**בעיית הריבועים הפחותים:** full rank, ,

אנו מחפשים וקטור כך ש- יהיה כמה שיותר קרוב ל-.

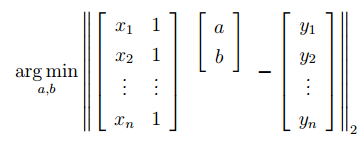
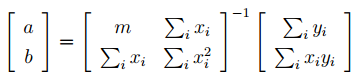
\* וקטור השארית.

\* הוא הוקטור האופטימלי או בכתיב מטריציוני (כאשר היא full rank)

הן המשוואות הנורמליות, אם אינה full rank אז יש להן אינסוף פתרונות.

\* עבור קבוצות נק' נרצה למצוא ישר העובר כמה שיותר קרוב לנק'.

נפתור זאת כך: ונקבל:



\* ניתן להוסיף משקולות לבעיה ע"י מטריצה שהיא ולקבל או

\* אם סינגולרית ניתן להוסיף רגולריזציה לבעיה באמצעות ומטריצה (בדר"כ ) ולקבל:

או

**טענה:** כל משוואה ריבועית ניתן לכתוב כך:

**מינימום ריבועים ע"י SVD:**

כאשר היא full rank אז

כאשר לא full rank אז

עבור

**פתרון מערכת משוואות:**

**טענה:** קיים פתרון יחיד אם , אחרת

יש אינסוף פתרונות או שאין פתרון כלל.

**Gaussian Elemination , LU , pivoting:** קיים פירוק ומתקיים: ,

,

הפיכת מטריצה: ,

\* אם הפיכה אז קיים לה פירוק יחיד

\* אם SPD אז קיים פתרון ללא Pivot

\*

**Cholesky למטריצות SPD:** אם היא SPD קיימת משולשית תחתונה עם כך ש-.

\*

**Thomas למערכת תלת אלכסונית:** האלגוריתם מצליח כאשר המטריצה היא או .

**פירוק QR:** עבור נקבל .

***מכפלה פנימית וקטורית:***

תקרא מכפלה פנימית לכל אם:

\* סימטריות:

\* לינאריות: וגם

\* אי שליליות: או

**מכפלות מוכרות:** , למטריצה מסוג

**נורמה וקטורית:** תקרא נורמה אם מתקיים:

\* אי שליליות: או

\* הומוגניות:

\* אי שוויון המשולש:

**קבוצת נורמות :**

למשל: ,

*נשים לב כי*

***נורמת אנרגיה:*** *לכל מטריצה מסוג תוגדר הנורמה*

***טענה:*** *מטריצה אורתוגונלית משמרת נורמה וקטורית ומתקיים:*

**טענה:** כל נורמה היא פונקציה רציפה (לא בהכרח גזירה).

***טענה:*** *לכל נורמות וקטוריות קיימים קבועים ומתקיים (כלומר כל הנורמות שקולות, אם באחת נתכנס ל-0 אז גם בשניה).*

***ע"ע ו-ו"ע:***  ו"ע ל-ע"ע ב-

**רדיוס ספקטרלי:**

**ערך סינגולרי:** עבור ע"ע של

**פירוק ספקטרלי:** אם ל- יש ו"ע עם ע"ע שונים

אז לכסינה:

**פירוק נורמלי:** אם נורמלית אז קיימת

אוניטרית: ,

,

**חיובית מוגדרת:**

(- מטריצה סימטרית חיובית מוגדרת )

**חיובית חצי מוגדרת:**

(מתקיים לכל כאשר מדרגה מלאה)

**מטריצה הפיכה:** קיים

לא סינגולרית היא full rank

**טענה:** סינגולרית או

**טענה:** עבור חיובית מוגדרת / הפיכה / לכסינה,

אם אז

**טענה:** עבור מהצורה אם

אז SPD

**נורמלית:**

**תלת אלכסונית:**

**strictly diagonally dominant: (SDD)** מטריצה

אשר מקיימת או

**דטרמיננטות:**

**\***

\* אם L היא מטריצה משולשית אז

\* עבור נקבל

**חוקי מטריצות:**

**\***  עבור ריבועיות

\*

\*

**סכום מטריצות ו-ע"ע:**

\* עבור מטריצות בעלי אותו ו"ע:

**פירוק SVD:** לכל , , קיימות מטריצות אורתוגונליות ומטריצה אלכסונית

( הם ערכים סינגולריים של ומתקיים ) כך שמתקיים:

* הם שורשי הע"ע של
* הם הו"ע של
* הם הו"ע של

**קירוב מטריצה עם מטריצה מדרגה קטנה ופירוק SVD:** נחפש הממזערת את כאשר .

נבחר את כאשר ל- נבחר את הערכים הסינגולריים הגדולים ביותר של .

**מטריצת Householder:** מטריצת Householder היא מטריצה אורתוגונלית מהצורה

לכל וקטור מתקיים   
עבור הוקטור מוגדר כך:

**תהליך Gram Schmidt:** ניצור מטריצה אורתוגונלית בהתבסס על עמודות כך:

**פירוק :** באמצעות גרהאם שמידט עם נירמול נפרק מטריצה למטריצה אורתוגונלית ו- משולשית עליונה.

Text

Description automatically generated with low confidence

**Power method:** למציאת ו"ע עם ע"ע גדול ביותר. בהינתן ניחוש התחלתי :

לכל לכסינה עם ע"ע :

**הערה:** עבור השיטה תחזיר את , ו"ע עם ע"ע קטן ביותר

**Inverse Power method:** מציאת ו"ע של ע"ע נתון.

בהינתן ניחוש לע"ע וניחוש התחלתי לו"ע :

פקטור ההתכנסות ל- I.P.Mהינו

**מספר מצב (Condition Number):** מסומן

מתקיים כאשר

ו- עבור ע"ע של . מטריצה עם מספר מצב גדול נקראת מטריצה "חולנית" ועלולה ליצור שגיאות גדולות.

**נורמת מטריצה:** (מקיימות את אותן תכונות של נורמות וקטוריות)

**נורמות מטריצות אנאלוגיות לנורמות וקטוריות:**

**\*** Forbenius Norm -

\*

\*

**נורמות מטריצות מושרות:**

יסומן

**דוגמאות:**

כאשר ע"ע של

**טענה:**

**טענה:**

**טענה:** עבור נורמלית מתקיים

**טענה:** לכל נורמה מושרית מתקיים

**נגזרות:**

\* גרדיאנט של הוא

\* אז

\* אז

\* אז

\* אז

\* אז

**שיטות איטרטיביות לבעיות אופטימיזציה ללא אילוצים:**

**שיטת steepest descend:** נבחר בכל איטרציה

(ניתן לבחור )

הוא כיוון ירידה כאשר מתקיים: .

כיוון הירידה הוא מינוס הגרדיאנט: , שהוא כיוון הירידה המירבי של הפונקציה .

כל כיוון כך ש- חיובית מוגדרת הוא כיוון ירידה.

**שיטת Newton:** אם הפיכה אז בכל איטרציה נבחר

עבור

**שיטת Quasi Neqton:** נחליף את ב- ואז בכל איטרציה נבחר עבור

(למשל שווה לאלכסון של וכך היא הפיכה)

**שיטות Line Search:** בכל איטרציה נבחר (כאן ההתייחסות היא לבחירת , בהנחה שאת אנו יודעים לבחור).

נרצה לבחור מינימלית, כלומר עבור נרצה למזער

**Backtracking line search using Armijo condition:** נבחר די גדול, (בדר"כ ) ו- (בדר"כ ).

בכל איטרציה נבחן עבור עד שמתקיים:

**שיטות Coordinate descent:** בכל איטרציה נעדכן את הקואורדינטה ה- של . נניח כי קבועים לכל ונבחר המקיים .

ניתן לעבור על ה- סדרתית או אקראית (העיקר לעדכן את כולם).

**Penalty and Barrier methods:**

את בעיות האופטימיזציה עם אילוצי אי שוויון ושוויון שראינו ב- נכתוב כך:

עבור ו- פונקציות סקלריות החסומות מלמטה ב-0 (לרוב ). עבור הפתרון יהיה זהה לפתרון של .

בפועל נפתור בצורה איטרטיבית עם ניחוש התחלתי את עם ובכל איטרציה הקלט יהיה מהאיטרציה הקודמת.

**משפטים שאני המצאתי:**

**משפט:** עבור כאשר הם ו"ע וע"ע מתקיים

**משפט:** עבור SPD מתקיים

וגם

**משפט:** בשיטות איטראטיביות לפתרון מערכת משוואות הוא זה שממזער את

**משפט:** אם מדרגה מלאה אז

**משפט:** אם SPD אז כל המינורים שלה הם גם SPD. בנוסף למינורים יש דטרמיננטות שונות מ-0.

**משפט:** אם חצי מוגדרת ו- חיובית מוגדרת אז חיובית מוגדרת.

בפרט אם full rank אז חיובית מוגדרת.

**משפט:** אם מכפלת מטריצות מדרגה מלאה נותנת מטריצה מסדר כאשר הוא המינימלי מבין כל הדרגות אז המטריצה מדרגה מלאה.

**The projected Steepest Descent method:** נדאג שבכל איטרציה נהיה בתוך התחום שמקיים את האילוצים (feasible). עבור ניחוש התחלתי בכל איטרציה נבצע:

עבור פונקציית הטלה:

**קמירות:**

**1)**  קבוצה קמורה אם לכל מתקיים

**2)**  פונקציה קמורה אם ה-domain שלה הוא קבוצה קמורה ולכל מתקיים

**3)**  פונקציה קמורה אם היא גזירה ולכל מתקיים

**4)**  פונקציה קמורה אם היא גזירה פעמיים ומתקיים , כלומר מטריצה חיובית חצי מוגדרת

**5)** אם קמורה אז כל מינימום locally או כל נקודה המקיימת היא מינימום globally

**אופטימיזציה עם אילוצי שוויון:**

**וקטור כופלי לגראנז':**

**פונקציית לגרז'יאן:**

נחפש ו- כך שמתקיים: וגם

נקודה עם כופלי לגראנז' מתאימים היא מינימום אם

לכל מתקיים

**אופטימיזציה עם אילוצי שוויון ואי שוויון:** הצורה הכללית היא .

**אילוץ אי שוויון פעיל:** קיים פתרון כך שמתקיים היא קבוצת האינדקסים של אילוצי אי שוויון פעילים

**אילוץ אי שוויון לא פעיל:** קיים פתרון כך שמתקיים

**אופטימיזציה עם אילוצים:** צורה כללית לאופטימיזציה עם אילוצים:

**Feasible set:** קבוצת הפתרונות שמקיימים את האילוצים.

ניתן להגדיר את הבעיה כך:

הוא פתרון locally ל- אם וגם קיים המקיימת

**שיטות איטרטיביות לפתרון מערכת משוואות:**

\* עבור ניחוש התחלתי השיטה תוגדר כך

\* **קצב ההתכנסות**

מגדיר את קצב ההתכנסות ו- הוא ה-Convergence factor.

\* ההתכנסות היא לינארית כאשר ו-.

\* **וקטור השגיאה:** (לא ניתן לחישוב כי נעלם)

\* **וקטור השארית:**

\* השארית אנכית לעמודות המטריצה, כלומר

\* נרצה שיתקיים

\* תנאי עצירה או

\* **כתיב מטריציוני:** ולכן ונפתור

**שיטת ריצ'רדסון (Richardson):**

עבור ,

**שיטת ג'ייקובי הממושקלת (Jacobi):** , אם נקבל את שיטת ג'ייקובי הרגילה, .

או ונקבל או

**שיטת גאוס זיידל ממושקלת (Gauss-Seidel):** , אם נקבל את שיטת גאוס זיידל הרגילה, .

או

או

**התכנסות שיטות איטראטיביות:**

\* וקטור השגיאה הינו .

\* אם אז השיטה מתכנסת לכל .

**טענה:** בגאוס זיידל וג'ייקובי, אם הינה אז השיטות מתכנסות.

**טענה:** אם היא לא סינגולרית אז גאוס זיידל מתכנסת.

**טענה:** אם היא חיובית מוגדרת אז גאוס זיידל מתכנסת.

**טענה:** גאוס זיידל שקולה למינימיזציה של

.

**שיטת Steepes Descent:** אם היא SPD אז עבור

השיטה תוגדר כך:

עבור שנבחר בכל איטרציה מחדש.

**התכנסות לשיטת Steepes Descent:**

עבור הנ"ל

ו- , פקטור ההתכנסות הוא

**שיטת Conjugate gradients (CG):**

טענה: אם היא SPD ו- full rank אז CG מתכנסת אחרי איטרציות לכל היותר, עבור כל ניחוש התחלתי.

**וקטור כופלי לגראנז':**

**פונקציית לגרז'יאן:**

נחפש , , כך שמתקיים

,

ו-

הערה: נחפש שכל איבריו ויש בו כמה שיותר אפסים.

נקודה עם , מתאימים היא מינימום אם לכל המקיים

מתקיים

**אופטימיזציה ללא אילוצים: 1)** נתונה פונקציה ונרצה למצוא

**טור טיילור חד מימדי:**

**טור טיילור דו מימדי:**

עבור , ,

(ב- מימדים ) נקבל:

**2)** אם היא וקטור של פונקציות (ניתן לחשוב על מטריצה בגודל ) , ,

\*

\* (עבור נקבל שוויון)

\* היא מטריצת יעקוביאן,

\* אם עבור אז

, כלומר

\* אם ו- (כלומר אם מיוצגת באמצעות מטריצה אז רק האיבר ה- שונה מ-) אז:

*ו- ,*

**פתרון בעיית אופטימיזציה ללא אילוצים:**

\* נחפש המקיימת והיא תוכל להיות נקודת מינימום.

\* מינימום locally:

\* מינימום gloabally:

\* אם חיובית מוגדרת, מינימום locally אם