The nonnegative least squares problem

Yueh-Cheng Kuo

Department of Applied Mathematics
National University of Kaohsiung, Taiwan

Joint work with
Ching-Sung Liu (National University of Kaohsiung).

Jan. 18, 2022

Outline

1. Introduction
2. Index search Method
3. Numerical experiments

The nonnegative least squares problem

The nonnegative least squares (NNLS) problem is given by

$$\min_{\mathbf{x}\geq 0} \frac{1}{2}\|A\mathbf{x}-\mathbf{b}\|^2, \tag{1}$$

where $\mathbf{b}\in\mathbb{R}^n$, $A\in\mathbb{R}^{n\times m}$, $\text{rank}(A)=m$. The NNLS problem (1) can be regarded as a quadratic optimization problem:

$$\min_{\mathbf{x}\geq 0} f(\mathbf{x}) = \min_{\mathbf{x}\geq 0} (\frac{1}{2}\mathbf{x}^T A^T A\mathbf{x} - \mathbf{x}^T A^T \mathbf{b}). \tag{2}$$

- This is a convex optimization problem. Hence, it has a unique optimal solution.
- **Applications:** The Nonnegative Matrix Factorization (NMF), text mining, support vector machines, $\cdots$, etc.

Methods for solving the NNLS problem

Active-set methods:

- lsqnonneg [1] in Matlab: This algorithm is a single principal pivoting algorithm, and it is proved that the iteration always converges.
- Fast NNLS (fnnls) [2]: It speeds the lsqnonneg by avoiding unnecessary computations.
- Block principal pivoting algorithm (blocknnls(p)) [3,4]: This algorithm is to control the number of infeasibilities set.
- Index search method (ISM): This algorithm is to control the objective value.

Iterative method:

- lsqlin [5]: The Matlab function lsqlin is the interior-point method that can be used to solve NNLS.

[1] C. L. Lawson and R. J. Hanson, Solving least squares problems, Philadelphia, Pa., Society for Industrial and Applied Mathematics, 1995.

[2] R. Bro and S. D. Jong, *A fast non-negativity-constrained least squares algorithm*, Journal of Chemometrics. Vol. 11, pp. 393–401, 1997.

[3] L. F. Portugal, J. J. Judice, and L. N. Vicente, *A comparison of block pivoting and interior-point algorithms for linear least squares problems with nonnegative variables*, Mathematics of Computation, Vol. 63, No. 208, pp. 625–643, 1994

[4] J. Cantarella, M. Piatek, *Tsnnls: A solver for large sparse least squares problems with non-negative variables*, ArXiv Computer Science e-prints, 2004.

[5] T. F. Coleman and Y. Li, *A reflective Newton method for minimizing a quadratic function subject to bounds on some of the variables*, SIAM Journal on Optimization, Vol. 6, No. 4, pp. 1040–1058, 1996.

The Lagrange dual problem

- The Lagrangian is

$$L(\mathbf{x}, \boldsymbol{u}) \equiv f(\mathbf{x}) - \boldsymbol{u}^T\mathbf{x} = \frac{1}{2}\mathbf{x}^T A^T A\mathbf{x} - \mathbf{x}^T A^T \mathbf{b} - \boldsymbol{u}^T\mathbf{x}. \tag{3}$$

  The dual function is $g(\boldsymbol{u}) = \min_{\mathbf{x}\in\mathbb{R}^m}\{L(\mathbf{x}, \boldsymbol{u})\}$.
- The Lagrange dual problem of (2) is

$$\max_{\boldsymbol{u}\geq 0} g(\boldsymbol{u}) = \max_{\boldsymbol{u}\geq 0} -\frac{1}{2}(A^T\mathbf{b} + \boldsymbol{u})^T(A^TA)^{-1}(A^T\mathbf{b} + \boldsymbol{u}).$$

- Duality gap

$$g(\boldsymbol{u}) \leqslant \frac{1}{2}\mathbf{x}^T A^T A\mathbf{x} - \mathbf{x}^T A^T\mathbf{b} - \mathbf{x}^T\boldsymbol{u} \leqslant f(\mathbf{x}).$$

Karush-Kuhn-Tucker (KKT) conditions

The Karush-Kuhn-Tucker (KKT) conditions of problem (2) are

(i) $A^T A\mathbf{x}_* - A^T \boldsymbol{b} - \boldsymbol{u}_* = 0$ (stationarity);

(ii) $\mathbf{x}_* \geq 0$ (primal feasibility);

(iii) $\mathbf{x}_*(i)\boldsymbol{u}_*(i) = 0$ for any i (complementary slackness);

(iv) $\boldsymbol{u}_* \geq 0$ (dual feasibility).

- The primal problem (2) is convex, the KKT conditions are necessary and sufficient conditions for the points to be primal and dual optimal.

Index search method for $\mathbf{x}_* = \text{argmin}_{\mathbf{x}\in\mathbb{R}^m} \|A\mathbf{x} - \mathbf{b}\|$

- Let $\mathbf{x} \in \mathbb{R}^m$ and $\mathtt{I}_{\mathbf{x}} = \{i \in \{1, \cdots, m\} \mid \mathbf{x}(i) > 0\}$.
- Let $\mathtt{I} \subseteq \{1, \cdots, m\}$, $A_{\mathtt{I}} = A(:, \mathtt{I})$. Denote

$$\mathcal{I} = \{\mathtt{I} \subseteq \{1, \cdots, m\} \mid (A_{\mathtt{I}}^T A_{\mathtt{I}})^{-1} A_{\mathtt{I}}^T \mathbf{b} > 0\}. \tag{4}$$

- The desired index set $\mathtt{I}_* \equiv \mathtt{I}_{\mathbf{x}_*} \in \mathcal{I}$. Then

$$\mathbf{x}_* \equiv P_{\mathtt{I}_*} \begin{bmatrix} \mathbf{x}_*(\mathtt{I}_*) \\ \mathbf{x}_*(\mathtt{I}_*^c) \end{bmatrix} = P_{\mathtt{I}_*} \begin{bmatrix} (A_{\mathtt{I}_*}^T A_{\mathtt{I}_*})^{-1} A_{\mathtt{I}_*}^T \mathbf{b} \\ 0 \end{bmatrix}, \tag{5}$$

  where $P_{\mathtt{I}_*}$ is a permutation matrix.
- $\boldsymbol{u}_* = A^T A\mathbf{x}_* - A^T \boldsymbol{b} \geq 0$ (dual feasibility).
- Our goal of this paper is to find the desired index set $\mathtt{I}_*$.

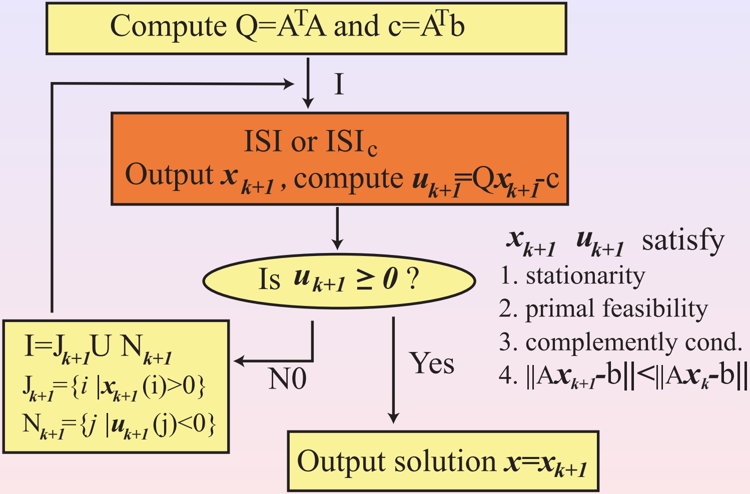
Index search method for $\mathbf{x}_* = \text{argmin}_{\mathbf{x}\geqslant 0}\|A\mathbf{x} - \mathbf{b}\|$

Now, we consider a new optimization problem:

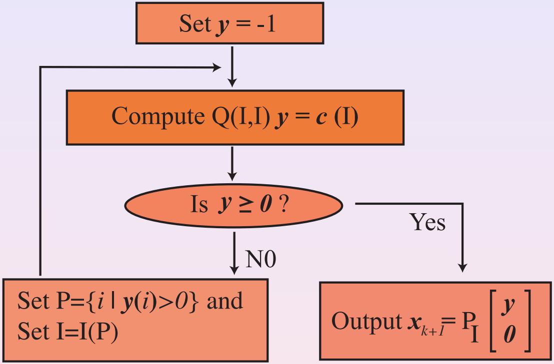
$$\mathtt{I}_* = \text{argmin}_{\mathtt{I}\in\mathcal{I}} \min_{\boldsymbol{y}\in\mathbb{R}^{|\mathtt{I}|}} \|A_{\mathtt{I}}\mathbf{y} - \mathbf{b}\|, \tag{6}$$

where $\mathcal{I}$ is defined in (4) and $A_{\mathtt{I}} = A(:, \mathtt{I})$.

Flowchart of Index search method (ISM)



Flowchart of Index search iteration (ISI) $\mathbf{x}_{k+1} = ISI(Q, \mathbf{c}, \mathtt{I})$



- Output vector $\mathbf{x}_{k+1}$ and the vector $\boldsymbol{u}_{k+1} = A^T A\mathbf{x}_{k+1} - A^T\mathbf{b}$ satisfy stationarity, primal feasibility, and complementary slackness.
- Does the inequality $\|A\mathbf{x}_k - \mathbf{b}\| < \|A\mathbf{x}_{k+1} - \mathbf{b}\|$ hold?. Ans: No.

Idea of ISI with convex combination (ISI_c)

Step1: Find a point $\hat{\mathbf{x}}$ with $\hat{\mathbf{x}}(\mathtt{I}) > 0$ and $\hat{\mathbf{x}}(\mathtt{I}^c) = 0$ such that

$$\|A\hat{\mathbf{x}} - \mathbf{b}\| < \|A\mathbf{x}_k - \mathbf{b}\|.$$

Step2: Take the $\hat{\mathbf{x}}$ obtained in **Step1** as an initial vector, and use active-set algorithm to get the primal feasible point $\mathbf{x}$.

- Note that $\hat{\mathbf{x}} = \mathbf{x}_k + \theta\mathbf{e}$, where

$$\theta = \frac{-\mathbf{e}^T\mathbf{u}_k}{\mathbf{e}^T A^T A\mathbf{e}}, \quad \mathbf{e} = P_{\mathtt{N}_k}\begin{bmatrix} \mathbf{1} \\ 0 \end{bmatrix} \text{ and } \mathtt{N}_k = \{i \in \mathbb{N} | \boldsymbol{u}_k(i) < 0\}.$$

ISI with convex combination: $\mathbf{x} = \mathrm{ISI}_c(Q, \mathbf{c}, \hat{\mathbf{x}})$

Given $\hat{\mathbf{x}} \geq 0 \in \mathbb{R}^m$, $Q = A^T A$ and $\mathbf{c} = A^T \mathbf{b}$.

Let $\mathbf{v} = \hat{\mathbf{x}}$ and $\mathtt{I} = \{i \in \mathbb{N} | \hat{\mathbf{x}}(i) > 0\}$. Solve $Q(\mathtt{I}, \mathtt{I})\mathbf{y} = \mathbf{c}(\mathtt{I})$.

Find an index $\mathtt{S} = \{i \in \mathbb{N} | \mathbf{y}(i) < 0\}$.

while $\mathtt{S} \neq \emptyset$

  Solve $[\alpha, p] = \min\{\frac{\mathbf{v}(\mathtt{I}(\mathtt{S}))}{\mathbf{v}(\mathtt{I}(\mathtt{S})) - \mathbf{y}(\mathtt{S})}\}$.

  Compute $\mathbf{v}(\mathtt{I}(\mathtt{S})) = (1 - \alpha)\mathbf{v}(\mathtt{I}(\mathtt{S})) + \alpha \mathbf{y}(\mathtt{S})$.

  Set $\mathtt{I} = \mathtt{I} \setminus \{\mathtt{I}(\mathtt{S}(p))\}$.

  Solve $Q(\mathtt{I}, \mathtt{I})\mathbf{y} = \mathbf{c}(\mathtt{I})$.

  Find an index $\mathtt{S} = \{i \in \mathbb{N} | \mathbf{y}(i) < 0\}$.

end

Set $\mathtt{J} = \mathtt{I}$ and $\mathbf{x} = P_{\mathtt{J}} \begin{bmatrix} \mathbf{y} \\ 0 \end{bmatrix}$.

ISI with convex combination: $\mathbf{x} = \mathrm{ISI}_c(Q, \mathbf{c}, \hat{\mathbf{x}})$

Theorem

Given an $\hat{\mathbf{x}} \geq 0 \in \mathbb{R}^m$ *with* $\hat{\mathbf{x}}(\mathtt{I}) > 0$ *and* $\hat{\mathbf{x}}(\mathtt{I}^c) = 0$. *Assume that* $(A_{\mathtt{I}}^T A_{\mathtt{I}})\hat{\mathbf{x}}(\mathtt{I}) \neq A_{\mathtt{I}}^T \boldsymbol{b}$ *and* $\mathbf{x} = \mathrm{ISI}_c(Q, \mathbf{c}, \hat{\mathbf{x}})$. *Let* $\mathbf{u} = A^T A\mathbf{x} - A^T\mathbf{b}$. *Then*

i $\mathbf{x} \geq 0$ *(primal feasibility);*

ii $\mathbf{x}(i)\mathbf{u}(i) = 0$ *for all* i *(complementary slackness);*

iii $\mathbf{u} = A^T A\mathbf{x} - A^T\mathbf{b}$ *(stationarity);*

iv $\|A\mathbf{x} - \mathbf{b}\| < \|A\hat{\mathbf{x}} - \mathbf{b}\|$.

Index search method

Algorithm 3.3 Index search method (Main algorithm)

Given $\mathbf{x}_0 = 0$ and $\mathsf{tol} > 0$. Compute $\mathbf{r}_0 = A\mathbf{x}_0 - \mathbf{b}$ and $\mathbf{u}_0 = A^T\mathbf{r}_0$.
Compute $Q = A^TA$ and $\mathbf{c} = A^T\mathbf{b}$.
For $k = 0, 1, 2, \ldots$
  Set $\mathtt{J}_k = \{i \in \mathbb{N} \mid \mathbf{x}_k(i) > 0\}$, $\mathtt{N}_k = \{i \in \mathbb{N} \mid \mathbf{u}_k(i) < 0\}$ and $\mathtt{I}_{k+1} = \mathtt{J}_k \cup \mathtt{N}_k$.
  Do the index search iteration: $\mathbf{x} = \mathrm{ISI}(Q, \mathbf{c}, \mathtt{I}_{k+1})$.
  Compute $\mathbf{r} = A\mathbf{x} - \mathbf{b}$.
  If $\|\mathbf{r}\| < \|\mathbf{r}_k\|$
    set $\mathbf{x}_{k+1} = \mathbf{x}$, $\mathbf{r}_{k+1} = \mathbf{r}$, and $\mathbf{u}_{k+1} = A^T\mathbf{r}_{k+1}$
  else
    set $\mathbf{e} = P_{\mathtt{N}_k}\begin{bmatrix} \mathbf{1} \\ 0 \end{bmatrix}$, where $\mathbf{1} = [1, \ldots, 1]^T \in \mathbb{R}^{|\mathtt{N}_k|}$.
    compute $\theta = (-\mathbf{e}^T\mathbf{u}_k)/\|A\mathbf{e}\|^2$ and $\hat{\mathbf{x}} = \mathbf{x}_k + \theta\mathbf{e}$.
    compute $\mathbf{x}_{k+1} = \mathrm{ISI}_c(Q, \mathbf{c}, \hat{\mathbf{x}})$.
    $\mathbf{r}_{k+1} = A\mathbf{x}_{k+1} - \mathbf{b}$ and $\mathbf{u}_{k+1} = A^T\mathbf{r}_{k+1}$.
  endIf
until $|\min(\mathbf{u}_{k+1})| \leq \mathsf{tol}$.(seek dual feasibility)

Numerical results

In the following numerical results,

- "Iter" is the number of iterations to achieves the optimal solution.
- "$\#\mathrm{Neqs}_1$" is the total number of solving normal equations in ISI.
- "$\#\mathrm{Neqs}_2$" is the total number of solving normal equations in ISI_c.
- "$\#\mathrm{Neqs} = \#\mathrm{Neqs}_1 + \#\mathrm{Neqs}_2$"
- "Time" is the execution time in second of algorithm.

Example 1

Consider a sparse matrix $A \in \mathbb{R}^{n \times m}$ and a sparse vector $\mathbf{b} \in \mathbb{R}^n$, where A and $\mathbf{b}$ are randomly generated by the Matlab function sprandn(n,m,ρ).

Table: Numerical results for $n = 10000$ in terms of $\rho = 10^{-2}$ and 10^{-3}.

	$\rho = 10^{-2}$				$\rho = 10^{-3}$			
m	Iter	$\#\mathrm{Neqs}_1$	$\#\mathrm{Neqs}_2$	Time	Iter	$\#\mathrm{Neqs}_1$	$\#\mathrm{Neqs}_2$	Time
500	3	5	0	0.016	7.1	12.6	3.5	0.009
1000	3.1	5.7	0	0.057	7.5	12.9	0.3	0.013
3000	4.1	9.3	0	0.671	6.2	13.4	0	0.081
5000	4.7	11.6	0	2.407	6.5	14.8	0	0.403
9000	5.5	13.06	0	13.06	7.3	19.2	0	2.328

Example 1

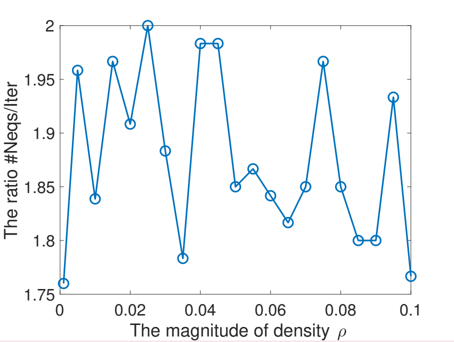


Figure: The magnitude of density ρ versus the ratio $\frac{\#\text{Neqs}}{\text{Iter}}$ for $m = 1000$.

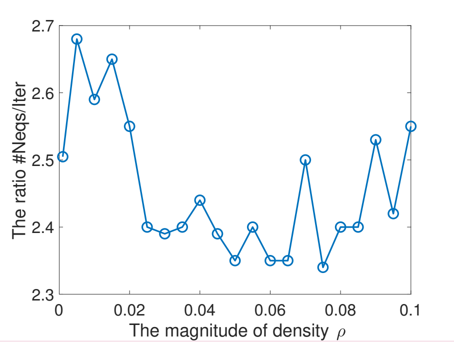


Figure: The magnitude of density ρ versus the ratio $\frac{\#\text{Neqs}}{\text{Iter}}$ for $m = 5000$.

Example 1

Table: Numerical results for $m = 5000$.

Algorithms	#Neqs for $n = 10^4$			#Neqs for $n = 10^5$		
	$\rho = 10^{-1}$	$\rho = 10^{-2}$	$\rho = 10^{-3}$	$\rho = 10^{-2}$	$\rho = 10^{-3}$	$\rho = 10^{-4}$
ISM	9.9	11.9	16.1	6.0	7.7	78.6
ISM_2	9.9	11.9	16.1	6.0	7.7	16.1
lsqnonneg	2468	2497	2511.4	2451.3	2523.4	$>$3000
fnnls	2468	2497	2511.4	2451.3	2523.4	$>$3000
blocknnls(3)	4.8	5.1	600.6	3.1	4.0	$>$3000
blocknnls(10)	4.8	5.1	7.4	3.1	4.0	265.5

Example 2

Consider a sparse matrix $A \in \mathbb{R}^{n\times m}$ and a sparse vector $\mathbf{b} \in \mathbb{R}^n$, where A and $\mathbf{b}$ are randomly generated by the Matlab function sprand(n,m,ρ)

Table: Numerical results for $n = 10000$ in terms of $\rho = 10^{-2}$ and 10^{-3}.

m	$\rho = 10^{-2}$				$\rho = 10^{-3}$			
	Iter	$\#\mathrm{Neqs}_1$	$\#\mathrm{Neqs}_2$	Time	Iter	$\#\mathrm{Neqs}_1$	$\#\mathrm{Neqs}_2$	Time
500	1	3.8	0	0.015	1	1	0	0.002
1000	1.1	4.2	0	0.045	1	1	0	0.002
3000	1.3	5.8	0	0.413	1	1.3	0	0.005
5000	1.7	6.9	0	1.134	1	1.8	0	0.010
9000	2.7	9.7	0	4.479	1	2.1	0	0.024

Example 2

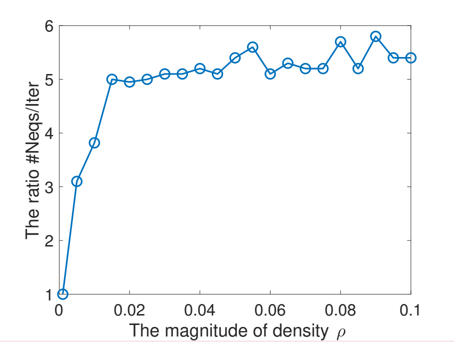


Figure: The magnitude of density ρ versus the ratio $\frac{\#\mathrm{Neqs}}{\mathrm{Iter}}$ for $m = 1000$.

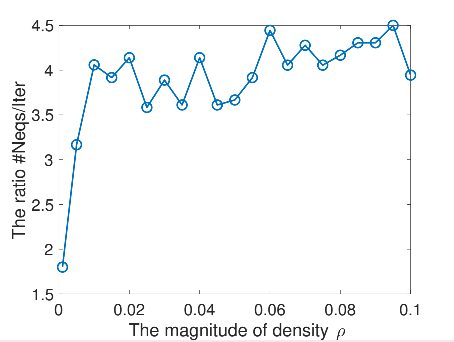


Figure: The magnitude of density ρ versus the ratio $\frac{\#\mathrm{Neqs}}{\mathrm{Iter}}$ for $m = 5000$.

Example 2

Table: Numerical results for $n = 10^4$.

Algorithms	#Neqs for $m = 4000$			#Neqs for $m = 8000$		
	$\rho = 10^{-1}$	$\rho = 10^{-2}$	$\rho = 10^{-3}$	$\rho = 10^{-1}$	$\rho = 10^{-2}$	$\rho = 10^{-3}$
ISM	8.8	6.5	1.7	10.8	9.1	2.1
ISM_2	8.8	6.5	1.7	10.8	9.1	2.1
lsqnonneg	246.6	372.6	36.3	287.7	445.4	72.0
fnnls	246.6	372.6	36.3	287.7	445.4	72.0
blocknnls(3)	7.2	6.0	62.5	7.9	7.1	2101
blocknnls(10)	7.2	6.0	2.7	7.9	7.1	4.8

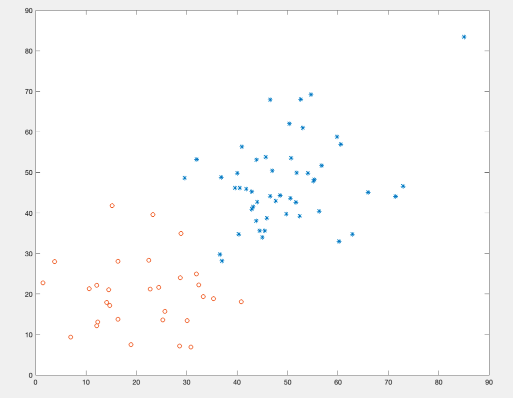
Example 3

Consider a matrix $A \in \mathbb{R}^{n\times m}$ and a vector $\mathbf{b} \in \mathbb{R}^n$, where A and $\mathbf{b}$ are randomly generated by the Matlab function rand(n,m) or randn(n,m).

Table: Numerical results for $n = 10^4$.

	Time for the case $A, \mathbf{b} \geqslant 0$				Time for the case $A, \mathbf{b} \ngeqslant 0$			
m	100	1000	3000	7000	100	1000	3000	7000
ISM	0.005	0.09	0.72	4.21	0.005	0.09	0.79	4.94
ISM_2	0.005	0.09	0.72	4.21	0.005	0.09	0.79	4.94
lsqlin	0.025	0.22	2.66	25.31	0.026	0.24	3.14	28.42

Example 4 Support vector machine



$$
\begin{aligned}
&\text{minimize } \tfrac{1}{2}\|\mathbf{w}\|^2\\
&\text{subject to}\\
&\qquad \mathbf{w}^\top \boldsymbol{u}_i - b \geqslant 1, \quad i = 1, \cdots, p\\
&\qquad -\mathbf{w}^\top \boldsymbol{v}_j + b \geqslant 1, \quad j = 1, \cdots, q\\
&\qquad \mathbf{w} \in \mathbb{R}^n.
\end{aligned}
$$

Example 4 Support vector machine

Let

$$X = [-\boldsymbol{u}_1, \ldots, -\boldsymbol{u}_p, \boldsymbol{v}_1, \ldots, \boldsymbol{v}_q] \in \mathbb{R}^{n\times(p+q)}, \quad \boldsymbol{c} = \begin{bmatrix} \mathbf{1}_p \\ -\mathbf{1}_q \end{bmatrix}.$$

- KKT condition:

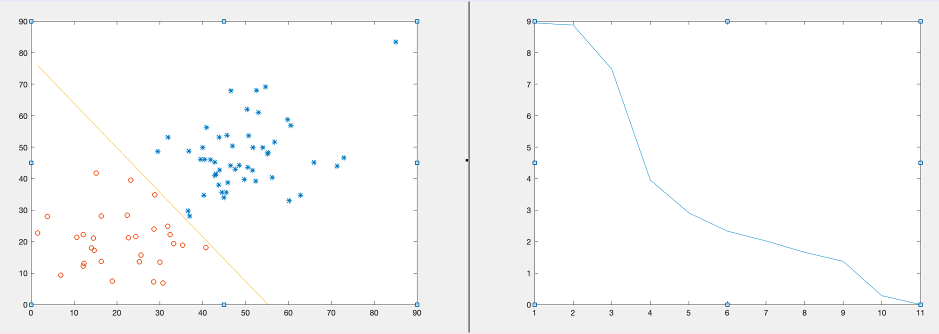
$$\begin{aligned} &-X^\top X \begin{bmatrix} \lambda \\ \mu \end{bmatrix} + b \begin{bmatrix} \mathbf{1}_p \\ -\mathbf{1}_q \end{bmatrix} + \mathbf{1}_{p+q} \leqslant \mathbf{0}_{p+q}, \\ &\textstyle\sum_{j=1}^q \mu_j - \sum_{i=1}^p \lambda_i = 0, \\ &\lambda \geqslant \mathbf{0}, \quad \mu \geqslant \mathbf{0}. \end{aligned}$$

- KKT condition ie equivalent to find $\min_{\mathbf{x}\geq 0} \frac{1}{2}\|A\mathbf{x} - \mathbf{b}\|^2$, where

$$A = \left[\begin{array}{c|ccc} -X^\top X & \boldsymbol{c} & -\boldsymbol{c} & I_n \\ -\boldsymbol{c}^\top & 0 & 0 & 0 \end{array}\right], \quad \mathbf{x} = \begin{bmatrix} \lambda \\ \mu \\ \hline b_1 \\ b_2 \\ \boldsymbol{y} \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} -\mathbf{1}_{p+q} \\ 0 \end{bmatrix}.$$

Example 4 $n = 2,\ p = 50,\ q = 30$

- Iter $=11$, Time=0.115s, $\#\text{Neqs}_1 = 59$ and $\#\text{Neqs}_2 = 195$.



Thank you for your attention!