

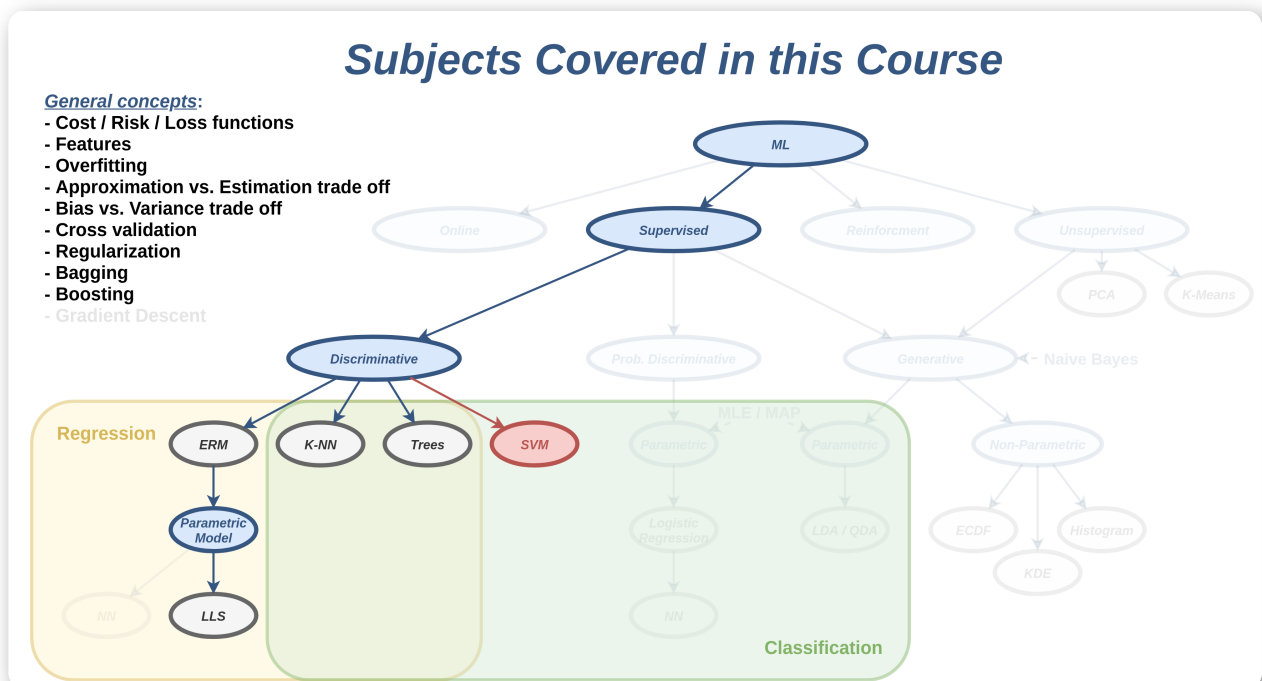
# הרצאה 6 - SVM ושיטות

## גרעין

Slides

PDF

מה נלמד היום



## סיווג לינארי

בפרק זה נעסוק בבעיית סיווג בינארי. לשם הנוחות נסמן את שתי המחלקות ב  $y = \pm 1$ . בפרט נעסוק בסוג מסוים של מסווגים מהצורה:

$$h(\mathbf{x}) = \text{sign}(\mathbf{w}^T \mathbf{x} + b) = \begin{cases} 1 & \mathbf{w}^T \mathbf{x} + b > 0 \\ -1 & \text{else} \end{cases}$$

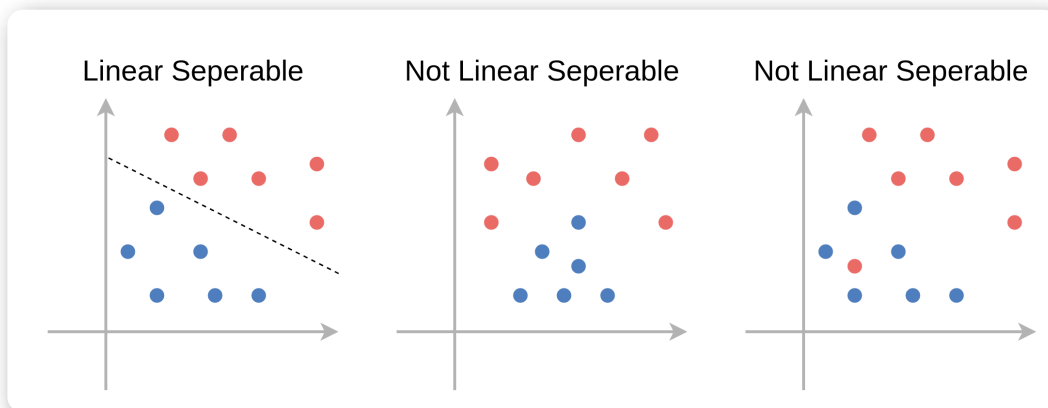
עם  $\mathbf{w}$  כל שהם. מסווגים אלו מכוונים **מסווגים לינאריים**. מסווגים מסוג זה מתקבלים לדוגמא בעבור LDA או linear logistic regression. מסווגים אלו מחלקים את המרחב לשני חלקים אשר נמצאים משני צידיו של על-מישור (hyperplane) המוגדר על ידי המשוואה  $\mathbf{w}^T \mathbf{x} + b = 0$  המכונה מישור ההפרדה.

על-מישור הוא הרחבה של מושג המישור למימדים שונים מ-2. במרחב מימד  $D$  על המישור יהיה מימד  $D - 1$ . לדוגמא, במימד אחד על-המישור הוא נקודה, ב-2 מימד על-המישור הוא קו ובתלת מימד על-המישור הוא מישור דו מימדי. לשם הפשטות, בקורס זה נשתמש בשם מישור גם כדי להתייחס לעל-מישורים.

**שימו לב:** חשוב לא להתבלבל בין המשוואה  $w^T x + b = 0$  לבין המשוואה  $ax + b = y$  אשר מגדירה משוואה לינארית במרחב של  $x$  ו  $y$ . שני הצורות האלה אומנם קרובות אך הם לא אותה משוואה וגם האיבר החופשי, שבשניהם מסומן לרוב ב  $b$  הוא לא אותו  $b$ .

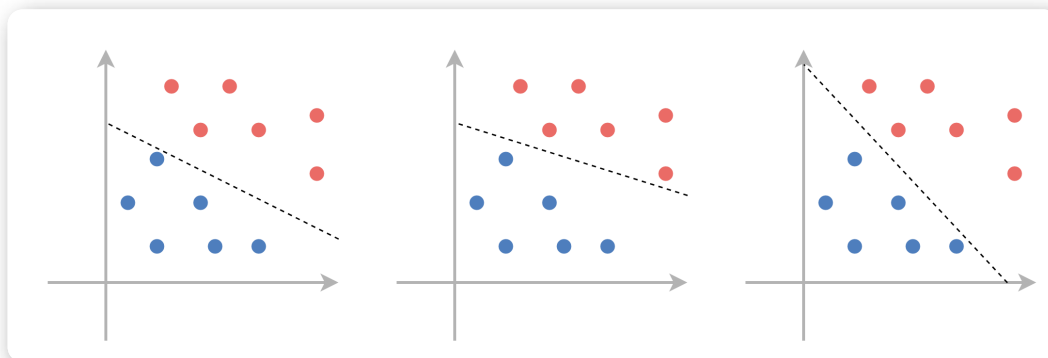
## פרידות לינארית (linear separability)

בהרצאה זו אנו נתייחס לשני מקרים, הראשון בו קיים מישור מפריד אשר מסווג את המדגם בצורה מושלמת (בלי טעויות סיווג) ושני בו לא ניתן למצוא מישור כזה. על מדגמים מהסוג הראשון נאמר שהם **פרידים לינארית**. להלן דוגמאות לשני סוגי המדגמים:



חשוב לציין שחלוקה זו לשני סוגי המדגמים רלוונטית רק לדיון התיאורטי, שכן לרוב לא נוכל לדעת מראש אם מדגם הוא פריד לינארית או לא.

למדגם פריד לינארית יהיה תמיד יותר ממשטח הפרדה אחד אשר יכול לסווג בצורה מושלמת את המדגם. לא נוכיח זאת, אך נראה דוגמא לכך על המדגם הבא:

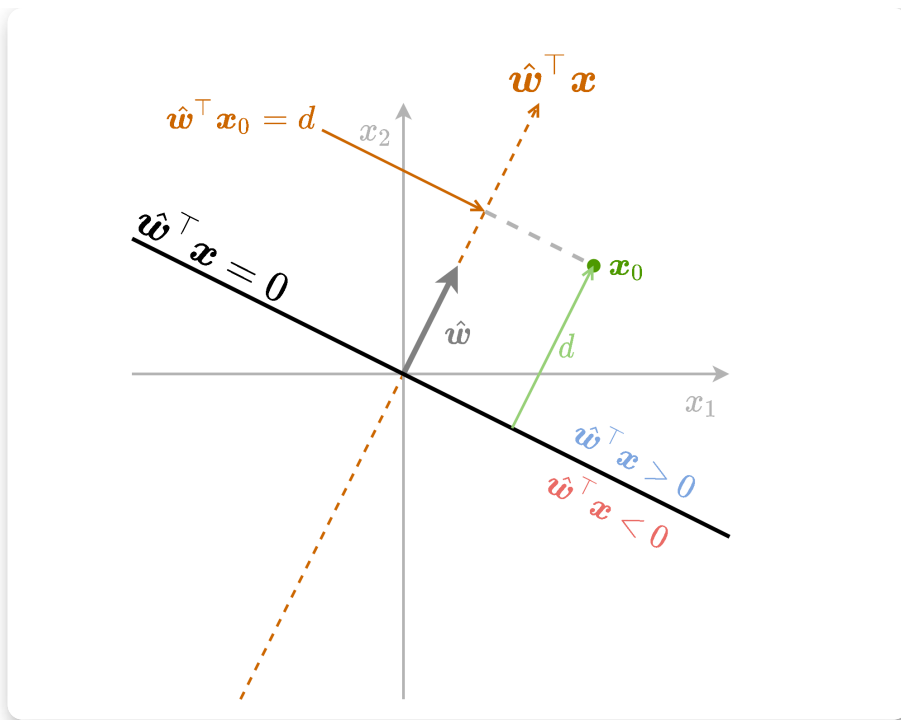


בחלקה הראשון של הרצאה זו נסתכל על המקרה של מדגם פריד לינארית ונציג את אלגוריתם ה  $hard\ SVM$  אשר מנסה לבחור בצורה חכמה את מישור ההפרדה הטוב ביותר. בחלקה השני של ההצגה נציג את אלגוריתם ה  $soft\ SVM$  אשר מרחיב את האלגוריתם גם למקרה שבו המדגם אינו פריד לינארית.

## תזכורת - גאומטריה של המישור

לפני שנציג את אלגוריתם ה  $SVM$  נתעכב לרגע על המשוואת המישור  $w^T x + b = 0$  כדי לקבל קצת אינטואיציה לגבי התפקיד של  $w$  ו  $b$  במשוואה זו.

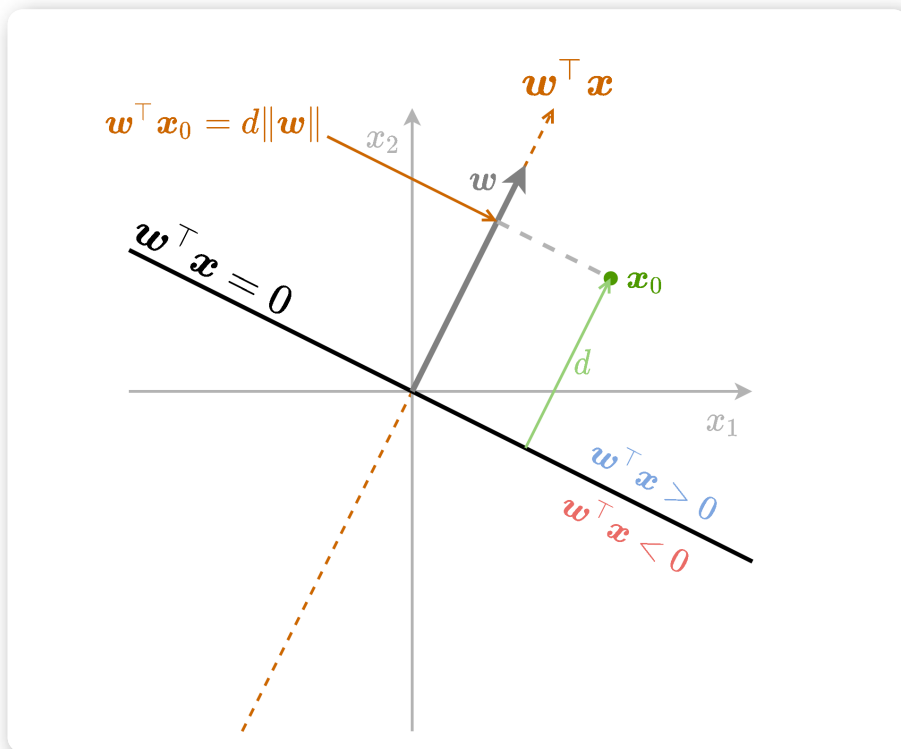
נתחיל מלהסתכל על גרסה פשוטה יותר של משוואת המישור שבה אין איבר היסט (איבר חופשי) ו  $w$  הוא וקטור יחידה  $\hat{w}$ . ספציפית, נסתכל על הפונקציה  $f(x) = \hat{w}^T x$ . משוואה זו מטילה נקודות במרחב על המישור המוגדר על ידי  $\hat{w}$  ומודד את המרחק של הטלה זו.



פונקציה זו למעשה מודדת את המרחק  $d$  של הנקודה כל שהיא  $x$  מהמישור של  $\hat{w}^\top x$  בתוספת של סימן אשר מציין את הצד של המישור בו נמצאת הנקודה. בחצי המרחב שעליו מצביע הוקטור  $\hat{w}$  הפונקציה חיובית והיא שלילית בחצי השני.

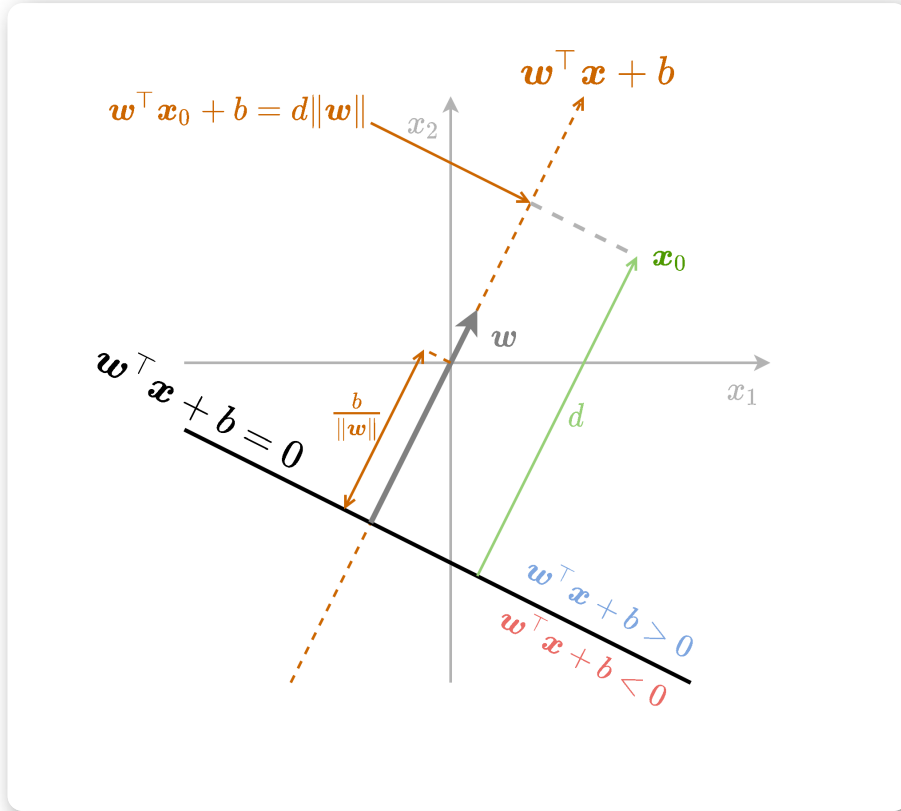
נשתמש בשם **signed distance (מרחק מסומן)** כדי להתייחס לשילוב של המרחק מהמישור בתוספת הסימן המתאים לצד של המישור.

כעת נחליף את הוקטור  $\hat{w}$  בוקטור  $w$  כל שהוא (שאינו וקטור יחידה). נקבל את הפונקציה  $f(x) = w^\top x$  שזהה לפונקציה הקודמת רק מוכפלת ב  $\|w\|_2$  (נורמת  $l_2$  של  $w$ . בחלק מהמקרים נשמיט את ה 2 ונרשום רק  $\|w\|$ ):



במקרה זה ה signed distance של נקודה כל שהיא  $\mathbf{x}_0$  מהמישור יהיה  $d = \frac{1}{\|\mathbf{w}\|} \mathbf{w}^\top \mathbf{x}_0$ .

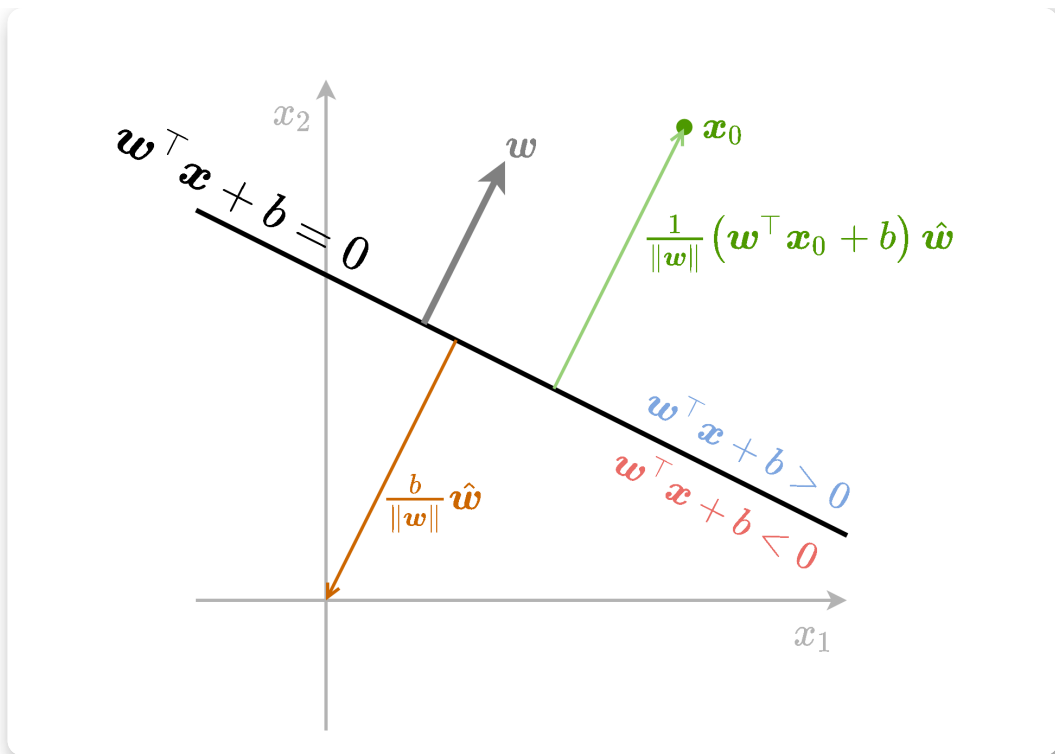
כאשר נוסיף לפונקציה גם איבר היסט  $b$  נקבל את הפונקציה  $f(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + b$ . ההוספה של הקבוע למעשה שקולה להזזה של נקודת ה-0 שממנה מודדים את ההטלה בכיוון ההפוך ל  $\mathbf{w}$ . המרחק שבו צריך להזיז את האפס לאורך הקו הינו  $\frac{b}{\|\mathbf{w}\|}$ .



כאן ה signed distance של נקודה כל שהיא  $\mathbf{x}_0$  ממישור  $\mathbf{w}^\top \mathbf{x} + b = 0$  הינו:

$$d = \frac{1}{\|\mathbf{w}\|} (\mathbf{w}^\top \mathbf{x}_0 + b)$$

נסכם את כל הנאמר לעיל בשרטוט הבא:



## אינוריאנטיות לכפל בסקלר

תכונה נוספת של על-מישורים הינה שהם אינוריאנטים (לא משתנים) תחת כפל בסקלר. זאת אומרת שאם נכפיל את גם את  $w$  וגם את  $b$  בקבוע כל שהוא  $\alpha$  שונה מאפס לא נשנה את מיקומו של המישור במרחב, זאת משום ש:

$$\begin{aligned} (\alpha w)^\top x + (\alpha b) &= 0 \\ \Leftrightarrow w^\top x + b &= 0 \end{aligned}$$

המשמעות של אינוריאנטיות זו הינה שיש מספר דרכים להגדיר את אותו המשווה הלינארי. לעובדה זו תהיה משמעות כאשר ננסה לרשום את בעיית האופטימיזציה שנרצה לפתור על מנת למצוא את מישור הפרדה הטוב ביותר.

## (Support Vector Machine (SVM

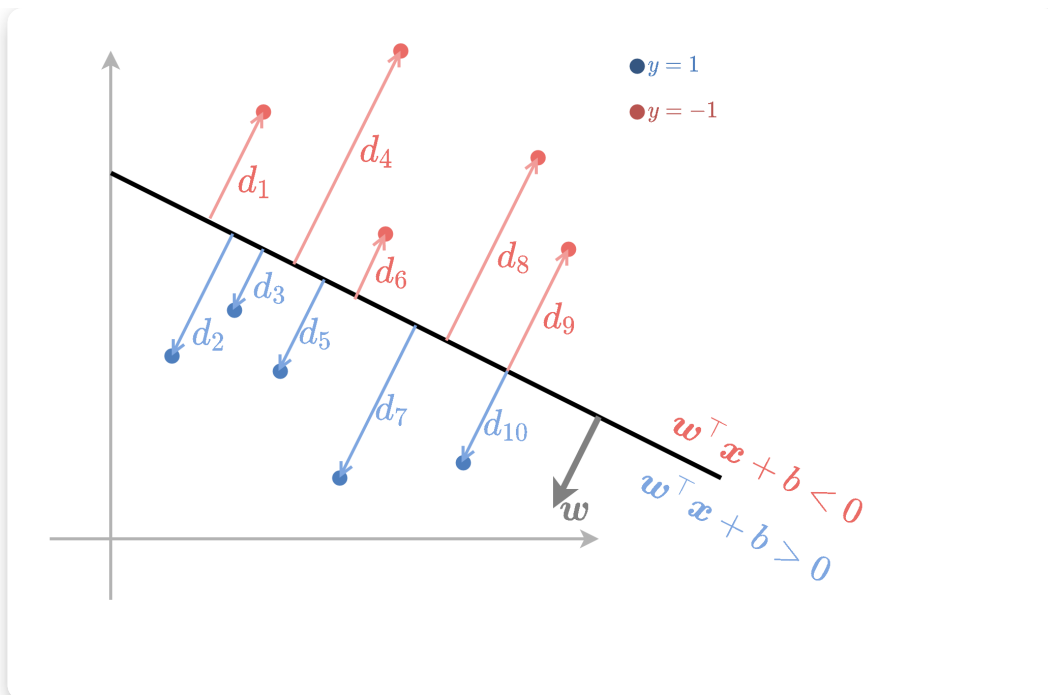
SVM הוא אלגוריתם דיסקרימינטיבי לסיווג בינארי אשר מחפש מסווג לינארי אשר יסווג בצורה טובה את המדגם. לאגוריתם זה יש שני גרסאות hard SVM אשר מחפש מסווג לינארי טוב לסיווג מדגם שהוא פריד לינארית, ו soft SVM אשר מרחיב את האלגוריתם למקרה שבו המדגם לא פריד לינארית. נתחיל בלהציג את hard SVM.

## Hard SVM

כפי שצינו קודם, במקרה שבו המדגם הינו פריד לינארית ישנו יותר ממישור הפרדה אחד אשר מסווג את המדגם באופן מושלם. נרצה למצוא מישור הפרדה אשר יכליל בצורה טובה גם לנקודות מחוץ למדגם. הנחה סבירה הינה שהפילוג הצפוי של הנקודות יתרכז באופן גס סביב הנקודות מהמדגם.

תחת היגיון זה hard SVM מנסה למצוא מישור הפרדה אשר יהיה רחוק ככל האפשר מהנקודות שבמדגם, או באופן יותר מדויק, נרצה שהמרחק מהמישור לנקודה הקרובה אליו ביותר יהיה מקסימאלי. נגדיר זאת באופן מתימטי.

נסתכל על המכפלה בין המרחקים המסומנים של הנקודות לתוויות שלהם:  $\frac{1}{\|w\|} (w^\top x^{(i)} + b) y^{(i)}$ . כדי לקבל סיווג מושלם נרצה שכל המכפלות האלה יהיו חיוביות. ב Hard SVM, בנוסף לניסיון לגרום לכל המכפלות להיות חיוביות, ננסה למקסם את המינימום של מכפלות אלו. דרישה זו תנסה להרחיק כמה שאפשר את הנקודות מהמישור.



בעיית האופטימיזציה שנרצה לפתור אם כן הינה:

$$\mathbf{w}^*, b^* = \arg \max_{\mathbf{w}, b} \min_i \left\{ \frac{1}{\|\mathbf{w}\|} (\mathbf{w}^\top \mathbf{x}^{(i)} + b) y^{(i)} \right\}$$

באופן כללי ניתן לנסות לפתור בעיה זו באופן ישיר על ידי אלגוריתם כמו gradient descent. בפועל העובדה שבבעיה מופיע  $\min$  על כל המדגם מאד מקשה לממש פתרון שיתכנס בזמן סביר. למזלנו ניתן לפשט את הבעיה ולמצוא בעיה שקולה, שאותה נכנה **הבעיה הפרימאלית**, שאותה שניתן יהיה לפתור באופן יעיל בשיטות נומריות אחרות.

## הפיתוח של הבעיה הפרימאלית

כפי שציינו קודם המשוואת המישור היא אינווריאנטית לכפל בקבוע. זאת אומרת שבבעיית האופטימיזציה נוכל לבחור באופן שרירותי קבוע כפלי להכפיל בו את  $\mathbf{w}$  ו  $b$  מבלי לפגוע במרחב של המסווגים שמתוכם אנו מחפשים את הפתרון. בפרט נוכל להוסיף דרישה ש:

$$\min_i \left\{ (\mathbf{w}^\top \mathbf{x}^{(i)} + b) y^{(i)} \right\} = 1$$

אם נוסיף את האילץ הזה לבעיית האופטימיזציה נקבל:

$$\begin{aligned}
\mathbf{w}^*, b^* &= \arg \max_{\mathbf{w}, b} \min_i \left\{ \frac{1}{\|\mathbf{w}\|} (\mathbf{w}^\top \mathbf{x}^{(i)} + b) y^{(i)} \right\} \\
&\text{s.t. } \min_i \left\{ (\mathbf{w}^\top \mathbf{x}^{(i)} + b) y^{(i)} \right\} = 1 \\
&= \arg \max_{\mathbf{w}, b} \min_i \frac{1}{\|\mathbf{w}\|} \left\{ (\mathbf{w}^\top \mathbf{x}^{(i)} + b) y^{(i)} \right\} \\
&\text{s.t. } \min_i \left\{ (\mathbf{w}^\top \mathbf{x}^{(i)} + b) y^{(i)} \right\} = 1 \\
&= \arg \max_{\mathbf{w}, b} \frac{1}{\|\mathbf{w}\|} \\
&\text{s.t. } \min_i \left\{ (\mathbf{w}^\top \mathbf{x}^{(i)} + b) y^{(i)} \right\} = 1 \\
&= \arg \min_{\mathbf{w}, b} \frac{1}{2} \|\mathbf{w}\|^2 \\
&\text{s.t. } \min_i \left\{ (\mathbf{w}^\top \mathbf{x}^{(i)} + b) y^{(i)} \right\} = 1
\end{aligned}$$

נוכל לפשט אף יותר את בעיית האופטימיזציה על ידי כך שנחליף את האילוץ של  $\min_i \{(\mathbf{w}^\top \mathbf{x}^{(i)} + b) y^{(i)}\} = 1$  באילוץ:

$$(\mathbf{w}^\top \mathbf{x}^{(i)} + b) y^{(i)} \geq 1 \quad \forall i$$

מכיוון שבעיית האופטימיזציה מנסה להקטין את הגודל של  $\mathbf{w}$  מובטח שלפחות עבור אחת מהדגימות במדגם האילוץ יתקיים בשיוון (אחרת אז ניתן להקטין עוד את  $\mathbf{w}$  ו  $b$  ובכך להקטין את ה objective), דבר אשר יגרור שיתקיים  $\min_i \{(\mathbf{w}^\top \mathbf{x}^{(i)} + b) y^{(i)}\} = 1$ . אם כן, בעיית האופטימיזציה הבאה שקולה לבעיה שמימנה התחלנו:

$$\begin{aligned}
\mathbf{w}^*, b^* &= \arg \min_{\mathbf{w}, b} \frac{1}{2} \|\mathbf{w}\|^2 \\
&\text{s.t. } (\mathbf{w}^\top \mathbf{x}^{(i)} + b) y^{(i)} \geq 1 \quad \forall i
\end{aligned}$$

הבעיה החדשה שקיבלנו נקראת **הבעיה הפרימאלית** ויש לה צורה מאד מיוחדת המוכנה **quadratic programming problem** והיה ניתנת לפתרון בשיטות נומריות מאד יעילות אותם לא נכסה בקורס זה. ניתן כעת פרשנות אינטואיטיבית לבעיה הפרימאלית.

## פרשנות

האילוץ בבעיית האופטימיזציה שקיבלנו דורשת שבעבור  $\mathbf{w}$  ו  $b$  כל שהם כל הנקודות במדגם יהיו מסווגות בצורה נכונה ועליהם להימצא מחוץ לתחום של:

$$1 \geq \mathbf{w}^\top \mathbf{x} + b \geq -1$$

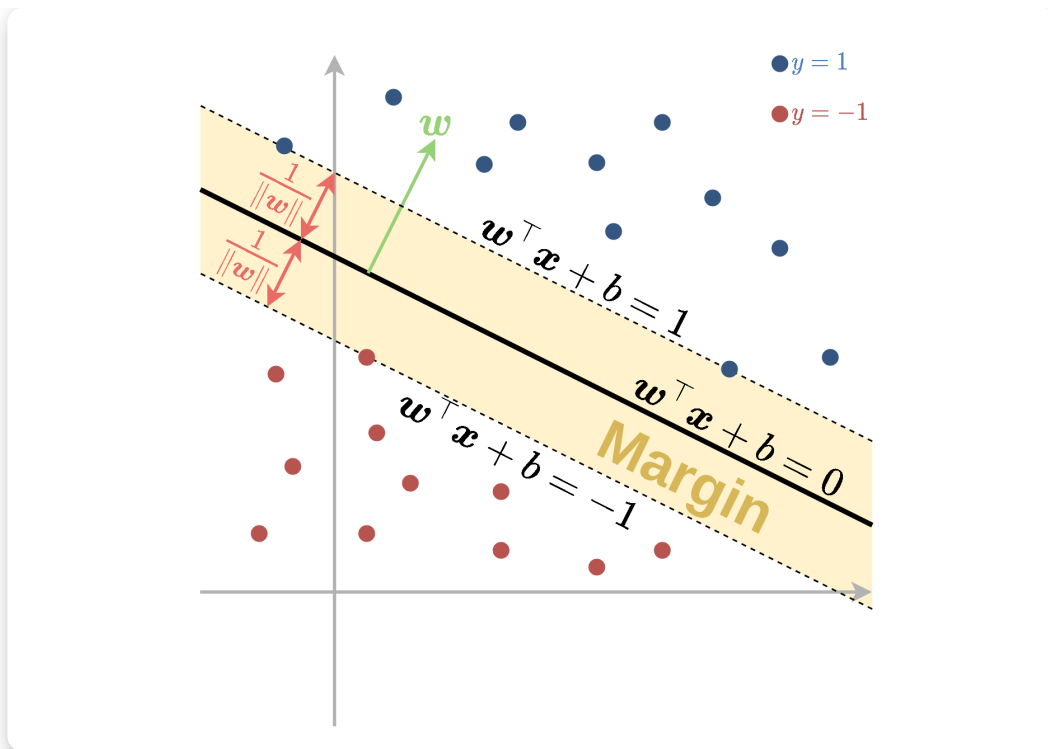
תחום זה אשר נמצא בין שני המישורים:

$$\mathbf{w}^\top \mathbf{x} + b = 1$$

1

$$\mathbf{w}^\top \mathbf{x} + b = -1$$

נקרא **השוליים (margin)** של המסווג ותפקידו להרחיק את הנקודות של המדגם ממישור ההפרדה.



המרחק בין מישור ההפרדה לשפה של ה margin שווה ל  $\frac{1}{\|w\|}$  והרוחב הכולל של ה margin הינו  $\frac{2}{\|w\|}$ . בעיית האופטימיזציה, שמנסה למזער את  $\|w\|$ , תחת האילוץ שתכנס לפרמטרים של המסווג בעל ה margin הגדול ביותר אשר מקיים תנאי זה.

## Support Vectors

משום ש SVM מנסה להרחיק את הנקודות הקרובות ביותר למישור נקבל שלרוב רק חלק קטן מאד של הנקודות ישפיעו על הבחירה של מישור ההפרדה. לדוגמא, נקודות שכנראה לא ישפיעו על מישור ההפרדה הם כאלה שנמצאות רחוק מהאיזור של התפר בין שני המחלקות. בפועל הנקודות היחידות אשר ישפיעו על התוצאה של בעיית האופטימיזציה הן הנקודות שבסופו של דבר ישבו על השפה של ה margin. נקודות אלו מקיימות  $y^{(i)}(w^T x^{(i)} + b) = 1$  והן מכונות **support vectors**. הסרה או הזזה אינפיניטסימאלית של נקודות שאינם support vectors לא תשפיע על הפתרון של בעיית האופטימיזציה.

## הבעיה הדואלית

נציג כעת דרך שקולה נוספת לרישום של בעיית האופטימיזציה אשר מכונה **הבעיה הדואלית**. בעיה זו לפעמים נוחה יותר לשימוש והיא גם תשרת אותנו בהמשך כאשר נציג את שיטת הגרעין. המעבר מהבעיה הפרימאלית נעשה על ידי שיטה המכונה תנאי **Karush-Kuhn-Tucker** אשר גם אותה לא נציג בקורס זה. נציג אבל את בעיית האופטימיזציה עצמה. הסבר מפורט יותר על אופטימיזציה של בעיות קמורות בהקשר של SVM והמעבר לבעיה הדואלית ניתן למצוא בנספח בנושא באתר המודל של הקרוס.

בבעיה זו נגדיר  $N$  משתני עזר נוספים  $\{\alpha_i\}_{i=1}^N$ . בעזרת משתנים אלו ניתן לרשום את הבעיה הדואלית באופן הבא:

$$\begin{aligned} \{\alpha_i\}^* &= \arg \max_{\{\alpha_i\}} \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} y^{(i)} y^{(j)} \alpha_i \alpha_j \mathbf{x}^{(i)\top} \mathbf{x}^{(j)} \\ \text{s.t.} \quad &\alpha_i \geq 0 \quad \forall i \\ &\sum_i \alpha_i y^{(i)} = 0 \end{aligned}$$

מתוך המשתנים  $\{\alpha_i\}_{i=1}^N$  ניתן לשחזר את  $w$  אופן הבא:

$$w = \sum_i \alpha_i y^{(i)} \mathbf{x}^{(i)}$$



ישנו קשר בין הערכים של  $\alpha_i$  לנקודות שהם support vectors. בפרט, בעבור נקודות שאינם support vectors יתקיים ש  $\alpha_i = 0$ . (בנוסף, במרבית המקרים, אם כי לא תמיד, בעבור נקודות שהם כן support vectors יתקיים ש  $\alpha_i > 0$ ).

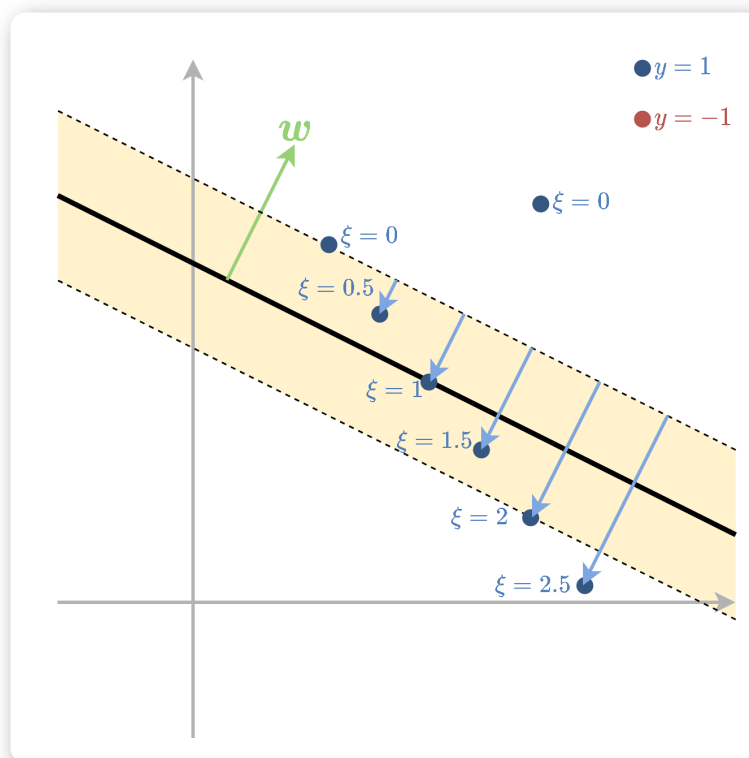
נסכם זאת בטבלה הבאה:

$\alpha_i = 0$	$y^{(i)} (\mathbf{w}^\top \mathbf{x}^{(i)} + b) > 1$	נקודות רחוקות מה margin
$\alpha_i \geq 0$	$y^{(i)} (\mathbf{w}^\top \mathbf{x}^{(i)} + b) = 1$	נקודות על ה margin (שהם support vectors)

על מנת לחשב את הפרמטר  $b$  של המישור נוכל לבחור נקודה מסוימת שבעבור ה  $\alpha_i > 0$ . נקודה כזו בהכרח תהיה support vectors ולכן היא תקיים  $y^{(i)} (\mathbf{w}^\top \mathbf{x}^{(i)} + b) = 1$ , מתוך משוואה זו ניתן לחלץ את  $b$ .

## Soft SVM

Soft SVM מתייחס למקרה שבו המדגם אינו פריד לינארית. במקרה זה עדיין מגדירים את השוליים בצורה דומה אך מאפשרים לנקודות המדגם להיכנס לתוך השוליים ואף לחצות אותם לצד הלא נכון של מישור ההפרדה. על כל חריגה כזו משלמים קנס בפונקציית המטרה שאותו מנסים למזער. את החריגה של הדגימה ה  $i$  נסמן ב  $\xi_i$ . לנקודות שהם בצד הנכון של המישור ומחוץ ל  $\xi_i$  יהיה 0.



המשתנים  $\xi_i$  נקראים **slack variables** ובעיית האופטימיזציה הפרימאלית תהיה

$$\mathbf{w}^*, b^*, \{\xi_i\}^* = \arg \min_{\mathbf{w}, b, \{\xi_i\}} \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^N \xi_i$$

$$\text{s.t. } y^{(i)} (\mathbf{w}^\top \mathbf{x}^{(i)} + b) \geq 1 - \xi_i \quad \forall i$$

$$\xi_i \geq 0 \quad \forall i$$

כאשר  $C$  הוא היפר-פרמטר אשר קובע את גודל הקנס בפונקציית המחיר על כל חריגה.

$$\begin{aligned} \{\alpha_i\}^* = \arg \max_{\{\alpha_i\}} & \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} y^{(i)} y^{(j)} \alpha_i \alpha_j \mathbf{x}^{(i)\top} \mathbf{x}^{(j)} \\ \text{s.t.} & 0 \leq \alpha_i \leq C \quad \forall i \\ & \sum_i \alpha_i y^{(i)} = 0 \end{aligned}$$

ה support vectors מוגדרים להיות הנקודות שמקיימות  $y^{(i)} (\mathbf{w}^\top \mathbf{x}^{(i)} + b) = 1 - \xi_i$  תכונות:

$\alpha_i = 0$	$y^{(i)} (\mathbf{w}^\top \mathbf{x}^{(i)} + b) > 1$	נקודות שמסוגות נכון ורחוקות מה margin
$0 \leq \alpha_i \leq C$	$y^{(i)} (\mathbf{w}^\top \mathbf{x}^{(i)} + b) = 1$	נקודות על ה margin (שהם support vectors)
$\alpha_i = C$	$y^{(i)} (\mathbf{w}^\top \mathbf{x}^{(i)} + b) = 1 - \xi_i$	נקודות שחורגות מה margin (גם support vectors)

כאשר המקרה האחרון כולל נקודות המסוגות נכון ולא נכון.

## פונקציות גרעין

### מאפיינים: תזכורת

נוכל תמיד להחליף את וקטור המשתנים  $\mathbf{x}$  שעליו פועל האלגוריתם בוקטור חדש  $\Phi(\mathbf{x})$ , כאשר  $\Phi$  היא פונקציה אשר נבחרה מראש ונקראת פונקציית המאפיינים שכן היא מחלצת מאפיינים רלוונטים מתוך  $\mathbf{x}$  שבהם נשתמש. אם הממד של  $\Phi$  מספיק גבוה, ניתן תמיד להגיע להפרדה לינארית במרחב הרב-ממדי (דורש הוכחה).

### פונקציות גרעין

במקרים רבים החישוב של  $\Phi(\mathbf{x})$  יכול להיות מסובך אך קיימת דרך לחשב בצורה יעילה את הפונקציה  $K(\mathbf{x}_1, \mathbf{x}_2) = \Phi(\mathbf{x}_1)^\top \Phi(\mathbf{x}_2)$  אשר נקראת פונקציית גרעין. יתרה מזאת, ייתכנו מצבים שבהם וקטור המאפיינים הוא אינסופי ועדיין פונקציית הגרעין היא פשוטה לחישוב.

ישנם קריטריונים תחתם פונקציה מסוימת  $K(\mathbf{x}_1, \mathbf{x}_2)$  היא פונקציית גרעין בעבור וקטור מאפיינים מסויים. בקורס זה לא נכנס לתאים אלו. נציג שתי פונקציות גרעין נפוצות:

- גרעין גאוסני:  $K(\mathbf{x}_1, \mathbf{x}_2) = \exp\left(-\frac{\|\mathbf{x}_1 - \mathbf{x}_2\|_2^2}{2\sigma^2}\right)$  כאשר  $\sigma$  פרמטר שיש לקבוע.
- גרעין פולינומיאלי:  $K(\mathbf{x}_1, \mathbf{x}_2) = (1 + \mathbf{x}_1^\top \mathbf{x}_2)^p$  כאשר  $p \geq 1$  פרמטר שיש לקבוע.

פונקציית המאפיינים שמתאימות לגרעינים אלו הם מסורבלות לכתיבה ולא נציג אותם כאן.

## Kernel Trick in SVM

הרעיון ב kernel trick הינו לעשות שימוש בפונקציית הגרעין על מנת להשתמש ב SVM עם מאפיינים מבלי לחשב את  $\Phi$  באופן ישיר. בעבור פונקציית מאפיינים  $\Phi$  עם פונקציית גרעין  $K$  הבעיה הדואלית של SVM הינה:

$$\begin{aligned} \{\alpha_i\}^* = \arg \max_{\{\alpha_i\}} & \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} y^{(i)} y^{(j)} \alpha_i \alpha_j K(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) \\ \text{s.t.} & \alpha_i \geq 0 \quad \forall i \\ & \sum_i \alpha_i y^{(i)} = 0 \end{aligned}$$

בעיית אופטימיזציה זו מגדירה את המשתנים  $\{\alpha_i\}$  בלי צורך לחשב את  $\Phi$  באופן מפורש בשום שלב.

הפרמטר  $w$  נתון על ידי:

$$w = \sum_i \alpha_i y^{(i)} \Phi(x^{(i)})$$

כדי לחשב את  $w$  באופן מפורש יש לחשב את  $\Phi$ , אך ניתן להמנע מכך עם מציבים את הנוסחא ל  $w$  ישירות לתוך המסווג:

כדי להמנע מהחישוב של  $\Phi$  גם במסווג נשתמש בעובדה ש:

$$\begin{aligned} h(x) &= \text{sign}(w^\top \Phi(x) + b) \\ &= \text{sign}\left(\sum_i \alpha_i y^{(i)} \Phi(x^{(i)})^\top \Phi(x) + b\right) \\ &= \text{sign}\left(\sum_i \alpha_i y^{(i)} K(x^{(i)}, x) + b\right) \end{aligned}$$

כך שגם בשלב החיזוי ניתן להשתמש בפונקציית הגרעין בלי לחשב את  $\Phi$  באופן מפורש.