Solving the normal equations system arising from interior point methods for linear programming by iterative methods

Aurelio Oliveira - aurelio@ime.unicamp.br

IMECC - UNICAMP

April - 2015

Partners

- Alessandro Coelho
- Carla Ghidini
- Frederico Campos
- Daniele Silva
- Danny Sorensen
- Jair Silva
- Jordi Castro
- Lino Silva
- Luciana Casacio
- Marilene Silva
- Marta Velazco
- Porfirio Suñagua
- Silvana Bocanegra

Summary

- Introduction
- Interior Point Methods
- Approaches for Solving the Linear Systems
- Iterative Methods
- Preconditioning
- Simple Algorithms
- Numerical Experiments
- Conclusions
- Future Works

Introduction

- Sophisticated codes for interior point methods
- Few iterations
- Expensive iterations
- Linear System Solution at each iteration
  - Choice between Augmented System and Normal Equations System
  - Choice between direct methods and iterative methods

Introduction

- Improve iteration time performance
- Reduce the number of iterations
- Specialized methods for a given class of problems

Main Goal

Improve iteration time performance using iterative methods on the normal equations system.

- Conjugate Gradient Method
- Preconditioners
- Simple Algorithms

Interior Point Methods

Standard LP form

$$\begin{aligned} \text{Min} \quad & c^T x \\ \text{s.t.} \quad & Ax = b \\ & x \geq 0, \end{aligned}$$

$A_{m \times n}$ - rank(A)= m
c, b, x column vectors

Interior Point Methods

Dual Problem

$$\begin{aligned} \text{Max} \quad & b^T y \\ \text{s.t.} \quad & A^T y + z = c \\ & z \geq 0, \end{aligned}$$

free variable y and slack dual variable z

Interior Point Methods

Optimality conditions

$$\begin{aligned} Ax - b &= 0, \\ A^T y + z - c &= 0, \\ XZe &= 0, \\ x, z &\geq 0. \end{aligned}$$

$X = diag(x), Z = diag(z)$ and e is the vector of all ones

Primal-dual like Newton´s method

Interior Point Methods

Search Directions

$$\begin{bmatrix} 0 & I & A^T \\ Z & X & 0 \\ A & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} = \begin{bmatrix} r_d \\ r_a \\ r_p \end{bmatrix}$$

$$\begin{aligned} r_d &= c - A^T y - z, \\ r_a &= -XZe, \\ r_p &= b - Ax \end{aligned}$$

Interior Point Methods

Primal-dual Interior Point Methods

Given y^0 and $(x^0, z^0) > 0$.
For $k = 0, 1, 2, \ldots,$ do

(1) Choose $\sigma^k \in [0, 1)$ and set $\mu^k = \sigma^k \left(\frac{(x^k)^t z^k}{n} \right)$.

(2) Solve the given linear system

(3) Choose a step length $\alpha^k = \min(1, \tau^k \rho_p^k, \tau^k \rho_d^k)$ for $\tau^k \in (0, 1)$ where

$$\rho_p^k = \frac{-1}{\min_i \left(\frac{\Delta x_i^k}{x_i^k} \right)} \quad \text{and} \quad \rho_d^k = \frac{-1}{\min_i \left(\frac{\Delta z_i^k}{z_i^k} \right)}.$$

Interior Point Methods

(4) Form the new iterate

$$(x^{k+1}, y^{k+1}, z^{k+1}) = (x^k, y^k, z^k) + \alpha^k (\Delta x^k, \Delta y^k, \Delta z^k).$$

Interior Point Methods

Predictor-Corrector Variant

Affine Direction

$$r_a^k = -X^k Z^k e$$

Final Direction

$$r_m^k = \mu^k e - X^k Z^k e - \Delta\tilde{X}^k \Delta\tilde{Z}^k e$$

Interior Point Methods

Eliminating Δz

$$\begin{bmatrix} -D & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} r_1 \\ r_p \end{bmatrix},$$

$D = X^{-1}Z.$

Augmented System

Eliminating Δx
Normal Equations System $AD^{-1}A^T\Delta y = r$

Linear Systems

- Augmented system
  - Symmetric
  - Indefinite
  - Matrix D
  - Sparse
  - Very Ill-conditioned
- Normal Equations System
  - Symmetric
  - Definite Positive
  - Matrix D
  - Usually Sparse
  - Very Very Ill-conditioned

Linear Systems

- Augmented system
  - LTL^T Factorization
  - Bunch-Parlett Factorization
  - Cholesky type Factorization
  - MINRES
  - SYMMLQ
  - QMR
  - Conjugate Gradient Method
- Normal Equations System
  - Cholesky Factorization
  - Conjugate Gradient Method

    COELHO, OLIVEIRA AND VELAZCO, TEMA, Accepted.

Preconditioning

Preconditioning

Given

$$Bx = b$$

solve

$$(MBN^t)\tilde{x} = \tilde{b}$$

where $\tilde{x} = N^{-t}x$ and $\tilde{b} = Mb$.

Preconditioning

Lemma

Given a Preconditioner for the Schur Complement

$$G(AD^{-1}A^t + E)H^t = GSH^t$$

There Exists an Equivalent Preconditioner for Augmented System

$$\begin{pmatrix} -D^{-\frac{1}{2}} & 0 \\ GAD^{-1} & G \end{pmatrix} \begin{pmatrix} -D & A^t \\ A & E \end{pmatrix} \begin{pmatrix} D^{-\frac{1}{2}} & D^{-1}A^tH^t \\ 0 & H^t \end{pmatrix} =$$

$$\begin{pmatrix} I & 0 \\ 0 & GSH^t \end{pmatrix}$$

Preconditioning

Lemma

Consider the augmented system given by

$$\begin{pmatrix} -D & A^t \\ A & E \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} = \begin{pmatrix} r_y - X^\dagger r_z + V^\dagger r_t \\ r_x + Y^\dagger r_w \end{pmatrix}$$

and its Schur complement $S = AD^{-1}A^t + E$ where D is nonsingular and A is full row rank. Then any symmetric block triangular preconditioner $\begin{pmatrix} H & 0 \\ F & G \end{pmatrix}$ leads to a preconditioned system independent of both: H and F.

Preconditioning

Proof

Modified Augmented System

$$\begin{pmatrix} -HDH^t & B^t \\ B & C \end{pmatrix} \begin{pmatrix} \Delta\tilde{x} \\ \Delta\tilde{y} \end{pmatrix} = \begin{pmatrix} H(r_p - X^{-1} r_c) \\ Fr_p + G(r_d - X^{-1} r_c) \end{pmatrix}$$

where, $B = -FDH^t + GAH^t$ and
$C = -FDF^t + FA^t G^t + GAF^t + GEG^t$.

Modified Schur Complement

$$GSG^t \Delta\tilde{y} = G(r_p + AD^{-1} r_d - AZ^{-1} r_c).$$

OLIVEIRA AND SORENSEN, Linear Algebra and Its Applications, 394 (2005).

Hybrid Preconditioner

- Matrix A full row rank
- Preconditioner based upon a basic solution
- First Iterations: Generic Preconditioner
- Final Iterations: Specially tailored preconditioner
- Middle Iterations

Bocanegra, Campos, and Oliveira, Computational Optimization and Applications, 36 (2007), pp. 149–164.

Bocanegra, Castro, and Oliveira, European Journal of Operational Research, 231 (2013), pp. 263–273.

Controlled Cholesky Factorization

A kind of incomplete Cholesky factorization

- $S = ADA^T = LL^T = \tilde{L}\tilde{L}^T + R$ where
  L complete factorization
  $\tilde{L}$ incomplete
  R remainder matrix
- $E = L - \tilde{L} \Longrightarrow L = E + \tilde{L}$

$$\tilde{L}^{-1}(ADA^T)\tilde{L}^{-T} = (\tilde{L}^{-1}L)(\tilde{L}^{-1}L)^T = (I + \tilde{L}^{-1}E)(I + \tilde{L}^{-1}E)^T$$

$$\tilde{L} \approx L \Longrightarrow E \approx 0 \Longrightarrow \tilde{L}^{-1}(ADA^T)\tilde{L}^{-T} \approx I$$

Controlled Cholesky Factorization

$$\text{minimize } \|E\|_F^2 = \sum_{j=1}^{m} c_j \text{ with } c_j = \sum_{i=1}^{m} |l_{ij} - \tilde{l}_{ij}|^2$$

$$c_j = \sum_{k=1}^{|S_j|+\eta} |l_{i_k j} - \tilde{l}_{i_k j}|^2 + \sum_{k=|S_j|+\eta+1}^{m} |l_{i_k j}|^2$$

- η extra entries allowed per column
- $\tilde{l}_{ij} \approx l_{ij}$ as $|S_j| + \eta$
- $\|E\|_F$ minimized increasing η (c_j decreases) and choosing the largest entries of $\tilde{L}_j$

Controlled Cholesky Factorization

- Generic
- Developed for Symmetric Positive Definite Systems
  - Conjugate Gradient Method
- No Need to Compute $S = ADA^t$
- Allows Drop-out ($\eta < 0$)
- Versatile
  - Diagonal Preconditioner
  - Complete Factorization
- Predictable Storage
- Loss of Positive Definiteness: Exponential Shift

F. CAMPOS *Analysis of Conjugate Gradients - type methods for solving linear equations* PHD THESIS, 1995.

Campos and Birkett, SIAM J. Sci. Comput., 19 (1998.), pp. 126–138.

Splitting Preconditioner

- $A = [B\ N]P$, where P permutation matrix and B nonsingular

$$
\begin{aligned}
ADA^t &= BD_B B^t + ND_N N^t \\
D_B^{-\frac{1}{2}} B^{-1} ADA^t B^{-t} D_B^{-\frac{1}{2}} &= I_m + D_B^{-\frac{1}{2}} B^{-1} ND_N N^t B^{-t} D_B^{-\frac{1}{2}}
\end{aligned}
$$

- Finding a Permutation P
  - Rules for Reordering the Columns of A
    - Diagonal value of D
    - $\|ADe_j\|$ Velazco, Oliveira and Campos, Pesquisa Operacional, 34 (2011), pp. 2553–2561.
  - Compute LU Factorization to Determine the Columns of B
  - Careful Implementation

Ghidini, Oliveira and Sorensen, Annals of Management Science, 3 (2014), pp. 44–66.

Splitting Preconditioner

- Normal Equations System
  - Conjugate Gradient Method
- No Need to Compute $S = ADA^T$
- Works Fine near a Solution
- Designed for Last IP Iterations
- It is not Efficient far from a Solution
  - First IP Iterations
- Hybrid Approach
  - Diagonal Preconditioner
  - Controlled Cholesky Preconditioner
  - Change of Phases

VELAZCO, OLIVEIRA AND CAMPOS, Optimization Methods and Software, 25 (2010), pp. 321–332.

Hydrid Approach

- No Need to Compute ADA^t
- Early Iterations
  - Preconditioned Conjugate Gradient Method
    - Diagonal Preconditioner
    - Controlled Cholesky Decomposition
  - Fast
- Transition
  - MINRES
    - Normal Equations System
    - Indefinite System
  - Robust
- Final Iterations
  - Splitting Preconditioner
  - MINRES
    - Indefinite System
    - Normal Equations System
  - Preconditioned Conjugate Gradient Method
  - Fast

GHIDINI, OLIVEIRA AND SILVA, TEMA, 15 (2014), pp. 1–15.

New Approach to Compute B

- Regularization (penalization) $ADA^t + \delta I$
- Compute LU Factorization of $D^{\frac{1}{2}}A^t$
  - Reorder columns for sparsity
  - Perform row permutation for stability
  - Threshold parameter σ
  - UMFPACK
- More stable that rectangular LU
- May need much more storage

P. SUÑAGUA, *Uma Nova Abordagem para Encontrar uma Base do Precondicionador Separador para Sistemas Lineares no Método de Pontos Interiores* PhD Thesis, 2014.

Simple Algorithms for Linear programming

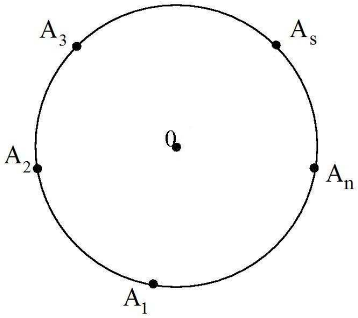
Consider

$$\begin{aligned} Ax &= 0, \\ e^T x &= 1, \\ x &\geq 0, \end{aligned}$$

$A \in R^{m \times n}$ and $||A_j|| = 1$

OBS: Any LP can be reduced to this form

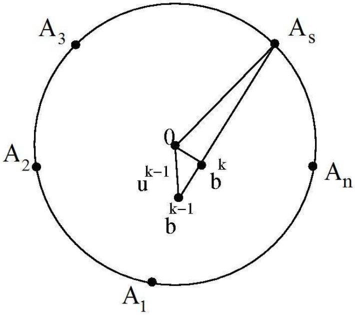
Simple Algorithms for Linear programming



Von Neumann Algorithm

- Simple but with slow convergence
- Find A column A_s with largest angle with the residual b^{k-1}. The next residual is given by the line connecting b^{k-1} to A_s

Von Neumann Algorithm



Adjustment Algorithm for p coordinates

Given: $x^0 \geq 0$, with $e^T x^0 = 1$. Compute $b^0 = Ax^0$.
For $k = 1, 2, 3, ...$ do:

1. Compute:

$\{A_{\eta_1^+}, \dots, A_{\eta_{s_1}^+}\}$ with largest angle with b^{k-1}.
$\{A_{\eta_1^-}, \dots, A_{\eta_{s_2}^-}\}$ with smallest angle with b^{k-1}
such that $x_i^{k-1} > 0, i = \eta_1^-, \dots, \eta_{s_2}^-$
$s_1 + s_2 = p$.
$v^{k-1} = \min_{i=1,\dots,s_1} A_{\eta_i^+}^T b^{k-1}$.

2. If $v^{k-1} > 0$, then **Stop.** The problem is infeasible

Adjustment Algorithm for p coordinates

3. Solve the subproblem

$$\begin{array}{ll} \text{Min} & \|\overline{b}\|^2 \\ \text{s.t.} & \lambda_0 \left(1 - \sum_{i=1}^{s_1} x_{\eta_i^+}^{k-1} - \sum_{j=1}^{s_2} x_{\eta_j^-}^{k-1}\right) + \sum_{i=1}^{s_1} \lambda_{\eta_i^+} + \sum_{j=1}^{s_2} \lambda_{\eta_j^-} = 1, \\ & \lambda_{\eta_i^+} \geq 0, \quad \text{for } i = 1, \ldots, s_1, \\ & \lambda_{\eta_i^-} \geq 0, \quad \text{for } j = 1, \ldots, s_2. \end{array}$$

$$\overline{b} = \lambda_0 \left(b^{k-1} - \sum_{i=1}^{s_1} x_{\eta_i^+}^{k-1} A_{\eta_i^+} - \sum_{j=1}^{s_2} x_{\eta_j^-}^{k-1} A_{\eta_j^-}\right) + \sum_{i=1}^{s_1} \lambda_{\eta_i^+} A_{\eta_i^+} + \sum_{j=1}^{s_2} \lambda_{\eta_j^-} A_{\eta_j^-}.$$

Adjustment Algorithm for p coordinates

4. Update

$$b^k = \lambda_0 \left(b^{k-1} - \sum_{i=1}^{s_1} x^{k-1}_{\eta_i^+} A_{\eta_i^+} - \sum_{j=1}^{s_2} x^{k-1}_{\eta_j^-} A_{\eta_j^-} \right) + \sum_{i=1}^{s_1} \lambda_{\eta_i^+} A_{\eta_i^+} + \sum_{j=1}^{s_2} \lambda_{\eta_j^-} A_{\eta_j^-},$$

$$u^k = ||b^k||,$$

$$x_j^k = \begin{cases} \lambda_0 x_j^{k-1}, & j \notin \{\eta_1^+, \ldots, \eta_{s_1}^+, \eta_1^-, \ldots, \eta_{s_2}^-\}, \\ \lambda_{\eta_i^+}, & j = \eta_i^+, i = 1, \ldots, s_1, \\ \lambda_{\eta_j^-}, & j = \eta_j^-, j = 1, \ldots, s_2. \end{cases}$$

$$k = k + 1.$$

Subproblem Solution

For $p = 2$: small number of cases in KKT conditions. For any p:

$$1 + 2C_1^p + 2C_2^p + 2C_3^p + \ldots + 2C_p^p = 2^{(p+1)} - 1.$$

Solution: Apply interior point methods

Small dimension problem if p is small.

GHIDINI, OLIVEIRA, SILVA AND VELAZCO, Linear Algebra and its Applications, 218 (2012), pp. 1267–1284.

Subproblem - Interior Point Method

$$\begin{array}{ll} \text{Min} & \frac{1}{2}||W\lambda||^2 \\ \text{s.t.} & c^T\lambda = 1, \\ & -\lambda \leq 0, \end{array}$$

$$W = \left[\overline{w}\, A_{\eta_1^+} \dots A_{\eta_{s_1}^+}\, A_{\eta_1^-} \dots A_{\eta_{s_2}^-}\right],$$

$$\overline{w} = b^{k-1} - \sum_{i=1}^{s_1} x_{\eta_i^+}^{k-1} A_{\eta_i^+} - \sum_{j=1}^{s_2} x_{\eta_j^-}^{k-1} A_{\eta_j^-},$$

$$\lambda = \left(\lambda_0, \lambda_{\eta_1^+}, \dots, \lambda_{\eta_{s_1}^+}, \dots, \lambda_{\eta_1^-}, \dots, \lambda_{\eta_{s_2}^-}\right),$$

$$c = (c_1, 1, \dots, 1),\, c_1 = 1 - \sum_{i=1}^{s_1} x_{\eta_i^+}^{k-1} - \sum_{j=1}^{s_2} x_{\eta_j^-}^{k-1}.$$

Subproblem - Interior Point Method

Solve the Linear System at each subproblem iteration

$$\begin{pmatrix} W^TW & c & -I \\ U & 0 & \Lambda \\ c^T & 0 & 0 \end{pmatrix} \begin{bmatrix} \Delta\lambda \\ \Delta I \\ \Delta\mu \end{bmatrix} = \begin{bmatrix} r_1 \\ r_2 \\ r_3 \end{bmatrix}$$

where

$$\begin{aligned} U &= diag(\mu)\ , \\ \Lambda &= diag(\lambda)\ , \\ r_1 &= \mu - cI - W^TW\lambda\ , \\ r_2 &= -I^T\lambda\ , \\ r_3 &= 1 - c^T\lambda\ . \end{aligned}$$

Subproblem - Interior Point Method

- Search Directions:

$$\Delta\mu = \Lambda^{-1} r_2 - \Lambda^{-1} U d\lambda,$$
$$\Delta\lambda = (W^T W + \Lambda^{-1} U)^{-1} r_4 - (W^T W + \Lambda^{-1} U)^{-1} c \Delta l,$$
$$c^T (W^T W + \Lambda^{-1} U)^{-1} c \Delta l = c^T (W^T W + \Lambda^{-1} U)^{-1} r_4 - r_3$$

- Linear Systems

$$(W^T W + \Lambda^{-1} U) v1 = c$$
$$(W^T W + \Lambda^{-1} U) v2 = r_4$$
$$r_4 = r_1 + \Lambda^{-1} r_2$$

Problems NETLIB – QAP – Kennington – Total: 189

Medium Problems			Large Problems
BL	fit1p	pds-30	ken18
BL2	fit2p	pilot87	kra30a[1]
chr22b	GE	qap12	kra30b[1]
chr25a	greenbeb	qap15	nug07-3rd
CO5	ken13	rou20	nug20[1]
CO9	NL	scr20	pds-70
CQ9	nug06-3rd	stocfor3	pds-80
cre-b	nug12		pds-90
cre-d	nug15		pds-100
dfl001	osa-60		ste36a[1]
els19	pds-10		ste36b[1]
ex09	pds-20		ste36c[1]

[1] Need large storage

Running Time (sec.)

Problem	LUrec	LUrec+Reg	LUstd	LUstd+Reg
pds-70	1350.78	1440.13	1460.42	1431.93
ken18	1181.18	1196.53	1573.41	1438.93
pds-80	1705.75	1860.69	1744.39	1872.91
pds-90	2140.66	2106.1	2143.03	2126.16
pds-100	2896.42	3099.77	3088.34	3153.13
kra30a			6693,08	6698.49
ste36a	14072.58	14563.07	7298.44	6911.57
ste36b			9497.97	8305.62
kra30b			6537.62	9326.45
ste36c			11286.5	9986.89
nug20			12245.43	10651.28
Remaining	2191.73	6060.74	3693.04	3591.50
Total	25539.1	30327.03	67261.67	65494.86
Common	25473.5	26249.59	19119.04	18726.05

Processing Time

	LUrec	LUrec+Reg	LUstd	LUstd+Reg
Solved	189	189	189	189
Problems	121	119	149	163
	64.02%	62.96%	78.84%	86.24%
not solved	68	70	40	26
	35.98%	37.04%	21.16%	13.76%
Total (sec)	25539.1	30327.03	67261.67	65494.86
109 Prob.	25473.5	26249.59	19119.04	18726.05
121 Prob.	25539.1			18793.51

Difference: $25539.1 - 18793.51 \approx 2h$.

LUstd more efficient then LUrec

Conclusions

- Iterative Methods
- Very large problems
- Preconditioners
- Normal Equations $\times$ Augmented System
- Sophisticated implementations
- Fast and Robust

Future Work

- Heuristics for changing Preconditioners
- More robust iterative methods in the change of phases
- New simple methods
- Augmented system
- New Preconditioners