

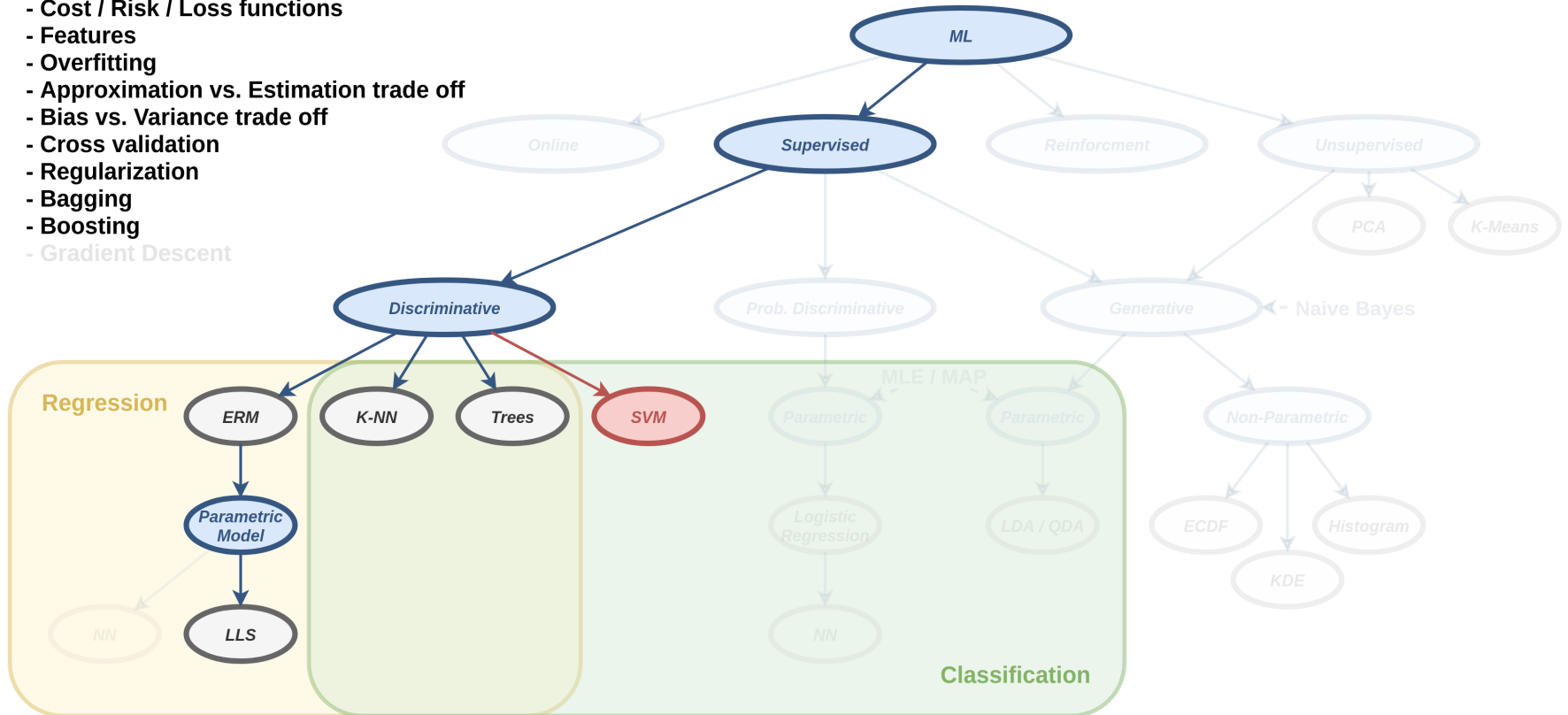
הרצאה 6 - SVM ושיטות

גרעין

Subjects Covered in this Course

General concepts:

- Cost / Risk / Loss functions
- Features
- Overfitting
- Approximation vs. Estimation trade off
- Bias vs. Variance trade off
- Cross validation
- Regularization
- Bagging
- Boosting
- Gradient Descent



- בפרק זה נעסוק בבעיית סיווג בינארי.

- נסמן את שתי המחלקות ב $y = \pm 1$.

- נעסוק במסווגים מהצורה:

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

- חלוקה של המרחב לשני צידיו של על-מישור (hyperplane):

$$\mathbf{w}^\top \mathbf{x} + b = 0$$

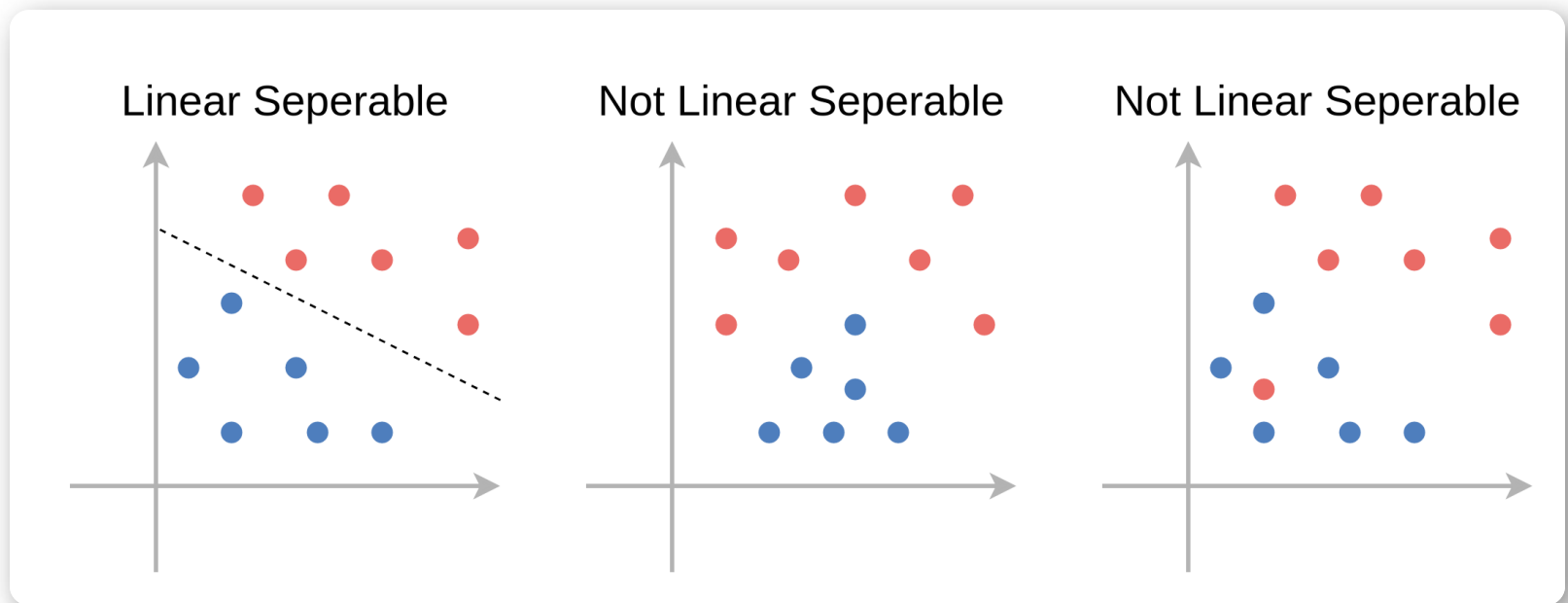
- מישור זה מכונה מישור ההפרדה.

על-מישור (hyperplane)

- הרחבה של מושג המישור למימדים שונים מ-2.
- במרחב ממימד D , על-המישור יהיה ממימד $D - 1$.
- בקורס זה נשתמש בשם מישור גם כדי להתייחס לעל-מישורים.
- לא להתבלבל בין $w^\top x + b = 0$ לבין $ax + b = y$.

פרידות לינארית (linear separability)

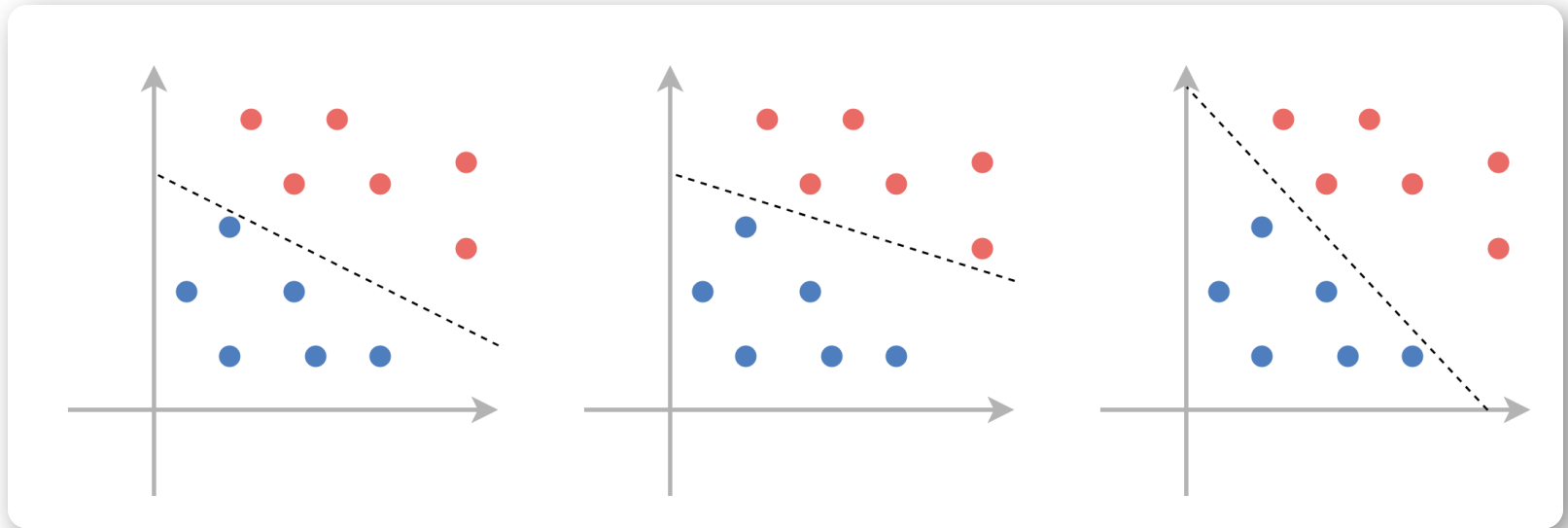
במקרה שבו קיים מישור מפריד אשר מסווג את המדגם בצורה מושלמת (בלי טעויות סיווג) נאמר שהמדגם **פריד לינארית**.



• לרוב לא נוכל לדעת מראש האם מדגם הוא פריד לינארית או לא.

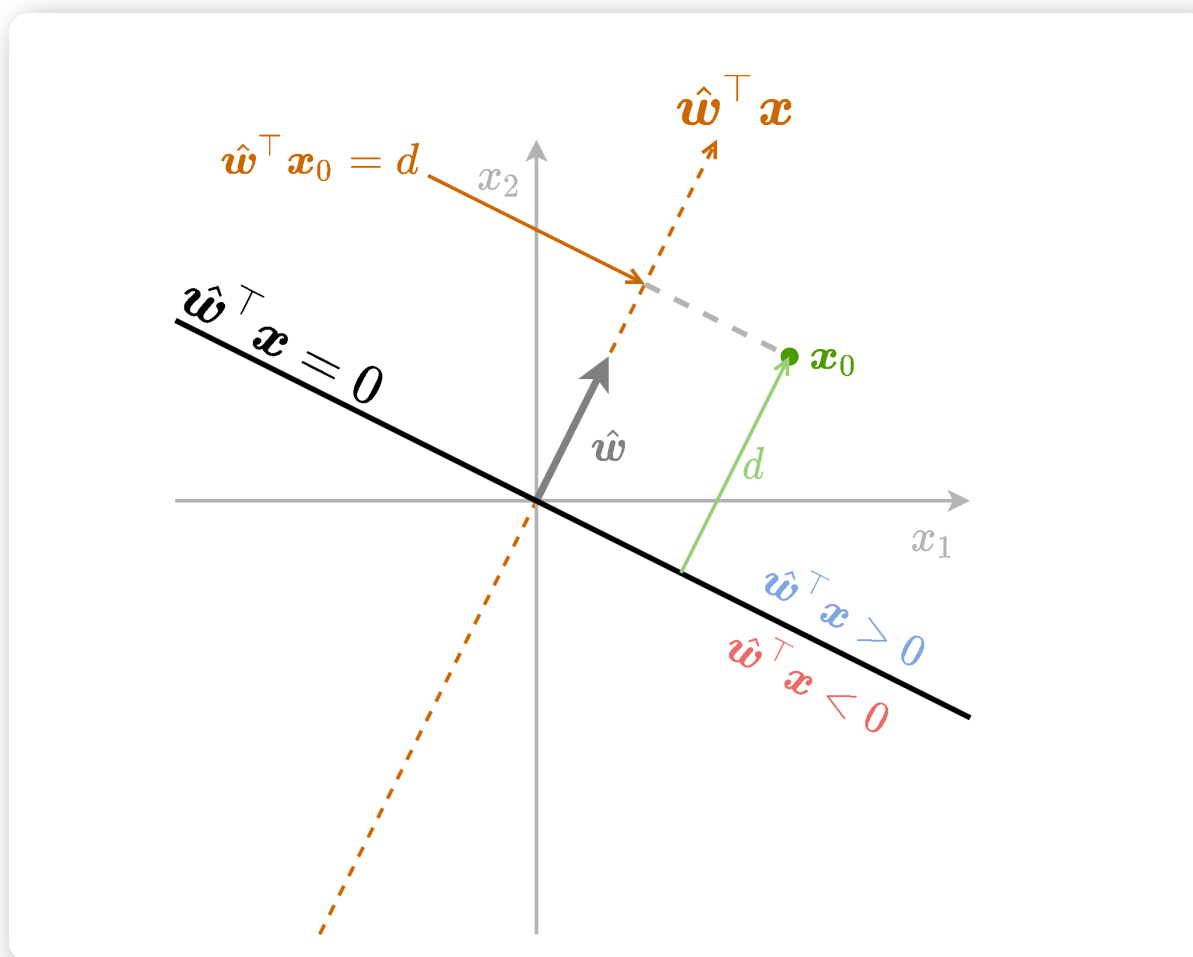
פרידות לינארית (linear separability)

למדגם פריד לינארית יהיה תמיד יותר ממשטח הפרדה אחד:

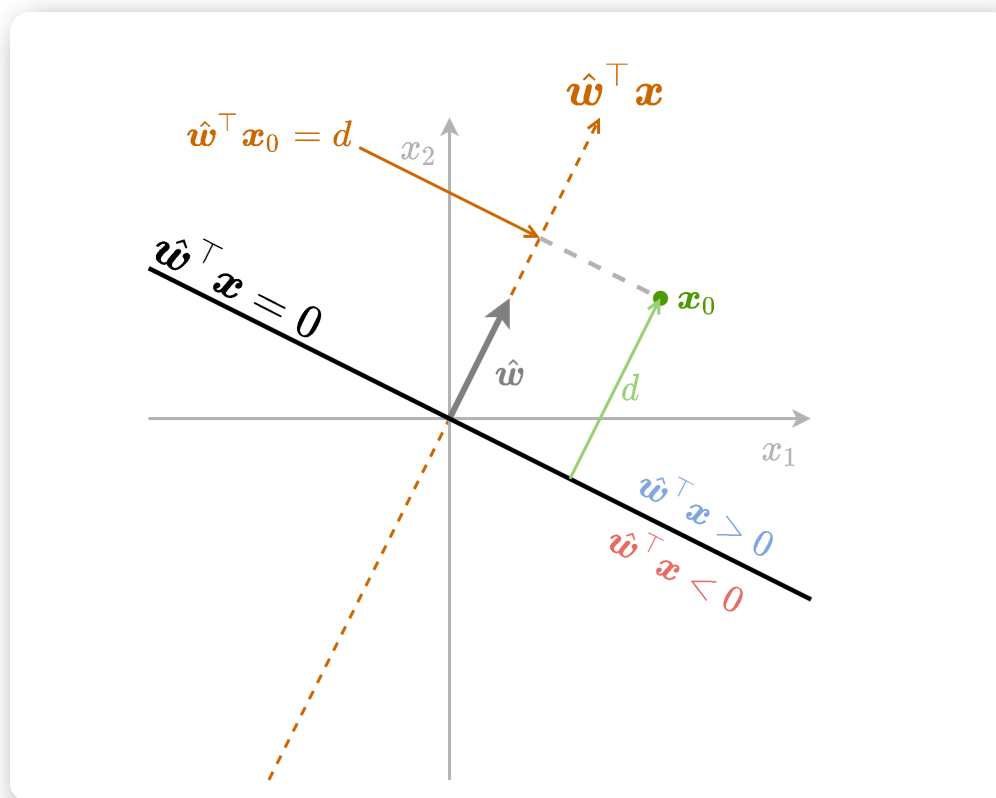


תזכורת - גאומטריה של המישור

נסתכל על הפונקציה $f(x) = \hat{w}^\top x$. משוואה זו מטילה נקודות במרחב על המישור המוגדר על ידי \hat{w} (וקטור יחידה בכיוון של w), ומודדת את האורך של הטלה זו.



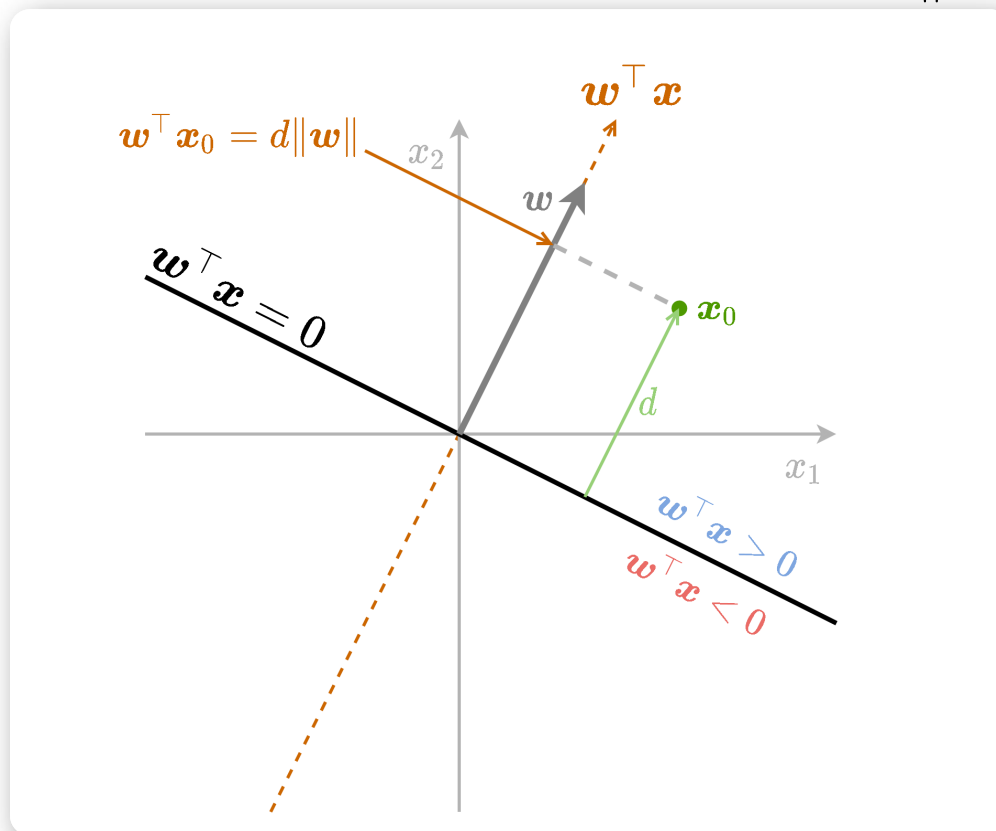
תזכורת - גאומטריה של המישור



- מודדת את המרחק מהמישור $\hat{w}^T x$ בתוספת של סימן אשר מציין את הצד של המישור.
- נשתמש בשם **signed distance** (מרחק מסומן) כדי להתייחס לשילוב של המרחק מהמישור בתוספת הסימן.

תזכורת - גאומטריה של המישור

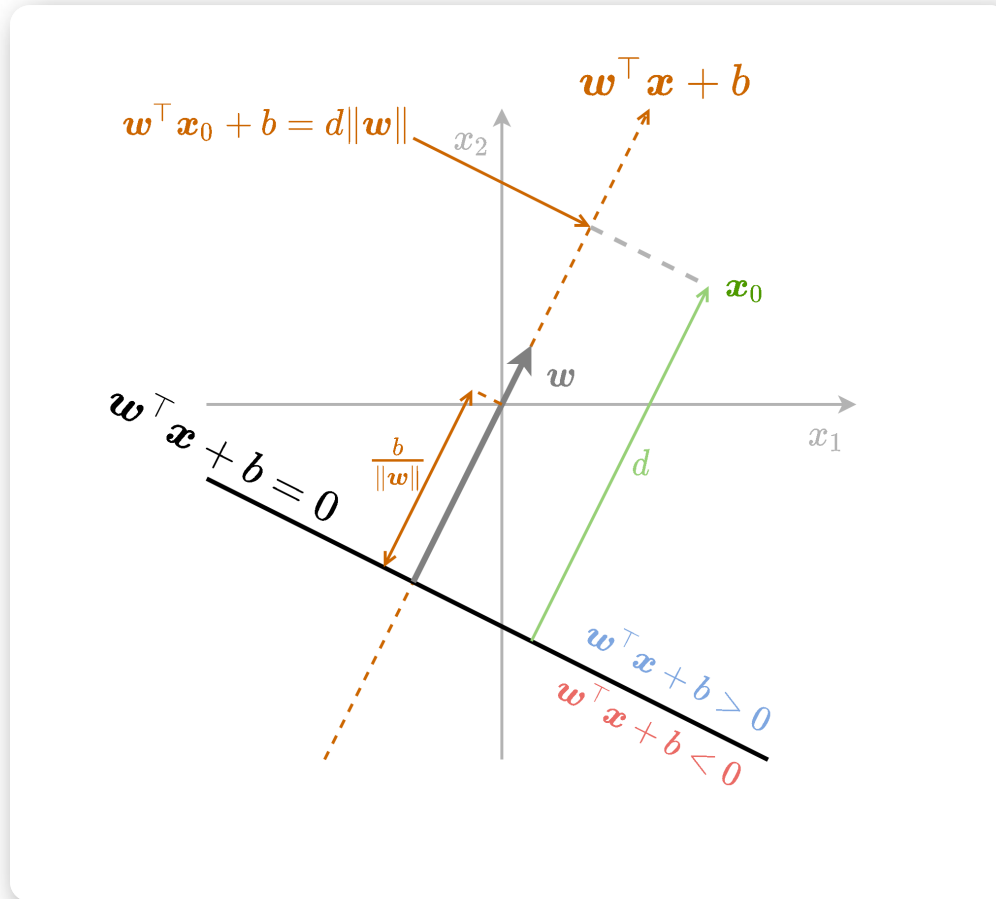
נחליף את הוקטור \hat{w} בוקטור w . נקבל פונקציה זחה המוכפלת ב $\|w\|_2$.



$d = \frac{1}{\|w\|} w^T x_0$ **signed distance** יהיה d .

תזכורת - גאומטריה של המישור

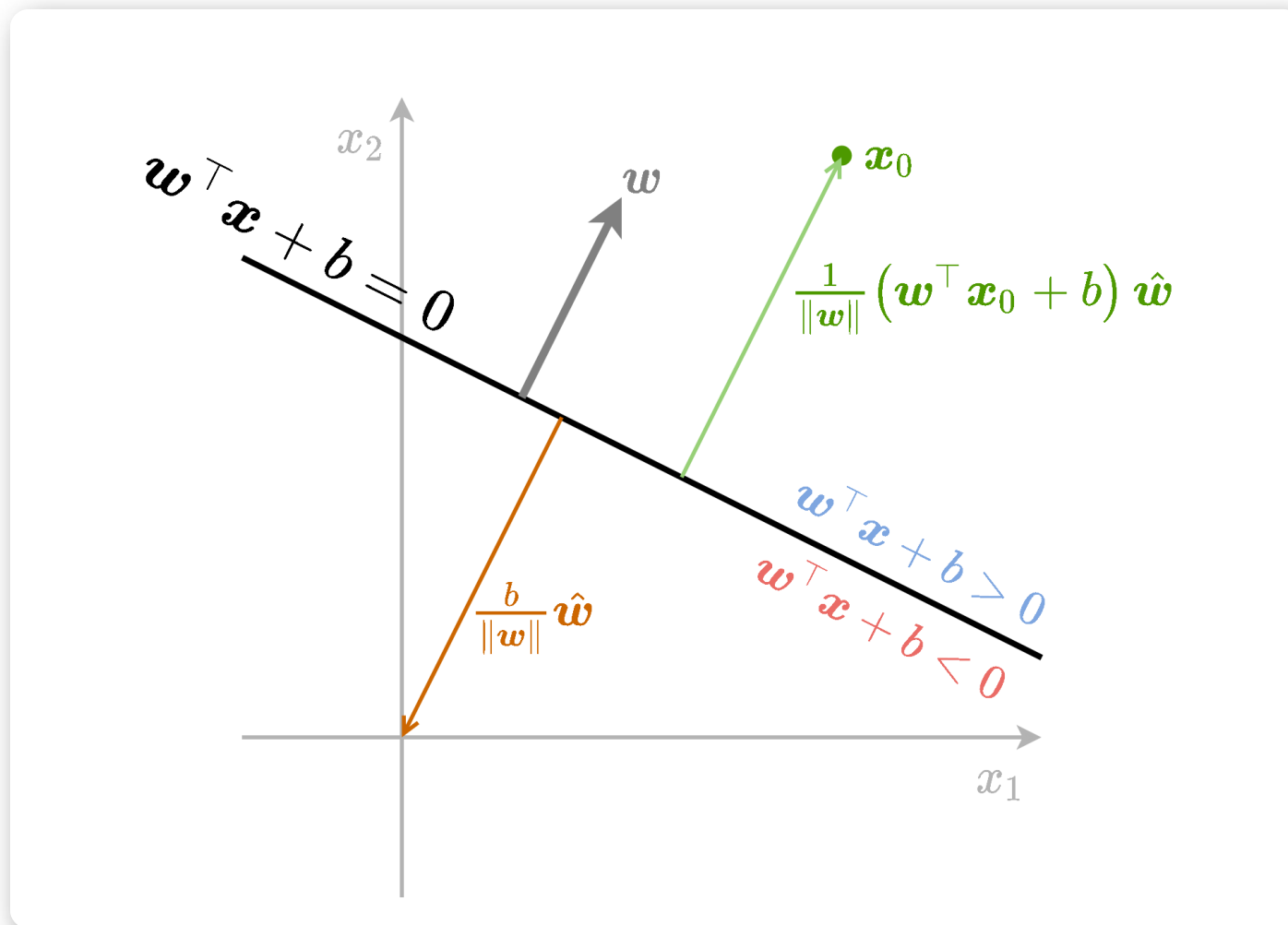
נוסיף לפונקציה גם איבר היסט b . ההוספה של הקבוע שקולה להזזה של נקודת ה-0.



$d = \frac{1}{\|w\|} (w^T x_0 + b)$ ה signed distance יהיה

תזכורת - גאומטריה של המישור

נסכם את כל הנאמר לעיל בשרטוט הבא:



אינווריאנטיות לכפל בסקלר

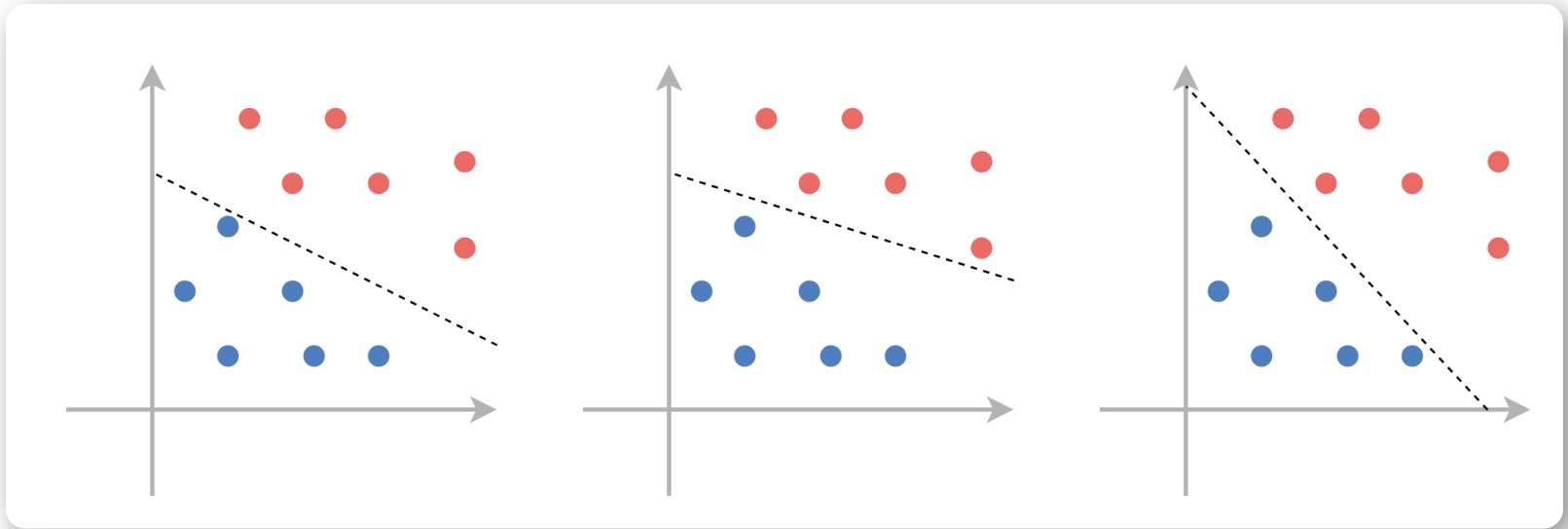
אם נכפיל את גם את w וגם את b בקבוע כל שהוא α שונה מאפס לא נשנה את מיקומו של המישור במרחב, זאת משום ש:

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

המשמעות הינה שיש מספר דרכים להגדיר את אותו מסווג לינארי.

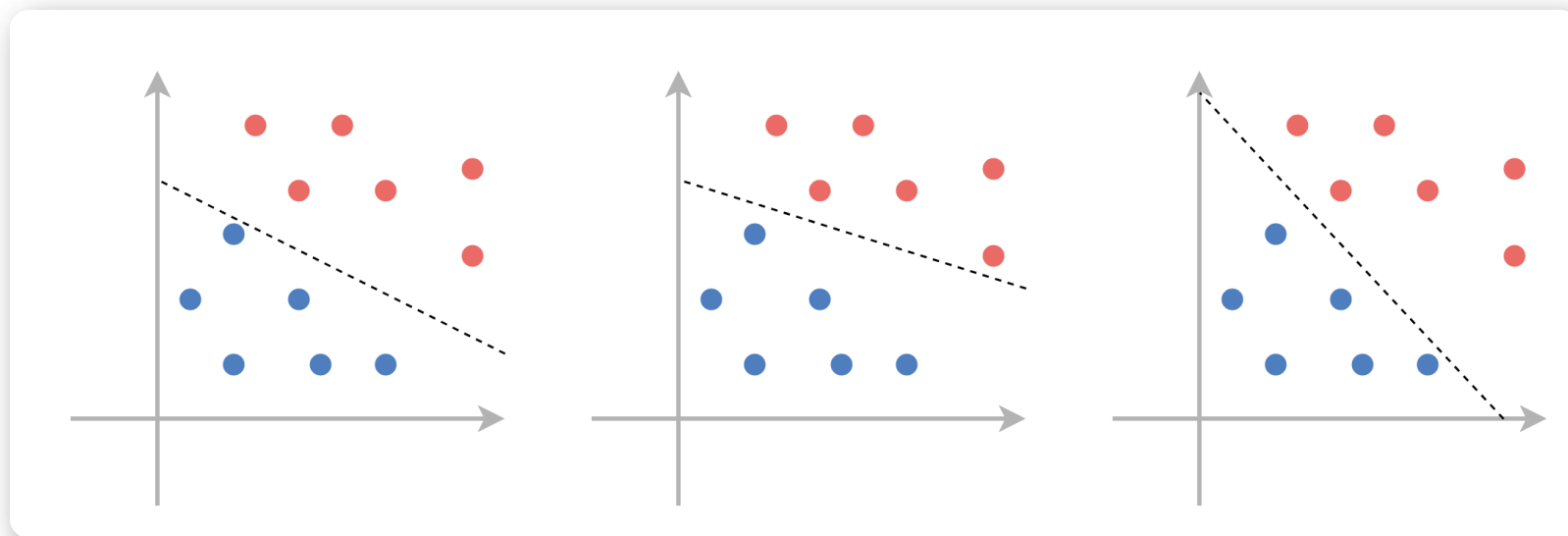
(Support Vector Machine (SVM

- אלגוריתם דיסקרימינטיבי לסיווג בינארי (מחפש מסווג טוב על המדגם).
- Hard SVM מחפש מסווג לינארי למדגם שהוא פריד לינארית.
- Soft SVM מרחיב את האלגוריתם למקרה שבו המדגם לא פריד לינארית.



- נרצה למצוא מישור הפרדה אשר יכליל בצורה טובה.

- הנחה סבירה הינה שהפילוג של הנקודות יתרכז סביב הנקודות מהמדגם.



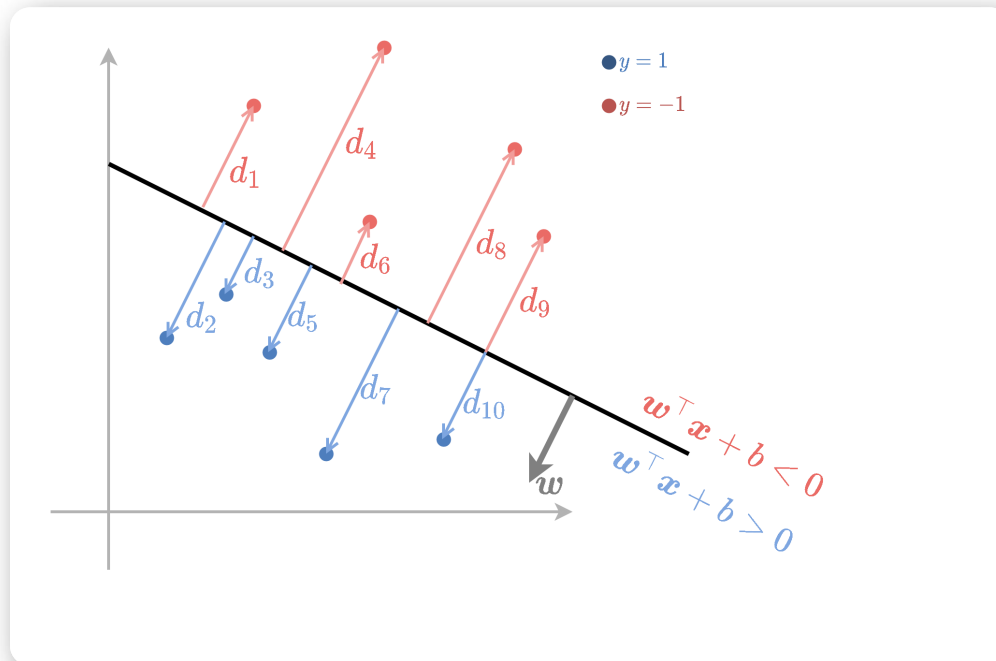
- **Hard SVM מנסה למצוא מישור הפרדה אשר יהיה רחוק ככל האפשר מהנקודות שבמדגם.**
- **או: נרצה שהמרחק מהמישור לנקודה הקרובה אליו ביותר יהיה מקסימאלי.**

שאלה: למה זה רעיון טוב אינטואיטיבית?

Hard SVM

נסתכל על המכפלה בין המרחקים המסומנים של הנקודות לתוויות שלהם $\frac{1}{\|w\|} (w^T x^{(i)} + b) y^{(i)}$.

- כדי לקבל סיווג מושלם נרצה שכל המכפלות יהיו חיוביות.
- בנוסף ננסה למקסם את המינימום של מכפלות אלו.



בעיית האופטימיזציה שנרצה לפתור אם כן הינה:

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

- ניתן לנסות לפתור באופן ישיר על ידי **gradient descent**.
- בפועל העובדה שבבעיה מופיע \min על כל המדגם מאד מקשה.
- ניתן לפשט את הבעיה ולמצוא בעיה שקולה, שאותה נכנה **הבעיה הפרימאלית**.
- את הבעיה הפרימאלית יהיה ניתן לפתור באופן יעיל בשיטות נומריות אחרות.

הפיתוח של הבעיה הפרימאלית

- נוכל לבחור באופן שרירותי קבוע כפלי להכפיל בו את w ו- b .
- בפרט נוכל להוסיף דרישה S :

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

הפיתוח של הבעיה הפרימאלית

אם נוסיף את האילוץ הזה לבעיית האופטימיזציה נקבל:

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

הפיתוח של הבעיה הפרימאלית

נוכל גם להחליף את האילוץ של $\min_i \{(\mathbf{w}^\top \mathbf{x}^{(i)} + b)y^{(i)}\} = 1$ באילוץ:

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

מובטח שלפחות עבור אחת מהנקודות האילוץ יתקיים בשיוויון: אם זה לא המצב, תמיד נוכל לכפול את w ו- b בקבוע חיובי קטן מספיק כך שהשיוויון יתקיים, וגם נשתפר בפתרון בעיית האופטימיזציה (שמנסה להקטין את $\|\mathbf{w}\|$).

הפיתוח של הבעיה הפרימאלית

קיבלנו את בעיית האופטימיזציה השקולה הבאה, היא הבעיה הפרימאלית:

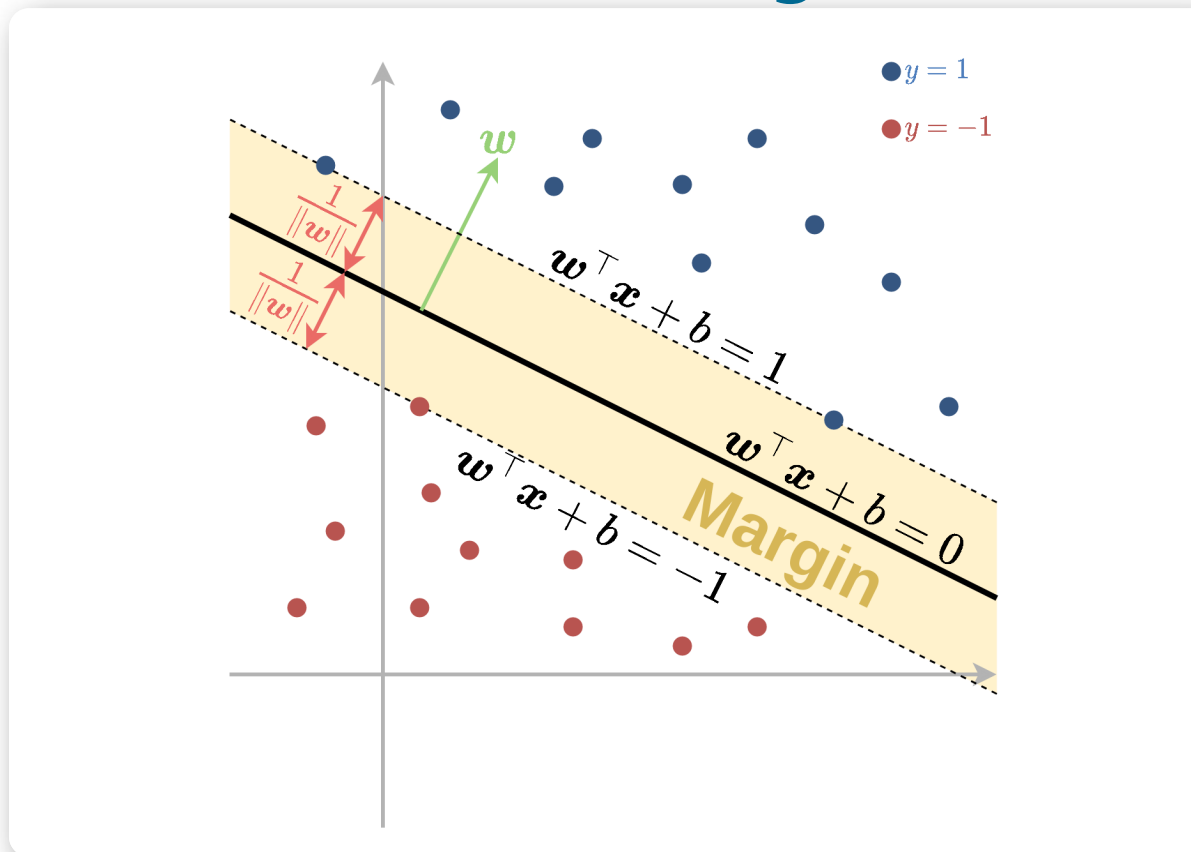
$$\begin{aligned} \boldsymbol{w}^*, b^* = \arg \min_{\boldsymbol{w}, b} \quad & \frac{1}{2} \|\boldsymbol{w}\|^2 \\ \text{s.t.} \quad & (\boldsymbol{w}^\top \boldsymbol{x}^{(i)} + b)y^{(i)} \geq 1 \quad \forall i \end{aligned}$$

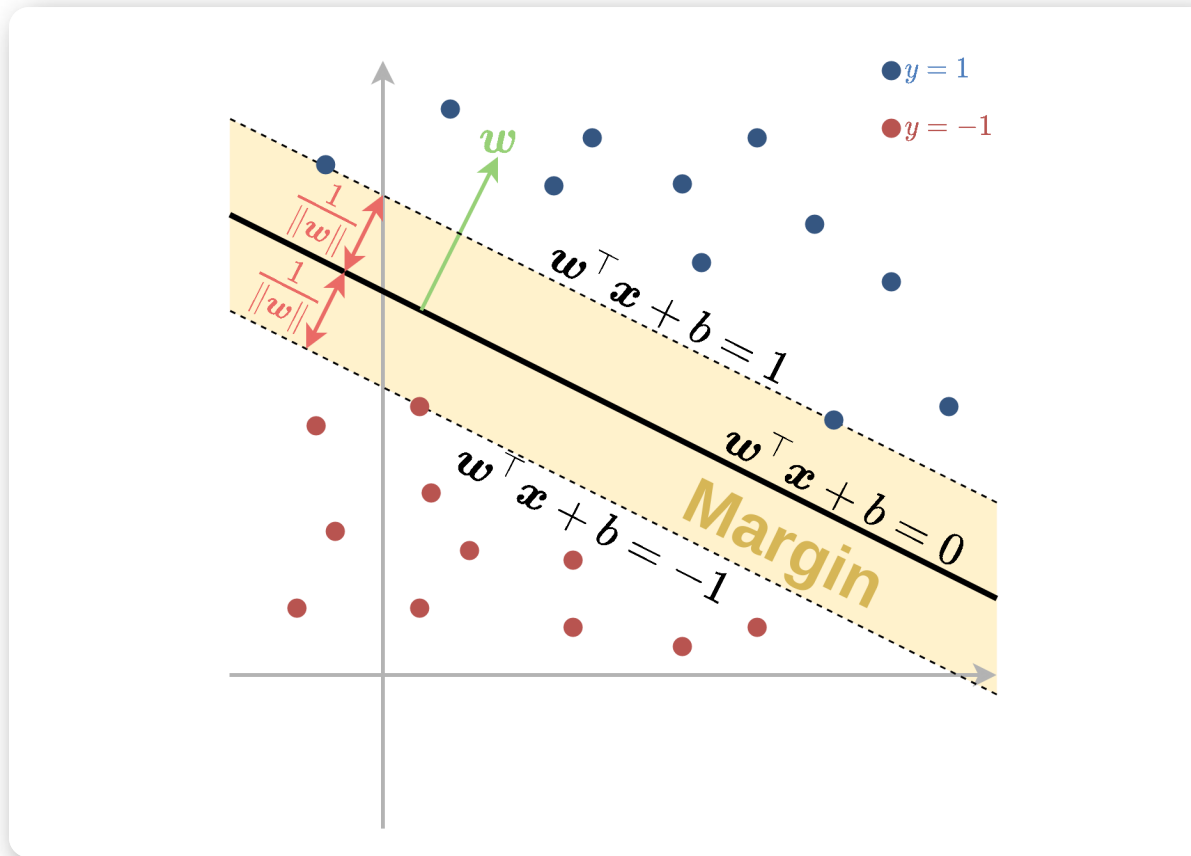
שימו לב למספר הגדול של האילוצים!

האילוץ דורש שהנקודות מהמדגם יהיו מחוץ לתחום:

$$1 > w^T x + b > -1$$

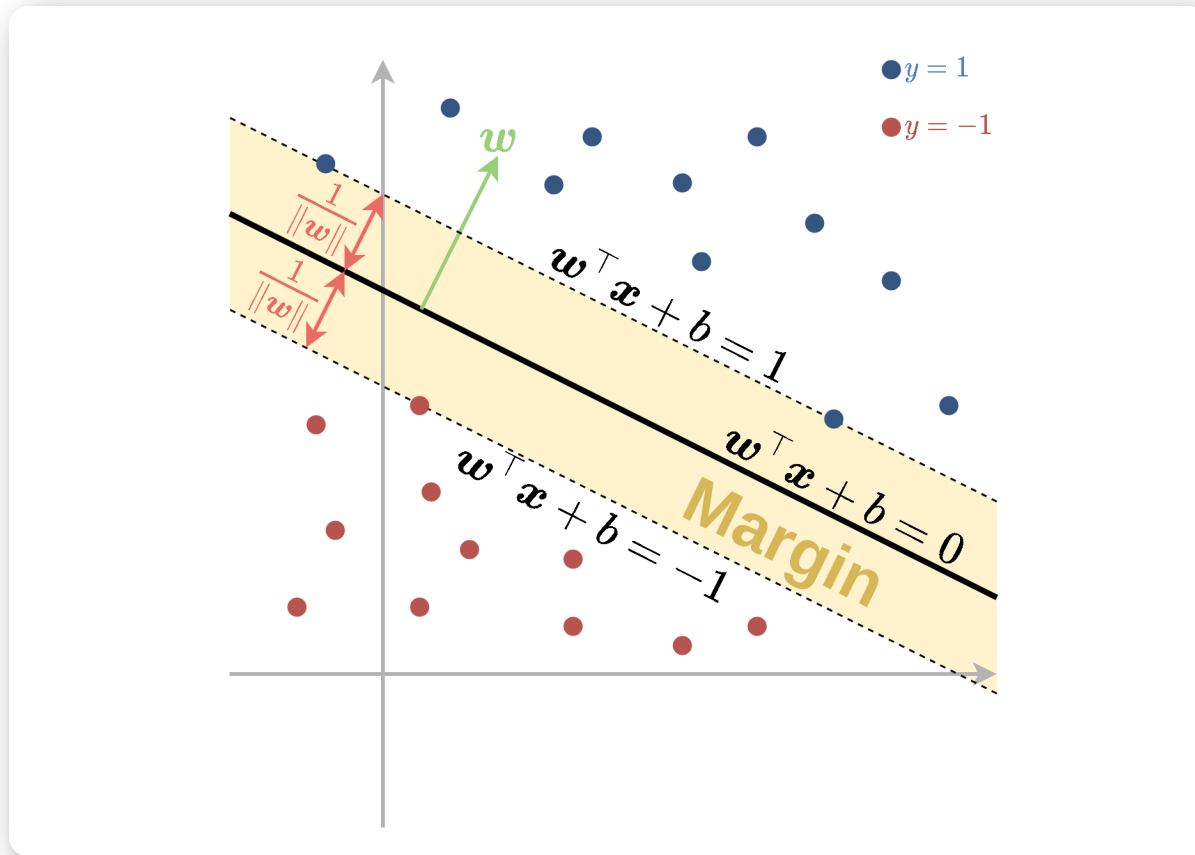
אשר נקרא השוליים (margin).





- המרחק בין מישור ההפרדה לשפה של ה margin שווה ל $\frac{1}{\|w\|}$

Support Vectors



• ה **support vectors** הן הנקודות שיושבות על ה-margin והן מקיימות $y^{(i)} (w^T x^{(i)} + b) = 1$.

• רק נקודות אלו ישפיעו על הפתרון של בעיית האופטימיזציה.

דרך שקולה נוספת לרישום של בעיית האופטימיזציה (ללא הוכחה):

נגדיר N משתני עזר נוספים $\{\alpha_i\}_{i=1}^N$ בעזרתם ניתן לרשום את הבעיה הדואלית באופן הבא:

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

יש מספר גדול של משתנים, אך מעט אילוצים. במודל ניתן למצוא נספח עם הסבר מפורט על אופטימיזציה קמורה בהקשר של SVM.

$$\{\alpha_i\}^* = \arg \max_{\{\alpha_i\}} \left[\sum_i \alpha_i - \frac{1}{2} \sum_{i,j} y^{(i)} y^{(j)} \alpha_i \alpha_j \mathbf{x}^{(i)\top} \mathbf{x}^{(j)} \right]$$

s.t. $\alpha_i \geq 0 \quad \forall i$

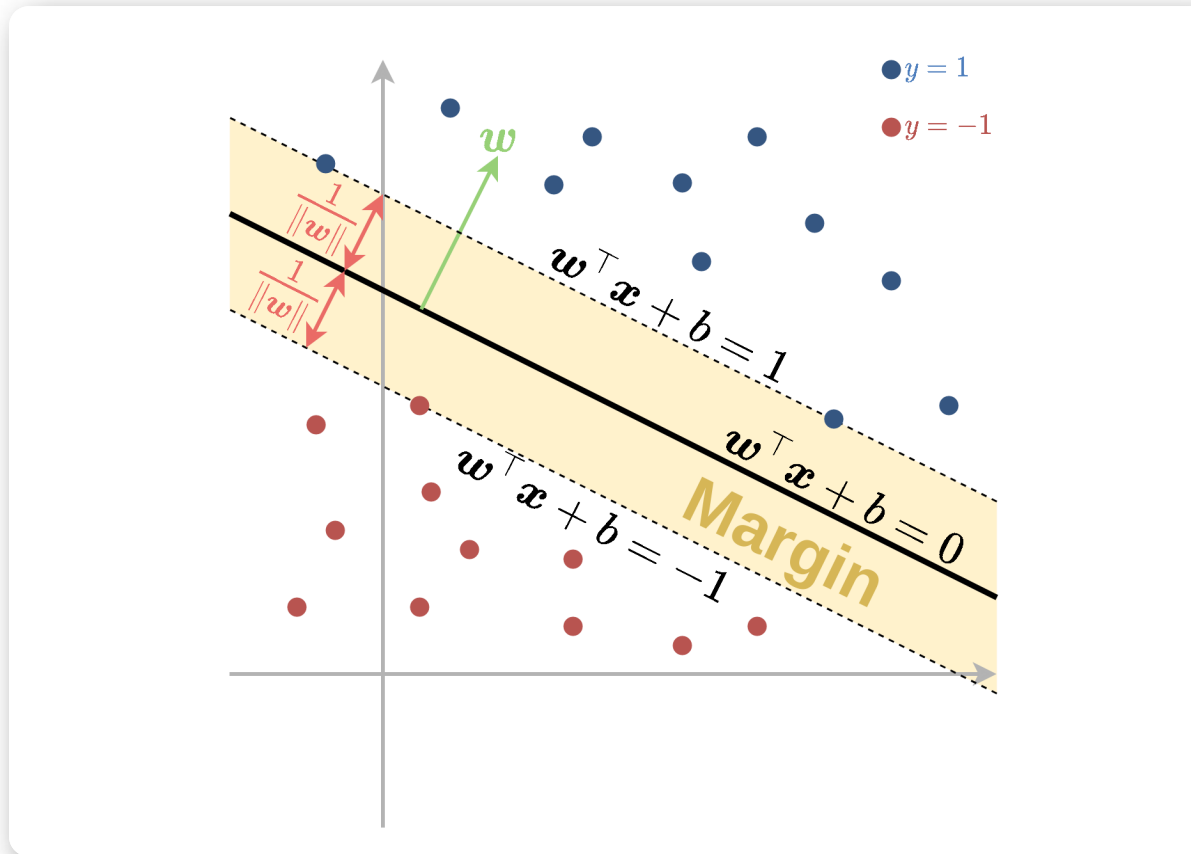
$$\sum_i \alpha_i y^{(i)} = 0$$

שימו לב שהתלות במאפיינים רק דרך מכפלות פנימיות.
מתוך המשתנים $\{\alpha_i\}_{i=1}^N$ ניתן לשחזר את w אופן הבא:

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

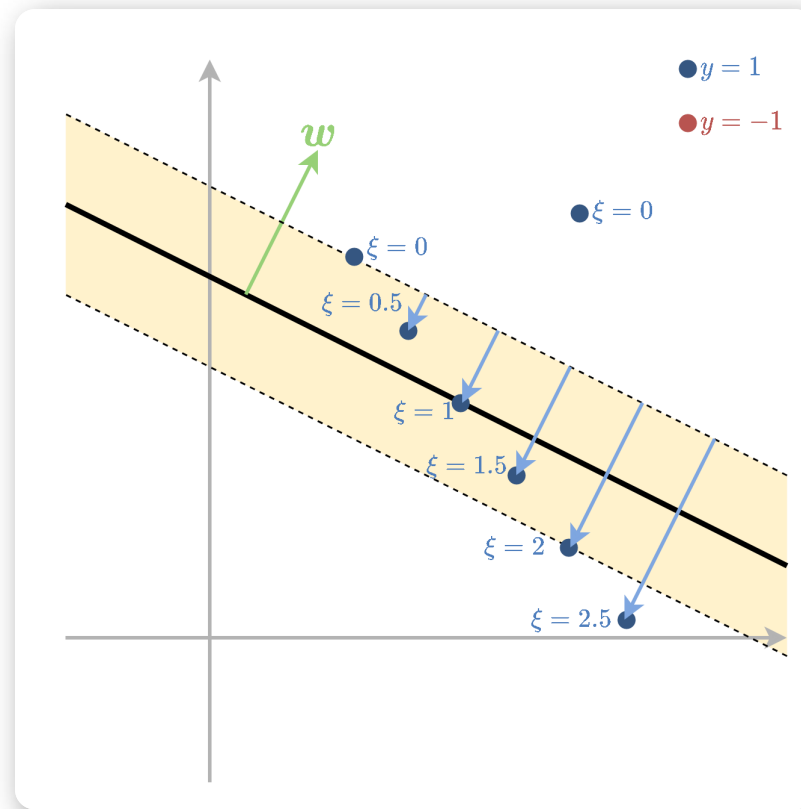
רק נקודות המדגם שעבורן α חיובי תורמות לסכום.
הערה (הרחבה למתעניינים): המשתנים α הם כופלי לגרנז' מהבעיה הפרימאלית.

הקשר בין α_i ו support vectors.

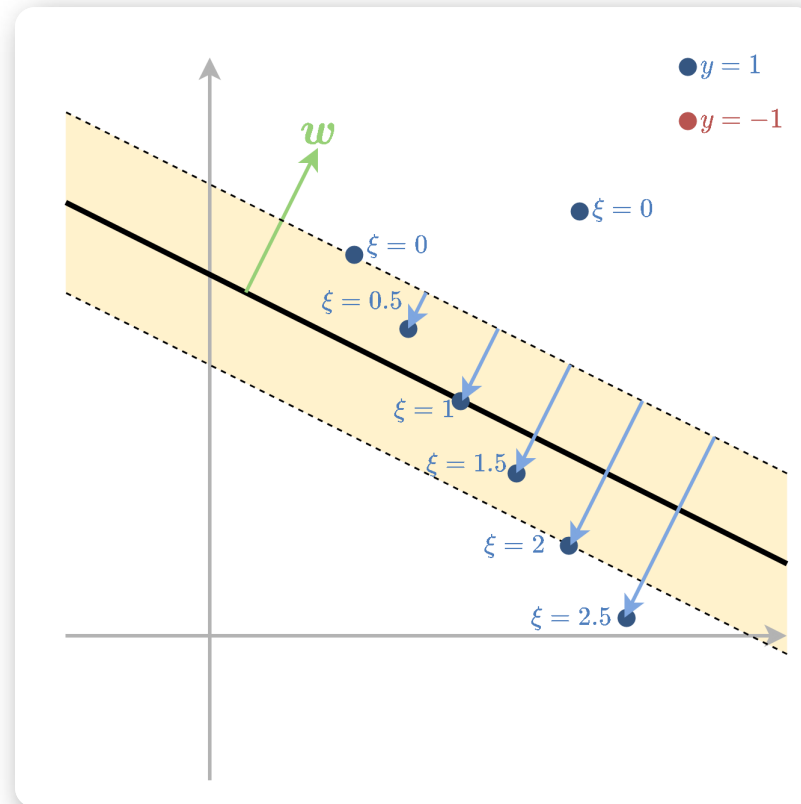


$\alpha_i = 0$	$y^{(i)} (\mathbf{w}^\top \mathbf{x}^{(i)} + b) > 1$	נקודות רחוקות מה margin
$\alpha_i \geq 0$	$y^{(i)} (\mathbf{w}^\top \mathbf{x}^{(i)} + b) = 1$	נקודות על ה margin (support vectors)

- נבחר נקודה מסוימת שעבורה $\alpha_i > 0$.
- נקודה כזו בהכרח תהיה **support vector** ותקיים $y^{(i)} (\mathbf{w}^\top x^{(i)} + b) = 1$.
- מתוך משוואה זו ניתן לחלץ את b .



- מתייחס למקרה שבו המדגם אינו פריד לינארית.
- מאפשרים לנקודות המדגם להיכנס לתוך השוליים ואף לחצות אותם.
- על כל חריגה כזו משלמים קנס בפונקציית המטרה.



- את החריגה של הדגימה i נסמן ב $\frac{1}{\|w\|} \xi_i$.
- המשתנים ξ_i נקראים **slack variables**.

ובעיית האופטימיזציה הפרימאלית תהיה

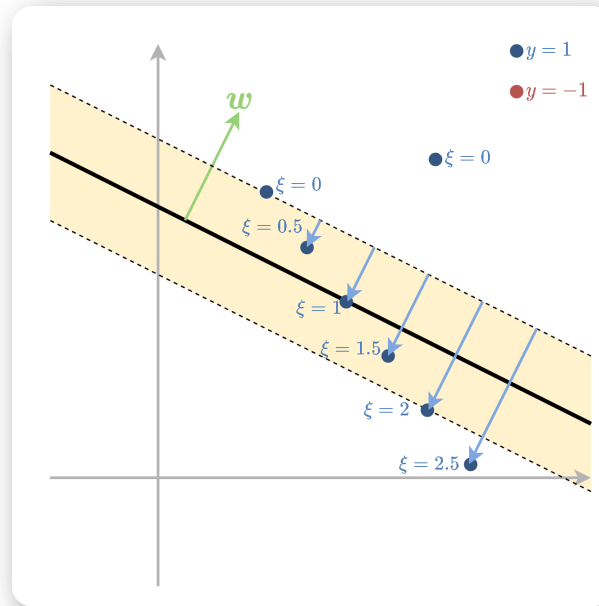
$$\begin{aligned} \mathbf{w}^*, b^*, \{\xi_i\}^* &= \arg \min_{\mathbf{w}, b, \{\xi_i\}} \left[\frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^N \xi_i \right] \\ \text{s.t.} \quad & y^{(i)} \left(\mathbf{w}^\top \mathbf{x}^{(i)} + b \right) \geq 1 - \xi_i \quad \forall i \\ & \xi_i \geq 0 \quad \forall i \end{aligned}$$

כאשר C הוא היפר-פרמטר אשר קובע את גודל הקנס בפונקציית המחיר על כל חריגה. בבעיה הפרימאלית לא היה היפר-פרמטר.

שאלה: מה ההשפעה של ערכים שונים של המשתנים ξ_i ?

הבעיה הדואלית הינה

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



עבור ה **support vectors** מתקיים: $y^{(i)} (\mathbf{w}^\top \mathbf{x}^{(i)} + b) = 1 - \xi_i$

תכונות:

α_i	$y^{(i)} (\mathbf{w}^\top \mathbf{x}^{(i)} + b)$	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.

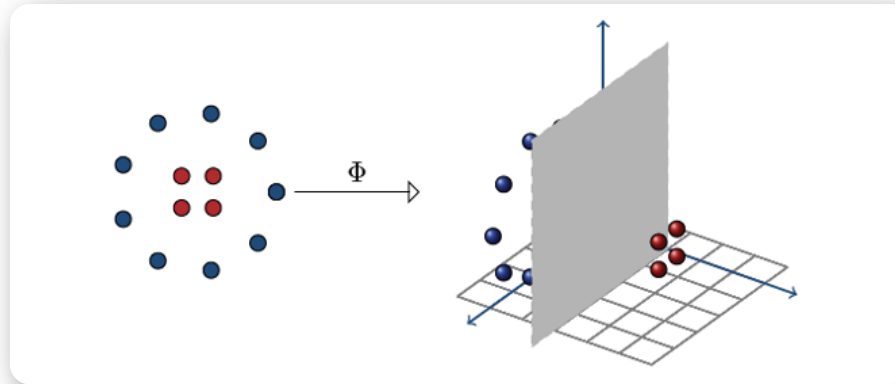
סיווג לינארי מוגבל, ולכן נרצה להרחיב לסיווג לא לינארי.

• נוכל תמיד להחליף את וקטור המשתנים x בוקטור חדש:

$$x_{\text{new}} = \Phi(x)$$

• Φ היא פונקציה אשר נבחרה מראש ונקראת פונקציית המאפיינים.

• אם הממד של Φ מספיק גבוה, ניתן תמיד להגיע להפרדה לינארית במרחב הרב-ממדי (דורש הוכחה).



**נשים לב שהמסווג האופטימלי הוא ריבועי במרחב המקורי.
נשתמש בפונקציית המאפיינים**

$$\Phi(x) = (x_1, x_2, x_1x_2, x_1^2, x_2^2, 1)$$

ונקבל $w^\top \Phi(x)$ שהוא לינארי ב- w .

האיור מתוך, Mohri et-al, Foundation of Machine Learning

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

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 σ 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.

Kernel Trick in SVM

הרעיון ב kernel trick הינו לעשות שימוש בפונקציית הגרעין Φ על מנת להשתמש ב SVM עם מאפיינים מבלי לחשב את Φ באופן ישיר.

עבור פונקציית מאפיינים Φ עם פונקציית גרעין K הבעיה הדואלית של SVM הינה:

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

Kernel Trick in SVM

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

שאלה: מה הקשר לבעיה הדואלית בשקף 25?
בעיית אופטימיזציה זו מגדירה את המשתנים $\{\alpha_i\}$ בלי צורך לחשב את Φ באופן מפורש בשום שלב.

Kernel Trick in SVM

באופן כללי, הפרמטר w נתון על ידי:

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

אשר מצריך חישוב של Φ . ניתן להימנע מכך על ידי הצבה של w כמו שהוא ישירות לחזאי.

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

בדרך זו אנו יכולים לאמן להשתמש בחזאי אשר אומן בעבור וקטור מאפיינים Φ מבלי לחשב בשום שלב את Φ באופן מפורש.

סיכום: תכונות מסוג SVM

במקרה הפריד לינארית:

- איטואיטיבי וקל להבנה - ייצוג פשוט של הפתרון
- מבטיח ביצועי הכללה טובים בזכות השוליים הרחבים (לא הראינו) - סוג של רגולריזציה
- יעיל חישובית

במקרה הלא פריד לינארית:

- מתווסף היפר-פרמטר שיש לכוון
- הבנה אינטואיטיבית פחותה
- מעבר פשוט ונוח למסווגים לא לינאריים ע"י שיטות גרעין