CSE 203B Week 2 Discussion

Po-Chun Wu

2025.01.15

Reminders

- Course website: https://cseweb.ucsd.edu//classes/wi25/cse203B-a/
- HW1 due on Jan 16 (Thursday) 11:50 PM
- HW2 due on Jan 23 (Thursday) 11:50 PM
- Late policy for homework: Piazza note @19

Resources

- Convex Optimization Textbook: https://stanford.edu/~boyd/cvxbook/
- Convex Optimization by Stephen Boyd (YouTube)
- Linear Algebra by Gilbert Strang (YouTube)

HW2

- Exercises
  - Grade by completion
  - Group work (up to four students)
- Assignments
  - Grade by content
  - Individual work

HW2 Assignments

- Solve $Ax = b$
- Support vector machine (SVM)

$$\begin{cases} x + y = 5 \\ 2x + 3y = 6 \end{cases}$$

$$\mathbf{A}\mathbf{x} = \mathbf{b}$$

$$\mathbf{A} \in \mathbb{R}^{m \times n}$$

$$\mathbf{x} \in \mathbb{R}^{n}$$

$$\mathbf{b} \in \mathbb{R}^{n}$$

Why not $x = A^{-1}b$?

- Computational inefficiency
- Numerical instability

How to solve Ax = b ?

- Direct
  - Gaussian elimination
  - LU decomposition
  - Cholesky decomposition
- Iterative
  - Jacobi method
  - Gauss-Seidel method
  - Conjugate gradient method (CG)
  - Generalized minimal residual method (GMRES)
- Hybrid
  - E.g., https://escholarship.org/uc/item/0j60b61v

Gaussian Elimination

$$\begin{cases} x + y = 5 \\ 2x + 3y = 6 \end{cases} \Rightarrow \begin{bmatrix} 1 & 1 \\ 2 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 5 \\ 6 \end{bmatrix}$$

Augmented Matrix

$$\left[\begin{array}{cc|c} 1 & 1 & 5 \\ 2 & 3 & 6 \end{array}\right]$$

LU Decomposition / LU Factorization

$$\mathbf{A} = \mathbf{L}\mathbf{U}$$

$$\mathbf{A}\mathbf{x} = \mathbf{b}$$

$$\mathbf{L}\mathbf{U}\mathbf{x} = \mathbf{b}$$

$$\mathbf{L}\mathbf{y} = \mathbf{b}$$

$$\mathbf{U}\mathbf{x} = \mathbf{y}$$

LU Decomposition

- Partial pivoting
  - PA = LU
  - Numerical stability
- Forward substitution
- Back substitution
- Non-square matrix?
- Parallel LU decomposition
  - E.g., https://faculty.cc.gatech.edu/~echow/pubs/parilu-sisc.pdf

$$\mathbf{A}\mathbf{x} = \mathbf{b}$$

$$\text{minimize } f(\mathbf{x}) = \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2$$

$$\text{minimize } f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T\mathbf{A}\mathbf{x} - \mathbf{x}^T\mathbf{b}$$

Conjugate Gradient Method (CG)

- https://nvlpubs.nist.gov/nistpubs/jres/049/jresv49n6p409_a1b.pdf
- https://www.cs.cmu.edu/~quake-papers/painless-conjugate-gradient.pdf
- https://cseweb.ucsd.edu//classes/sp24/cse291-e/slides/ConjugateGradient.pdf

In case the matrix A is *symmetric* and *positive definite*, the following formulas are used in the conjugate gradient method:

$$p_0 = r_0 = k - Ax_0 \quad (x_0 \text{ arbitrary}) \tag{3:1a}$$

$$a_i = \frac{|r_i|^2}{(p_i, Ap_i)}, \tag{3:1b}$$

$$x_{i+1} = x_i + a_i p_i, \tag{3:1c}$$

$$r_{i+1} = r_i - a_i A p_i, \tag{3:1d}$$

$$b_i = \frac{|r_{i+1}|^2}{|r_i|^2}, \tag{3:1e}$$

$$p_{i+1} = r_{i+1} + b_i p_i. \tag{3:1f}$$

In place of the formulas (3:1b) and (3·1e) one may use

$$a_i = \frac{(p_i, r_i)}{(p_i, Ap_i)}, \tag{3:2a}$$

$$b_i = -\frac{(r_{i+1}, Ap_i)}{(p_i, Ap_i)}. \tag{3:2b}$$

Generalized Minimal Residual Method (GMRES)

- More generalized compared to CG
- https://web.stanford.edu/class/cme324/saad-schultz.pdf

ALGORITHM 3: The generalized minimal residual method (GMRES).

1. *Start*: Choose x_0 and compute $r_0 = f - Ax_0$ and $v_1 = r_0/\|r_0\|$.
2. *Iterate*: For $j = 1, 2, \cdots, k, \cdots$, until satisfied do:

   $h_{i,j} = (Av_j, v_i)$, $i = 1, 2, \cdots, j$,

   $\hat{v}_{j+1} = Av_j - \sum_{i=1}^{j} h_{i,j} v_i$,

   $h_{j+1,j} = \|\hat{v}_{j+1}\|$, and

   $v_{j+1} = \hat{v}_{j+1} / h_{j+1,j}$.
3. *Form the approximate solution*:

   $x_k = x_0 + V_k y_k$, where y_k minimizes (7).

Key Concepts

- Absolute error
- Relative error
- Residual
- Condition number
- Machine precision
- https://www.sciencedirect.com/science/article/pii/0045790673900104

Condition Number of a Matrix

$$\begin{bmatrix} 1.001 & 0.999 \\ 0.999 & 1.001 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$$

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} 1.001 & 0.999 \\ 0.999 & 1.001 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 2.004 \\ 1.996 \end{bmatrix}$$

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 3 \\ -1 \end{bmatrix}$$

$$\mathbf{A} = \begin{bmatrix} 1.001 & 0.999 \\ 0.999 & 1.001 \end{bmatrix}$$

```
---numpy.linalg package---
Condition number of A: 1000.0000000001692
Euclidean norm of A: 2.00000099999975
Euclidean norm of A inverse: 500.0002499999684
Product of norms: 1000.0010000000618
Largest eigenvalue: 2.0
Smallest eigenvalue: 0.0020000000000000018
Ratio of largest to smallest eigenvalue: 999.9999999999991
```

Machine Precision

- IEEE 754: https://ieeexplore.ieee.org/document/4610935
- Decimal module in Python: https://docs.python.org/3/library/decimal.html

```
x = 1.3
y = 2.6

print(x + y)
```

```
3.9000000000000004
```

```
from decimal import Decimal

print(Decimal('1.3') + Decimal('2.6'))
```

```
3.9
```

Discussions

- Limitations and assumptions
- Computational complexity of solving $Ax = b$
- Number of solutions to $Ax = b$
  - $r = n = m$
  - $r = n < m$
  - $r = m < n$
  - $r < m, \ r < n$
  - https://ocw.mit.edu/courses/18-06sc-linear-algebra-fall-2011/f5a74578196a8afc2fd2ba8581acb17f_MIT18_06SCF11_Ses1.8sum.pdf

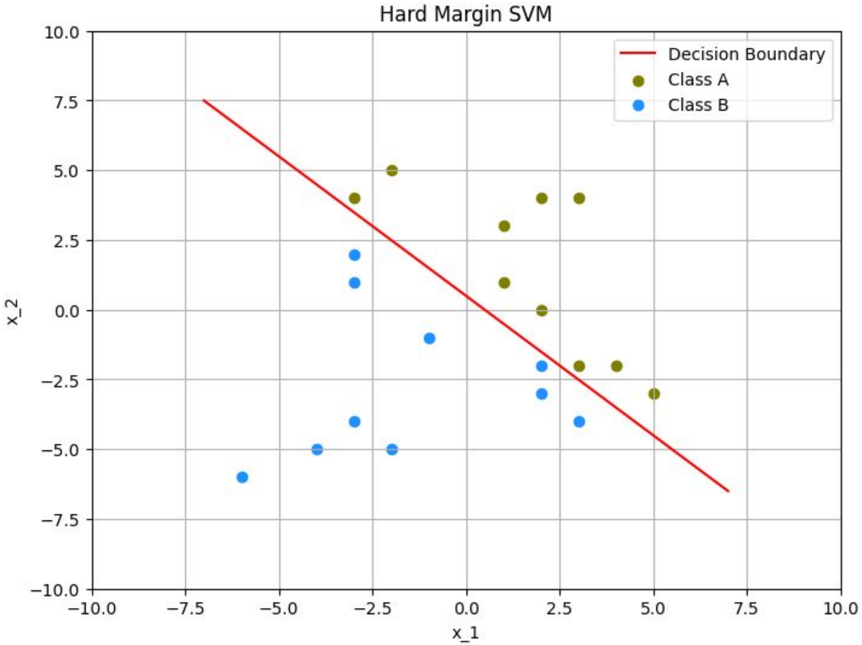
Tools

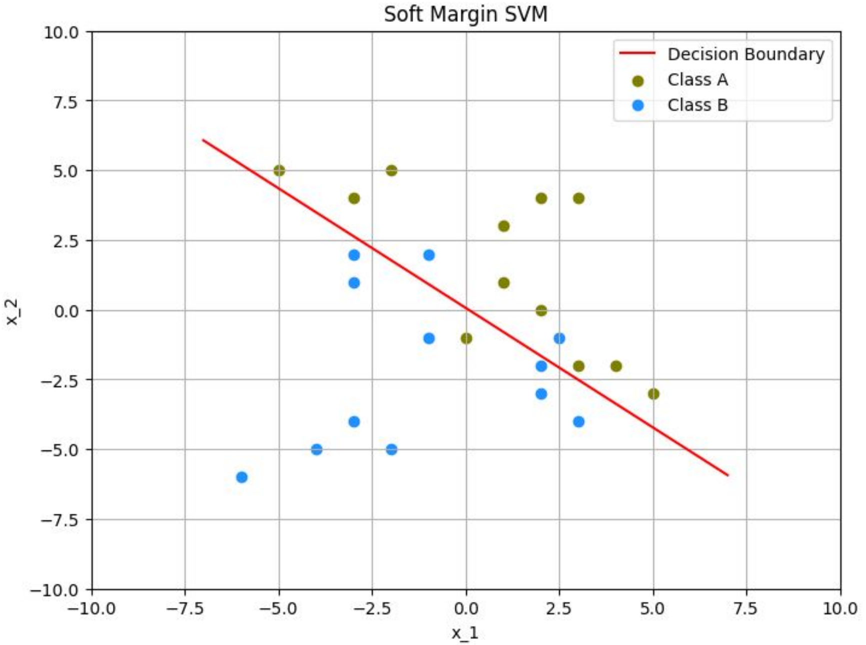
- Python (e.g., NumPy, SciPy)
- MATLAB

Support Vector Machine (SVM)

- Goal: find the hyperplane with maximum margin
- Hard margin SVM
    - Linearly separable
- Soft margin SVM
    - Introduce slack variables to allow for misclassifications

Hard Margin SVM vs. Soft Margin SVM





Good luck on HW1 and HW2!