

1. Introduction

Bandit: decision-making model where a learner chooses from multiple actions each round and receives a reward. The goal is to choose the optimal action every round and maximize cumulative reward or minimize cumulative regret.

SCLB: the learner observes additional context and applies it to the actions to get feature vectors. The corresponding reward is a linear function of this feature vector and an unknown parameter vector θ^* .

Sparsity: θ^* is a sparse vector. Only few feature are relevant to computing the reward.

p : warmup rounds before feature selection
 k : number of feature selected

Many domains like personalized medicine and online advertising are represented by SCLBs in sparse, high-dimensional setting.

Problem: accurate prediction requires lot of data, which is costly to collect, and the true relevance of candidate features for computing rewards is uncertain.

It is important to study SCLBs under high-dimensional sparse regime and exploit this sparsity.

Our strategy is to estimate rewards with sparse regression and to embed a feature selection routine within SCLBs.

Sparse Sequential Learning: Exploring Stochastic Contextual Linear Bandit and Feature Selection Combinations for Fixed Reduced Dimensions

2. Research Question

How do the base bandits and combinations of SCLBs and feature selection methods compare in a high-dimensional sparse setting when constrained to a fixed number of features?

3. Methodology

Implemented SCLB-FS algorithms that run a feature selection routine mid-bandit.

Explored the algorithms via a (p, k) grid search.

Compared best SCLB-FS configs against base SCLB counterparts and across themselves.

4. Conclusion

Feature selection embedded in bandits yield similar or improved performance over their base counterparts, with only a fraction of the data-collection cost, provided the feature selection is done accurately.

5. Results

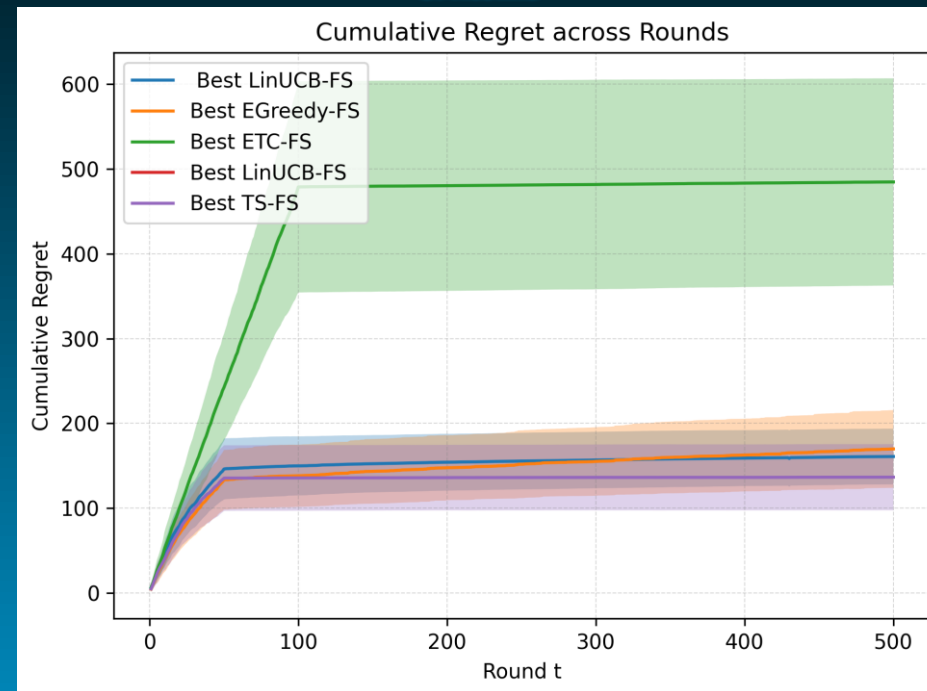


Figure 1: cumulative regret graph: ϵ -Greedy-FS, ETC-FS, LinUCB-FS & TS-FS

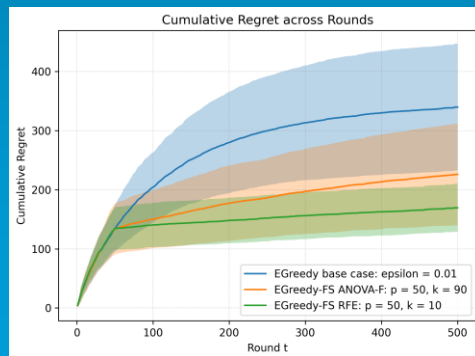


Figure 2: ϵ -Greedy vs ϵ -Greedy-FS

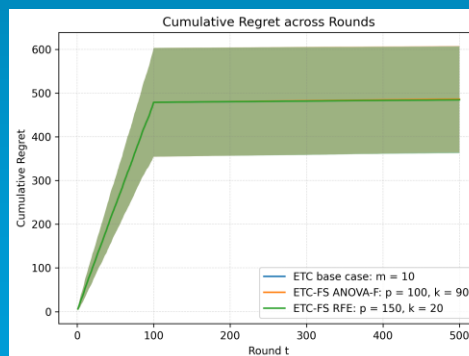


Figure 3: ETC vs ETC-FS

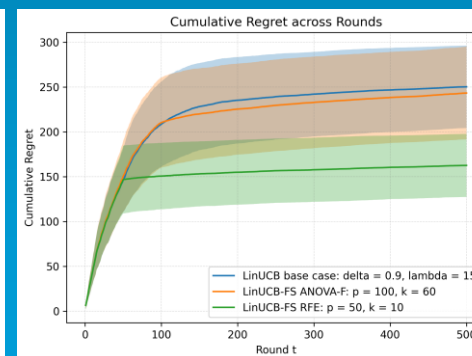


Figure 4: LinUCB vs LinUCB

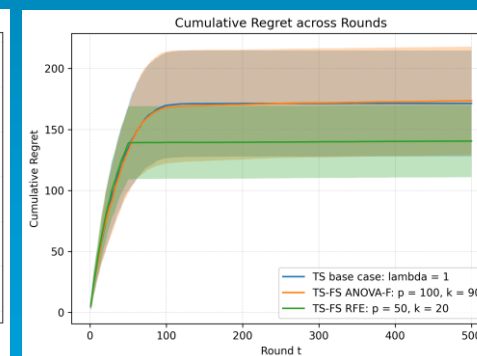


Figure 5: FS Ts vs TS-FS