Personalized Recommendation via Parameter-Free Contextual Bandits

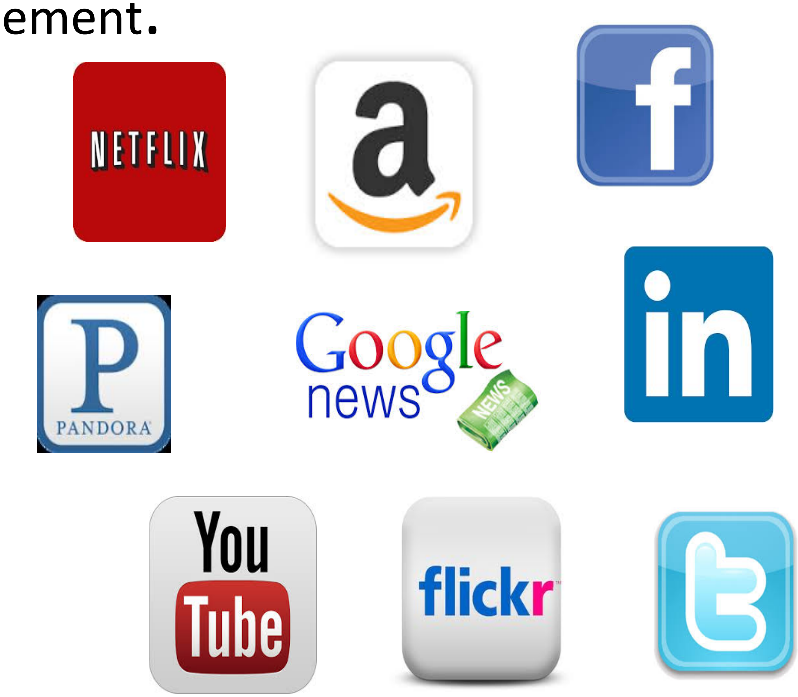
Liang Tang, Yexi Jiang, Lei Li, Chunqiu Zeng, Tao Li
Email: {ltang002, yjian004, lli003, czeng001, taoli}@cs.fiu.edu

Outline

- Introduction
- Motivation
- Solution
- Experiment
- Conclusion
- Q&A

What is Personalized Recommendation?

- Personalized Recommendation help users find interesting items based the individual interest of each item.
  - Ultimate Goal: maximize user engagement.



All the images are downloaded from Google Image.

What is Cold Start Problem?

FLORIDA INTERNATIONAL UNIVERSITY
Miami's public research university

- Do not have enough observations for new items or new users.
  - How to predict the preference of users if we do not have data?

- Many practical issues for offline data
  - Historical user log data is biased.
  - User interest may change over time.

Solving Cold Start Problem

- Feature based modeling
  - How about if the new items have new features?

- Exploration and Exploitation (Our paper)

Exploitation vs Exploration

- Exploitation:
  – Show “best” items to maximize the user’s engagement.
- Exploration:
  – Show new items to explore the user’s preference.
- Goal:
  – Maximize the overall user’s engagement.
- Tradeoff:
  – Only exploitation, you will have bad estimation for “best” items.
  – Only exploration, you will have low user’s engagement.

Bandit Algorithm in Recommender Systems

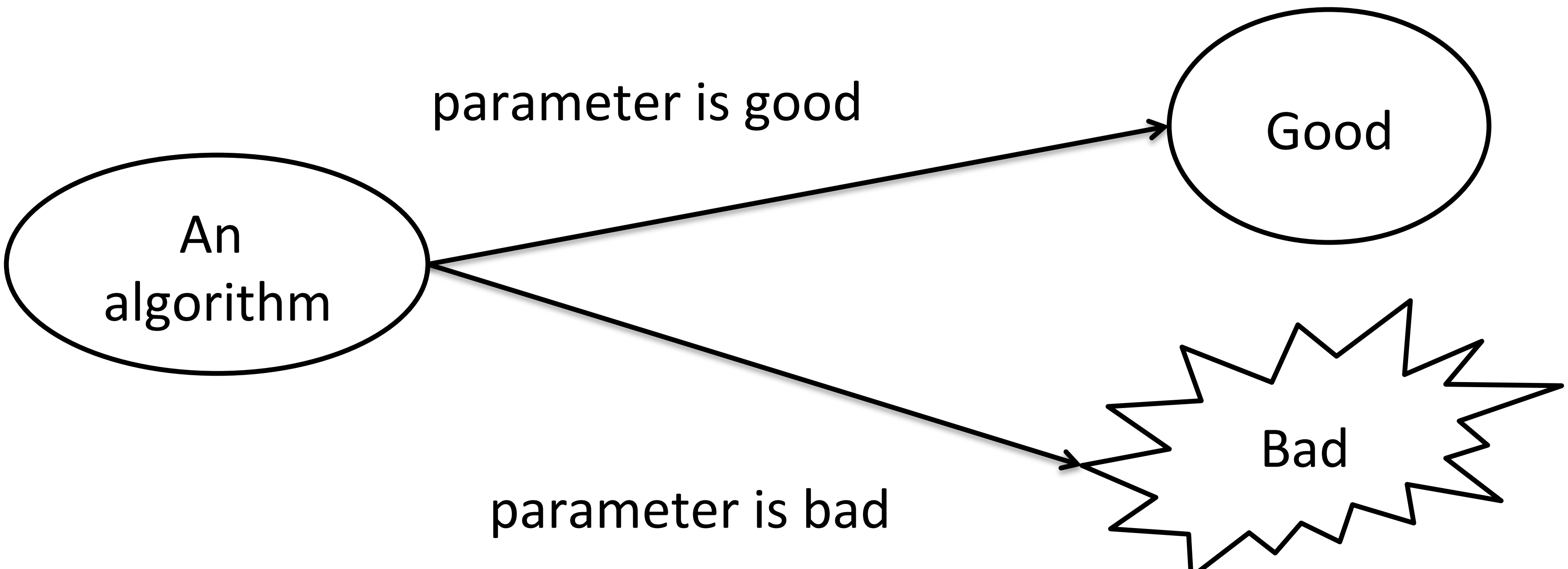
- Bandit algorithm is a framework to balance the tradeoff of Exploitation and Exploration **[2]**.

- Multi-armed Bandit Algorithms **[4]**
  - Estimate the reward of each item based on the click and impression counts. E.g., ε-greedy **[34]**, UCB**[19]**, Bernoulli Thomson Sampling **[14]**.

- Contextual Bandit algorithms **[35]**
  - Estimate the reward of each item based on a feature-based prediction model, where the context is seen as a feature vector.

How to Balance Tradeoff

- Performance is mainly determined by the tradeoff. Existing algorithms find the tradeoff by user input parameters and data characteristics (e.g., variance of the estimated reward).
- Existing algorithms are all parameter-sensitive.



Chicken-and-Egg Problem for Existing Bandit Algorithms

- Why we use bandit algorithms?
  - Solve the cold start problem (No enough data for estimating user preferences).
- How to find the best input parameters?
  - Tune the parameters online or offline.

If you already have the data to tune the parameters, why do you need bandit algorithms?

Our Work

- Parameter-free:
  - It can find the tradeoff by data characteristics automatically.

- Robust:
  - Existing algorithm can have very bad performance if the input parameter is not appropriate.

Solution

- Thompson Sampling
  – Randomly select a model coefficient vector from posterior distribution and find the “best” item.
  – Prior is the input parameter for computing posterior.
- Non-Bayesian Thompson Sampling (**Our Solution**)
  – Randomly select a bootstrap sample to find the MLE of model coefficient and find the “best” item.
  – Bootstrapping has no input parameter.

Bootstrap Bandit Algorithm

Input : a feature vector $\boldsymbol{x}$ of the context.
Output: an item to show

if each article has sufficient observations **then** {
 for each article i=1,…, k
  i. $D^i \leftarrow$ randomly sample $\boldsymbol{n_k}$ impression data of article i with replacement // Generate a bootstrap sample.
  ii. $\boldsymbol{\theta}_i \leftarrow$ MLE coefficient of D^i // Model estimation on bootstrap sample
 select the article $i^* = \text{argmax}(f(\boldsymbol{x}, \boldsymbol{\theta}_i))$, i=1,…, k. to show.
}

Prediction function

else {

 randomly select an article that has no sufficient observations to show.
}

Online Bootstrap Bandits

- Why Online Bootstrap?
  - Inefficient to generate a bootstrap sample for each recommendation.

- How to online bootstrap?
  - Keep the coefficient estimated by each bootstrap sample in memory.
  - No need to keep all bootstrap samples in memory.
  - When a new data arrives, incrementally update the estimated coefficient for each bootstrap sample **[23]**.

Experiment Data

- Two public data sets
  - News recommendation data (Yahoo! Today News)
    - News displayed on the Yahoo! Front Page from Oct. 2nd, 2011 to Oct. 16th 2011.
    - 28,041,015 user visit events.
    - 136 dimensions of feature vector for each event.
  - Online advertising data (KDD Cup 2012, Track 2)
    - The data set is collected by a search engine and published by KDD Cup 2012.
    - 1 million user visit events.
    - 1,070,866 dimensions of the context feature vector.

Offline Evaluation Metric and Methods

FLORIDA INTERNATIONAL UNIVERSITY
Miami's public research university

- Performance Metric
  - Overall CTR (average reward of a trial).
- Evaluation Method
  - The experiment on Yahoo! Today News is evaluated by *replay* **[20]**.
  - The reward on KDD Cup 2012 AD data is simulated with a weight vector for each AD **[8]**.

Experimental Methods

- Our method
  1. Bootstrap(*B*), where *B* is the number of bootstrap samples.
- Baselines
  1. Random: it randomly selects an arm to pull.
  2. Exploit: it only consider the exploitation without exploration.
  3. ε-greedy(ε): ε is the probability of exploration **[34]**.
  4. LinUCB(α): it pulls the arm with largest score defined by the parameter α **[19]**.
  5. TS(q_0): Thompson sampling with logistic regression, where q_0^{-1} is the prior variance, 0 is the prior mean**[8]**.
  6. TSNR(q_0): Similar to TS(q_0), but the logistic regression is not regularized by the prior.

Experiment(Yahoo! News Data)

- All numbers are relative to the random model.

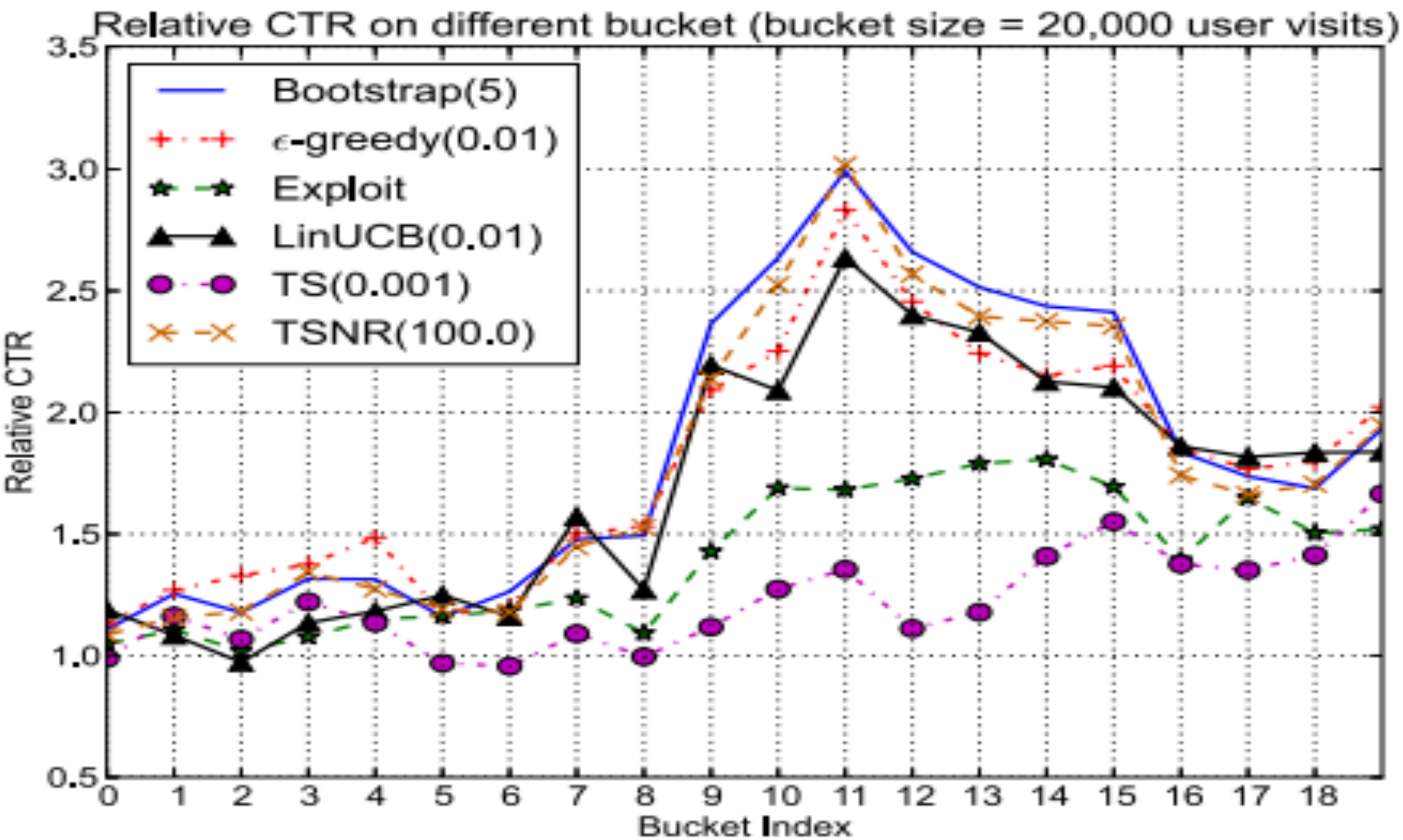
Algorithm	Cold Start				Warm Start			
	mean	std	min	max	mean	std	min	max
Bootstrap(1)	1.7350*	0.08327	1.6032	1.9123	1.7029*	0.1392	1.4299	1.8358
Bootstrap(5)	**1.8025**	0.07676	1.6526	1.9127	1.8366	0.07996	1.7118	1.9514
Bootstrap(10)	1.7536	0.07772	1.6338	1.8814	**1.8403**	0.08518	1.6673	1.9296
Bootstrap(30)	1.7818	0.08857	1.6092	1.9025	1.8311	0.08699	1.7230	1.9396
ϵ-greedy(0.01)	**1.7708**	0.09383	1.6374	1.9503	**1.8466**	0.05494	1.7846	1.9755
ϵ-greedy(0.1)	1.7375	0.04992	1.6452	1.8003	1.8132	0.03502	1.7621	1.8721
ϵ-greedy(0.3)	1.5486	0.03703	1.4812	1.5930	1.5976	0.02739	1.5591	1.6491
ϵ-greedy(0.5)	1.3819*	0.02341	1.3489	1.4169	1.3753*	0.02884	1.3173	1.4020
Exploit	**1.1782***	0.2449	0.9253	1.5724	**1.1576***	0.00198	1.1554	1.1607
LinUCB(0.01)	**1.6349**	0.08967	1.4849	1.7360	**1.8103**	0	1.8103	1.8103
LinUCB(0.1)	1.2037	0.02321	1.1682	1.2577	1.2394	0	1.2394	1.2394
LinUCB(0.3)	1.1661	0.01073	1.1552	1.1926	1.1650	1.863e-08	1.1650	1.1650
LinUCB(0.5)	1.1462	0.01215	1.1136	1.1571	1.1752	1.317e-08	1.1752	1.1752
LinUCB(1.0)	1.1361*	0.01896	1.0969	1.1594	1.1594*	1.317e-08	1.1594	1.1594
TS(0.001)	**1.2203**	0.026	1.1842	1.2670	**1.2725**	0.03175	1.2301	1.3422
TS(0.01)	1.1880	0.02895	1.1585	1.2466	1.2377	0.01886	1.2132	1.2713
TS(0.1)	1.1527	0.01988	1.1289	1.1811	1.1791	0.02225	1.1437	1.2169
TS(1.0)	1.1205	0.0142	1.1009	1.1472	1.1362	0.02203	1.0971	1.1599
TS(10.0)	0.7669*	0.1072	0.5445	0.9526	0.8808*	0.01557	0.8483	0.9031
TSNR(0.01)	1.2173*	0.03369	1.1430	1.2561	1.2972*	0.02792	1.2479	1.3394
TSNR(0.1)	1.2285	0.01948	1.1915	1.2610	1.3028	0.02121	1.2701	1.3461
TSNR(1.0)	1.2801	0.02365	1.2558	1.3303	1.3250	0.03148	1.2486	1.3634
TSNR(10.0)	1.6657	0.03285	1.6025	1.7125	1.6153	0.05608	1.5210	1.7128
TSNR(100.0)	**1.7816**	0.07609	1.7093	1.9278	1.8399	0.1134	1.5240	1.9200
TSNR(1000.0)	1.7652	0.09946	1.6123	1.9346	**1.8769**	0.03731	1.8409	1.9656

Experiment(AD KDD Cup' 12)

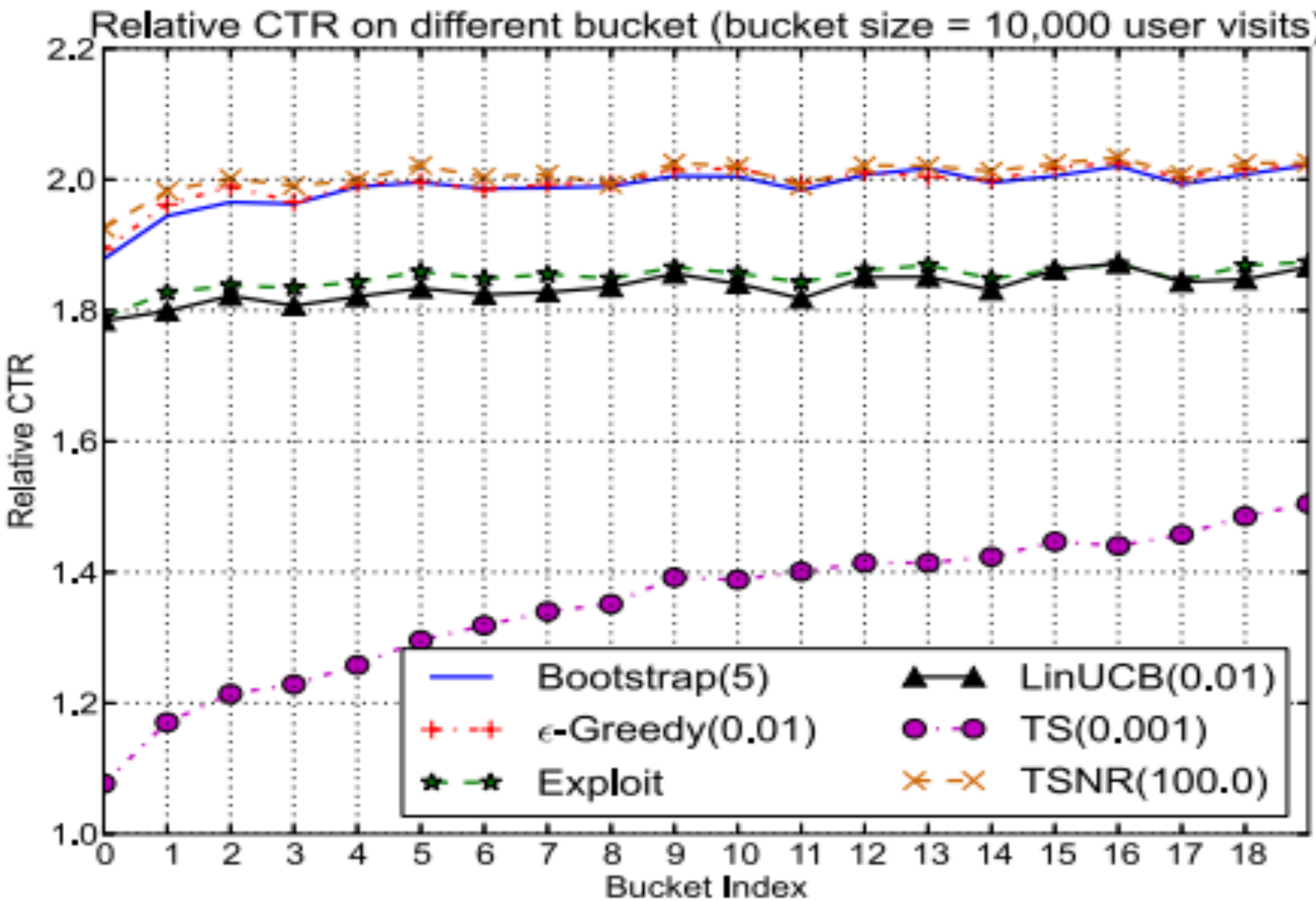
- All numbers are relative to the random model.

Algorithm	Cold Start				Warm Start			
	mean	std	min	max	mean	std	min	max
Bootstrap(1)	**1.9933**	0.01291	1.9692	2.0098	**1.9990**	0.005678	1.9878	2.0083
Bootstrap(5)	1.9883	0.01106	1.9686	2.0012	1.9964	0.004983	1.9848	2.0022
Bootstrap(10)	1.9862	0.009128	1.9672	1.9977	1.9890	0.005434	1.9829	2.0003
Bootstrap(30)	1.9824*	0.01492	1.9566	2.0088	1.9886*	0.006086	1.9753	1.9954
ϵ-greedy(0.01)	**1.9941**	0.007293	1.9834	2.0060	**1.9971**	0.004908	1.9886	2.0038
ϵ-greedy(0.1)	1.9089	0.004887	1.8965	1.9145	1.8952	0.002741	1.8910	1.8986
ϵ-greedy(0.3)	1.7039	0.003797	1.6990	1.7101	1.6973	0.009368	1.6834	1.7193
ϵ-greedy(0.5)	1.5018*	0.004335	1.4965	1.5114	1.4983*	0.006319	1.4845	1.5067
Exploit	**1.8185***	0.05235	1.7228	1.8934	**1.9241***	0.007046	1.9152	1.9370
LinUCB(0.01)	1.8551	0.03543	1.7977	1.9059	**1.9279**	0.006951	1.9178	1.9371
LinUCB(0.1)	**1.9168**	0.005466	1.9070	1.9267	1.9202	0.004434	1.9112	1.9266
LinUCB(0.3)	1.8665	0.003644	1.8609	1.8726	1.8610	0.003271	1.8550	1.8661
LinUCB(0.5)	1.7808	0.007009	1.7669	1.7913	1.7903	0.0051	1.7823	1.7988
LinUCB(1.0)	1.6693*	0.004738	1.6634	1.6762	1.6742*	0.003179	1.6704	1.6792
TS(0.001)	1.3587	0.009703	1.3366	1.3736	1.3518	0.01002	1.3297	1.3673
TS(0.01)	1.4597	0.007215	1.4504	1.4749	1.4891	0.006421	1.4771	1.4994
TS(0.1)	**1.5714**	0.004855	1.5647	1.5791	**1.5905**	0.004176	1.5826	1.5967
TS(1.0)	1.5345	0.003435	1.5262	1.5384	1.5421	0.003741	1.5376	1.5480
TS(10.0)	0.9388*	0.4236	0.3064	1.5675	1.3174*	0.003157	1.3115	1.3212
TSNR(0.01)	1.4856*	0.01466	1.4657	1.5078	1.5700*	0.02163	1.5499	1.6298
TSNR(0.1)	1.7931	0.01284	1.7774	1.8167	1.8716	0.01035	1.8518	1.8870
TSNR(1.0)	1.9826	0.005853	1.9704	1.9921	1.9952	0.006996	1.9833	2.0047
TSNR(10.0)	**2.0118**	0.007808	1.9941	2.0208	2.0095	0.005107	2.0022	2.0198
TSNR(100.0)	2.0039	0.008942	1.9912	2.0215	**2.0097**	0.004586	2.0022	2.0187
TSNR(1000.0)	2.0047	0.01022	1.9894	2.0228	2.0088	0.004644	1.9966	2.0151

CTR over Time Bucket (Yahoo! News Data)

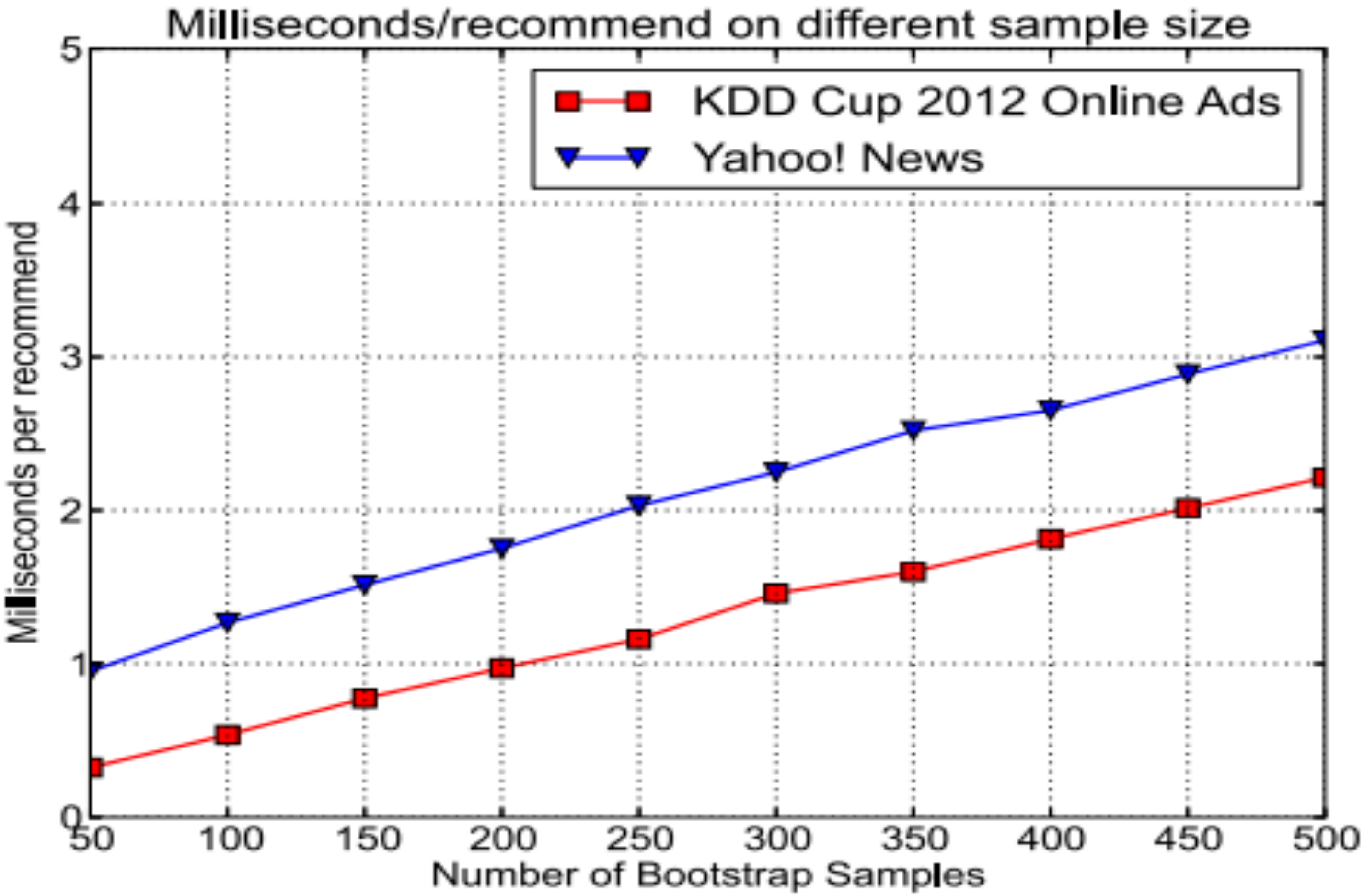


CTR over Time Buckets (KDD Cup Ads Data)



Efficiency

- Time cost on different bootstrap sample sizes



Summary of Experiment

- Summary
  - For solving the contextual bandit problem, the algorithms of ϵ-greedy and LinUCB can achieve the optimal performance, but the input parameters that control the exploration need to be tuned carefully.
  - The probability matching strategies highly depend on the selection of the prior.
  - Our proposed algorithm is a safe choice of building predictive models for contextual bandit problems under the scenario of cold-start.

Conclusion

- Propose a non-Bayesian Thompson Sampling method to solve the personalized recommendation problem.
- Give both theoretical and empirical analysis to show that the performance of Thompson sampling depends on the choice of the prior.
- Conduct extensive experiments on real data sets to demonstrate the efficacy of the proposed method and other contextual bandit algorithms.

Question and Answer

Thanks!