INF580 LARGE SCALE MATHEMATICAL OPTIMIZATION

Project 6 random projection report

March 16, 2022

Yunhao Chen & Ming Cheng

1 INTRODUCTION

In mathematics and statistics, random projection is a technique used to reduce the dimension of a set of points which lie in Euclidean space. Random projection methods are known for their power and simplicity compared to other methods. According to experimental results, random projection preserves distances well. Its theoretical support is the well-known Johnson-Lindenstrauss lemma (JLL). The lemma states that there are random matrices with small amount of rows that approximately preserve Euclidean distances among a set of points.

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 $O(\epsilon^{-2} \ln n)$.

Therefore, any set of n points can be projected to a subspace with dimension logarithmic in n (and the dimension is independent of the original dimension m), such that no pairwise distance is distorted too much.

2 PROBLEM FORMULATION

In this project, the random projection is employed to solve the linear programming(LP) problems. More precisely, we consider LPs in standard form:

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

where A is an $m \times n$ matrix, c is a vector in $\mathbb{R}^n$, b is a vector in $\mathbb{R}^m$, x is in $\mathbb{R}^n$. The dual problem can be written as follow:

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

A random projection $k \times m$ matrix T is sampled from a normal distribution $\frac{1}{\sqrt{k}}\mathcal{N}(0,1)$. The random projection version of the primal problem P can be formulated as follow:

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

Moreover, we can define the dual projection problem:

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

2.1 Solution retrieval

Notation:

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

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

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

Theorem 2 proves that the objective function value of projected problem $v(TP)$ is a good approximation of objective value of primal problem $v(P)$.
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.
Our objective is then to retrieve one solution of primal problem from solution of TP and/or TD. Imagine we have one solution x' of TP, by resolving the equation $A\hat{x} = b - Ax'$, we get a vector $\hat{x}$. Then, we can get an approximate solution x of primal problem:

$$x = x' + \hat{x}$$

Now we introduce how to retrieve a primal solution from TD. Imagine we have a solution of TD, denoted by u. Then $y = T^{\mathrm{T}}u$ is a solution of dual(P). Denote x, one solution of primal problem P. By applying the complementary slackness, we obtain the equations to be satisfied by x and y:

$$y_i(A_i x - b_i) = 0 \ \forall 1 \leq i \leq m$$

$$((A^{\mathrm{T}}y)_j - c_j)x_j = 0 \ \forall 1 \leq j \leq n$$

In practice, it is hard to see that y_i equals exactly to 0, similarly for $((A^{\mathrm{T}}y)_j - c_j)$. Therefore, we have loosened the condition by:

$$|y_i| \geq \epsilon \implies A_i x = b_i \ \forall 1 \leq i \leq m$$

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

The ϵ is a hyperparameter. Here $\epsilon = 1e - 3$.

3
IMPLEMENTATION DETAILS

3.1 Generate LP instances

- The $m \times n$ matrix A is sampled pointwise from the uniform distribution $[-1, 1]$;
- The $n \times 1$ vector c and x are both sampled pointwise from the uniform distribution $[0, 1]$.
- The $k \times m$ matrix T is sampled pointwise from the normal distribution $\mathcal{N}(0, \frac{1}{k})$.

Then, let $b = Ax$. In this way, the feasible linear programming instances are generated, so do their corresponding random projection problems.

3.2 Program introduction

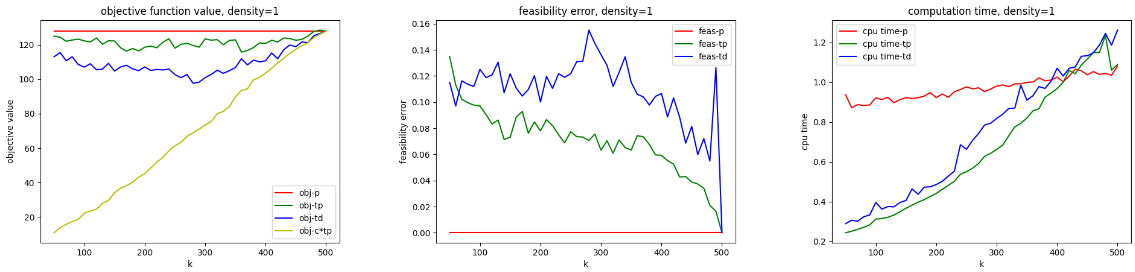
We implemented three ampl programs to solve these three different problems. Every program shares the same data file which is generated automatically by python. These three programs will run one after another. After solving the problem, the ampl program returns the optimal solution by writing a csv file. The python program will then read these solutions and retrieve the primal solution from TP and TD. And the python program will finally return the feasibility error, the optimal objective value and the cpu time (in second). The definition of feasibility error is:

$$\frac{\|Ax - b\|_1}{\|b\|_1}$$

In order to remain the result reproducible, we fix the random seed as 0.
The computer we use has a M1 chip from Apple and 8GB RAM.
The experiment is done in three parts. First, we fix $m = 500$, $n = 700$ and set the sparsity of A as 1 and change the value of k from 150 to 300. During this process, matrix A is not re-sampled. That's why the objective function value remain the same when k change. We then fix $n = 700$, $k = 300$ without changing the sparsity of A, we adjust m (the number of rows in A), from 350 to 700. We choose $k = 300$ because at this point, the objective value of **P**, **TP** and **TD** are very close to each other which can reduce the impact of k. Finally, we repeat the first two steps while changing the sparsity of A, which is 0.8, 0.6 and 0.4. The results are shown as following. For clarity, red lines stand for results of P, green lines stand for results of TP, blue lines stand for results of TD.

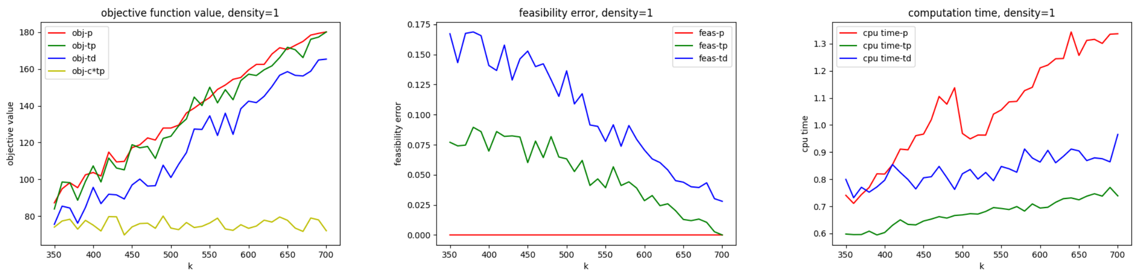
3.2.1 • First Setup

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



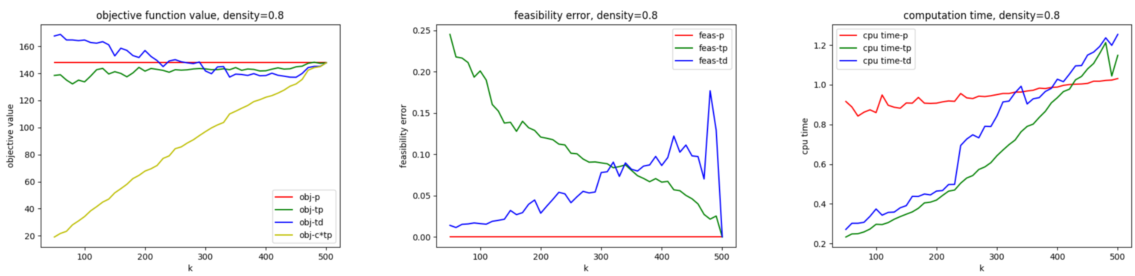
3.2.2 • Second Setup

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



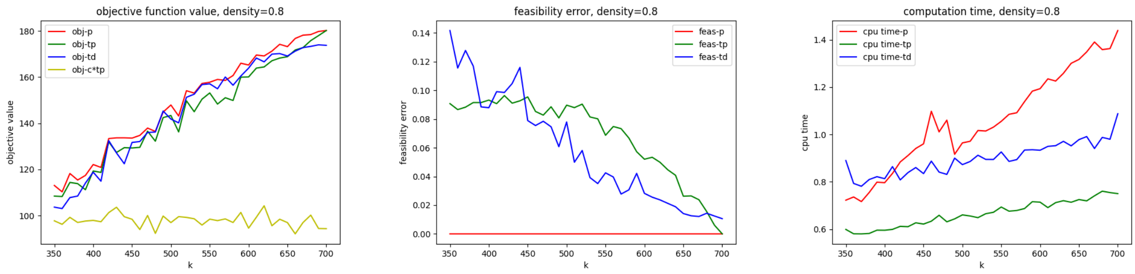
3.2.3 • Third Setup

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



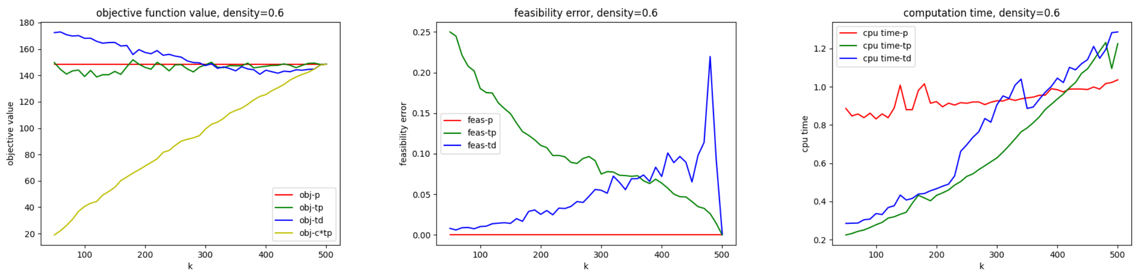
3.2.4 • Fourth Setup

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



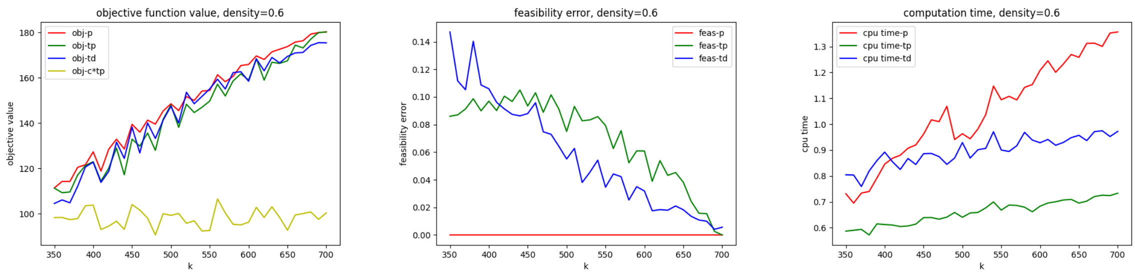
3.2.5 • Fifth Setup

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



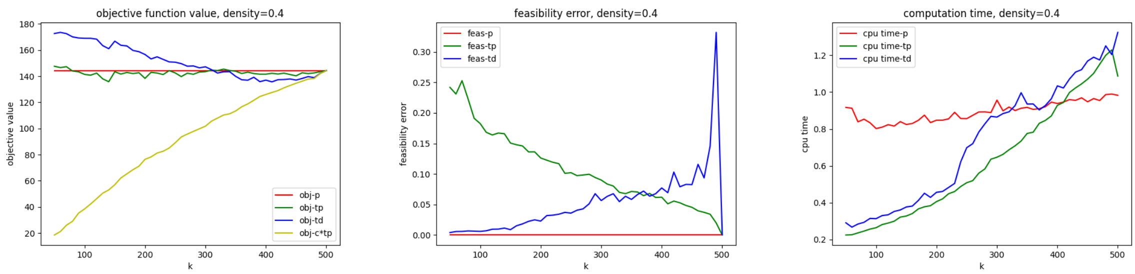
3.2.6 • Sixth Setup

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



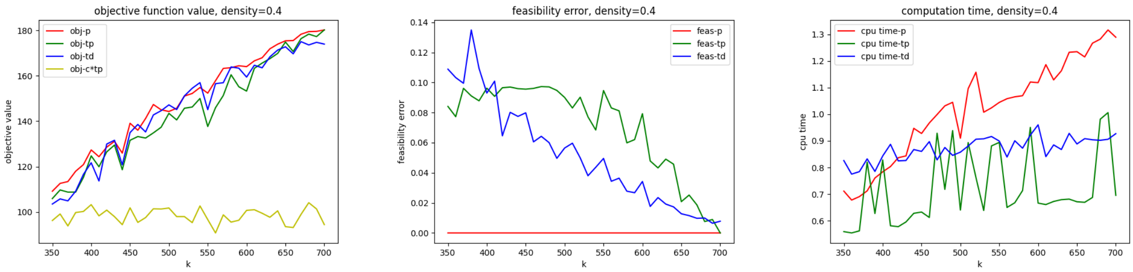
3.2.7 • Seventh Setup

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



3.2.8 • Eighth Setup

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



3.3 Interpretation

The figures give a lot of hints through different aspects. We will discuss them in four dimensions.

3.3.1 • objective values

As we can see, the objective values of problem **P** are almost always larger than the objective values of problem **TP**, no matter the height and the sparsity of the matrix A. This result agrees with the Theorem 2 in section 2.1

When density equals to 1, the objective value of **TD** is always smaller than **TP**'s. This is true due to the propriety of duality:

$$v(\text{dual}) \leq v(\text{primal})$$

However, when sparsity is smaller than 1, the objective values of **TD** and **TP** behave differently. **TD**'s objective value is larger than **P**'s value. By having k augment, **TD**'s objective value decreases until it is smaller than **P**, and **TP**'s value augment until it has almost the same level as **TD** has. They then somehow have the similar behavior and both converge to **P** when $k \to m$.

This result cannot be observed while density = 1.

We also find out that when m augment, the objective values of all these three problems increase. This is a reasonable result. When m increase, which means A is taller, more constraints are added to x, so the object value is higher.

3.3.2 • feasibility error

In all cases, the feasibility error for **P** is zero. It is normal because the primal solution must satisfy the constraints. Moreover, the feasibility error for **TP** is also zero. In the process of solution retrieval from **TP**, we have $A(x' + \hat{x}) = b$. But for solution retrieved from **TD**, the feasibility error is slightly bigger than zero. It is due to the approximation in the retrieval and we did not add the constraint $Ax = b$ in the retrieval process. It decrease while k increase.

So it is reasonable to say that **TP** has fewer error than **TD**.

3.3.3 • computation time

In general, resolve **TP** and **TD** is time-saving compared to the primal problem. We notice that the computation time for **TP** and **TD** has an obvious increasing trend when k increases and finally larger than **P** while **P**'s computation time does not increase (the increment of cpu time might due to load file). And at some point, they finally get larger than **P**.

However, the increment of m shows a much larger impact to the calculation of problem **P** than the other two problems.

We can conclude that the larger m/k is, the more efficient **TP**, **TD** is.

3.3.4 • Sparsity

In general, as density decrease, the cpu time decrease and the objective value augment. This is not an astonishing fact. The larger the sparsity is, the less complex the calculation is, but, the more specific the constraints are which leads to stricter constraints.

REFERENCES

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