

## CSE 203B WI25 Homework 2 Suggested Solution

### Assignments

#### Part One

##### 1.1 Gaussian Elimination

Step 1: Form the Augmented Matrix

$$\left[ \begin{array}{ccc|c} 4 & 12 & -8 & 12 \\ 12 & 37 & -19 & 59 \\ -8 & -19 & 50 & 127 \end{array} \right].$$

Step 2: Perform Gaussian Elimination

- Row 1: Normalize the pivot element (Divide Row 1 by 4):

$$\left[ \begin{array}{ccc|c} 1 & 3 & -2 & 3 \\ 12 & 37 & -19 & 59 \\ -8 & -19 & 50 & 127 \end{array} \right].$$

- Row 2: Eliminate the first element (Row 2  $\rightarrow$  Row 2  $-$  12  $\times$  Row 1):

$$\left[ \begin{array}{ccc|c} 1 & 3 & -2 & 3 \\ 0 & 1 & 5 & 23 \\ -8 & -19 & 50 & 127 \end{array} \right].$$

- Row 3: Eliminate the first element (Row 3  $\rightarrow$  Row 3  $+$  8  $\times$  Row 1):

$$\left[ \begin{array}{ccc|c} 1 & 3 & -2 & 3 \\ 0 & 1 & 5 & 23 \\ 0 & 5 & 34 & 151 \end{array} \right].$$

- Row 2: Normalize the pivot element (unchanged):

$$\left[ \begin{array}{ccc|c} 1 & 3 & -2 & 3 \\ 0 & 1 & 5 & 23 \\ 0 & 5 & 34 & 151 \end{array} \right].$$

- Row 3: Eliminate the second element (Row 3  $\rightarrow$  Row 3  $-$  5  $\times$  Row 2):

$$\left[ \begin{array}{ccc|c} 1 & 3 & -2 & 3 \\ 0 & 1 & 5 & 23 \\ 0 & 0 & 9 & 36 \end{array} \right].$$

- Row 3: Normalize the pivot element (Divide Row 3 by 9):

$$\left[ \begin{array}{ccc|c} 1 & 3 & -2 & 3 \\ 0 & 1 & 5 & 23 \\ 0 & 0 & 1 & 4 \end{array} \right].$$

Step 3: Back Substitution

- From Row 3:

$$x_3 = 4.$$

- From Row 2:

$$x_2 + 5x_3 = 23 \implies x_2 + 20 = 23 \implies x_2 = 3.$$

- From Row 1:

$$x_1 + 3x_2 - 2x_3 = 3 \implies x_1 + 9 - 8 = 3 \implies x_1 = 2.$$

Final Solution:

$$\mathbf{x} = \begin{bmatrix} 2 \\ 3 \\ 4 \end{bmatrix}$$

## 1.2 LU Decomposition

$$\mathbf{A} = \mathbf{LU} = \begin{bmatrix} 1 & 0 & 0 \\ 3 & 1 & 0 \\ -2 & 5 & 1 \end{bmatrix} \begin{bmatrix} 4 & 12 & -8 \\ 0 & 1 & 5 \\ 0 & 0 & 9 \end{bmatrix}$$

Step 1: Solve  $\mathbf{Ly} = \mathbf{b}$

First, we solve for  $\mathbf{y}$  in the system  $\mathbf{Ly} = \mathbf{b}$ :

$$\begin{bmatrix} 1 & 0 & 0 \\ 3 & 1 & 0 \\ -2 & 5 & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 12 \\ 59 \\ 127 \end{bmatrix}.$$

The solution for  $\mathbf{y}$  is:

$$\mathbf{y} = \begin{bmatrix} 12 \\ 23 \\ 36 \end{bmatrix}.$$

Step 2: Solve  $\mathbf{Ux} = \mathbf{y}$

Now we solve for  $\mathbf{x}$  in the system  $\mathbf{Ux} = \mathbf{y}$ :

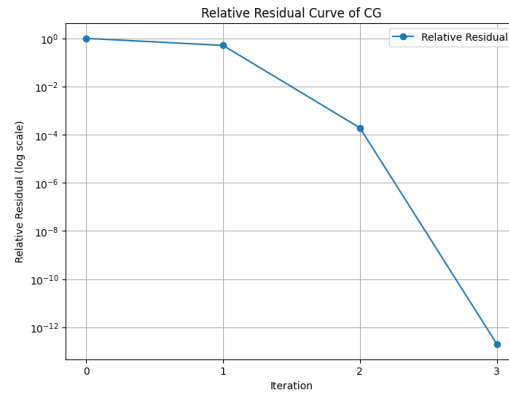
$$\begin{bmatrix} 4 & 12 & -8 \\ 0 & 1 & 5 \\ 0 & 0 & 9 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 12 \\ 23 \\ 36 \end{bmatrix}.$$

The solution for  $\mathbf{x}$  is:

$$\mathbf{x} = \begin{bmatrix} 2 \\ 3 \\ 4 \end{bmatrix}.$$

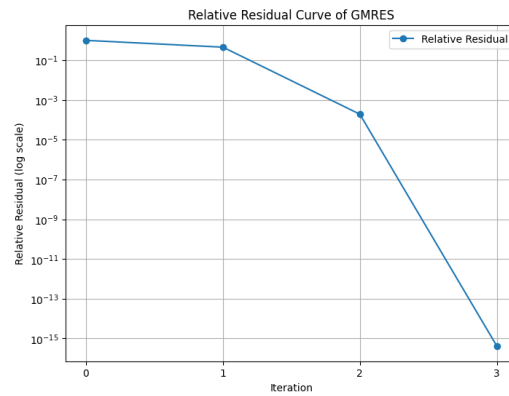
### 1.3 Conjugate Gradient Method

Iteration	$\hat{\mathbf{x}}^T$
1	[0.3683, 1.8106, 3.8974]
2	[0.8308, 3.3520, 3.9467]
3	[2.0000, 3.0000, 4.0000]



### 1.4 Generalized Minimal Residual Method

Iteration	$\hat{\mathbf{x}}^T$
1	[0.2923, 1.4373, 3.0939]
2	[0.8308, 3.3520, 3.9467]
3	[2.0000, 3.0000, 4.0000]

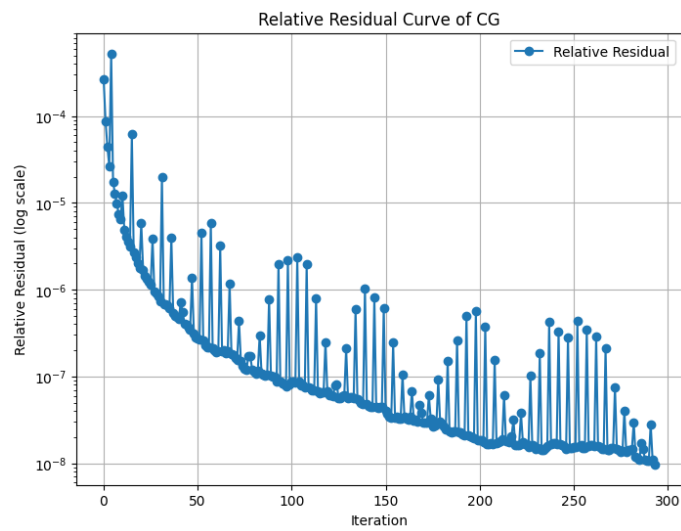


## 1.5 Large Matrix Equation

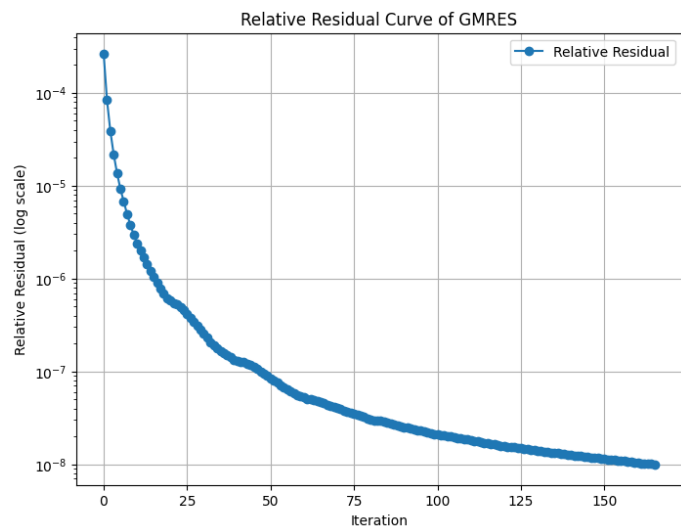
1.5.1  $\|\mathbf{x}\|_2 \approx 19.08700702357971$

1.5.2  $\text{tr}(\mathbf{U}) \approx 42664.32665486095$

1.5.3 CG



1.5.4 GMRES



### 1.5.5 Machine Precision

- 32-bit (single-precision floating-point format):  $2^{-23} \approx 1.19 \cdot 10^{-7}$
- 64-bit (double-precision floating-point format):  $2^{-52} \approx 2.22 \cdot 10^{-16}$

### 1.5.6 Condition number $\kappa(A) \approx 3112127456.697697$

1.5.7 The condition number  $\kappa(A) = \|A\| \cdot \|A^{-1}\|$  measures how sensitive the output is due to perturbation or errors in the input. A matrix with small  $\kappa(A)$  is well-conditioned, meaning small perturbations or errors in the input result in small changes in solution  $x$ , while a large  $\kappa(A)$  indicates ill-conditioning, where the solution is highly sensitive to such changes. Also, we usually define a singular matrix's condition number as  $\infty$ . For iterative solvers, smaller condition numbers typically lead to faster convergence.

### 1.5.8 Error, residual, machine precision, and condition number.

Let  $\mathbf{x}$  be the solution to  $\mathbf{Ax} = \mathbf{b}$  and  $\hat{\mathbf{x}}$  be the solution to  $\mathbf{A}\hat{\mathbf{x}} = \mathbf{b} + \delta\mathbf{b}$  where  $\delta\mathbf{b}$  indicates the perturbation or errors in the input data.

$$\begin{aligned}\mathbf{Ax} &= \mathbf{b} \\ \mathbf{A}\hat{\mathbf{x}} &= \mathbf{b} + \delta\mathbf{b} \\ \delta\mathbf{x} &= \hat{\mathbf{x}} - \mathbf{x}\end{aligned}$$

Thus,

$$\begin{aligned}\mathbf{A}\hat{\mathbf{x}} &= \mathbf{A}(\mathbf{x} + \delta\mathbf{x}) \\ \mathbf{Ax} + \mathbf{A}\delta\mathbf{x} &= \mathbf{b} + \delta\mathbf{b} \\ \mathbf{A}\delta\mathbf{x} &= \delta\mathbf{b}\end{aligned}$$

According to our definitions:

$$\begin{aligned}\text{Relative error: } & \frac{\|\hat{\mathbf{x}} - \mathbf{x}\|}{\|\mathbf{x}\|} \\ \text{Relative residual: } & \frac{\|\mathbf{b} - \mathbf{A}\hat{\mathbf{x}}\|}{\|\mathbf{b}\|} \\ \text{Condition number: } & \|A\| \cdot \|A^{-1}\|\end{aligned}$$

Now, let's see their relationship:

$$\begin{aligned}\frac{\frac{\|\hat{\mathbf{x}} - \mathbf{x}\|}{\|\mathbf{x}\|}}{\frac{\|\mathbf{b} - \mathbf{A}\hat{\mathbf{x}}\|}{\|\mathbf{b}\|}} &= \frac{\|\mathbf{b}\| \|\hat{\mathbf{x}} - \mathbf{x}\|}{\|\mathbf{x}\| \|\mathbf{b} - \mathbf{A}\hat{\mathbf{x}}\|} \\ &= \frac{\|\mathbf{Ax}\| \|\mathbf{A}^{-1}(\mathbf{A}\hat{\mathbf{x}} - \mathbf{b})\|}{\|\mathbf{x}\| \|\mathbf{b} - \mathbf{A}\hat{\mathbf{x}}\|} \\ &\leq \frac{\|\mathbf{A}\| \|\mathbf{x}\| \|\mathbf{A}^{-1}\| \|\mathbf{A}\hat{\mathbf{x}} - \mathbf{b}\|}{\|\mathbf{x}\| \|\mathbf{b} - \mathbf{A}\hat{\mathbf{x}}\|} \\ &= \|A\| \cdot \|A^{-1}\|\end{aligned}$$

Finally, we have the inequality:

$$\text{Relative error} \leq \text{Condition number} \cdot \text{Relative residual}$$

Notice that even if an algorithm is guaranteed to converge and produce a small residual, it does not guarantee that the relative error will be small. The relative error will be small only if the condition number of the matrix  $\mathbf{A}$  is small. Thus, it is important to calculate the condition number to check if the input matrix is well-conditioned before solving the problem.

There are multiple definitions of machine precision or machine epsilon. One common definition states that the machine precision  $\epsilon$  is the smallest positive machine number (e.g., floating point number) such that  $1 + \epsilon > 1$ . In this case, machine precision provides an upper bound on the relative error due to rounding.

It is also crucial to clarify the definitions of error, residual, and machine precision as they may vary in different contexts. The primary objective of this assignment is not to delve into the details of formal mathematical proofs, but rather to gain hands-on experience with various methods and understand basic concepts. For example, this exercise illustrates that the condition number plays a vital role in determining the accuracy of the solution. Additionally, for iterative methods, the solution  $\mathbf{x}$  may not be exact, as these methods often terminate once a predefined tolerance of the relative residual is reached. Consequently, the relative error of the solution can stem not only from the machine precision  $\epsilon$  but also from the level of tolerance set during computation.

1.5.9 The runtime depends on the implementation. You will receive full credit as long as you report the times and state which method is the fastest.

1.5.10 The four methods discussed (Gaussian elimination, LU decomposition, CG, and GMRES) each have distinct advantages and limitations. Direct methods like Gaussian elimination and LU decomposition are robust and suitable for small to moderately sized problems. However, these methods can become computationally expensive for very large systems due to their high memory and computational requirements. Iterative methods, such as CG and GMRES, are often more efficient for large, sparse systems. CG is specifically designed for symmetric positive definite matrices, while GMRES can handle nonsymmetric matrices but requires additional memory to store Krylov subspace vectors.

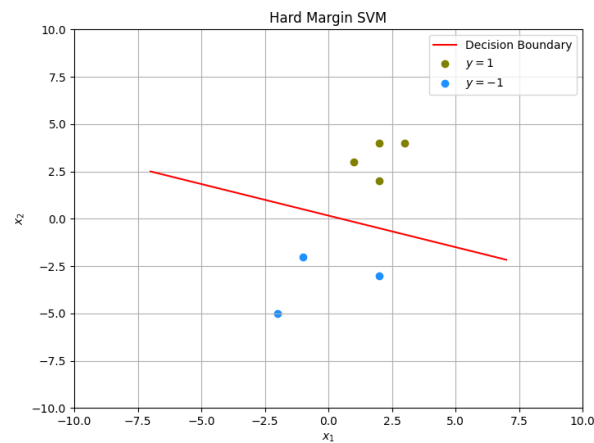
In practice, hybrid methods combining direct and iterative techniques are frequently employed to leverage the strengths of both approaches. The key idea is to find a good preconditioner  $\mathbf{P}$ , a matrix such that  $\mathbf{P}^{-1}\mathbf{A}$  has a smaller condition number than  $\mathbf{A}$ . By solving the transformed system  $\mathbf{P}^{-1}\mathbf{A}\mathbf{x} = \mathbf{P}^{-1}\mathbf{b}$ , the convergence of iterative methods can be significantly accelerated. This highlights the importance of preconditioning in modern numerical linear algebra, where reducing the condition number of the problem is often a central objective to ensure both efficiency and accuracy.

## Part Two

### 2.1 Hard Margin SVM

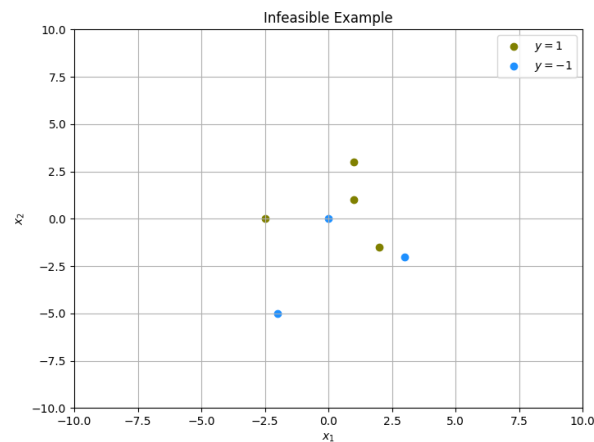
There are multiple interpretations of when a hard-margin SVM has a valid solution. One common interpretation is that the dataset must be linearly separable. Specifically, there exists a hyperplane defined by  $\mathbf{a} \in \mathbb{R}^n$  and  $b \in \mathbb{R}$  such that the constraints  $y_i(\mathbf{a}^T \mathbf{x}_i - b) \geq 1$  for all  $i = 1, \dots, m$  are satisfied. This implies no overlap between the positive and negative classes in the feature space, and the data points can be perfectly separated by a hyperplane. Another interpretation is that valid solutions exist only if the convex hulls of the sets  $\{\mathbf{x}_i \mid y_i = -1, i = 1, \dots, m\}$  and  $\{\mathbf{x}_i \mid y_i = 1, i = 1, \dots, m\}$  are disjoint, which also ensures a hyperplane can separate them.

### 2.2 Feasible Solution



### 2.3 Infeasible Solution

In this example, no hyperplane can separate the two classes without errors.



## 2.4 Soft Margin SVM

Note that multiple valid solutions exist. One approach to reformulate the problem involves introducing slack variables  $\xi_i \geq 0$ , which accommodate misclassification.

$$\min_{\mathbf{a}, b, \boldsymbol{\xi}} \frac{1}{2} \|\mathbf{a}\|_2^2 + C \sum_{i=1}^m \xi_i, \quad \mathbf{a} \in \mathbb{R}^n, b \in \mathbb{R}, \boldsymbol{\xi} \in \mathbb{R}^m$$
$$\text{s.t. } y_i(\mathbf{a}^T \mathbf{x}_i - b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad i = 1, \dots, m.$$

Here,  $C > 0$  is a regularization parameter that balances the trade-off between maximizing the margin and minimizing the classification error. The slack variables  $\xi_i$  relax the constraints, enabling the formulation to handle non-separable data.

