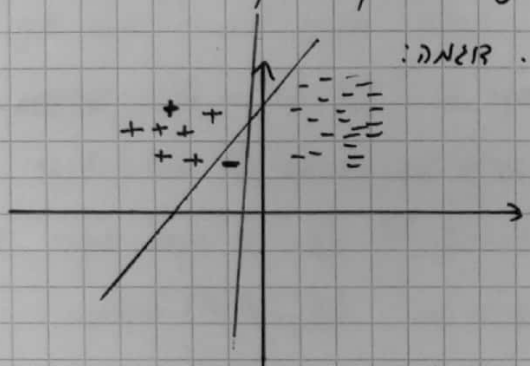
מבוא ללמידה מכונה - תרגיל 3

נושאים:

- Soft-Margin SVM
- Kernel Methods

Soft SVM

SVM ה"נקי" דורש כדי לקבל מאוד - שהדאטא יהיה ניתן להפרדה ע"י מישור.
זה מגביל לדברים כמו האומן כלל, ובכל מיני רוצים להפיק את המיטב ממה ש"ל.
משהו נוסף הוא שלפעמים הפתרון ה"נכון" למבוקה בעיה שלה כמו שאנחנו מישור שלבר לא נראה לנו הגיוני. דוגמה:



המפריד במשהו של שיעור שלבר נראה כמו המסומן בשחור.
מצד שני, זה לא נשמה הגיוני להפריד כך דאטא, רק בגלל נקודה אחת "רועשת".
הרבה יותר הגיוני לעשות לאו נכון את הנקודה הצו ולקבל ישר שנראה כמו המסומן בכחול.

כדי להתמודד עם שני הבעיות באלה, מנסים לבנות הרכלה משתנים נוספים ξ_i או שלפים, ומשנים את האילוץ ל- $y_i(\bar{w}\cdot\bar{x}_i+b) \geq 1-\xi_i$ (עבור i).
ξ_i הם פרמטרים המודדים כמה חריגה מההפרדה שהתבקשה ע"י הישר.
$\xi_i \geq 0$ אומר שהנקודה מקיימת נכון, אבל נמצאת קרוב לישר (מעבר שמו אומר להיות רווח שלמה נקודות). $\xi_i < 1$ אומר שהנקודה מסווגת בצורה שגויה.

הבעיה עכשיו מנוסחת כך:

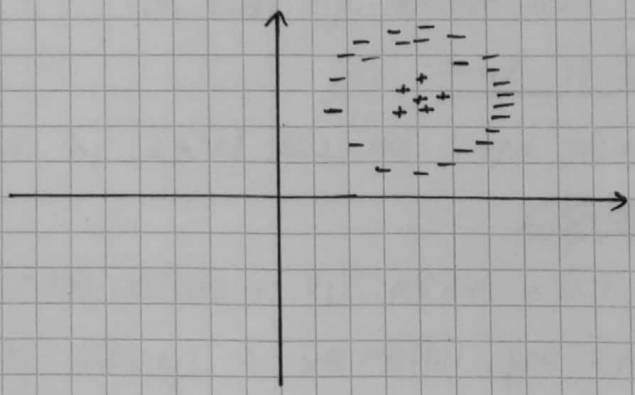
$$\underset{w,b,\xi}{\operatorname{argmin}} \quad \|w\|^2 + C\sum_i \xi_i$$
$$\text{s.t.} \quad y_i(\bar{w}\cdot\bar{x}_i+b) \geq 1-\xi_i \quad \forall i$$
$$\xi_i \geq 0$$

הערות:
- C - מכונה גם "Box Constraint", קובע כמה אנחנו מוכנים לסבול ניוד לא טוב של הדאטא.
C גבוה אומר שנרצה פחות חריגות (כי מקטין הרבה את הנורמה). עלול להוביל ל-Overfitting.
C נמוך אומר שאין בעיה שהדאטא יחרוג מהניוד שלו. עלול להוביל ל- Underfitting.
- למעשה, C משחק כמו קבוע רגולריזציה, והמשתנים ξ_i כולה שהתחלק בכל פ' רגולריזציה שלבה.
- את הבעיה פותרים ע"י מעבר לבעיה הדואלית והפעלת שיטות אופטימיזציה לפתרון בעיות ריבועיות.

Scanned by CamScanner

Kernel Methods

לעיתים יהיה קשה למצוא הפרדה לינארית. לדוגמה:



כדי להתגבר על הבעיה, נרצה להטיל את הדאטא שלנו למרחב אחר (שנסמן ב-F) מרחב הילברט, אולי בכלל $\mathbb{R}^n$ אחר n כלשהו, או סתם מספיק) כך שיהיה ניתן להפריד את הדאטא ע"י מפריד לינארי.

נגדיר פונקציה $\psi: X \to F$ כש- X התמונה המקורי ו- F המרחב החדש, ונרצה שיהיה אפשר להפריד את $(y_i, \psi(x_i))$ בקלות.

איך נה מתבצע?

הטריק הוא פונקציית גרעין $K(x_i, x_j) := \langle \psi(x_i), \psi(x_j) \rangle$.

כל פונקציה כזו היא מטריצה חיובית (למעלה, כל הערכים העצמיים שלה אי-שליליים) ולפי משפט מאוליצה פונקציונלית (לא לפחות, אם אני לא טועה) כל מטריצה חיובית כזו מייצגת מכפלה פנימית בין הוקטורים שיושבו מרחב הילברט.

הערה - לא כתבתי את זה כאן, אבל בהרצאה ראיתי איפה מתקיים את $\langle x_i, x_j \rangle$ ב-$K(x_i, x_j)$.

Kernels לשעבר: - פולינומיאלי: $K(x,x') = (\langle x, x' \rangle + 1)^k$. (כדי להוכיח שזה אכן Kernel, צריך למצוא את ψ המתאימה שלה מקיים $K(x,x') = \langle \psi(x), \psi(x') \rangle$).

בעצרת הגרעין זה אפשר למשל להפריד את הדאטא ע"י מפריד שמוגדר פולינום מדרגה k.

- גאוסיאני (ידוע גם כ-RBF - Radial Basis Function): $K(x,x') = e^{-\frac{\|x-x'\|^2}{2\sigma^2}}$

כש- $\|\cdot\|$ נורמה - 2 (סכום ריבועים הרכיבים, ואז שורש).

הגרעין הזה, נקודות x, x' קרובות זו לזו יחזירו ערך קרוב ל-1 ונקודות x, x' רחוקות זו מזו יחזירו ערך $K(x,x')$ קרוב ל- 0.

σ הוא פרמטר שקובע מהו כמה קרוב נחשב "קרוב". ערכים גדולים של σ נותנים $\frac{1}{2\sigma^2}$ קטן.

Kernels נותנים לנו אפשרות להכניס ידע מוקדם על הבעיה אל המודל.

Scanned by CamScanner

כלומר, אם מהסתכלות על הדאטא או כל ידע קודם כלשהו נדע שהוא עם צורת הדאטא במרחב (למשל, שהוא מסודר במעגלים כמו בדוגמה שבציור בעמוד הקודם), נוכל להשתמש ב- Kernel שיצא לה טלה למרחב מתאים.

שימוש ב- Kernels בסיווג Soft-SVM

מהרצאה ראינו שאת הפתרון של Soft-SVM מוצאים ע"י סיווג בעזרת פונקציות (ואם לא - תזכורת: הפתרון המקורי נבחר עם נוסחה שראינו בפועל משקולות האילוצים. הפתרון הדואלי הוא כזה שבה מוצאים את נוסחה שראינו של הפתרון המקורי), שבה הדאטא המקורי x_i מופיע אך ורק במכפלות פנימיות: $x_i^T x_j$.

במקום לבצע את המכפלות הנ"ל, אפשר להחליף ונוכן בכל מכפלה פנימית של איבר אחר, כלומר בפונקציה $K(x_i, x_j)$.

כך, אנחנו עוברים למרחב אחר מן הדאטא (התבע) "ח", ושם סיווגים את הפתרון ה- Soft-SVM.

Scanned by CamScanner