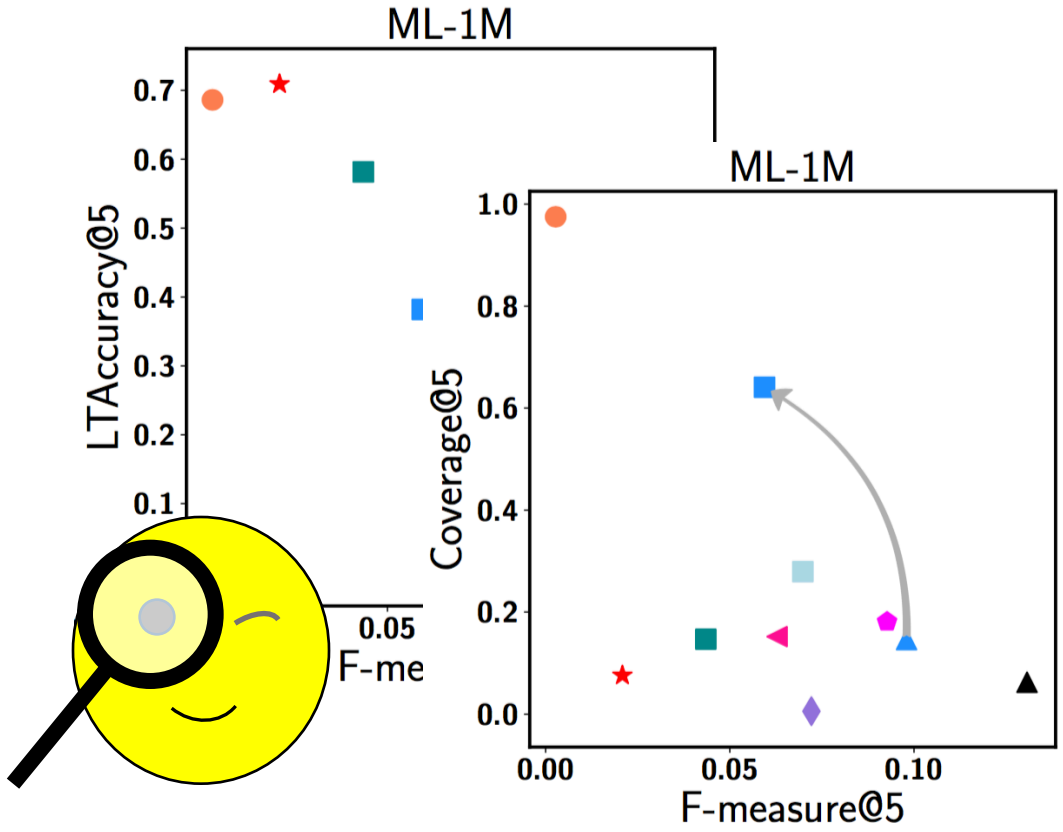
Challenges: Accuracy, novelty, and coverage trade-offs

- ✓Promoting long-tail item can increase novelty [Ste11]
  - Long-tail items are more likely to be unseen
- ✓Promoting long-tail items increases coverage [Ste11]
  - Generates revenue for providers of items
- ➢Long-tail promotion can reduce accuracy [Ste11]
  - ➢Not all users receptive of long-tail items



Recommendation system evaluation

- Need to assess multiple aspects
  - Accuracy, novelty, and coverage
  - No single measure that combines all aspects. Report trade-offs?
- Need to consider real-world settings
  - Datasets are sparse
  - Users provide little feedback
- Test ranking protocol [Ste13, CKT10]
  - Do not reward popularity-biased algorithms
  - Offline accuracy should be close to what user experiences in real-world

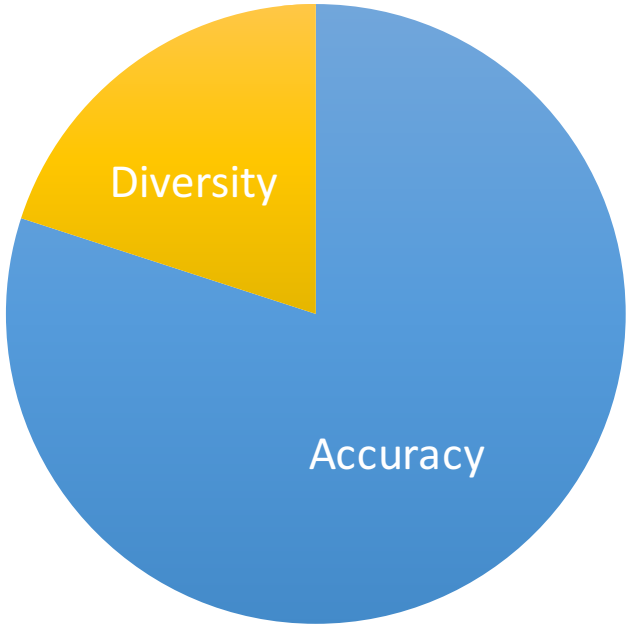


Contributions

- We study models for estimating user long-tail novelty from interaction data
- We introduce GANC, a generic re-ranking framework
- We conduct an extensive empirical study
  - Study performance from accuracy, coverage, and novelty perspectives
  - Consider the impact of dataset density
- Our results confirm performance of re-ranking models is impacted by the base recommender algorithm
  - In dense settings, using the same base recommender as existing models, we improve upon all metrics
  - In sparse settings, we plugin a more suitable base recommender
    - GANC is competitive with existing top-N recommendation models

Related work: Re-ranking frameworks

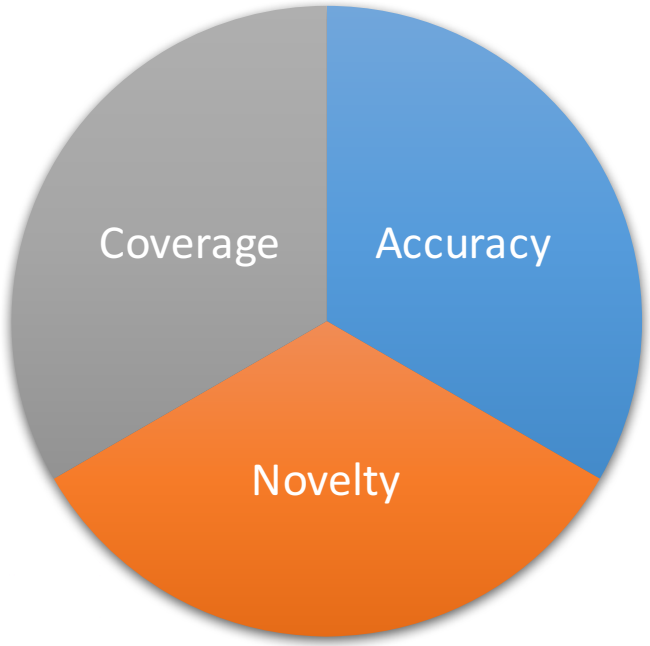
- Re-rank predictions of a base recommender to optimize for additional objectives [AK12, HCH14]
- Advantage
  - Computationally efficient
- Limitations
  1. Trade-off parameters are not personalized per user
     - But users have varying levels of preference for different objectives
  2. Often limited to a specific base recommender that may be sensitive to dataset density
     - Datasets are pruned and problem is examined in dense settings.



Solution overview: GANC

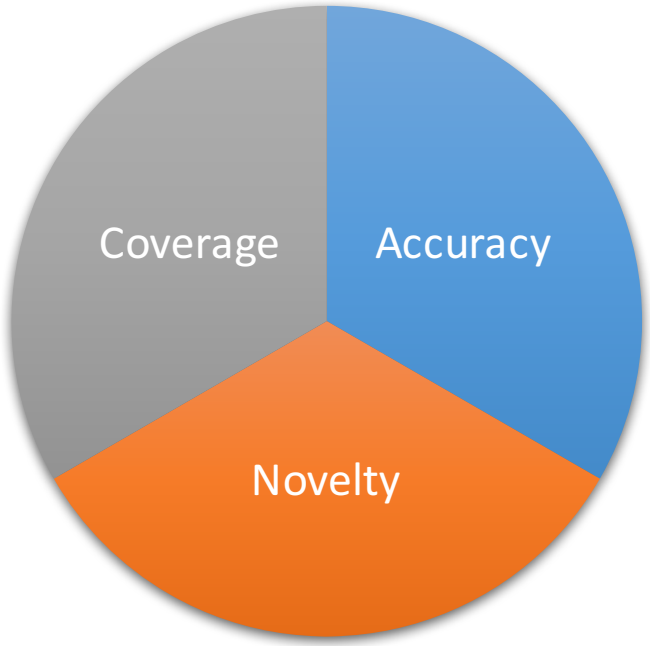
- A Generic top-N recommendation framework that provides customized balanced between Accuracy, Novelty, and Coverage
- Objective: Assign top-N sets to all users
- Find $\mathcal{P} = \{\mathcal{P}_u\}_{u=1}^{|\mathcal{U}|}$, the collection of top-N sets to maximize

$$
\begin{aligned}
v(\mathcal{P}) &= \sum_u v_u(\mathcal{P}_u) \\
&= \sum_u (1-\theta_u) a(\mathcal{P}_u) + \theta_u c(\mathcal{P}_u) \\
&= \sum_u (1-\theta_u) \sum_{i \in \mathcal{P}_u} a(i) + \theta_u \sum_{i \in \mathcal{P}_u} c(i)
\end{aligned}
$$

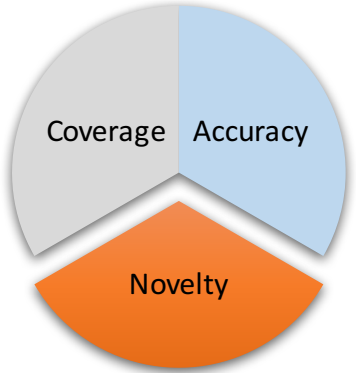


Solution overview: GANC

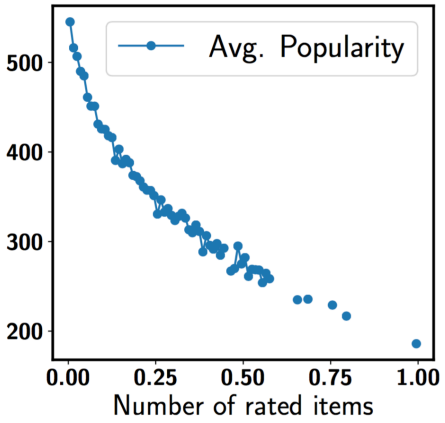
- Main features of our solution
  1. Directly infer user long-tail novelty preference θ_u from interaction data
     - Customize trade-off parameters per user
  2. Integrate θ_u into a generic re-ranking framework
     - θ_u independent of any base recommender
     - Plugin a suitable base recommender w.r.t. factors such as dataset density



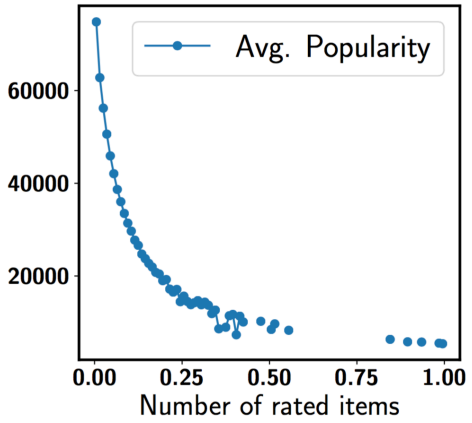
Long-Tail novelty preference model (Θ_u)



- Activity
  - Number observations in the train set (e.g., number of rated items)
  - Does not distinguish between long-tail and popular items



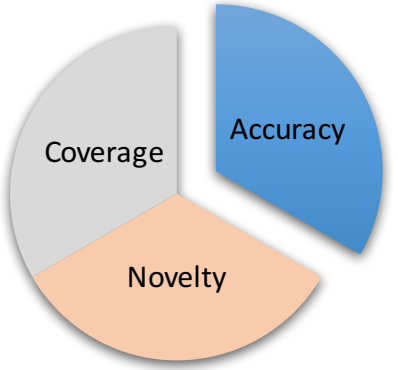
(a) ML-1M



(b) Netflix

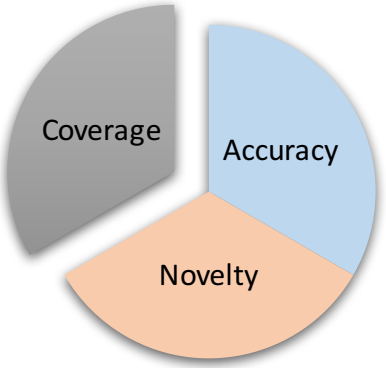
- Normalized long-tail measure
  - Ratio of long-tail items they have rated in train set
  - Does not consider whether user liked the item
- TFIDF-Measure
  - Incorporates rating and popularity of items
  - Does not consider view of other users
- Generalized measure
  - Optimization approach
  - Incorporates rating information, popularity of items, and view of other users

GANC: Accuracy recommender



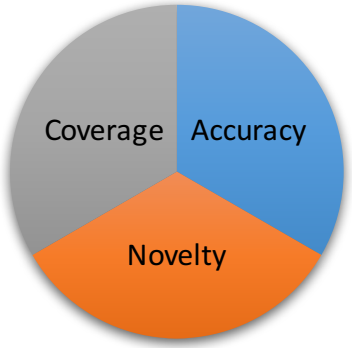
- Focuses on making accurate suggestions
- Used existing models from literature
  - Regularized SVD [KBV09]
    - Rating prediction model
  - PureSVD [CKT10]
    - Top-N recommendation algorithm
  - Most Popular [CKT10]
    - Suggests accurate, yet trivial top-N sets

GANC: Coverage recommender



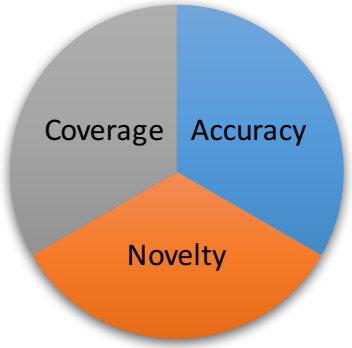
- Focus on increasing coverage
  - Random
  - Static
    - Consider how many times the item was rated in the past
      - Gain of recommending an item is proportionate to the inverse of its frequency in train set
    - GANC with Static coverage results in a modular set function optimization problem
  - Dynamic
    - Consider how many times item has been recommended so far
      - Gain of recommending an item is proportionate to the inverse of item recommendation frequency
    - GANC with Dynamic coverage is submodular across users
      - Submodular function maximization s.t a partition matroid constraint
      - Locally greedy is not scalable!

GANC with Dynamic coverage

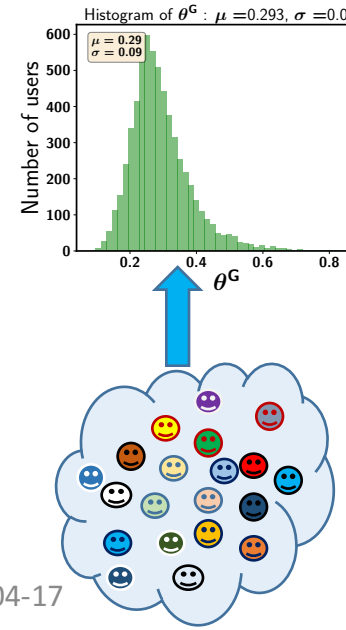


- Dynamic coverage leads to scalability problem
- Make parallel for the purpose of scalability
  - Design a sampling-based locally greedy algorithm

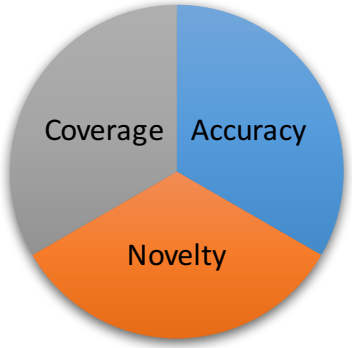
GANC with Dynamic coverage



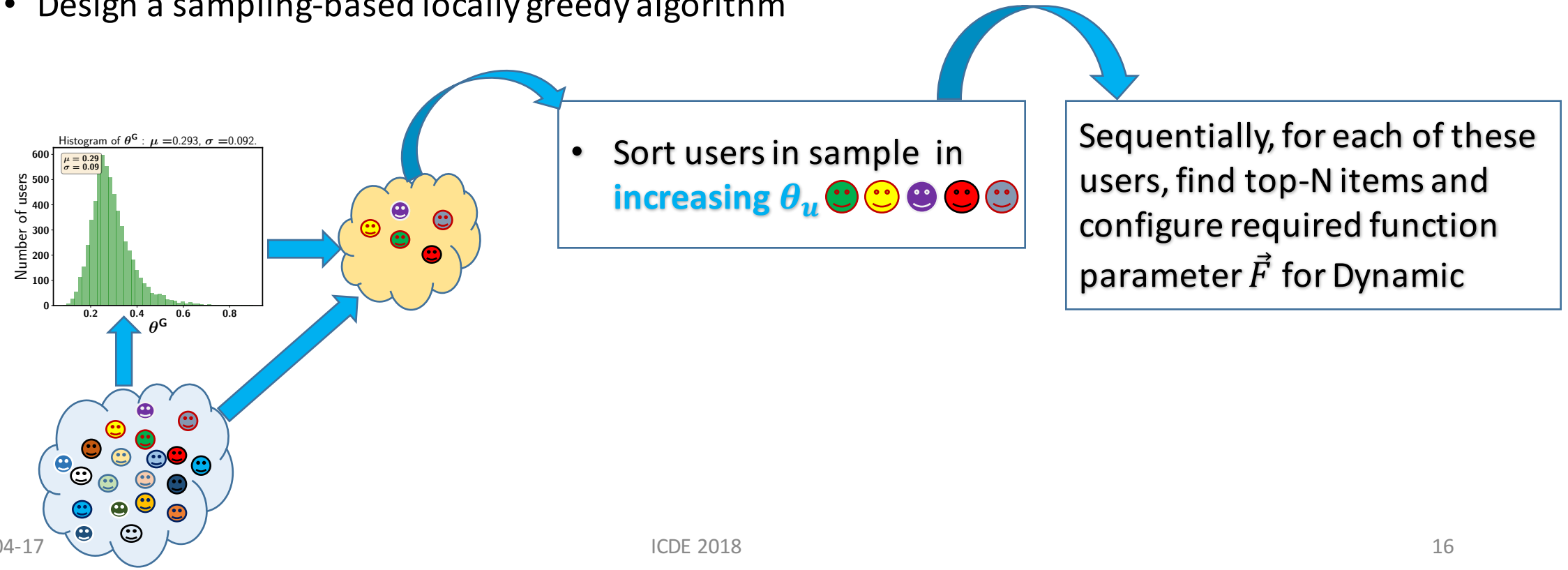
- Dynamic coverage leads to scalability problem
- Make parallel for the purpose of scalability
  - Design a sampling-based locally greedy algorithm



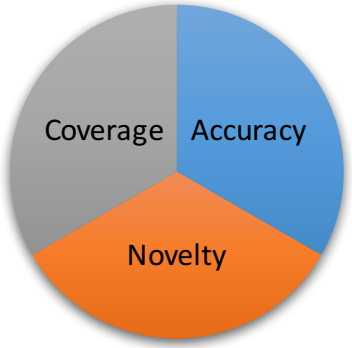
GANC with Dynamic coverage



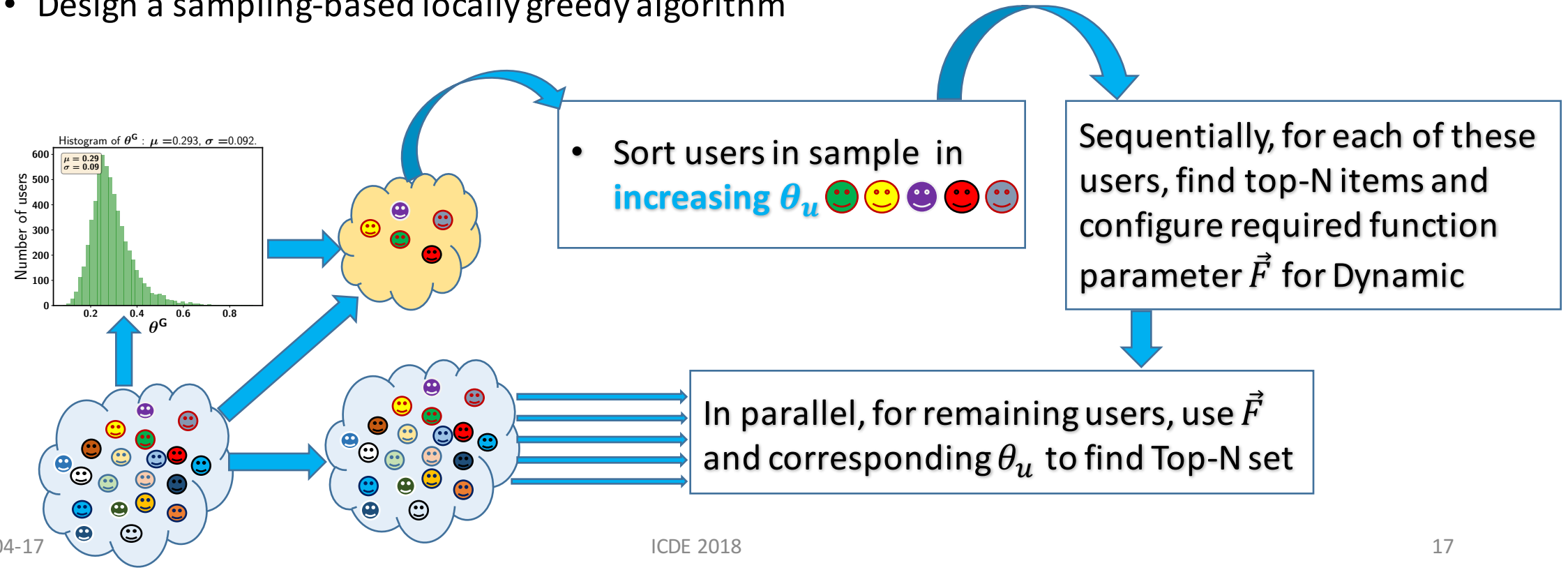
- Dynamic coverage leads to scalability problem
- Make parallel for the purpose of scalability
  - Design a sampling-based locally greedy algorithm



GANC with Dynamic coverage



- Dynamic coverage leads to scalability problem
- Make parallel for the purpose of scalability
  - Design a sampling-based locally greedy algorithm



Empirical evaluation

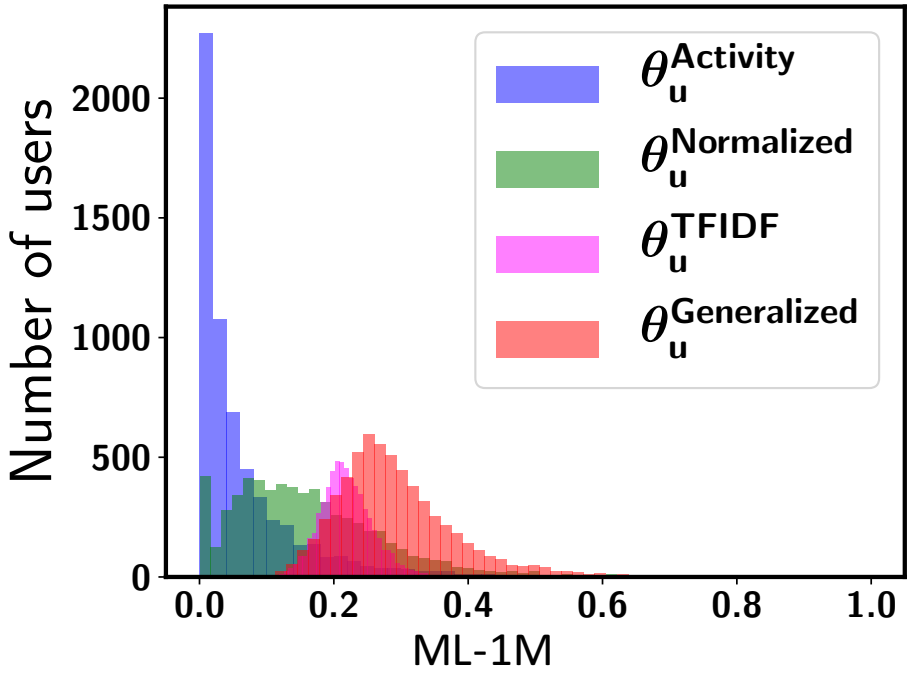
Dataset	#Ratings	#Users	#Items	Density	Long-Tail %
ML-100K	100K	943	1682	6.30	66.98
ML-1M	1M	6,040	3,706	4.47	67.58
ML-10M	10M	69,878	10,677	1.34	84.31
MT-200k	172,506	7,969	13,864	0.16	86.84
Netflix	98,754,394	459,497	17,770	1.21	88.27

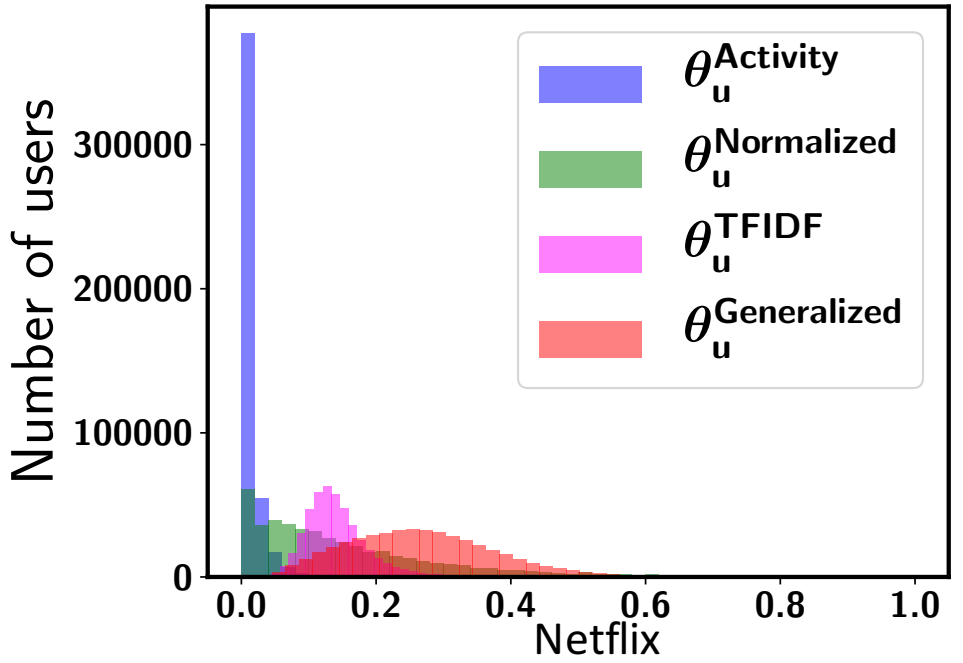
- ML = MovieLens, MT = MovieTweetings
- ML, MT, and Netflix are common recommender datasets
- Datasets have varying level of density
- Long-tail items correspond to approximately 85% in three datasets

Empirical evaluation

- Performance metrics
  - Local ranking accuracy metrics
    - Precision, Recall, F-measure
  - Long-tail promotion metrics
    - LTAccuracy (emphasizes novelty and coverage), Stratified Recall (emphasizes novelty and accuracy)
  - Coverage metrics
    - Coverage, Gini
- Test ranking protocol [Ste13, CKT10]
  - "All unrated items test ranking protocol"
    - Generate the top-N set of each user, by ranking all items that do not appear in the train set of that user

Histograms of long-tail novelty preference estimates





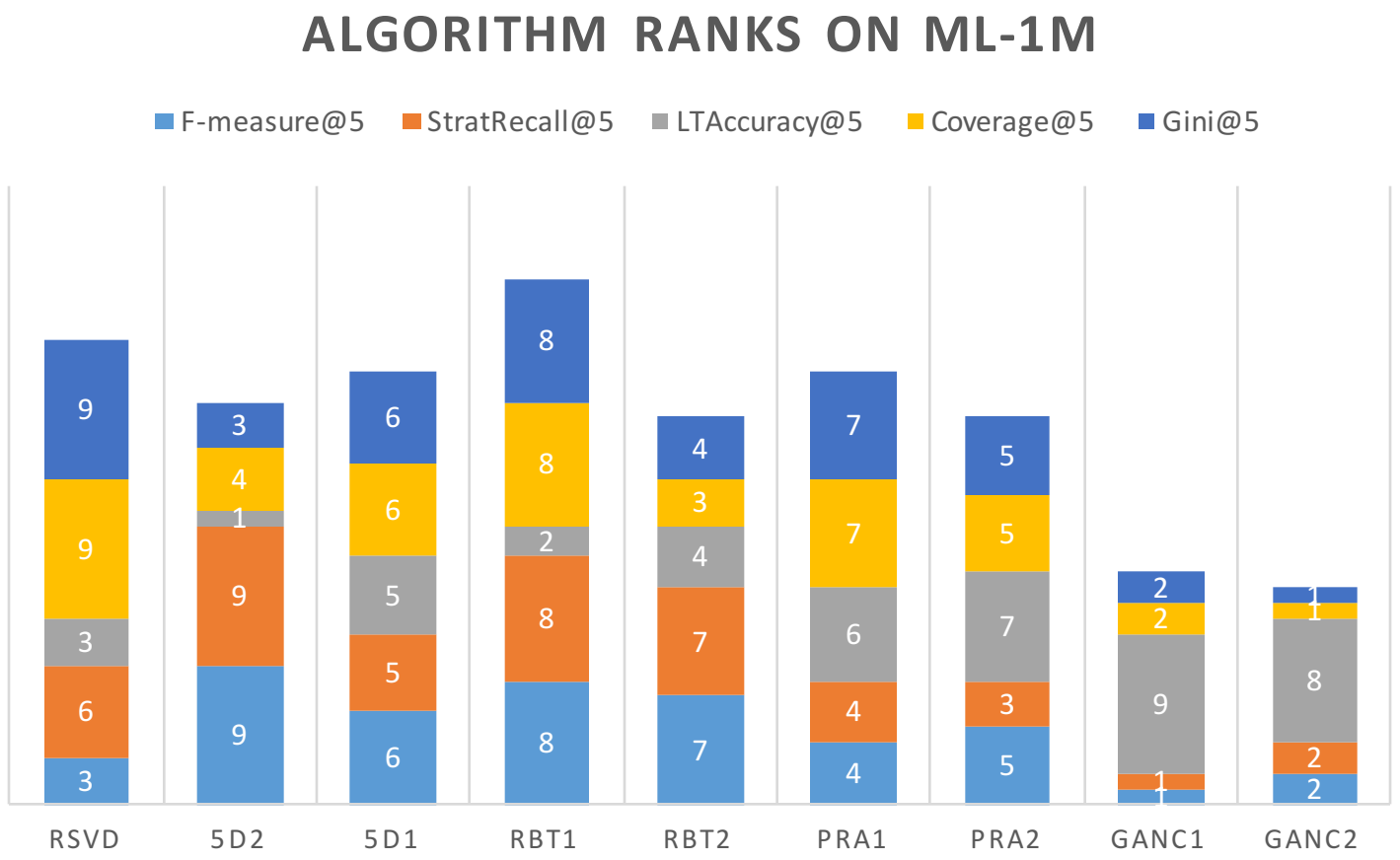
- $\theta_u^{Activity}$ is skewed to the right
  - Majority of users provide little feedback and have small activity
- $\theta_u^{Generalized}$ increases variance and identifies more categories of users

Comparison with re-rankings models for rating-prediction

- Rating prediction base accuracy recommender
  - Regularized SVD (RSVD)
- Baselines
  - RSVD
  - Resource Allocation (5D)
  - Ranking-based Techniques (RBT)
  - Personalized Ranking Adaptation (PRA)
- Report results for two variants of each algorithm

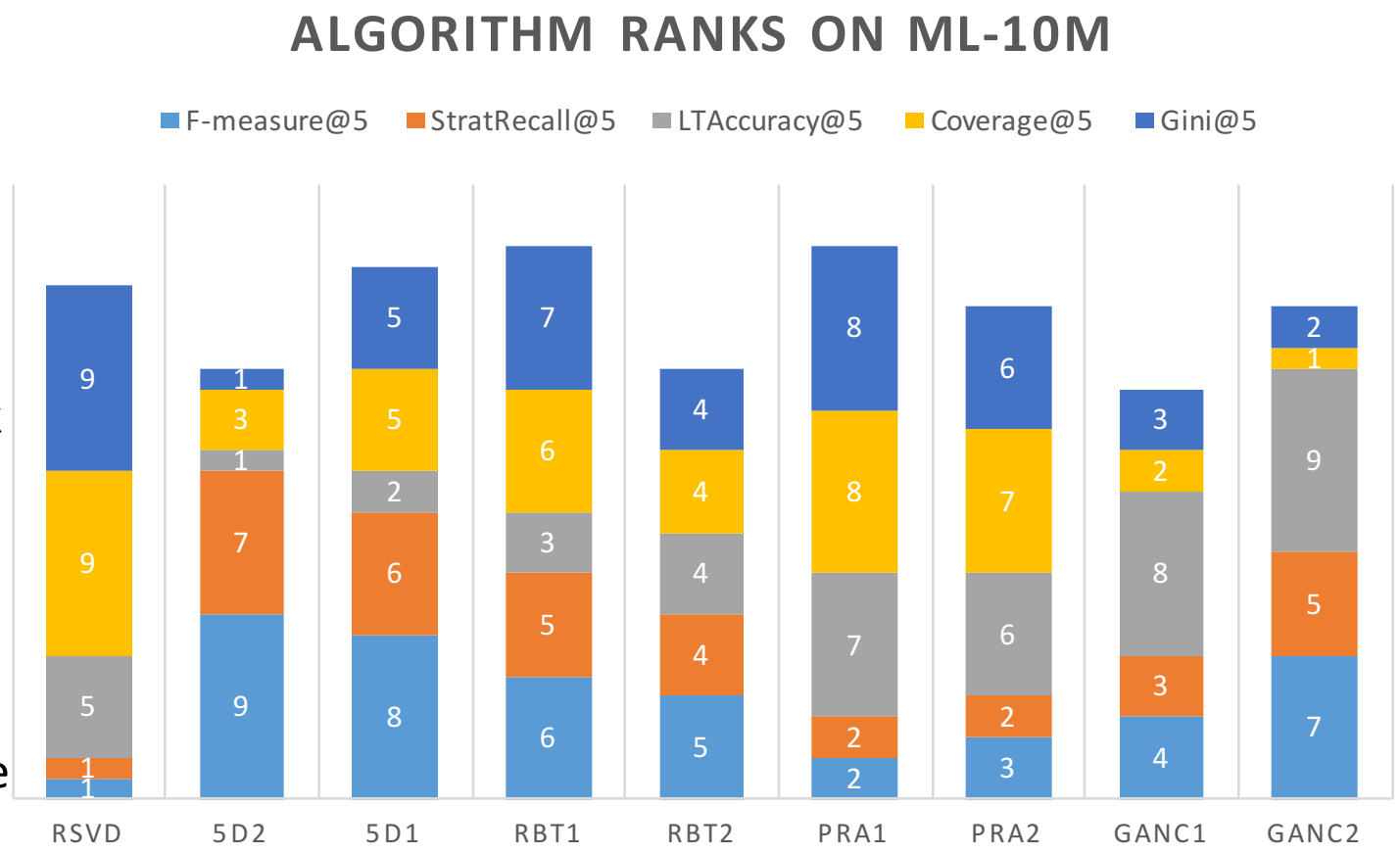
Comparison with re-rankings models for rating-prediction

- Dense dataset
  - ML-1M
- RSVD is base accuracy recommender
- Lower height is better
  - Corresponds to better rank
- GANC
  - Outperforms RSVD in 4 metrics, including accuracy
  - Obtains best average performance



Comparison with re-rankings models for rating-prediction

- Sparse dataset
  - ML-10M
- RSVD is base accuracy recommender
- Lower height is better
  - Corresponds to better rank
- Performance of all models degrades
  - RSVD has low accuracy to begin with.
  - But for sparse datasets, we can plugin a different accuracy recommender …

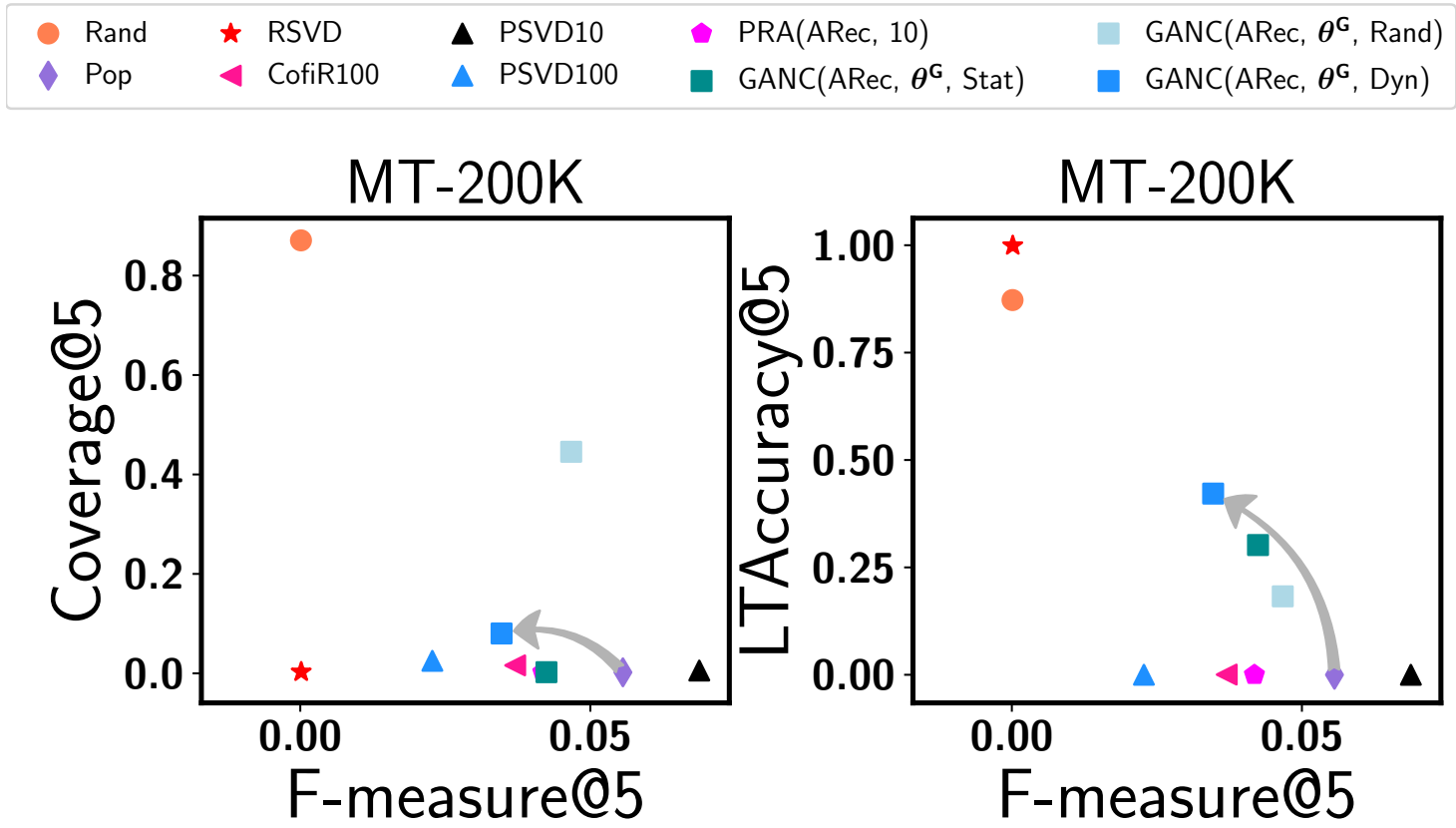


Comparison with top-N item recommendation models

- Base accuracy recommender
  - Most popular (Pop)
  - No longer in the domain of rating prediction
  - Modify baselines
- Top-N recommendation baselines
  - Pop
  - Random (Rand)
  - Regularized SVD (RSVD)
  - CofiRank (CofiR100)
  - PureSVD with 10 factors (PSVD10)
  - PureSVD with 100 factors (PSVD100)
  - Personalized Ranking adaptation (PRA)

Comparison with top-N item recommendation models

- GANC
  - Plugin the non-personalized algorithm Pop as accuracy recommender
  - Competitive with PSVD100 and more sophisticated algorithms like CofiR100



Selected related work

- Accuracy-focused models
  - KBV09- Koren, Yehuda, Robert Bell, and Chris Volinsky. "Matrix factorization techniques for recommender systems." *Computer*42.8 (2009).
  - WKL+08- Weimer, Markus, et al. "Cofi rank-maximum margin matrix factorization for collaborative ranking." *Advances in neural information processing systems*. 2008.
- Re-ranking frameworks
  - AK12- Adomavicius, Gediminas, and YoungOk Kwon. "Improving aggregate recommendation diversity using ranking-based techniques." *IEEE Transactions on Knowledge and Data Engineering* 24.5 (2012): 896-911.
  - HCH14- Ho, Yu-Chieh, Yi-Ting Chiang, and Jane Yung-Jen Hsu. "Who likes it more?: mining worth-recommending items from long tails by modeling relative preference." *Proceedings of the 7th ACM international conference on Web search and data mining*. ACM, 2014.
- Evaluation of top-N recommendation
  - CKT10- Cremonesi, Paolo, Yehuda Koren, and Roberto Turrin. "Performance of recommender algorithms on top-n recommendation tasks." *Proceedings of the fourth ACM conference on Recommender systems*. ACM, 2010.
  - Ste11- Steck, Harald. "Item popularity and recommendation accuracy." *Proceedings of the fifth ACM conference on Recommender systems*. ACM, 2011.
  - Ste13- Steck, Harald. "Evaluation of recommendations: rating-prediction and ranking." *Proceedings of the 7th ACM conference on Recommender systems*. ACM, 2013.

Contributions

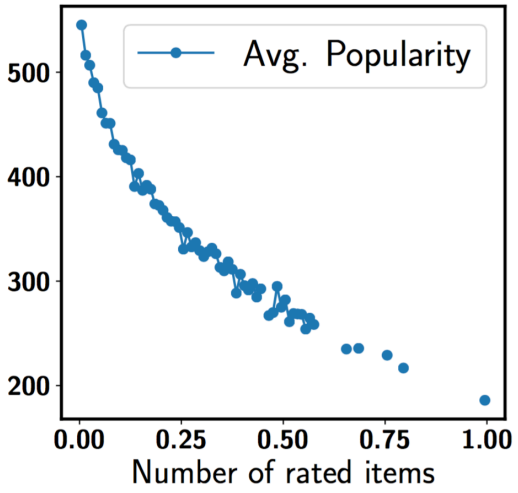
- We study models for estimating user long-tail novelty from interaction data
- We introduce GANC, a generic re-ranking framework
- We conduct an extensive empirical study
  - Study performance from accuracy, coverage, and novelty perspectives
  - Consider the impact of dataset density
- Our results confirm performance of re-ranking models is impacted by the base recommender algorithm
  - In dense settings, using the same base recommender as existing models, we improve upon all aspects
  - In sparse settings, we plugin a more suitable base recommender
    - GANC is competitive with existing top-N recommendation models

$$
\begin{aligned}
v(\mathcal{P}) &= \sum_u v_u(\mathcal{P}_u) \\
&= \sum_u (1-\theta_u) a(\mathcal{P}_u) + \theta_u c(\mathcal{P}_u) \\
&= \sum_u (1-\theta_u) \sum_{i \in \mathcal{P}_u} a(i) + \theta_u \sum_{i \in \mathcal{P}_u} c(i) \\
&= \sum_u (1-\theta_u) \sum_{i \in \mathcal{P}_u} \hat{r}_{ui} + \theta_u \sum_{i \in \mathcal{P}_u} \frac{1}{\sqrt{1+f_i^{\mathcal{P}}}}
\end{aligned}
$$

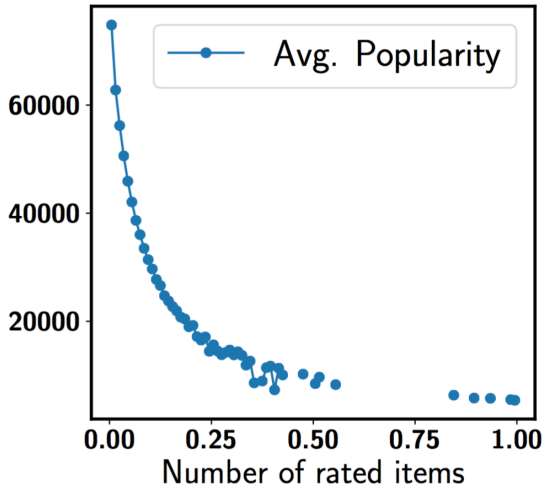
Submodularity and Monotonicity. Let $\mathcal{I}$ denote a ground set of items. Given a set function $f : 2^{\mathcal{I}} \rightarrow \mathbb{R}$, $\delta(i|\mathcal{A}) := f(\mathcal{A} \cup \{i\}) - f(\mathcal{A})$ is the marginal gain of f at $\mathcal{A}$ with regard to item i. Furthermore, f is submodular if and only if $\delta(i|\mathcal{A}) \geq \delta(i|\mathcal{B}), \forall \mathcal{A} \subseteq \mathcal{B} \subseteq \mathcal{I}, \forall i \in \mathcal{I} \backslash \mathcal{B}$. It is modular if $f(\mathcal{A} \cup i) = f(\mathcal{A}) + f(i)$, $\forall \mathcal{A} \subset \mathcal{I}, i \in \mathcal{I} \setminus \mathcal{A}$. In addition, f is monotone increasing if $f(\mathcal{A}) \leq f(\mathcal{B}), \forall \mathcal{A} \subseteq \mathcal{B} \subseteq \mathcal{I}$. Equivalently, a function is monotone increasing if and only if $\forall \mathcal{A} \subseteq \mathcal{I}$ and $i \in \mathcal{I}$, $\delta(i|\mathcal{A}) \geq 0$ [57]. Submodular functions have the following concave composition property:

Simple Long-Tail novelty preference models

- User Activity $\theta_u^A = |\mathcal{I}_u^{\mathcal{R}}|$



(a) ML-1M



(b) Netflix

- Normalized long-tail measure

$$\theta_u^N = \frac{|\mathcal{I}_u^{\mathcal{R}} \cap \mathcal{L}|}{|\mathcal{I}_u^{\mathcal{R}}|}$$

- TFIDF-Measure

$$\theta_u^T = \frac{1}{|\mathcal{I}_u^{\mathcal{R}}|} \sum_{i \in \mathcal{I}_u^{\mathcal{R}}} r_{ui} \log \left(\frac{|\mathcal{U}|}{|\mathcal{U}_i^{\mathcal{R}}|} \right)$$

Learning Long-Tail novelty preference

- Rewriting TFIDF-Measure

$$\theta_u^T = \frac{1}{|\mathcal{I}_u^{\mathcal{R}}|} \sum_{i \in \mathcal{I}_u^{\mathcal{R}}} \theta_{ui} = \frac{\sum_{i \in \mathcal{I}_u} w_i \theta_{ui}}{\sum_{i \in \mathcal{I}_u} w_i}$$

- Where $w_i = 1$ for all items.

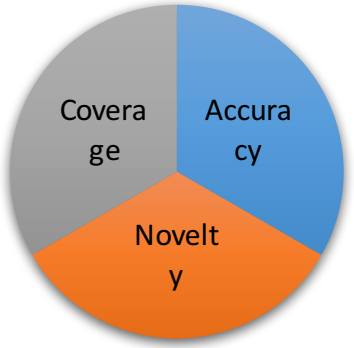
Learning Long-Tail novelty preference

$$O(\mathbf{w}, \boldsymbol{\theta}^G) = \sum_{i \in \mathcal{I}^{\mathcal{R}}} w_i \Big[\sum_{u \in \mathcal{U}_i^{\mathcal{R}}} 1 - \left(\theta_{ui} - \theta_u^G\right)^2 \Big] = \sum_{i \in \mathcal{I}^{\mathcal{R}}} w_i \epsilon_i$$

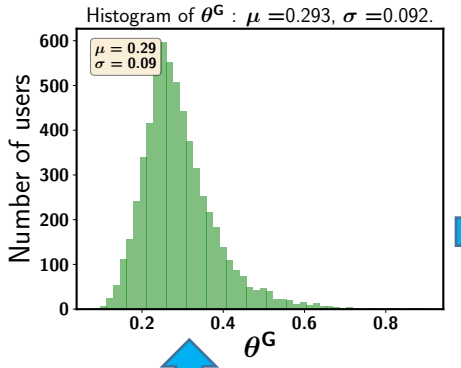
$$\min_{\mathbf{w}} \max_{\boldsymbol{\theta}^G} \qquad O(\mathbf{w}, \boldsymbol{\theta}^G) - \lambda_1 \sum_{i \in \mathcal{I}^{\mathcal{R}}} \log w_i$$

- Solve iteratively

GANC with Dynamic coverage



- Make algorithm parallel for the purpose of scalability



- Initialize $\vec{F}$: the item recommendation frequency counter
- Sort users in sample in **increasing θ_u**

Sequentially, for each user in

1. Find Top-N Items
2. Update $\vec{F}$
3. Store $\vec{F}$ and corresponding θ_u

In parallel, for remaining users

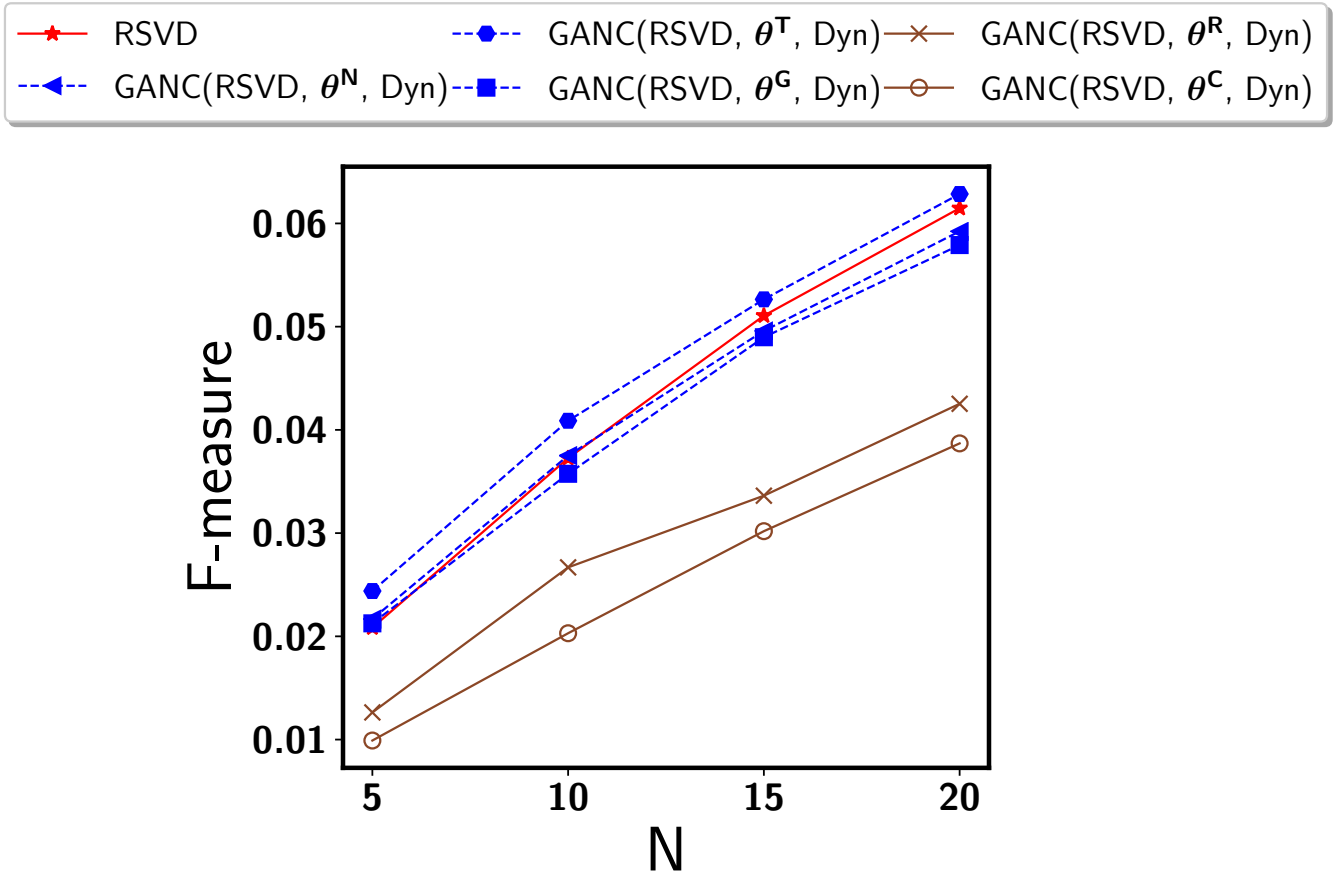
1. Use $\vec{F}$ and corresponding θ_u to find Top-N set

Empirical Evaluation: Performance metrics

Local Ranking Accuracy Metrics	$\text{Precision@N} = \frac{1}{N\lvert\mathcal{U}\rvert}\sum_{u\in\mathcal{U}}\lvert\mathcal{I}_u^{\mathcal{T}+}\cap\mathcal{P}_u\rvert$
	$\text{Recall@N} = \frac{1}{\lvert\mathcal{U}\rvert}\sum_{u\in\mathcal{U}}\frac{\lvert\mathcal{I}_u^{\mathcal{T}+}\cap\mathcal{P}_u\rvert}{\lvert\mathcal{I}_u^{\mathcal{T}+}\rvert}$
	$\text{F-measure@N} = \frac{\text{Precision@N.Recall@N}}{\text{Precision@N+Recall@N}}$
Longtail Promotion	$\text{LTAccuracy@N} = \frac{1}{N\lvert\mathcal{U}\rvert}\sum_{u\in\mathcal{U}}\lvert\mathcal{L}\cap\mathcal{P}_u\rvert$
	$\text{StratRecall@N} = \frac{\sum_{u\in\mathcal{U}}\sum_{i\in\mathcal{I}_u^{\mathcal{T}+}\cap\mathcal{P}_u}\left(\frac{1}{f_i^{\mathcal{R}}}\right)^{\beta}}{\sum_{u\in\mathcal{U}}\sum_{i\in\mathcal{I}_u^{\mathcal{T}+}}\left(\frac{1}{f_i^{\mathcal{R}}}\right)^{\beta}}$
Coverage Metrics	$\text{Coverage@N} = \frac{\lvert\cup_{u\in\mathcal{U}}\mathcal{P}_u\rvert}{\lvert\mathcal{I}\rvert}$
	$\text{Gini@N} = \frac{1}{\lvert\mathcal{I}\rvert}(\lvert\mathcal{I}\rvert + 1 - 2\frac{\sum_{j=1}^{\lvert\mathcal{I}\rvert}(\lvert\mathcal{I}\rvert+1-j)f[j]}{\sum_{j=1}^{\lvert\mathcal{I}\rvert}f[j]})$

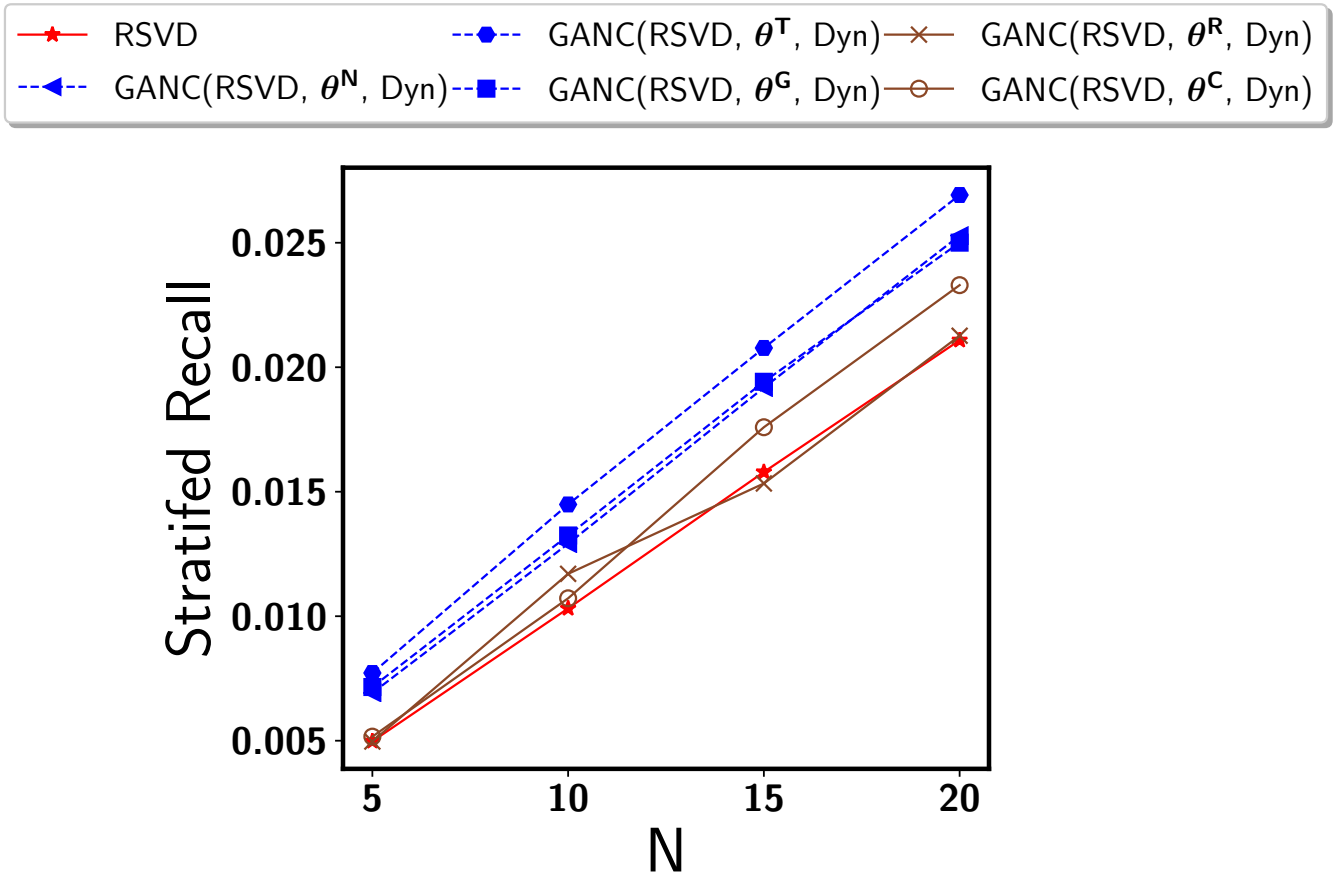
Performance of GANC with Dynamic coverage

- Using the popular RSVD model as base accuracy recommender
- Comparing against random and constant coverage
- F-measure increases



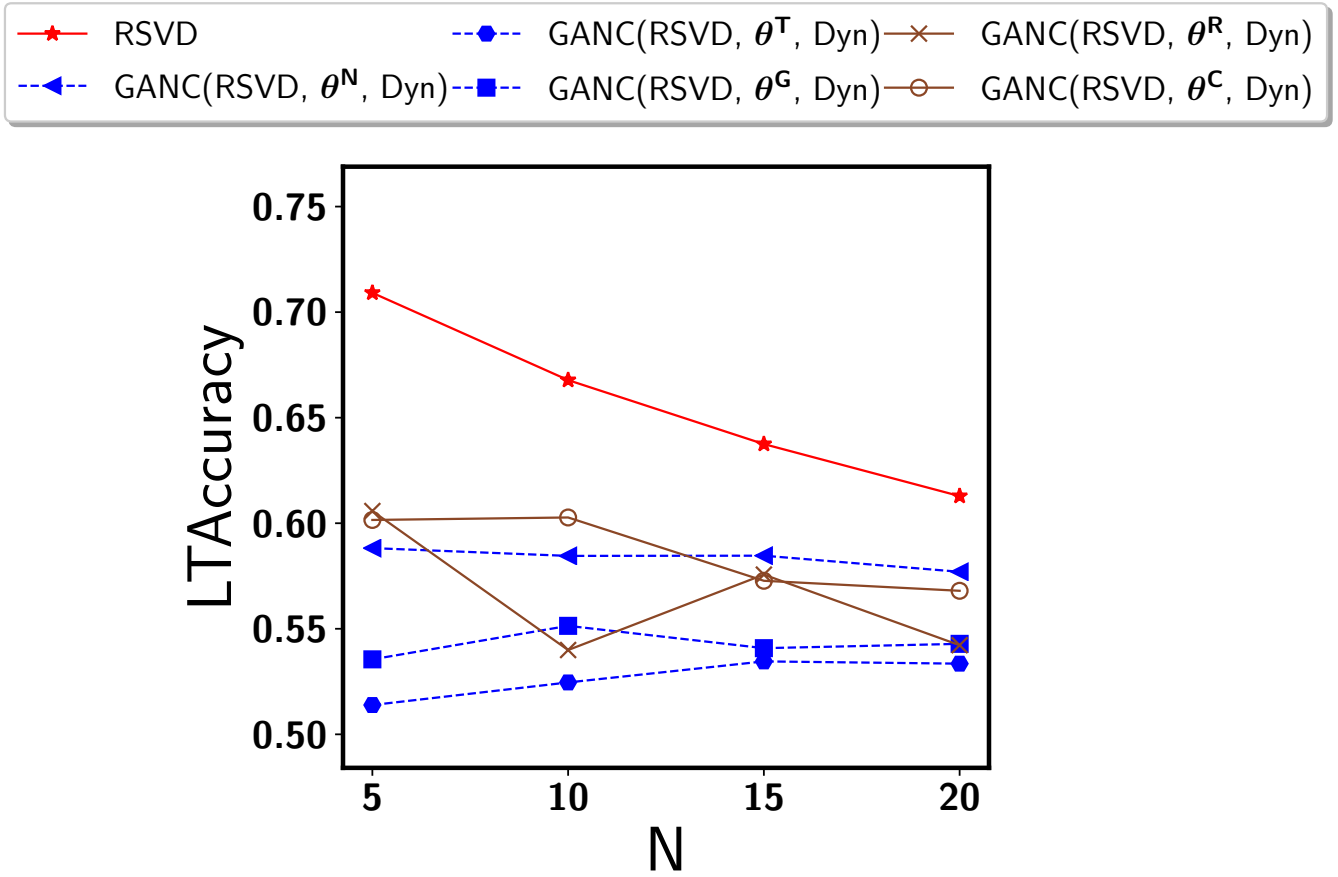
Performance of GANC with Dynamic coverage

- Stratified recall emphasizes novelty and accuracy
- Stratified recall improves across N



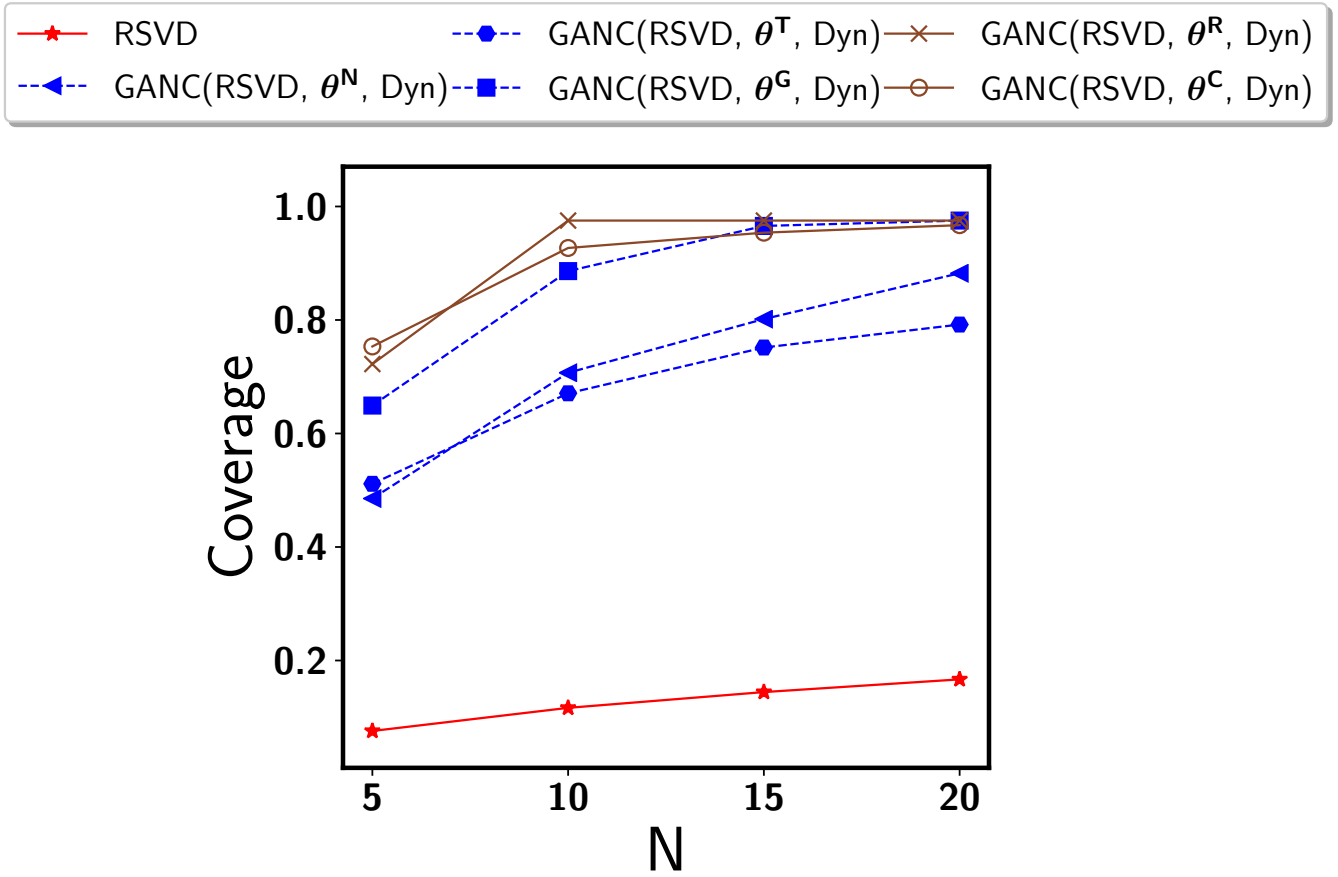
Performance of GANC with Dynamic coverage

- LTAccuracy emphasizes novelty and coverage
- RSVD recommends the same long-tail items to all users



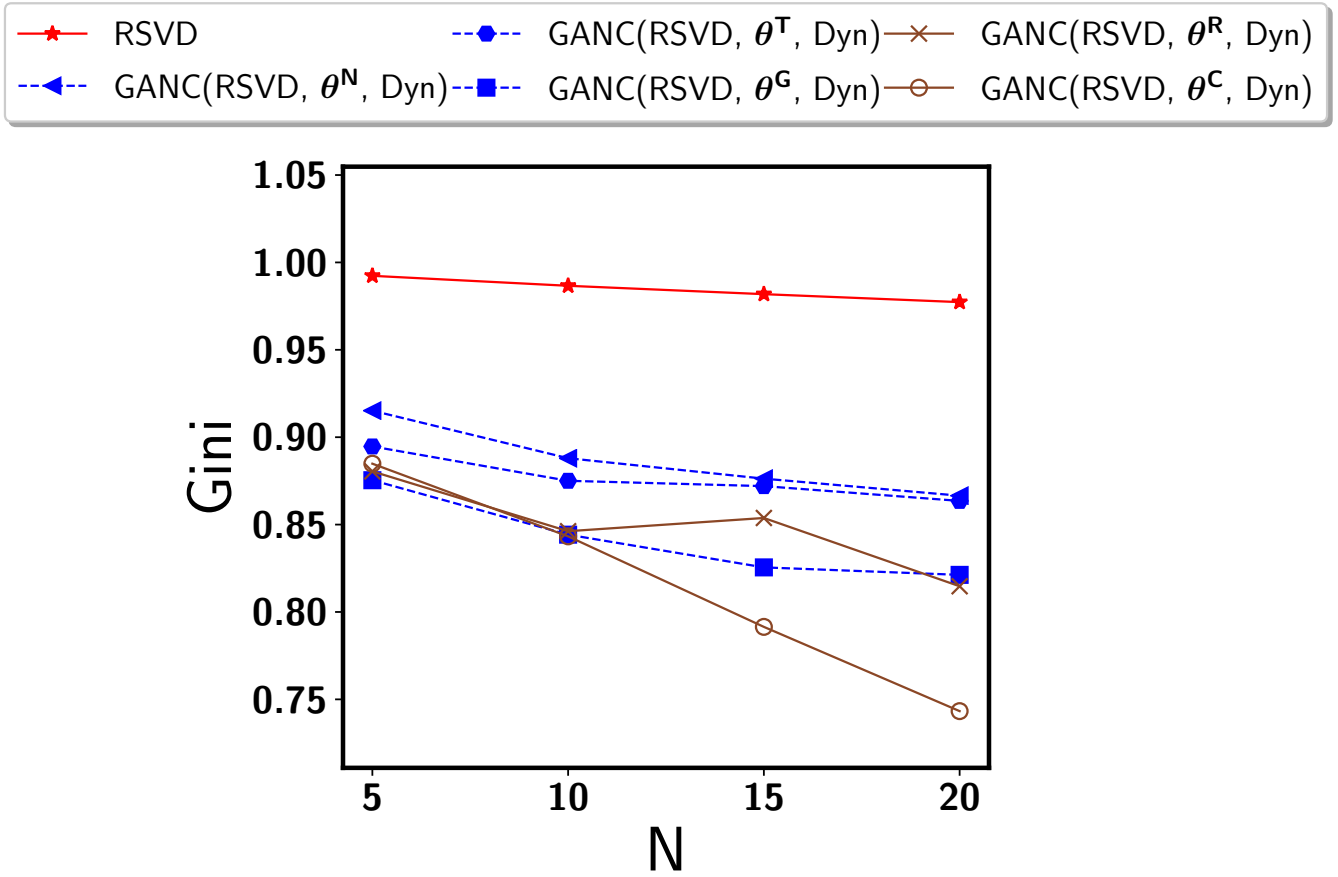
Performance of GANC with Dynamic coverage

- Coverage improves
- This result is the same for all base recommenders and all datasets

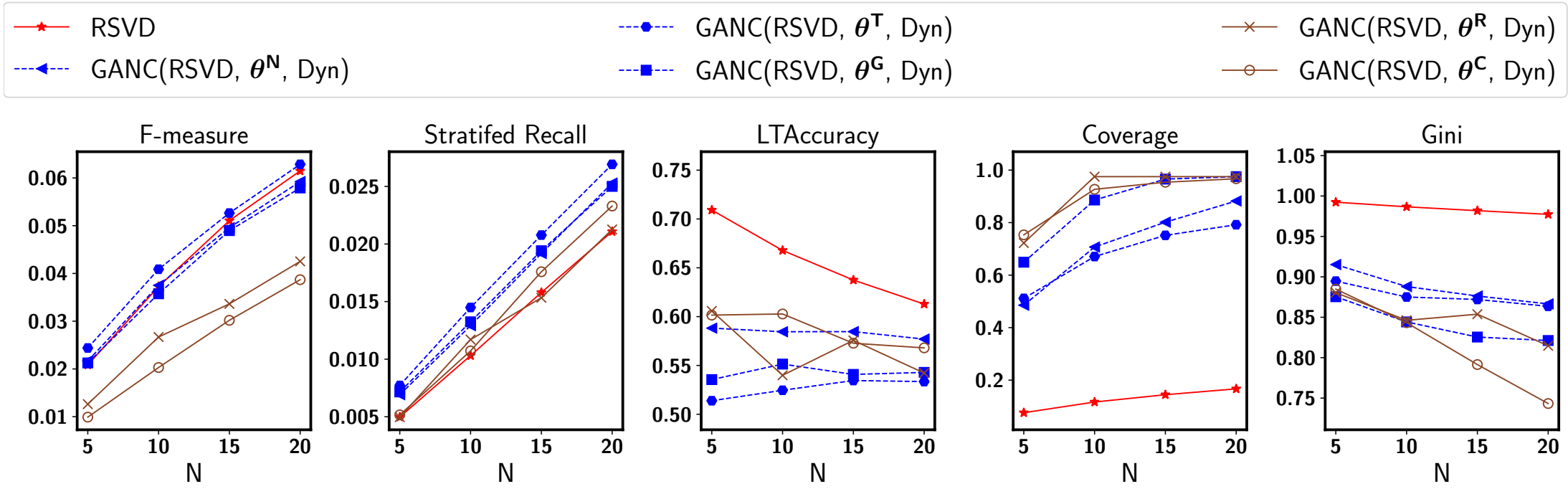


Performance of GANC with Dynamic coverage

- Lower gini shows balance in recommendations
- Random long-tail novelty preference and constant obtain best performance

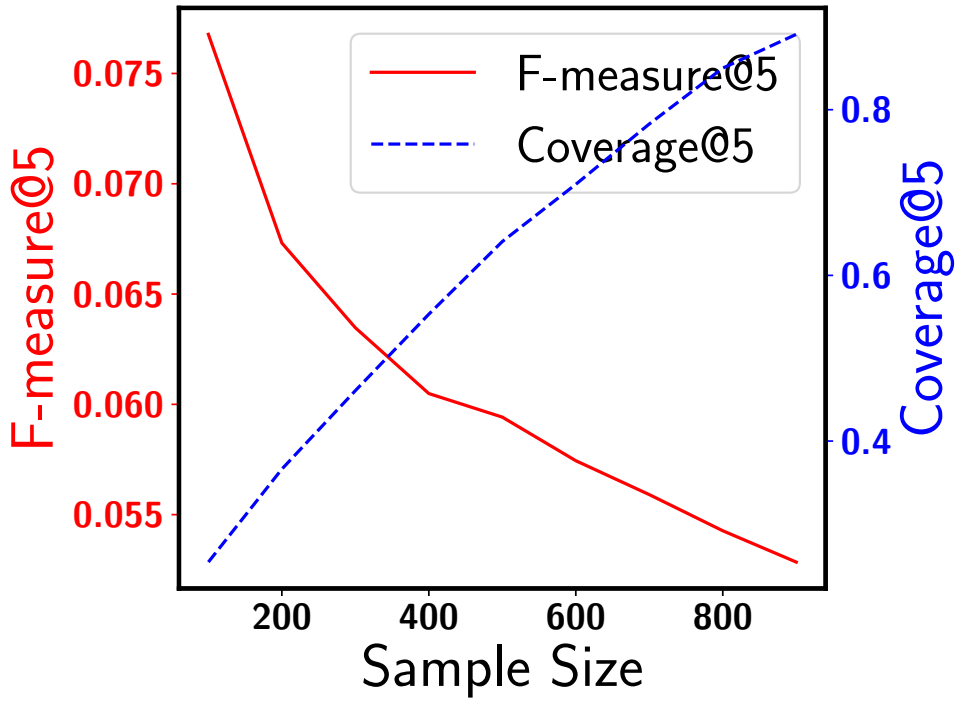


Performance of GANC with Dynamic coverage

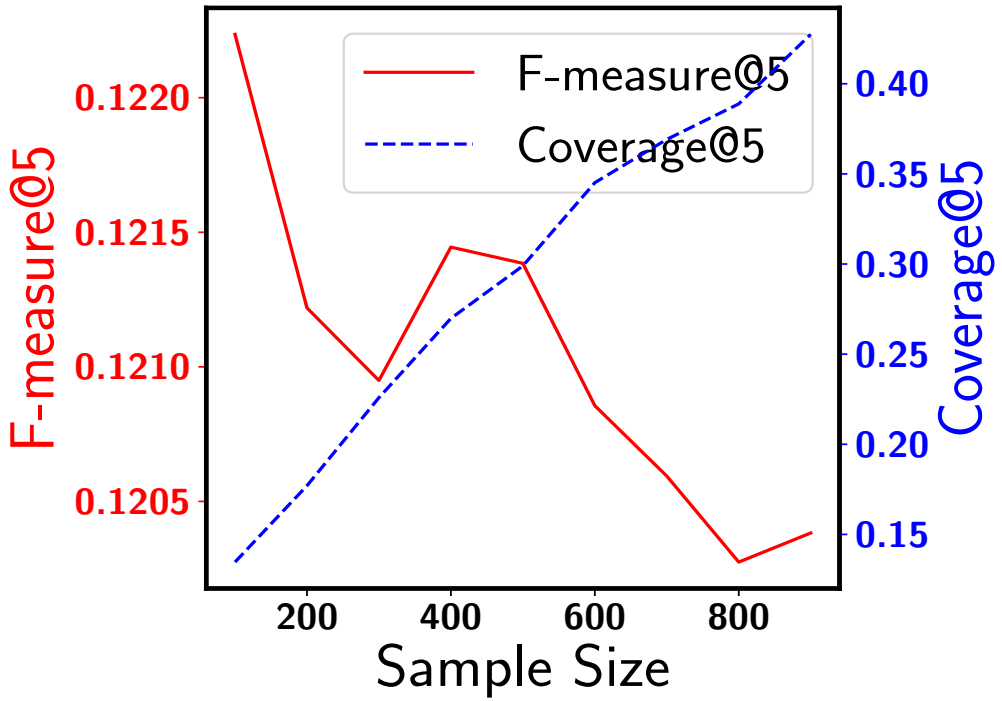


- Considering all metrics at the same time

Performance of GANC with Dynamic coverage

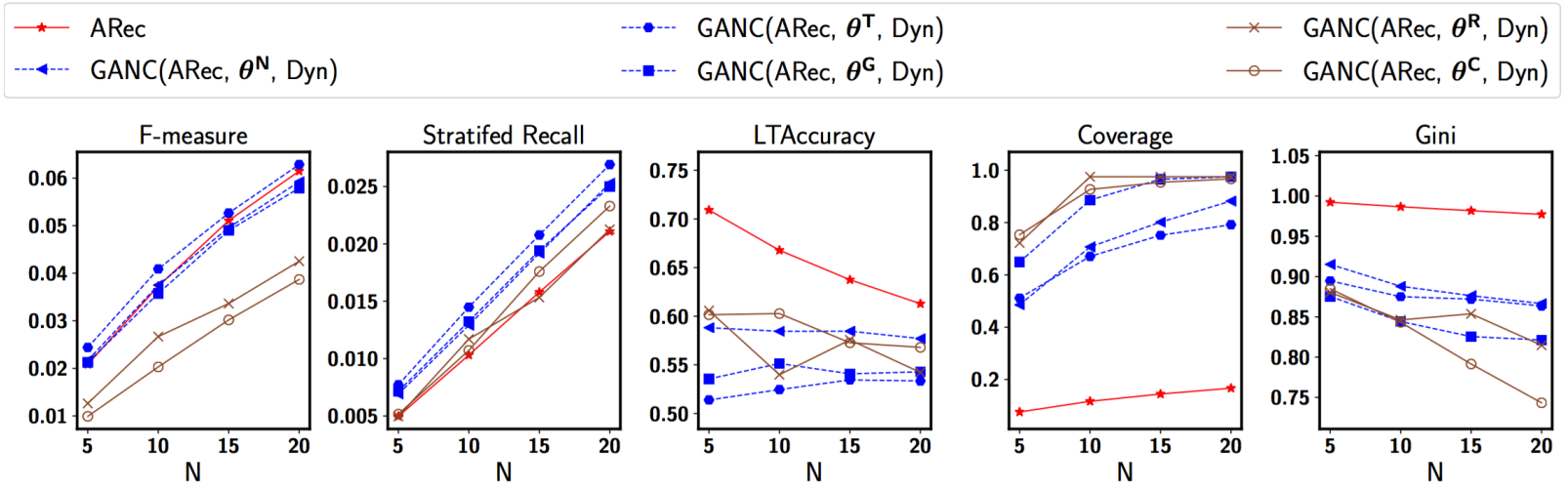


- Accuracy recommender is PSVD100
- Increasing sample size decreases accuracy but increases coverage

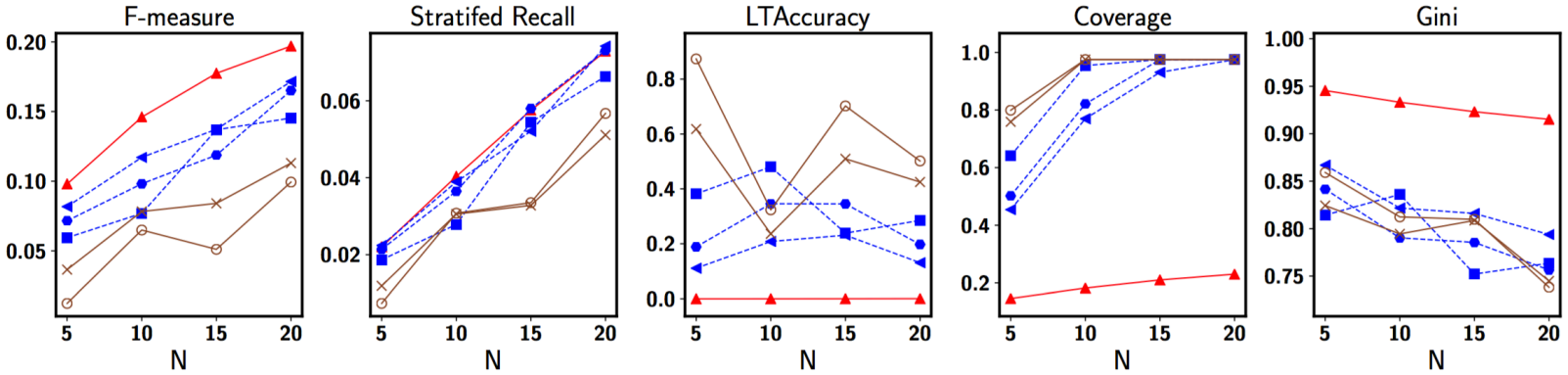


- Accuracy recommender is PSVD10
- The bump is due to the base recommender PSVD10

Performance of GANC with Dynamic coverage



(a) Accuracy recommender (ARec) is RSVD

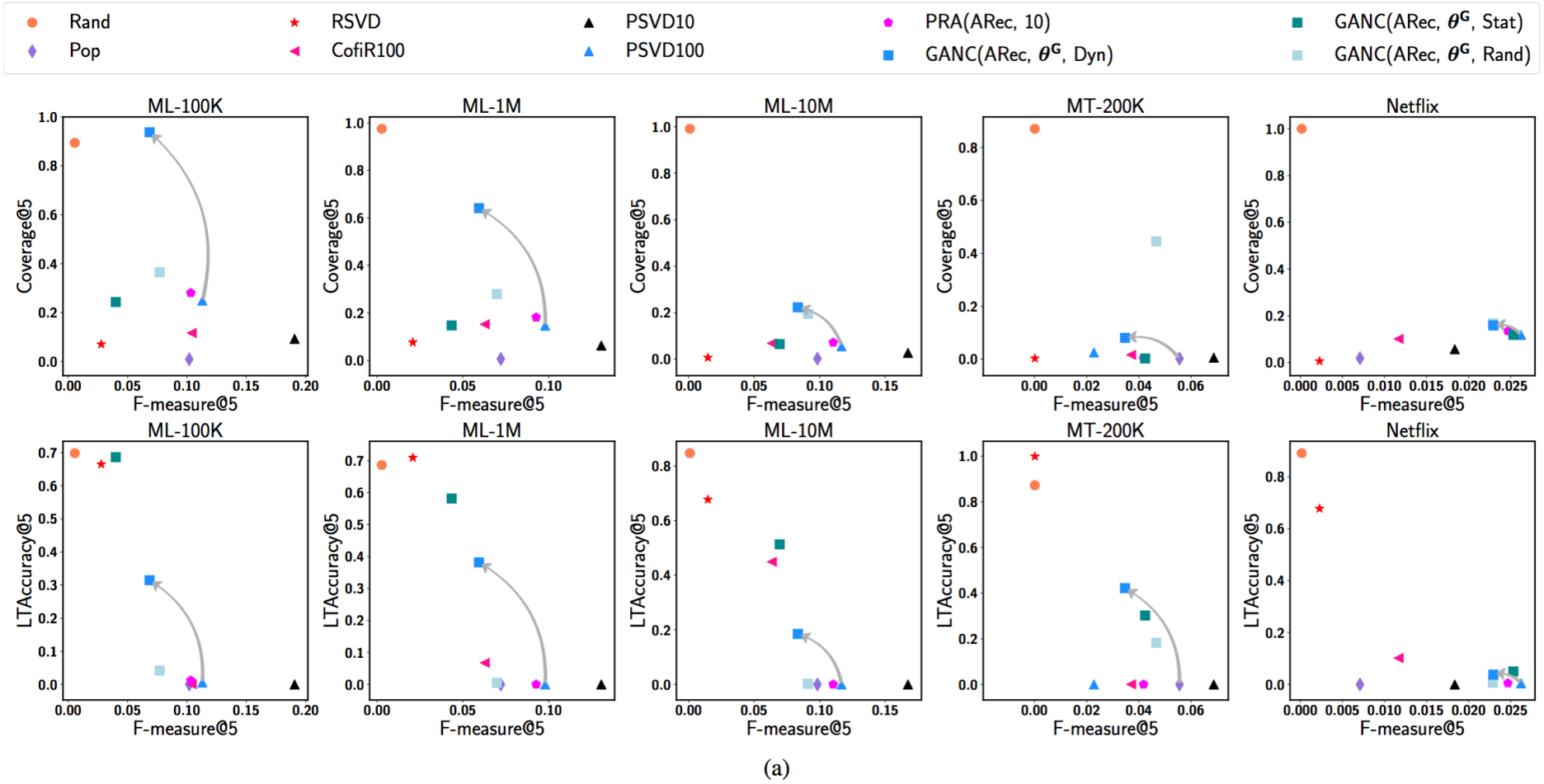


(b) Accuracy recommender (ARec) is PSVD100

Comparison with re-rankings models for rating-prediction

	Alg.	F@5	S@5	L@5	C@5	G@5	Score
ML-1M	RSVD	0.0208 (3)	0.0050 (6)	0.7091 (3)	0.0758 (9)	0.9923 (9)	6.0 (6)
	5D(RSVD)	0.0008 (9)	0.0006 (9)	**0.9579** (1)	0.1927 (4)	0.9468 (3)	5.2 (4)
	5D(RSVD, A, RR)	0.0167 (6)	0.0052 (5)	0.6649 (5)	0.1360 (6)	0.9752 (6)	5.6 (5)
	RBT(RSVD, Pop)	0.0091 (8)	0.0022 (8)	0.8019 (2)	0.1125 (8)	0.9872 (8)	6.8 (7)
	RBT(RSVD, Avg)	0.0155 (7)	0.0044 (7)	0.6816 (4)	0.2261 (3)	0.9704 (4)	5.0 (3)
	PRA(RSVD, 10)	0.0207 (4)	0.0053 (4)	0.6268 (6)	0.1171 (7)	0.9800 (7)	5.6 (5)
	PRA(RSVD, 20)	0.0205 (5)	0.0055 (3)	0.5976 (7)	0.1436 (5)	0.9714 (5)	5.0 (3)
	GANC(RSVD, θ^T, Dyn)	**0.0244** (1)	**0.0077** (1)	0.5139 (9)	0.5113 (2)	0.8947 (2)	3.0 (2)
	GANC(RSVD, θ^G, Dyn)	0.0213 (2)	0.0072 (2)	0.5355 (8)	**0.6492** (1)	**0.8754** (1)	2.8 (1)
ML-10M	RSVD	**0.0147** (1)	**0.0021** (1)	0.6775 (5)	0.0066 (9)	0.9992 (9)	5.0 (4)
	5D(RSVD)	0.0000 (9)	0.0000 (7)	**1.0000** (1)	0.1248 (3)	**0.9609** (1)	4.2 (2)
	5D(RSVD, A, RR)	0.0024 (8)	0.0007 (6)	0.9421 (2)	0.0489 (5)	0.9968 (5)	5.2 (5)
	RBT(RSVD, Pop)	0.0086 (6)	0.0012 (5)	0.8062 (3)	0.0210 (6)	0.9973 (7)	5.4 (6)
	RBT(RSVD, Avg)	0.0087 (5)	0.0013 (4)	0.8039 (4)	0.0614 (4)	0.9945 (4)	4.2 (2)
	PRA(RSVD, 10)	0.0116 (2)	0.0020 (2)	0.5888 (7)	0.0085 (8)	0.9978 (8)	5.4 (6)
	PRA(RSVD, 20)	0.0110 (3)	0.0020 (2)	0.5992 (6)	0.0115 (7)	0.9972 (6)	4.8 (3)
	GANC(RSVD, θ^T, Dyn)	0.0091 (4)	0.0019 (3)	0.5861 (8)	0.2158 (2)	0.9920 (3)	4.0 (1)
	GANC(RSVD, θ^G, Dyn)	0.0057 (7)	0.0012 (5)	0.5704 (9)	**0.2477** (1)	0.9910 (2)	4.8 (3)

Comparison with top-N item recommendation models



(a)

Comparison with top-N recommendation algorithms

- Sparse dataset
  - MT-200K
- Pop is base accuracy recommender
- Three variations of GANC competitive with more PSVD100 and Cofi100