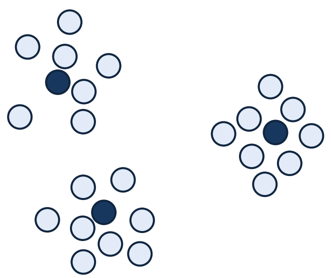
Instance Specific Approximations for Submodular Maximization

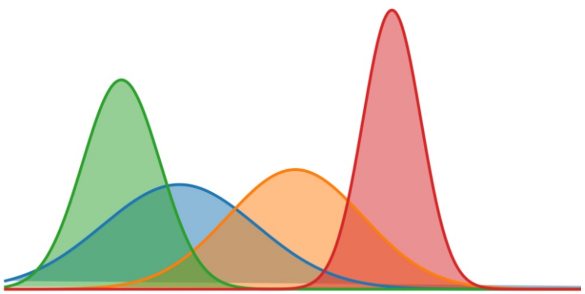
Eric Balkanski, Sharon Qian, Yaron Singer

ICML 2021

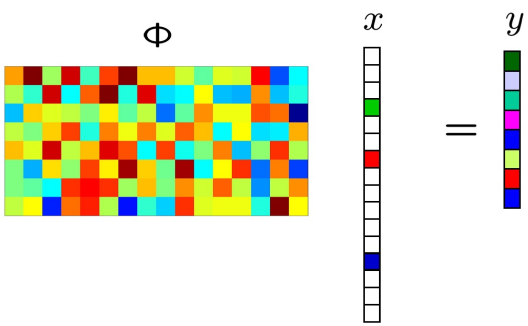
Computational Intractability in Machine Learning



Clustering



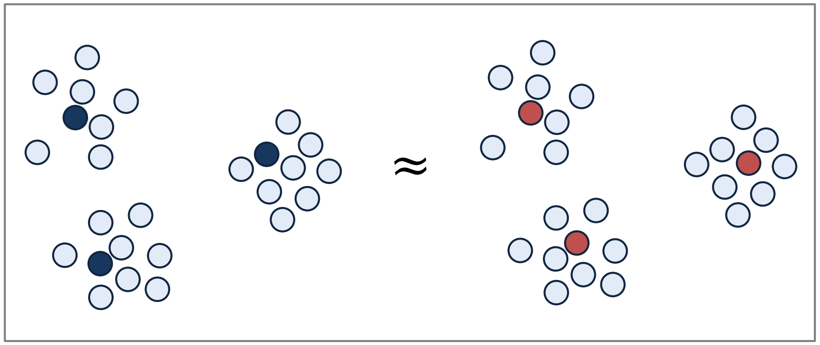
Maximum Likelihood Estimation

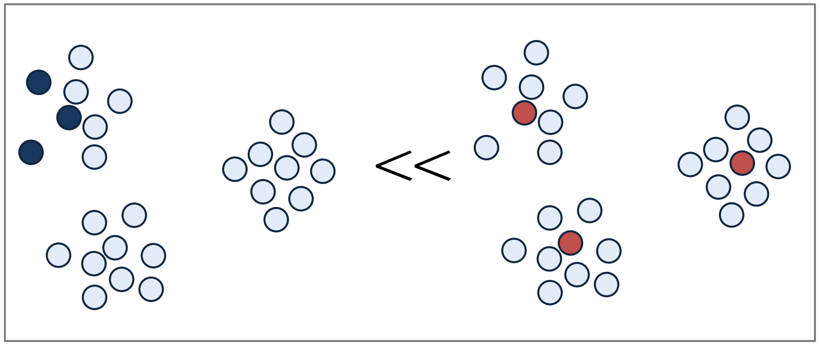


Sparse Recovery

Evaluating Algorithm Performance

Compare solution found to value of an optimal solution





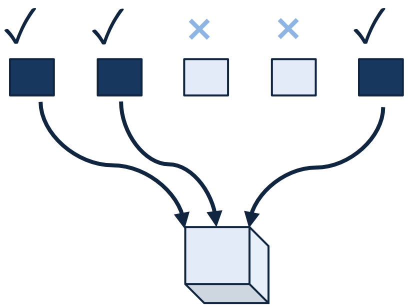
Issue: we do not know the optimal solution!

Main Question

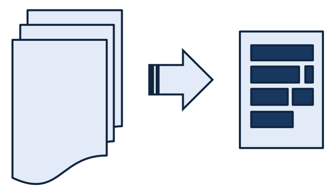
How do we measure algorithm performance on instances that are computationally intractable?

Submodular Optimization

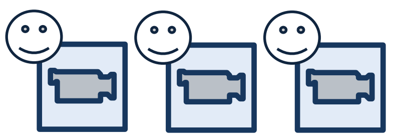
In practice



Feature Selection
[Das-Kempe 11]



Data Summarization
[Lin-Bilmes 11]



Recommender Systems
[Mirzasoleiman-Badanidiyuru-Karbasi 16]

In theory

Theorem (informal): There is no polynomial time algorithm that can obtain better than a $1\text{-}1/e \approx 0.63$ approximation guarantee.

Explanations for Greedy Performance in Practice

Properties that allow improved approximations for greedy:

- Curvature [Conforti and Cornuejols 84]
- Stability [Chatziafratis-Roughgarden-Vondrak 17]
- Sharpness [Torrico-Singh-Pokutta 20]
- Budget-smoothness [Rubinstein-Zhao 21]

Main Question for Submodular Maximization

How do we measure the performance of greedy in practice?

Contribution

- **Dual:** Efficient method to compute instance specific approximations for submodular maximization

- Based on a novel dual approach

- Use Dual to show that greedy and other algorithms perform extremely well in practice

Methodology

<u>Goal</u>: Upper bound the optimal value of f to compute instance-specific approximations.

set S

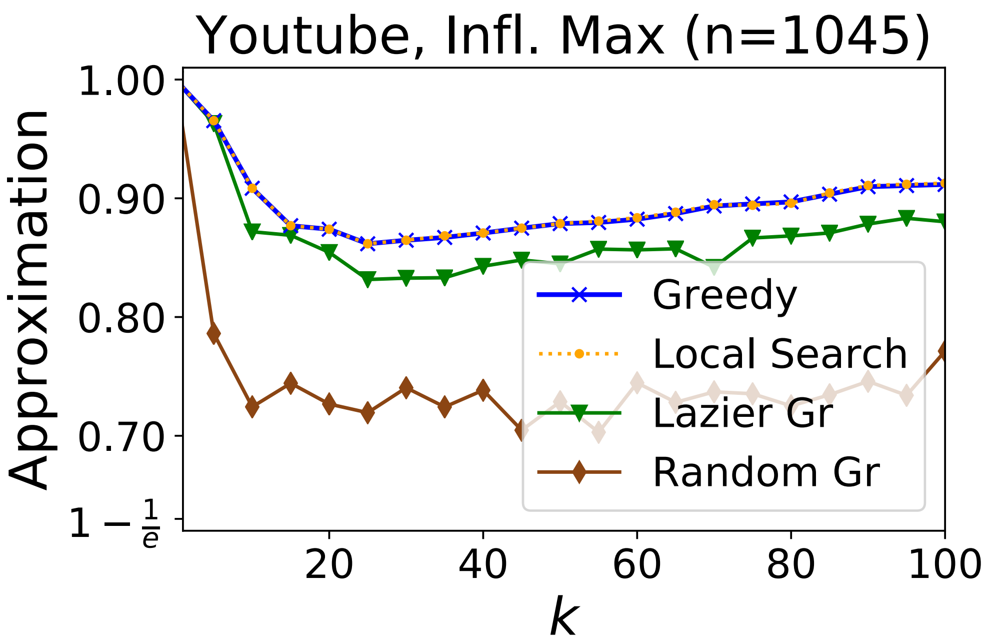
$$f : 2^N \to \mathbb{R} \quad \longleftrightarrow \quad g : \mathbb{R} \to \mathbb{R}$$

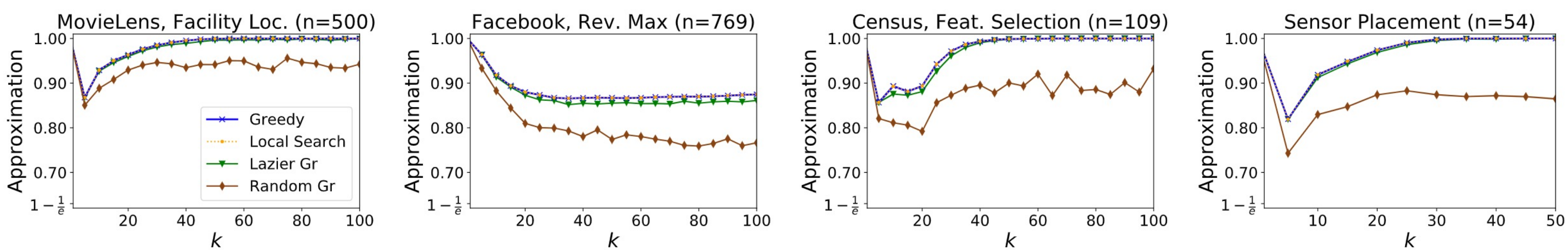
function value v

value v

min number of elements $|S|$

Result: Algorithms Perform Well





Result: Dual beats Benchmarks