NEURAL INFORMATION PROCESSING SYSTEMS

Recruitment Strategies that Take a Chance

Gregory Kehne[†], Ariel Procaccia[†], and Jingyan Wang[*]

[†] Harvard University

[*] Georgia Tech

Our Setting

We take the perspective of a hiring algorithm in a batch setting

Each candidate $i \in [n]$ has a known probability and value (x_i, p_i)

(x_1, p_1) (x_2, p_2) (x_3, p_3) (x_4, p_4) (x_5, p_5)

Hiring algorithm chooses some $S \subseteq [n]$ to make offers to:

Candidates independently accept with probabilities p_i, algorithm receives reward

Our goal: How to choose subsets $S \subseteq [n]$?

Target/Constraint

There is some target number of acceptances M

$M = 3$

Hiring algorithm chooses $\{1,2,3\} \subseteq [n]$ to make offers to:

$S_Z = 2$

Candidates independently accept with probabilities p_i, algorithm receives reward,

minus some penalty for missing its target.

The Penalty and Objective

What form does this penalty for missing M take?

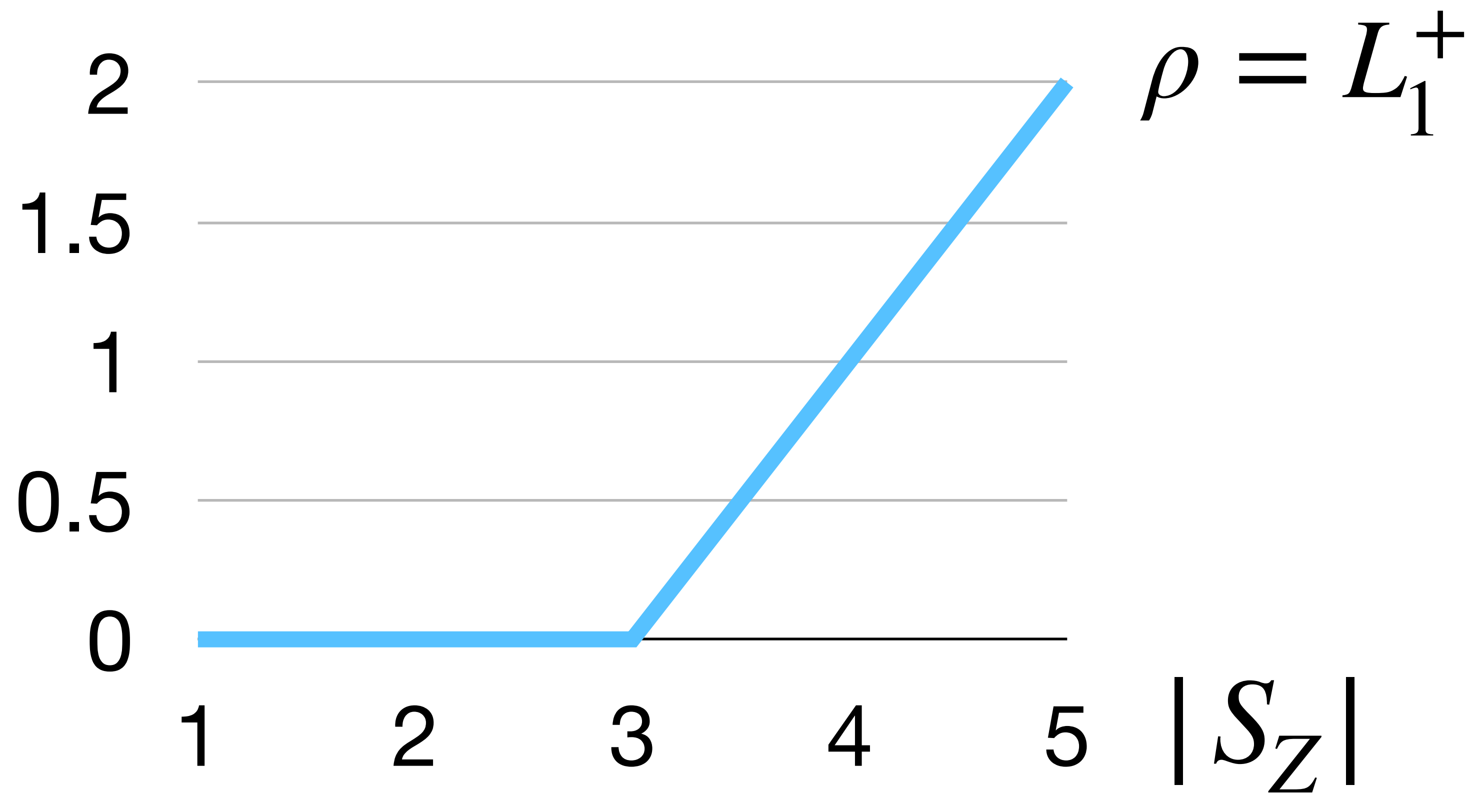
A number of natural choices.

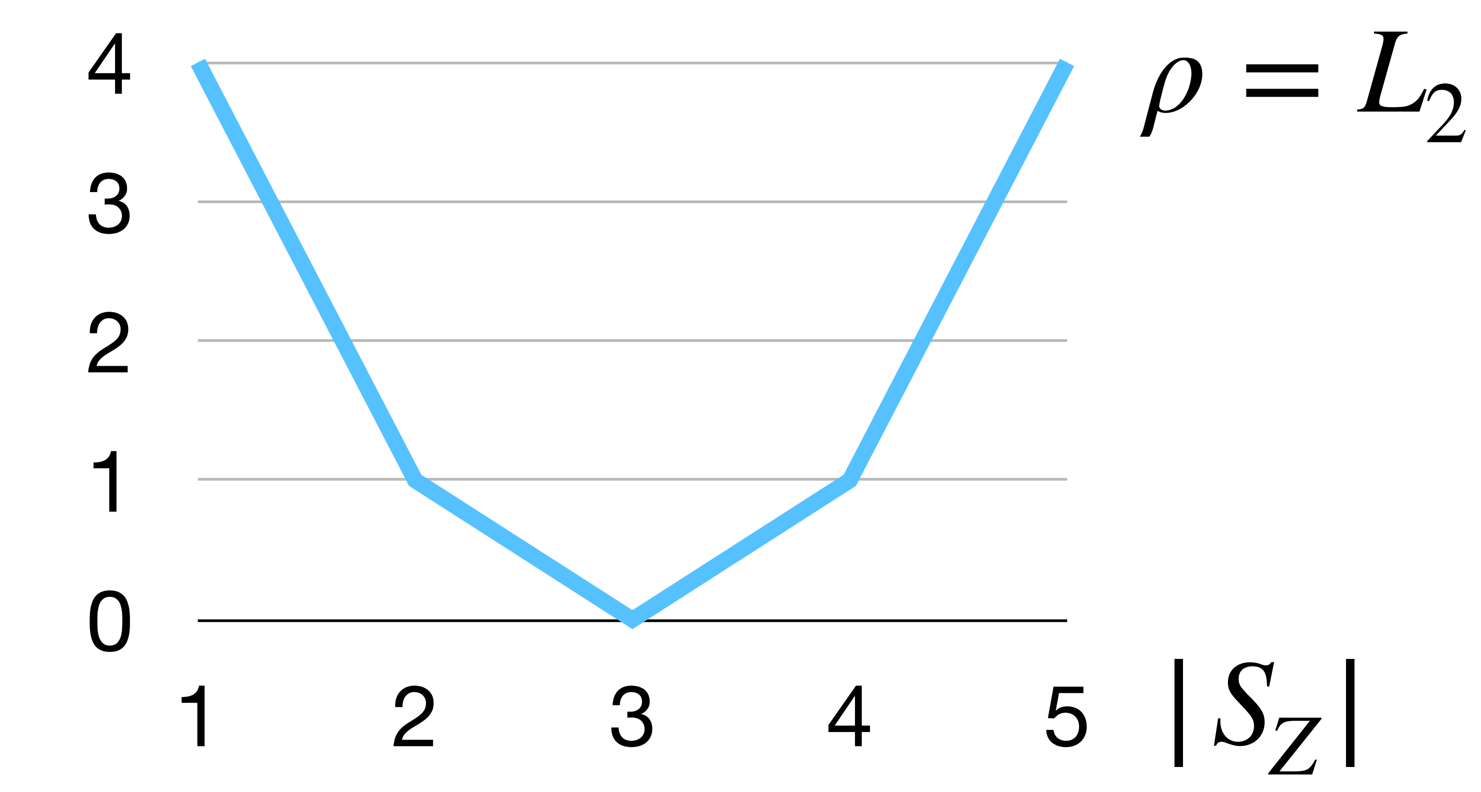
Here are two: linear one-sided loss (L_1^+) and MSE loss (L_2)

$$U(S) = \sum_{i \in S} x_i p_i - \lambda \cdot \mathbb{E}_Z \left[\rho(|S_Z|, M) \right]$$

λ is a regularizing term: reflects how important the target is relative to the candidate values.

$M = 3$

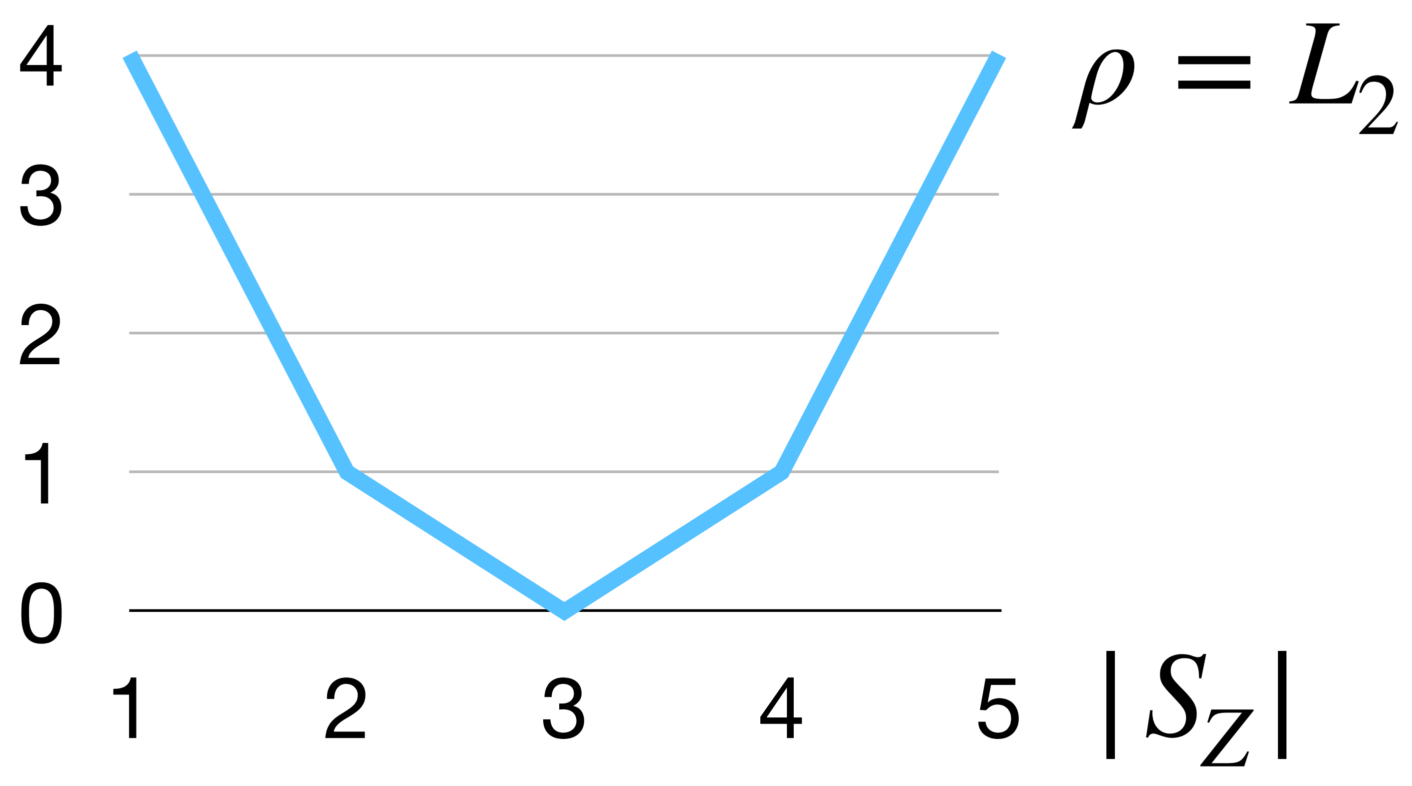




L_2 Loss

For mean squared error (MSE) loss, for a given subset $S \subseteq [n]$,

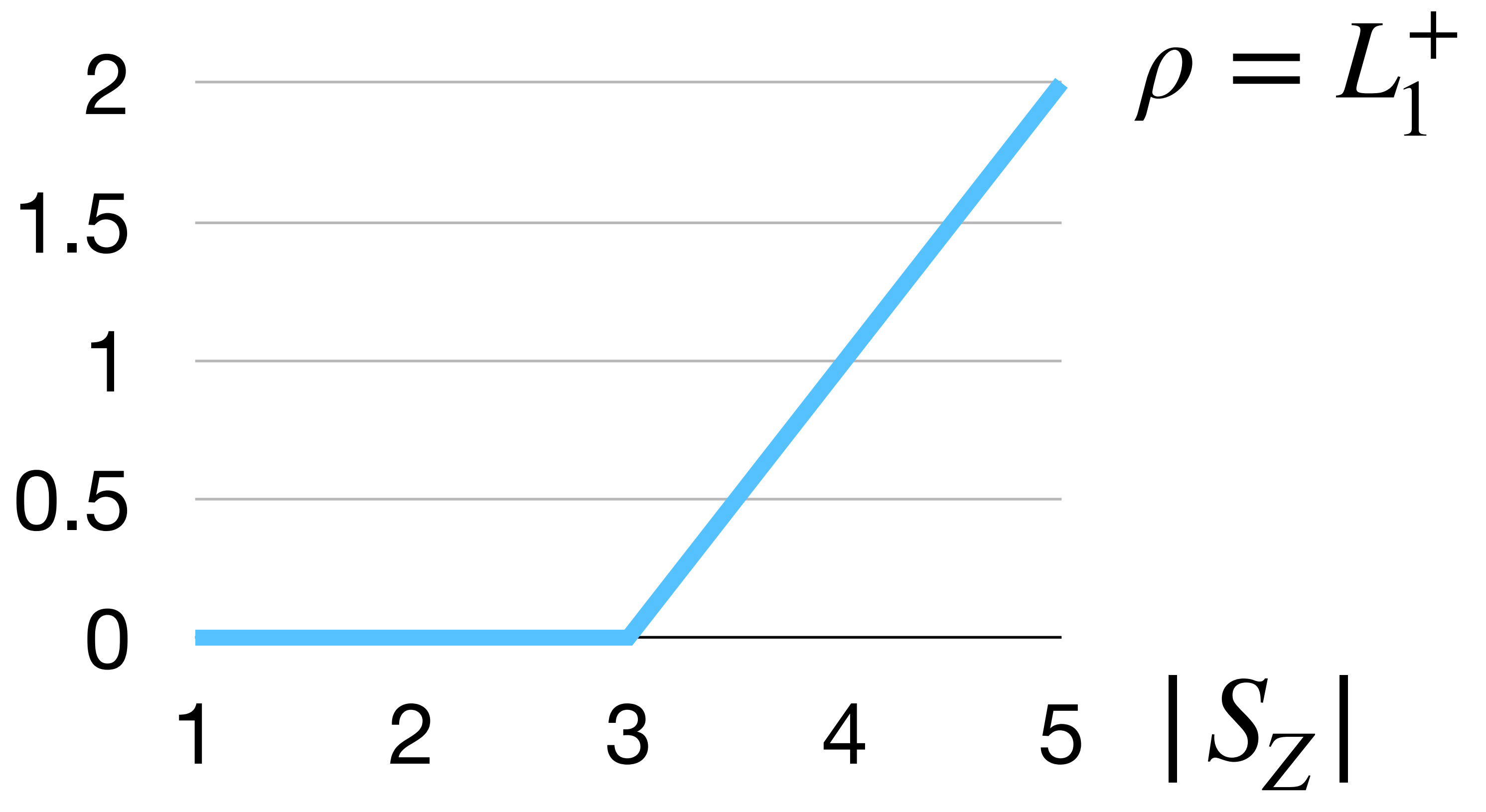
$$U(S) = \sum_{i \in S} x_i p_i - \lambda \cdot \mathbb{E}_Z \left[(|S_Z| - M)^2 \right]$$



Theorem: There is an FPTAS which finds some $S \subseteq [n]$ within ϵ of the optimal solution, in $poly(n, M, \lambda, \epsilon^{-1})$

L_1^+ Loss

For one-sided linear loss, for a given subset $S \subseteq [n]$,



$$U(S) = \sum_{i \in S} x_i p_i - \lambda \cdot \mathbb{E}_Z \left[(|S_Z| - M)_+ \right]$$

Greedy Algorithms

$$p_{min} = \min_i p_i$$

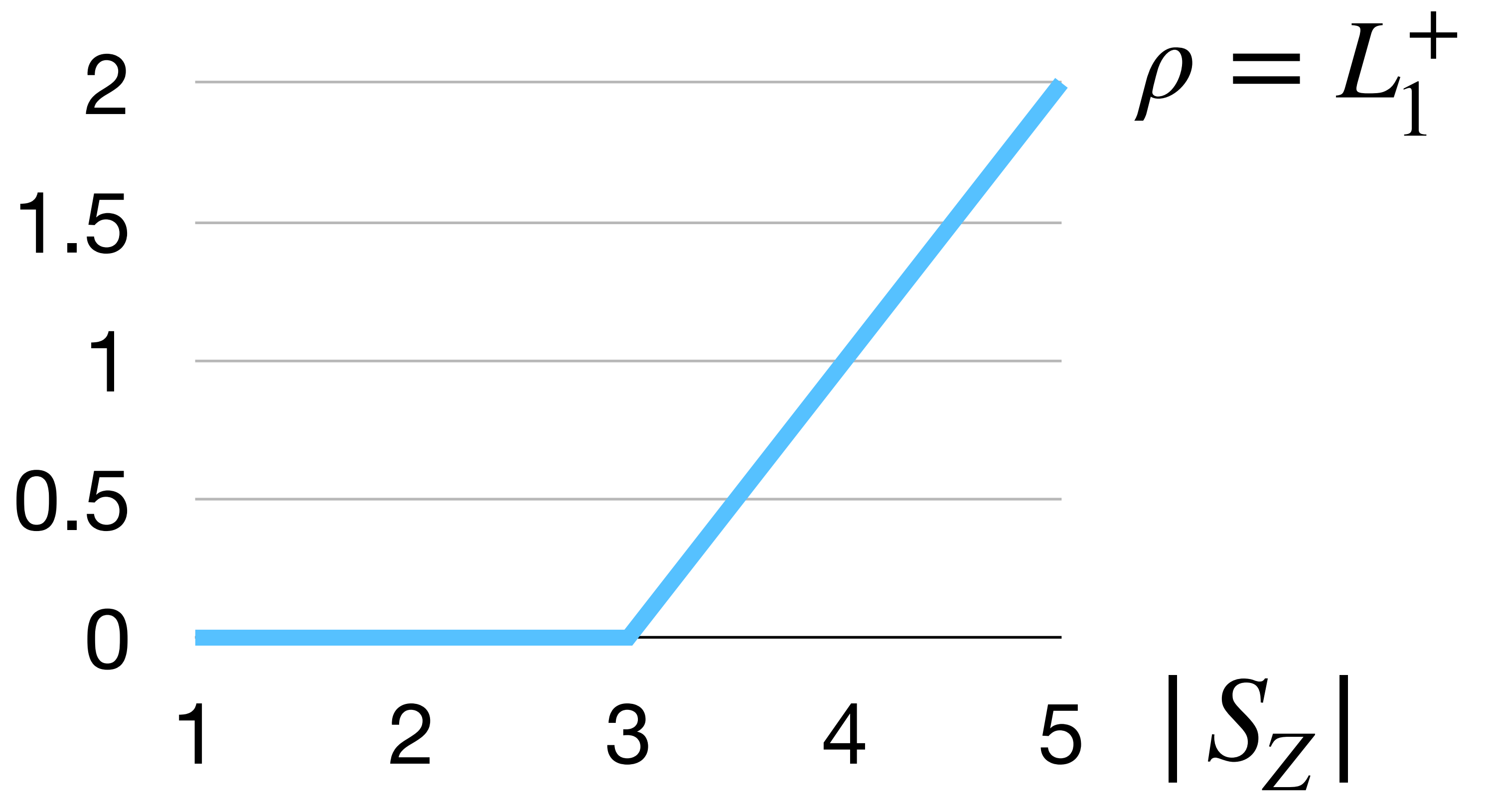
xGreedy: candidates i are added to S in the order of decreasing value x_i

xpGreedy: candidates i are added to S in the order of decreasing expected value $x_i p_i$

Theorem: xpGreedy is a $\Theta(p_{min})$ approximation, and xGreedy is an $\Omega(p_{min}^2)$ and $O(p_{min})$ approximation to this objective

L_1^+ Loss: a new algorithm

$$U(S) = \sum_{i \in S} x_i p_i - \lambda \cdot \mathbb{E}_Z \left[(|S_Z| - M)_+ \right]$$



Approach:

Divide $[n]$ into three groups, depending on value x_i relative to penalty weight λ

$i : x_i \leq \lambda \cdot (1 - p_{min}/4)$

$i : \lambda \cdot (1 - p_{min}/4) \leq x_i \leq \lambda$

$i : x_i \geq \lambda$

Compute a constant-factor approximation on each group separately

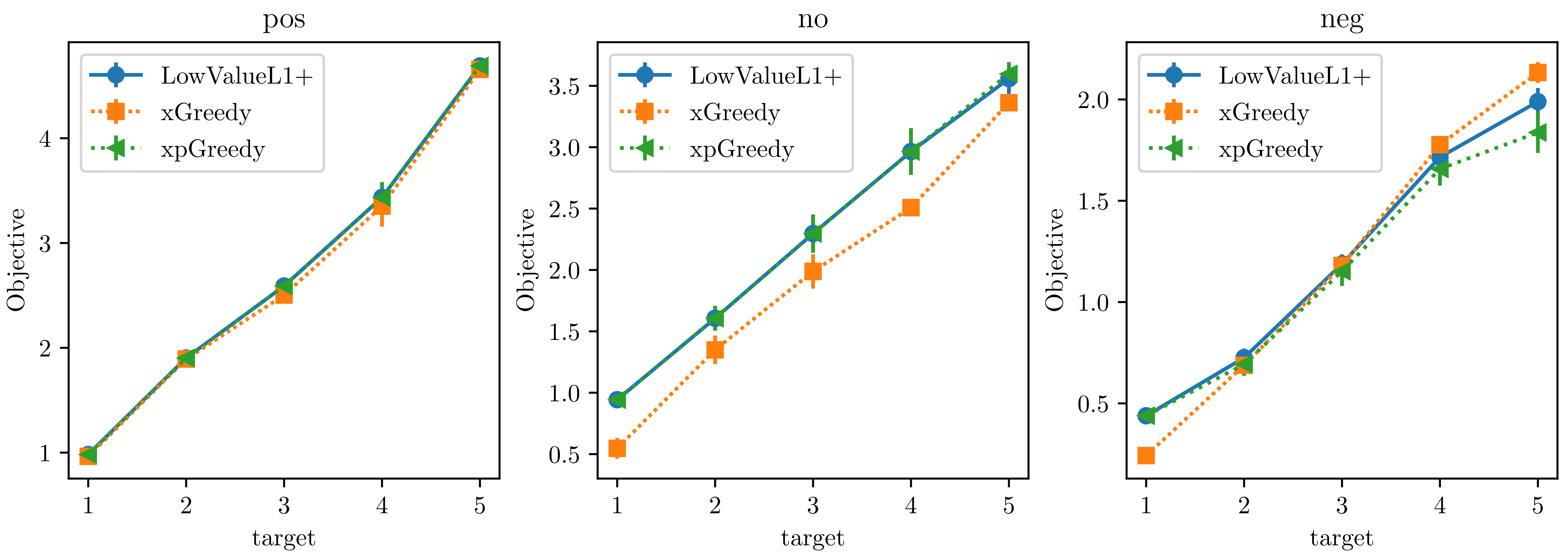
Theorem: This is a constant-factor approximation to this objective in $O(n^{f(p_{min},\, \lambda/x_{max})})$, where $f(p_{min}, \lambda/x_{max}) = O\left(p_{min}^{-2} \cdot \max(1,\ \log p_{min}^{-1},\ \log \lambda/x_{max})\right)$

Experiments

How does our algorithm perform against the greedy algorithms? ($n = 50$)

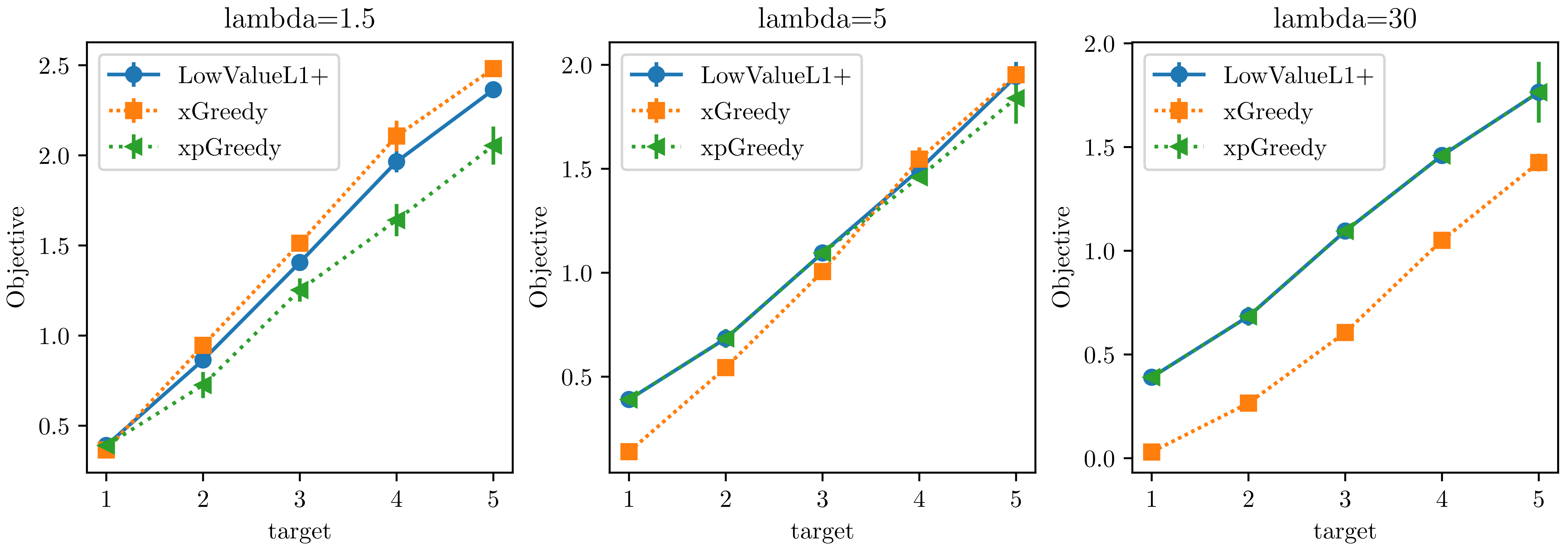
Varying correlation between

x_i and p_i



Increasing penalty regularizer

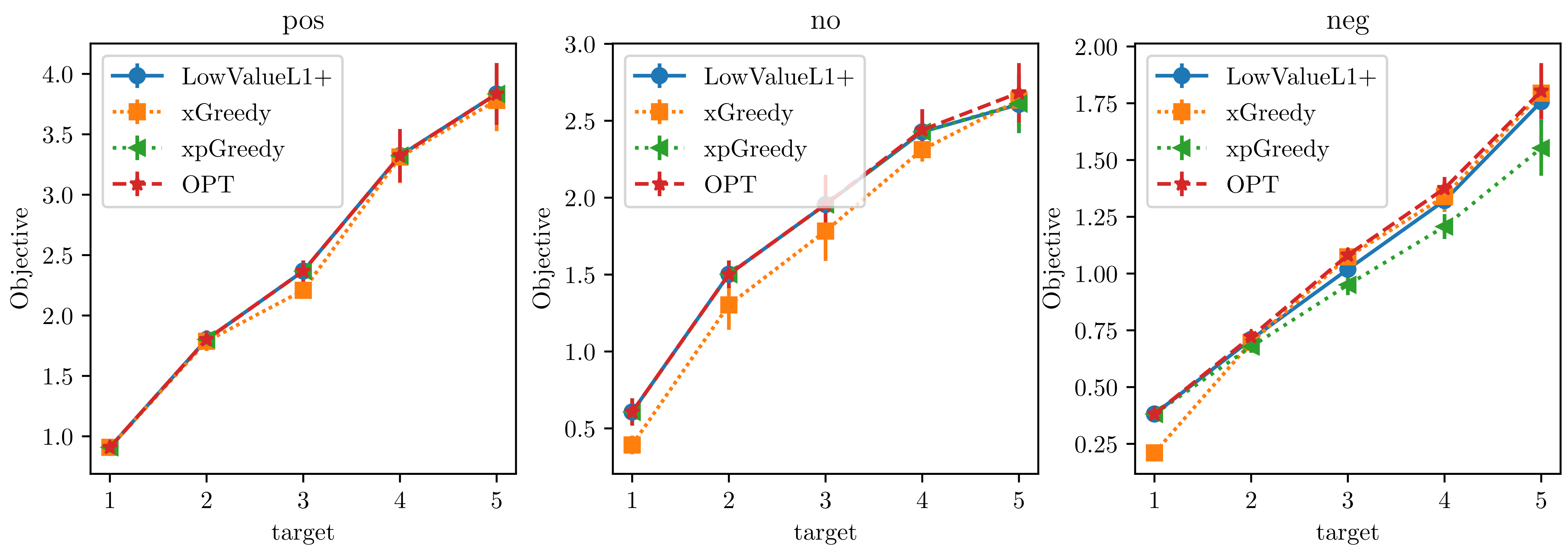
λ relative to the x_i



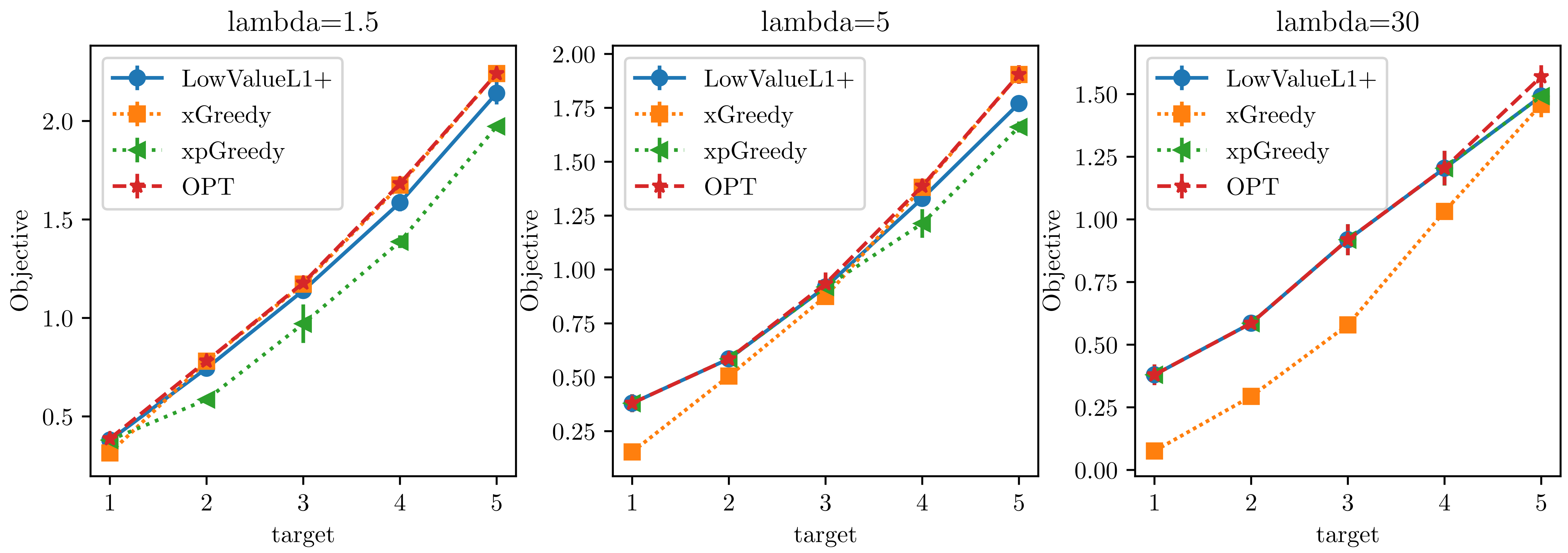
Experiments (contd.)

How do our algorithm and the greedy algorithms perform against OPT? ($n = 20$)

Varying correlation between x_i and p_i



Increasing penalty regularizer λ relative to the x_i



Thank you!

Please contact the authors* with any questions,

or ask in person at the poster session!

*gkehne@g.harvard.edu