INF580 LARGE SCALE MATHEMATICAL OPTIMIZATION

Instructor: Leo Liberti

Project 6: random projection

Ming Cheng
Yunhao Chen

16/03/2022

Summary

1. Problem formulation
2. Solution retrieval
3. Implementation details
4. Results

Problem formulation

Random projection + Linear programming

$$P \equiv min\{c^{\mathrm{T}}x \mid Ax = b, x \geq 0\}$$

$$D \equiv max\{b^{\mathrm{T}}y \mid A^{\mathrm{T}}y \leq c\}$$

$$TP \equiv min\{c^{\mathrm{T}}x \mid TAx = Tb, x \geq 0\}$$

$$TD \equiv max\{(Tb)^{\mathrm{T}}u \mid (TA)^{\mathrm{T}}u \leq c\}$$

Dimension:

A (m,n)
x, c (n,1)
y, b (m,1)
T (k,m) -> random projection matrix
u (k,1)

k<m<n

Problem formulation

Theorem 1 (Johnson-Lindenstrauss Lemma). *Given* $\epsilon \in (0,1)$ *and an* $m \times n$ *matrix* A, *there exists a* $k \times m$ *matrix* T *such that:*

$$\forall 1 \leq i \leq j \leq n \quad (1-\epsilon)\|A_i - A_j\| \leq \|TA_i - TA_j\| \leq (1+\epsilon)\|A_i - A_j\|$$

where k *is* $O(\epsilon^{-2} \ln n)$.

The “distance structure” between columns of **TA** remains approximately the same as columns in **A**.

Solution of TP is a good approximation of solution of P?

Solution retrieval

From TP to P:

$$\mathcal{F}(P) \equiv \{x \in \mathbb{R}^n \mid Ax = b, x \geq 0\}$$

$$(TP)_\theta \equiv min\left\{c^{\mathrm{T}}x \mid TAx = Tb, \sum_{j=1}^{n} x_j \leq \theta, x \geq 0\right\}$$

Theorem 2. *Assume $\mathcal{F}(P)$ is bounded and non-empty. Let y^* be an optimal dual solution of P of minimal Euclidean norm. Given $0 \leq \delta \leq |v(P)|$, we have*

$$v(P) - \delta \leq v((TP)_\theta) \leq v(P)$$

with probability at least $p = 1 - 4ne^{-C(\epsilon^2 - \epsilon^3)k}$, where $\epsilon = O\left(\frac{\delta}{\theta^2 \|y^*\|}\right)$. v(P) = optimal objective function value of P

The objective function value of projected problem v(TP) is a good approximation of objective value of primal problem v(P).

Solution retrieval

From TP to P:

On the other hand, given x' the solution of TP, we can prove that $Ax' \neq b$ almost surely. (cf Chapter 5 in paper [1]). Therefore, the projected problem directly gives us an approximate optimal objective function value, but not the optimum itself.

$$\boxed{x'} \longrightarrow \boxed{A\hat{x} = b - Ax'} \longrightarrow \boxed{x = x' + \hat{x}} \longrightarrow \boxed{Ax = b}$$

Given one solution of TP

Solve the equation, get x_hat

Sum them, get one solution of P

x satisfies the constraint

Solution retrieval

From TD to P:

$$u$$

Given one solution of TD

$$y = T^{\mathrm{T}} u$$

Get one solution y of dual problem D

$$y_i(A_i x - b_i) = 0 \ \forall 1 \leq i \leq m$$
$$((A^{\mathrm{T}} y)_j - c_j) x_j = 0 \ \forall 1 \leq j \leq n$$

complementary slackness must be satisfied

$$|y_i| \geq \epsilon \implies A_i x = b_i \ \forall 1 \leq i \leq m$$
$$|(A^{\mathrm{T}} y)_j - c_j| \geq \epsilon \implies x_j = 0 \ \forall 1 \leq j \leq n$$

Loosened conditions

Epsilon is a hyperparameter, taken as 1e-3

$$\mathbf{x}$$

Get one solution of primal problem P

Implementation details

Generate a feasible problem:

We generate A,x,T and c where:
A ~ U[-1,1[with size(m,n)
x,c ~ U[0,1[with size (n,1)
T ~ N(0,1/k) with size(k,m)

B is calculated by A*x with size (m,1)
TA is calculated by T*A with size (k,n)
TB is calculated by T*B with size (k,1)

k<m<n

$$P \equiv min\{c^{\mathrm{T}}x \mid Ax = b, x \geq 0\}$$

$$TP \equiv min\{c^{\mathrm{T}}x \mid TAx = Tb, x \geq 0\}$$

$$TD \equiv max\{(Tb)^{\mathrm{T}}u \mid (TA)^{\mathrm{T}}u \leq c\}$$

Program Introduction

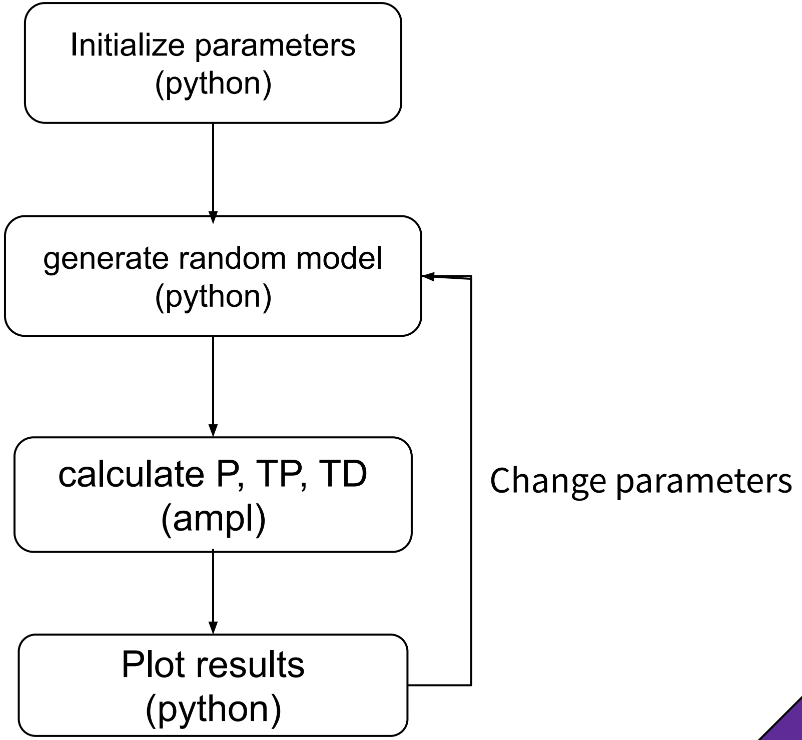
We set basicly m=500, n=700, k=300.

We analysed:

1. objective function value
2. feasibility error: $\frac{\|Ax-b\|_1}{\|b\|_1}$
3. cpu time (s)

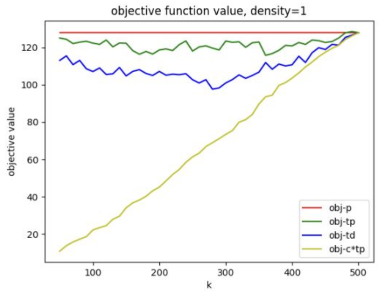
We studied the impact of parameter changes:

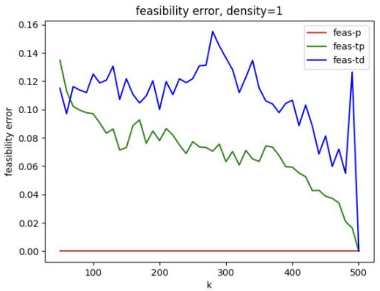
1. k: from 50 to 300
2. m: from 150 to 500
3. density: 1, 0.8, 0.6, 0.4

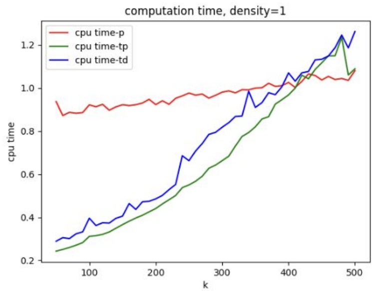


Results (1) : density 1

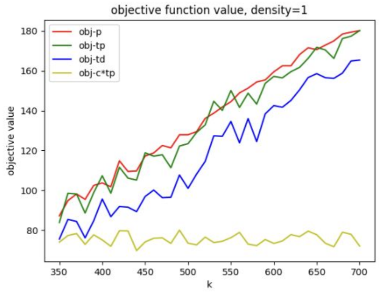
The first setup is: $m = 500, n = 700, k = 50 \to 500, \text{density} = 1$

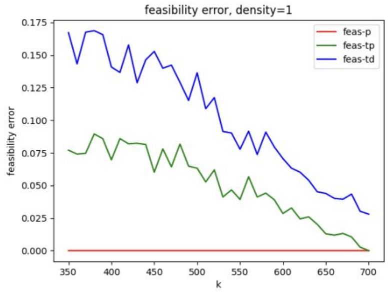


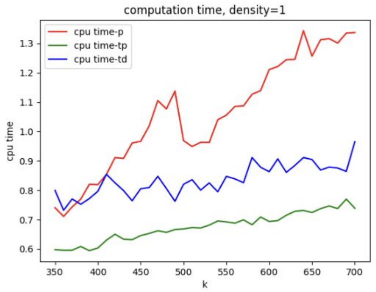




The second setup is: $m = 350 \to 700, n = 700, k = 300, \text{density} = 1$

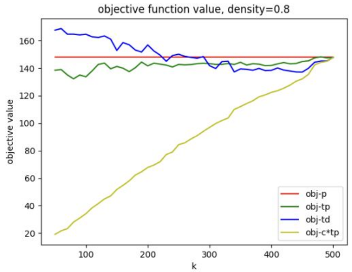


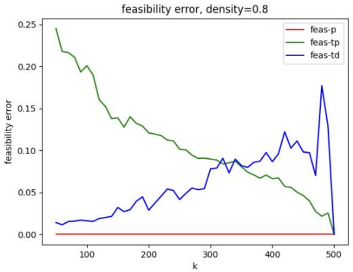


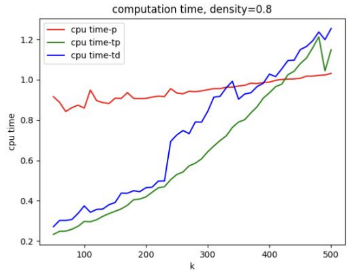


Results (2): density 0.8

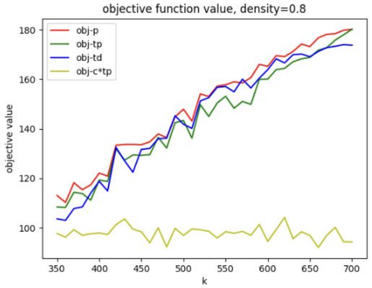
The Third setup is: $m = 500, n = 700, k = 50 \to 500, \text{density} = 0.8$

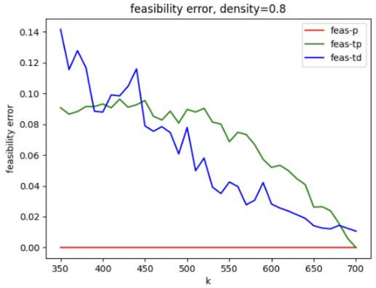


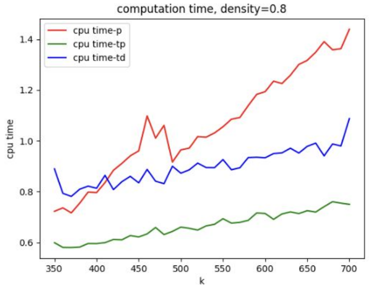




The fourth setup is: $m = 350 \to 700, n = 700, k = 300, \text{density} = 0.8$

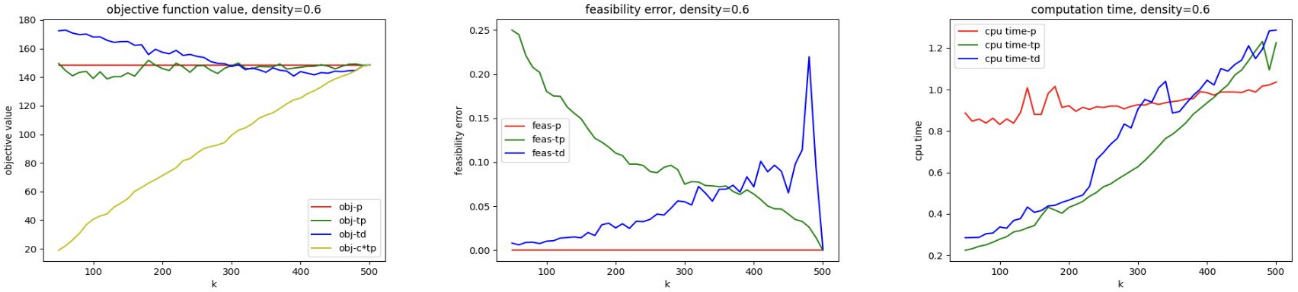




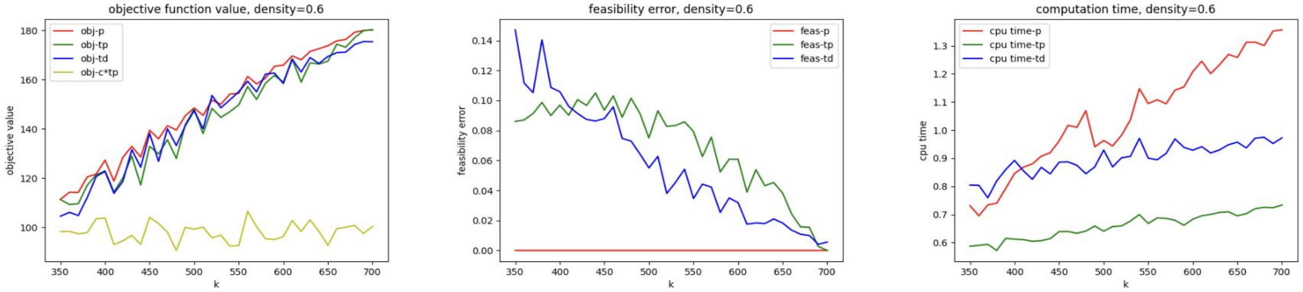


Results (3): density 0.6

The fifth setup is: $m = 500, n = 700, k = 50 \to 500, \text{density} = 0.6$

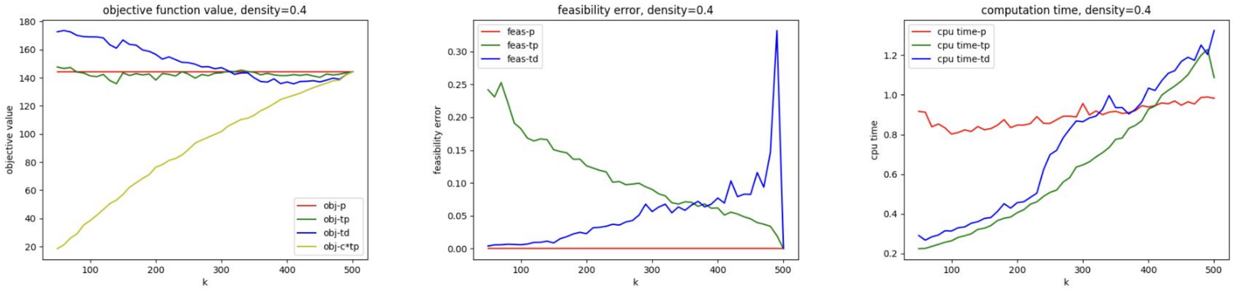


The sixth setup is: $m = 350 \to 700, n = 700, k = 300, \text{density} = 0.6$

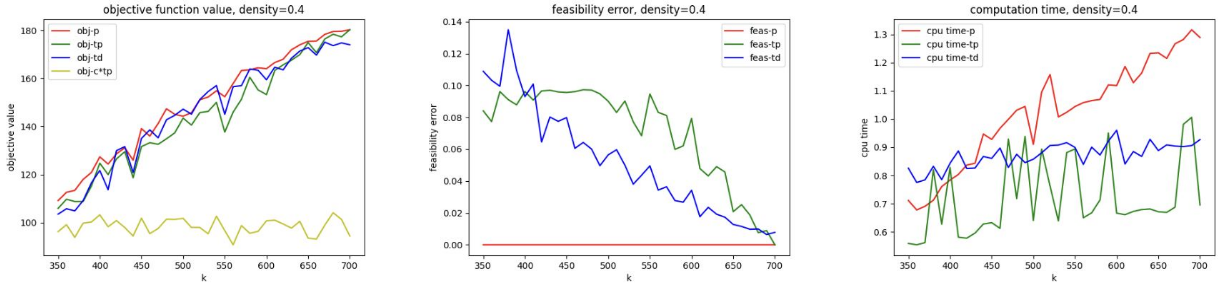


Results (4): density 0.4

The fifth setup is: $m = 500, n = 700, k = 50 \to 500, \text{density} = 0.4$



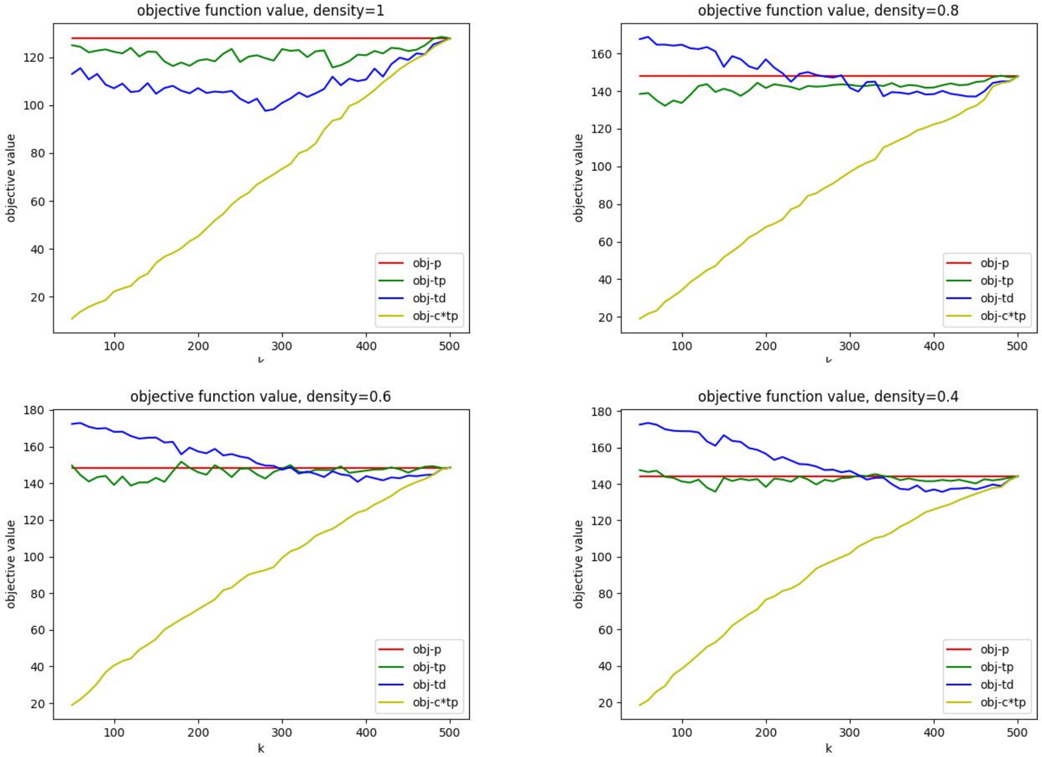
The sixth setup is: $m = 350 \to 700, n = 700, k = 300, \text{density} = 0.4$



Results—interpretation (objctive function value)

m=500,

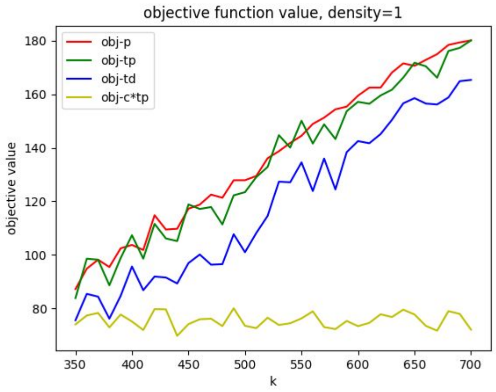
the effect of k:

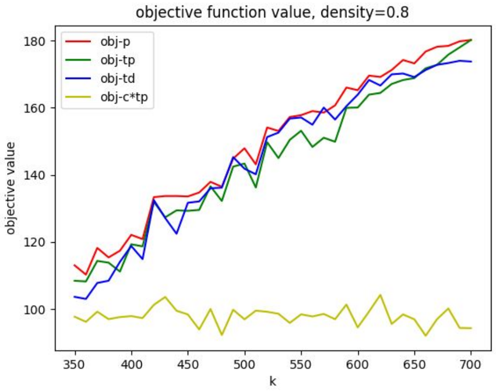


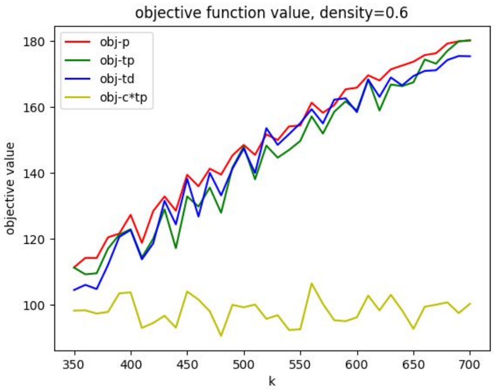
Results—interpretation (objctive function value)

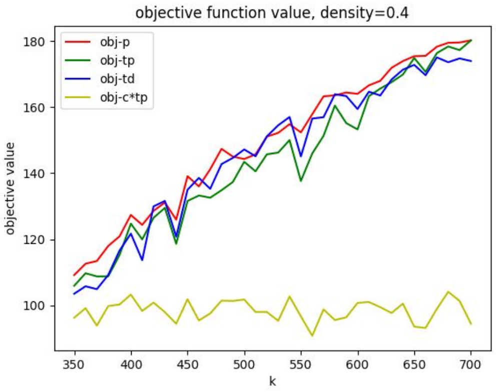
k=300,

the effect of m:







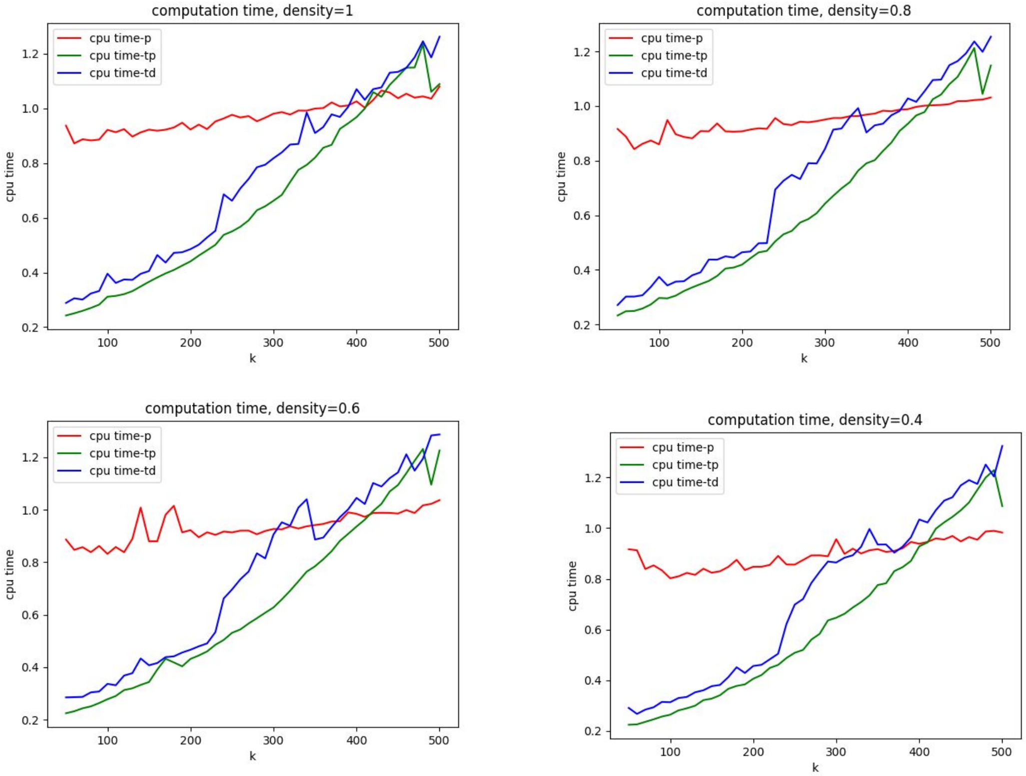


Results—interpretation (feasibility error)

the effect of k and m:

Results—interpretation (computation time)

The effect of k:



Results—interpretation (computation time)

The effect of m:

References

[1] Ky Vu, Pierre-Louis Poirion, and Leo Liberti. Random projections for linear programming. arXiv e-prints, page arXiv:1706.02768, June 2017.