Machine learning

SVM

Lecture X

פיתוח:
ד"ר יהונתן שלר
משה פרידמן

מה ראינו עד כה?

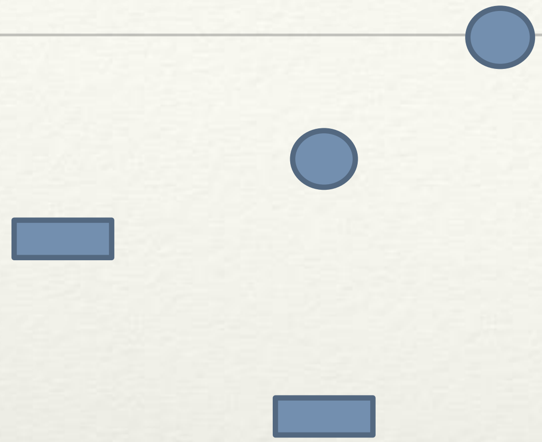
- מסווג לפי "שכנים" – kNN
- עץ החלטה
- מסווג הסתברותי – ביסיאני – NB
- רשתות עצביות

- היום נרצה ללמוד מסווג נוסף – מאד יעיל SVM (בתוצרתו הבסיסית – מסווג ליניארי)

מאמרים רלוונטיים

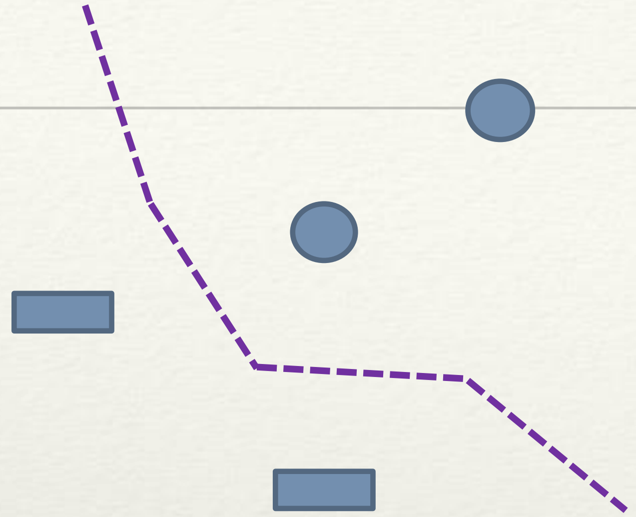
- Burges, Christopher J. C.;A Tutorial on Support Vector Machines for Pattern Recognition, Data Mining and Knowledge Discovery 2:121–167, 1998
- Joachims, Thorsten. "Svmlight: Support vector machine." SVM-Light Support Vector Machine http://svmlight. joachims. org/, University of Dortmund 19.4 (1999).
- sk-learn https://scikit-learn.org/stable/modules/svm.html

כללי



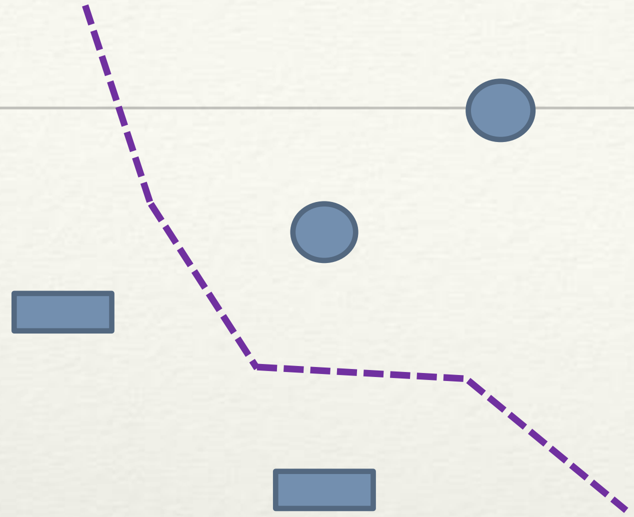
KNN (K=1)

כללי

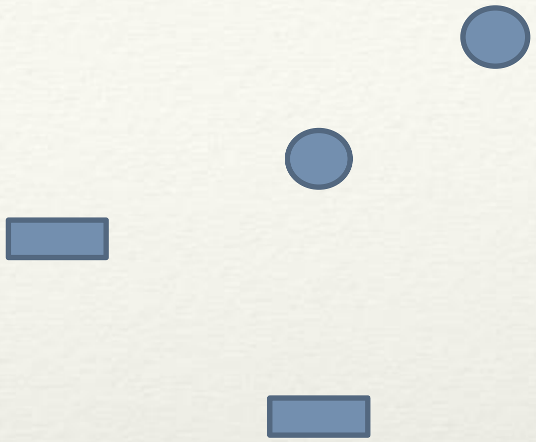


KNN (K=1)

כללי

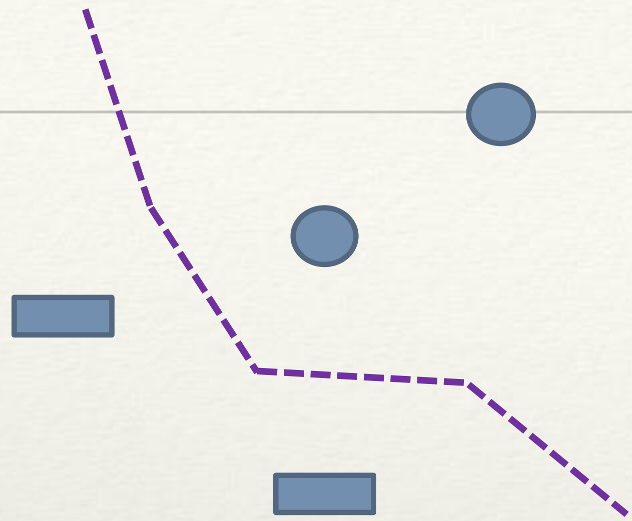


KNN (K=1)

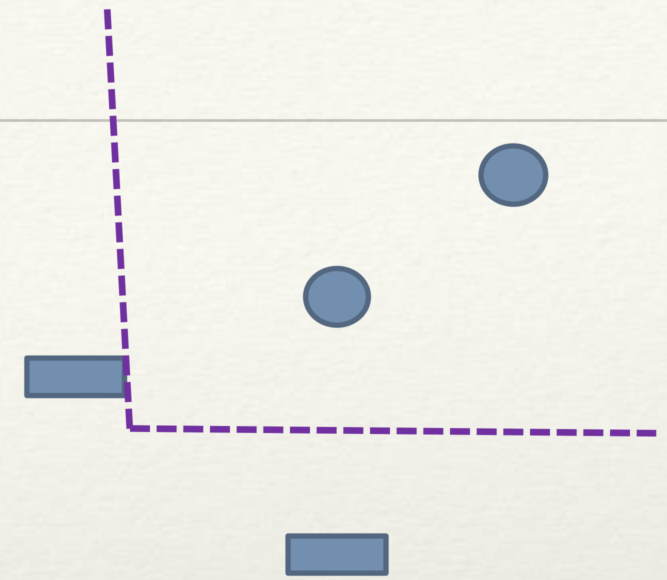


Decision Trees (ID3)

כללי

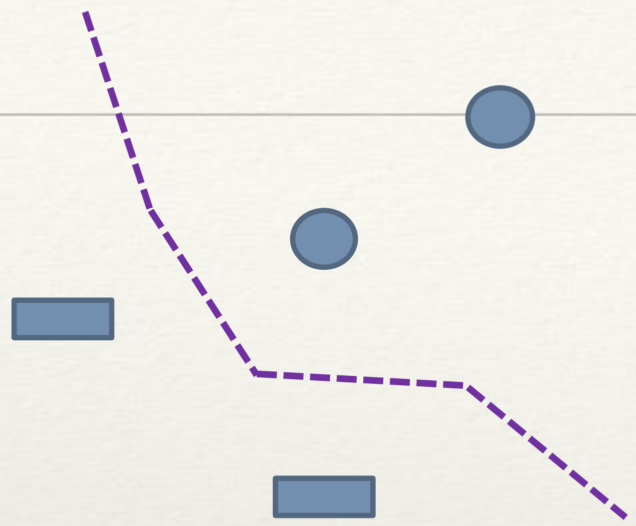


KNN (K=1)

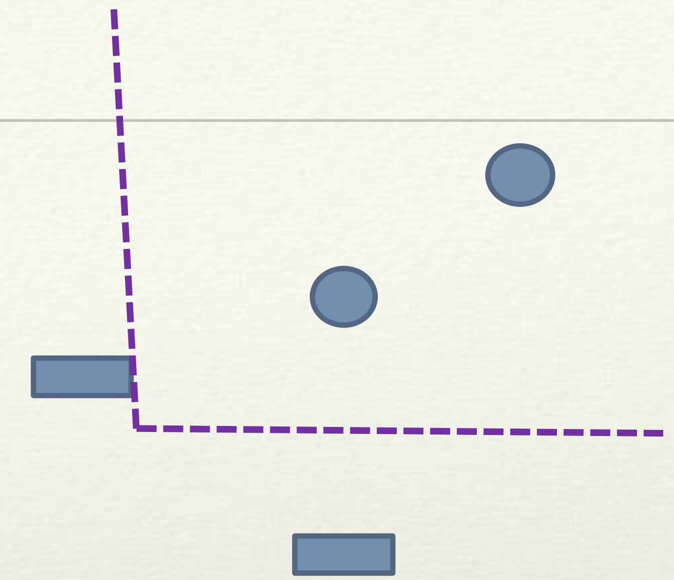


Decision Trees (ID3)

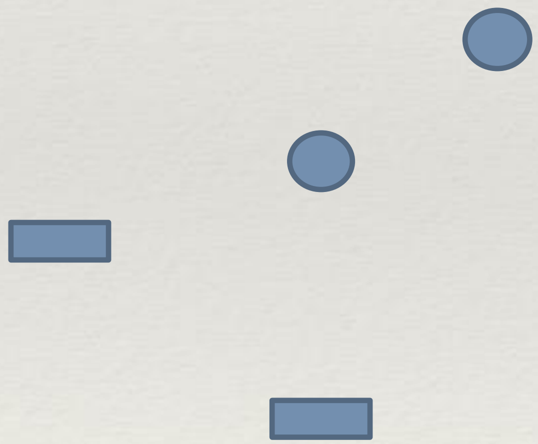
כללי



KNN (K=1)

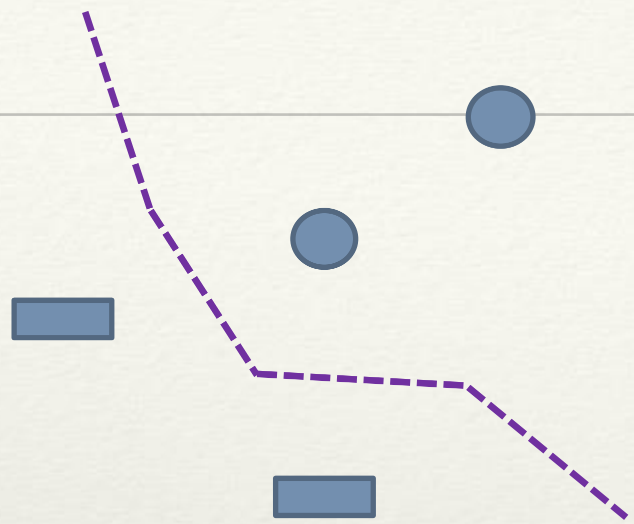


Decision Trees (ID3)

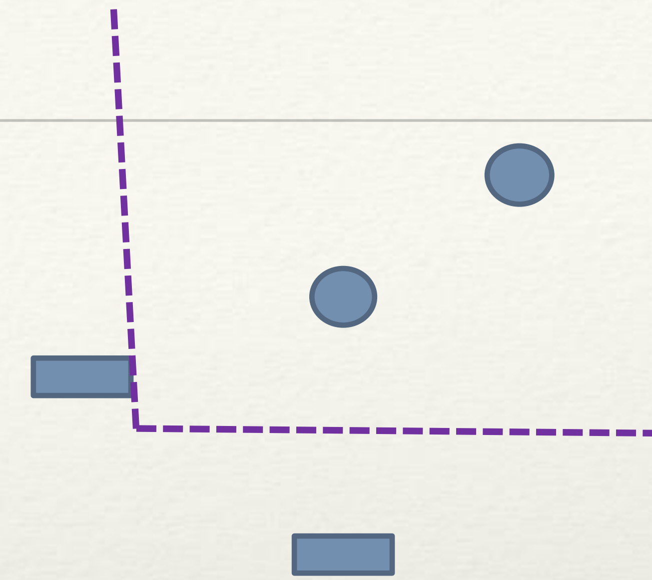


PERCEPTRON

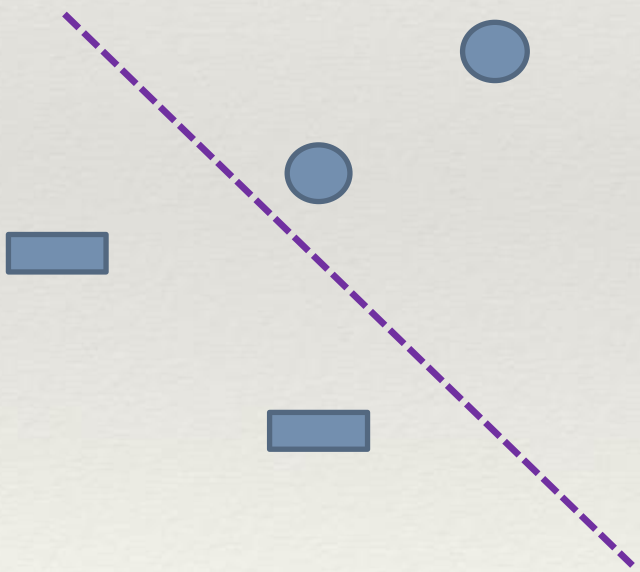
כללי



KNN (K=1)

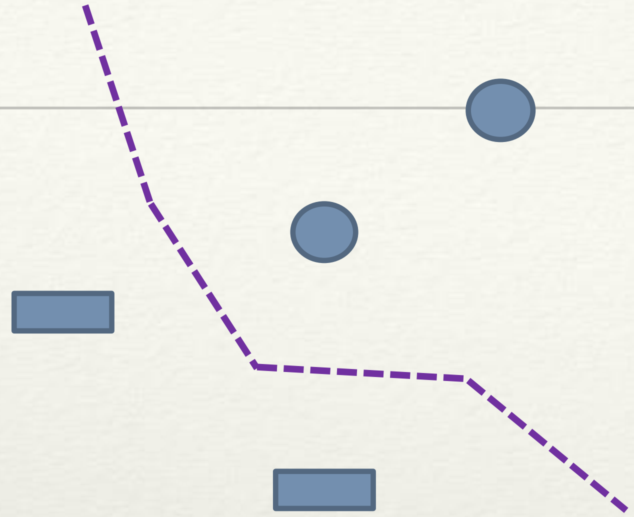


Decision Trees (ID3)

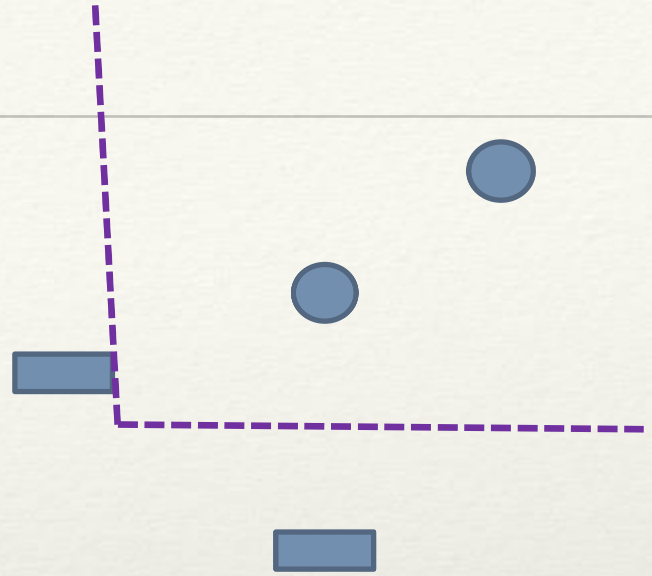


PERCEPTRON

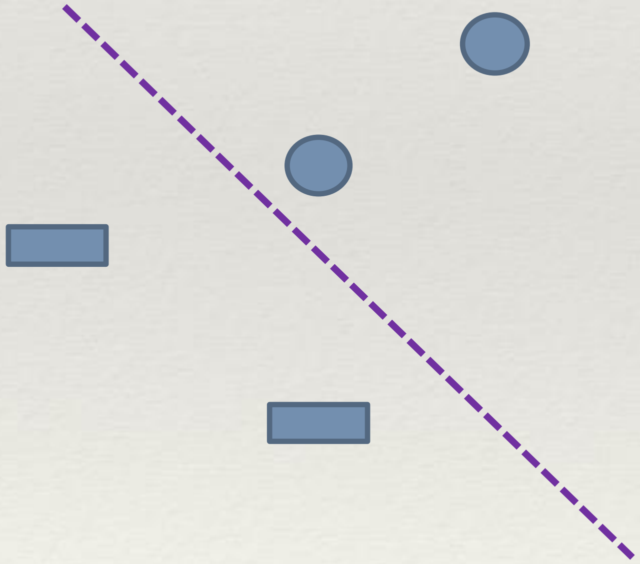
כללי



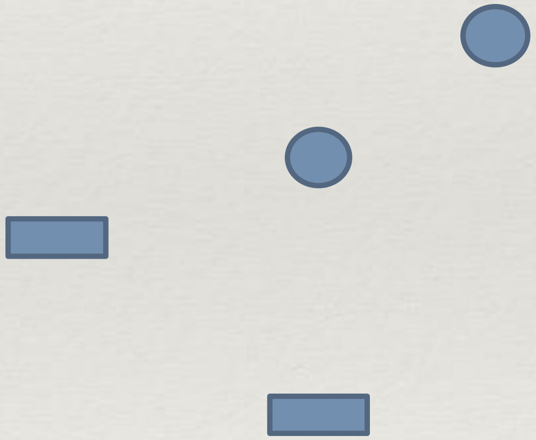
KNN (K=1)



Decision Trees (ID3)

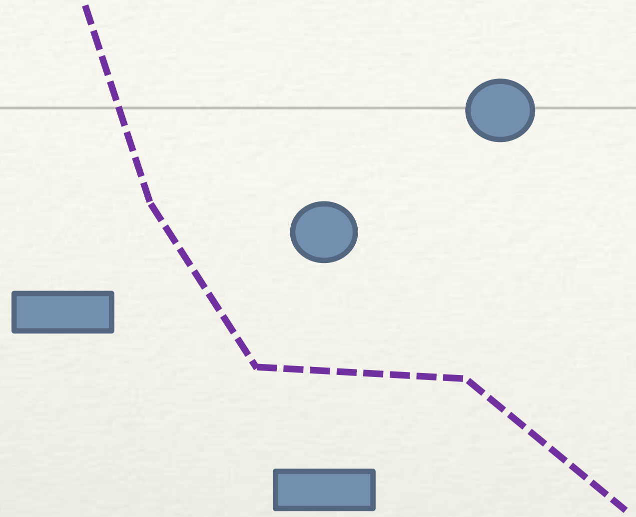


PERCEPTRON

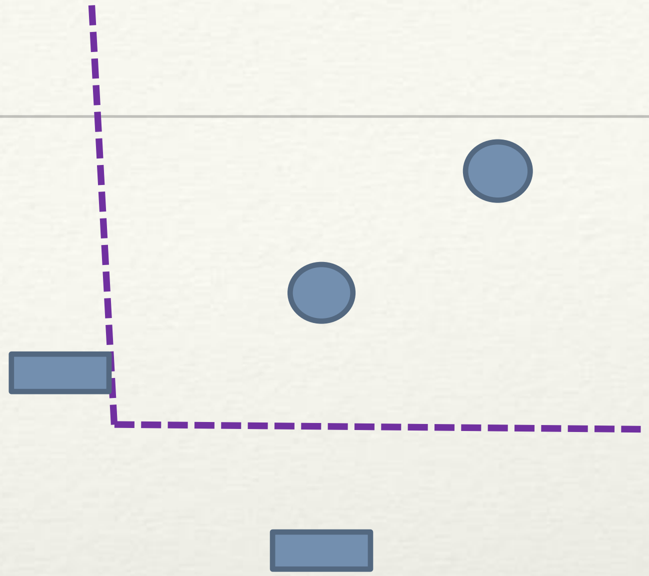


Neural Network

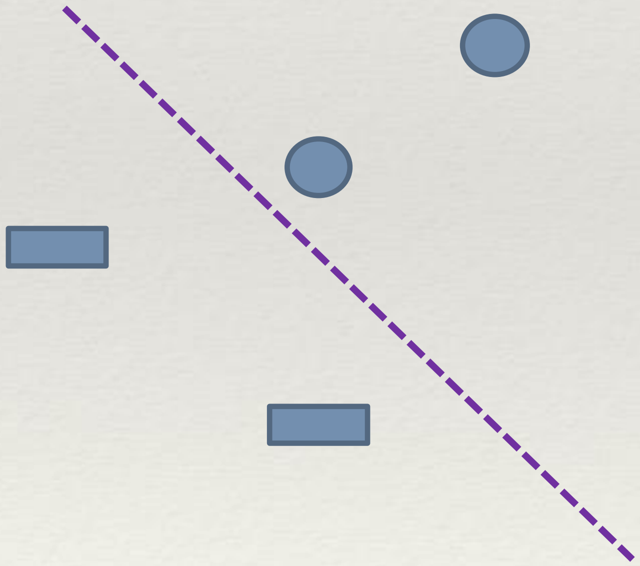
כללי



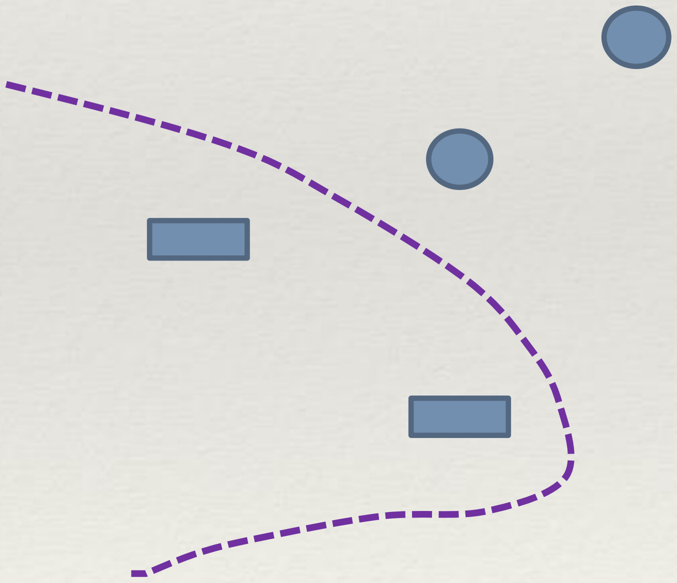
KNN (K=1)



Decision Trees (ID3)

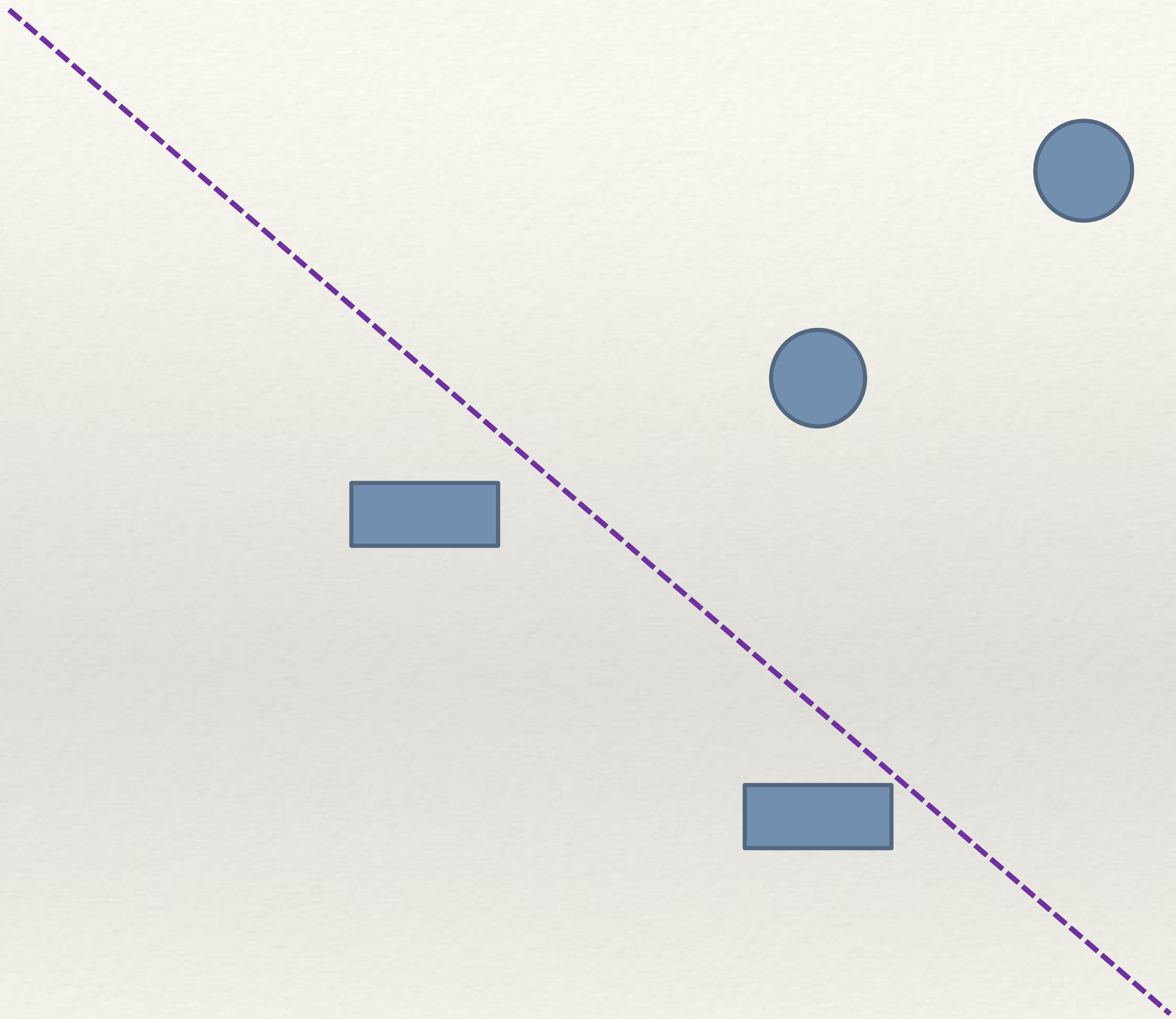


PERCEPTRON

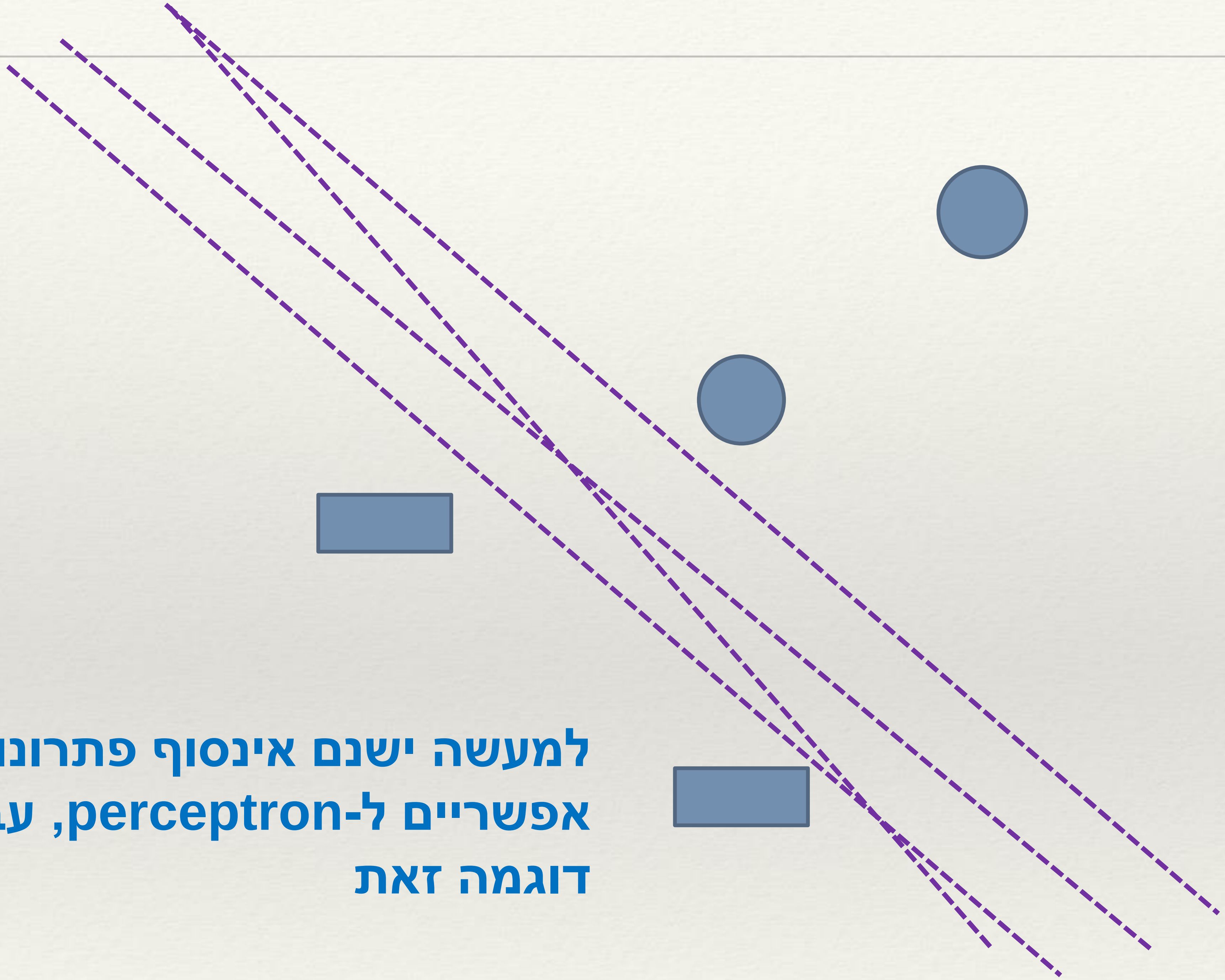


Neural Network

Perceptron – Different possible solutions



Perceptron – Different possible solutions



למעשה ישנם אינסוף פתרונות
אפשריים ל-perceptron, עבור
דוגמה זאת

פונקצית הפסד (Loss function) או
פונקצית עלות (Cost function) - תזכורת

פונקצית מחיר או פונקצית עלות - פונקציה הממפה מאורע או ערכים של משתנה אחד או יותר למספר ממשי המייצג "עלות" של מאורע. נסמן את הפונקציה ע"י J.

- בלמידה נסתכל למשל עלות של טעות בסיווג למשל

נסמן: l או loss – כפונקצית ההפסד

J(misclassification of i) – מחיר ההפסד של סיווג לא נכון

- סך ההפסד בלמידה, משתמש בשיטות כפי שראינו בשערוך (evaluation)

פונקצית מטרה (objective function) –

בהקשר שלנו - פונקצית המטרה תהווה, בדרך כלל, פונקצית ההפסד או פונקצית הטעות, אותה נגדיר.

בעיית אופטימיזציה (optimization problem)

ב- SVM – במקום להגדיר את הסיווג בצד הלא נכון של המפריד הלינארי, מקשיחים את המטרה, כך שדוגמאות חיוביות ושליליות יהיו בצד הנכון של ה-margin.

בעיות אופטימיזציה:

מזעור (minimization) – $\mathbf{x}_0 \in A$ such that $f(\mathbf{x}_0) \leq f(\mathbf{x})$ for all $\mathbf{x} \in A$

- בבעיות למידה, נרצה לעיתים קרובות לעשות מזעור של פונקצית ההפסד (פונקציית המטרה)

מקסום (maximization) - $\mathbf{x}_0 \in A$ such that $f(\mathbf{x}_0) \geq f(\mathbf{x})$ for all $\mathbf{x} \in A$

בעיית סיווג

- כלומר, אנו רוצים למצוא היפותזה במרחב ההיפותזות H –

$$H \in \Re^D \longrightarrow \{-1, 1\}$$

- כך שבהינתן "מחירון ענישה" לטעות

$$loss(y, H(x))$$

- אנו נמצא את h שממזערת לנו את הטעות על ה"עולם האמיתי". כלומר יכולת "הכללה"

SVM

- עד כה ראינו, למשל באלגוריתם הפרספטרון, איך אנו מוצאים היפותזה h שמפרידה בין המחלקות השונות. לא הראינו יכולת הכללה.

תזכורת

- נניח שנתונים לנו k ווקטורים (דוגמאות) $X_i \in R^n$

כל עמודה במטריצה מייצגת דוגמא בודדת

סה"כ K דוגמאות *

$$\begin{bmatrix} x_{01} & x_{02} & x_{03} & \cdots & \cdots & \cdots & x_{0k} \\ x_{11} & x_{12} & x_{13} & & & & x_{1k} \\ x_{21} & x_{22} & x_{23} & & & & x_{2k} \\ \cdots & \cdots & \cdots & & & & \\ \cdots & \cdots & \cdots & & & & \\ \cdots & \cdots & \cdots & & & & \\ x_{n1} & x_{n2} & x_{n3} & & & & x_{nk} \end{bmatrix}$$

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \\ \\ \\ y_k \end{bmatrix}$$

- לכל ווקטור נתון לנו הסיווג $y_i \in \{0,1\}$

* הערה: צורך ההצגה כאן משוחלפת (transposed),
כדי שנוכל להכפיל בווקטור המשקולות

תזכורת: ייצוג מתמטי

- אנו מחפשים קבוצת משקלים כך ש-

$$\mathrm{sgn}(w_0x_{01} + w_1x_{11} + w_2x_{21} + \ldots.. + w_nx_{n1}) = y_1$$
$$\mathrm{sgn}(w_0x_{02} + w_1x_{12} + w_2x_{22} + \ldots.. + w_nx_{n2}) = y_2$$

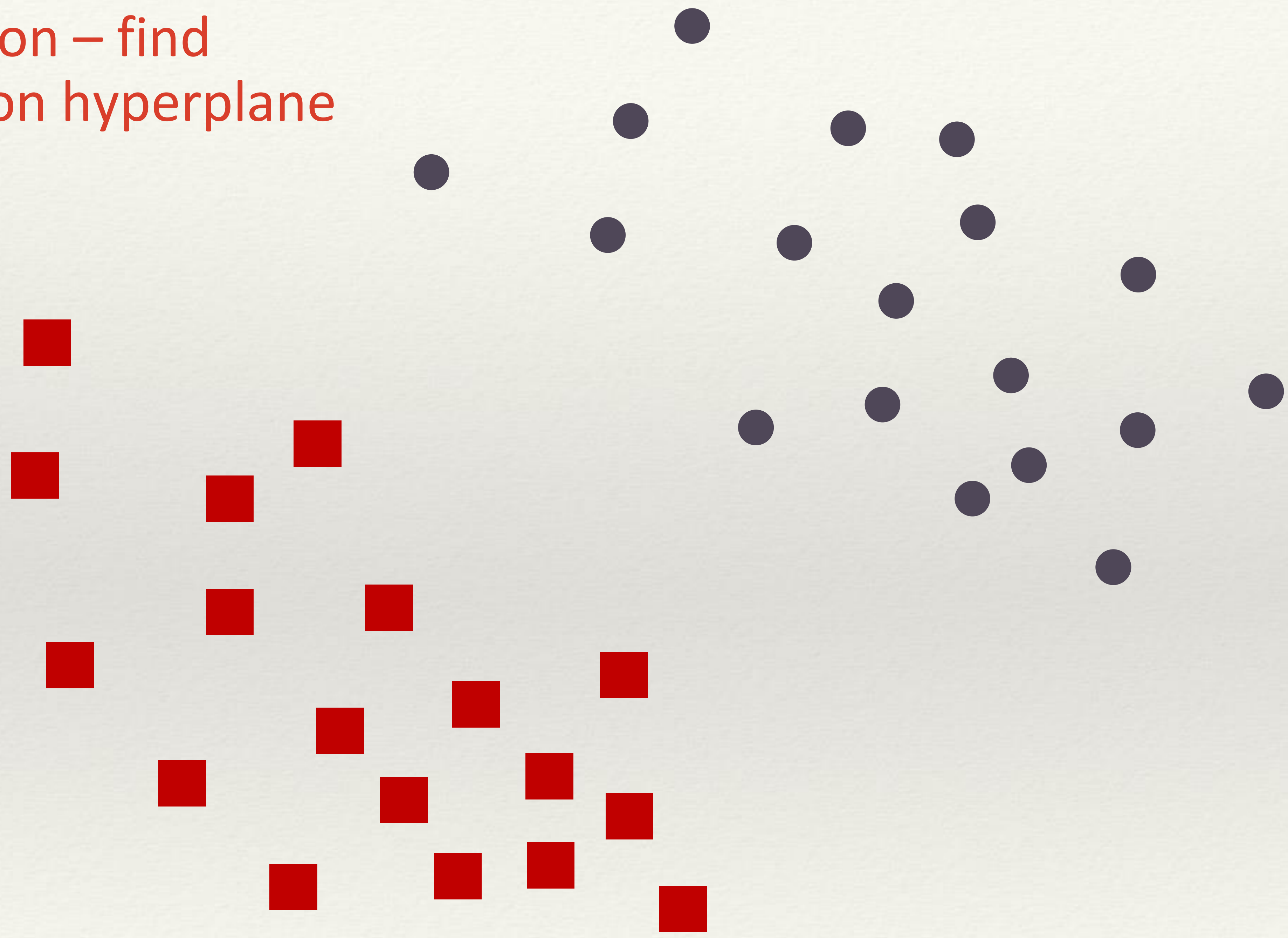
.....

.....

$$\mathrm{sgn}\left(\left[w_{0,} w_1, w_2 \ldots\ldots w_n\right] \begin{bmatrix} x_{01} & x_{02} & x_{03} & \cdots & \cdots & \cdots & x_{0k} \\ x_{11} & x_{12} & x_{13} & \cdots & & & x_{1k} \\ x_{21} & x_{22} & x_{23} & \cdots & & & x_{2k} \\ \cdots & \cdots & \cdots & & & & \\ \cdots & \cdots & \cdots & & & & \\ \cdots & \cdots & \cdots & & & & \\ x_{n1} & x_{n2} & x_{n3} & & & & x_{nk} \end{bmatrix} \right) = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ . \\ . \\ . \\ y_k \end{bmatrix}^{\mathrm{T}}$$

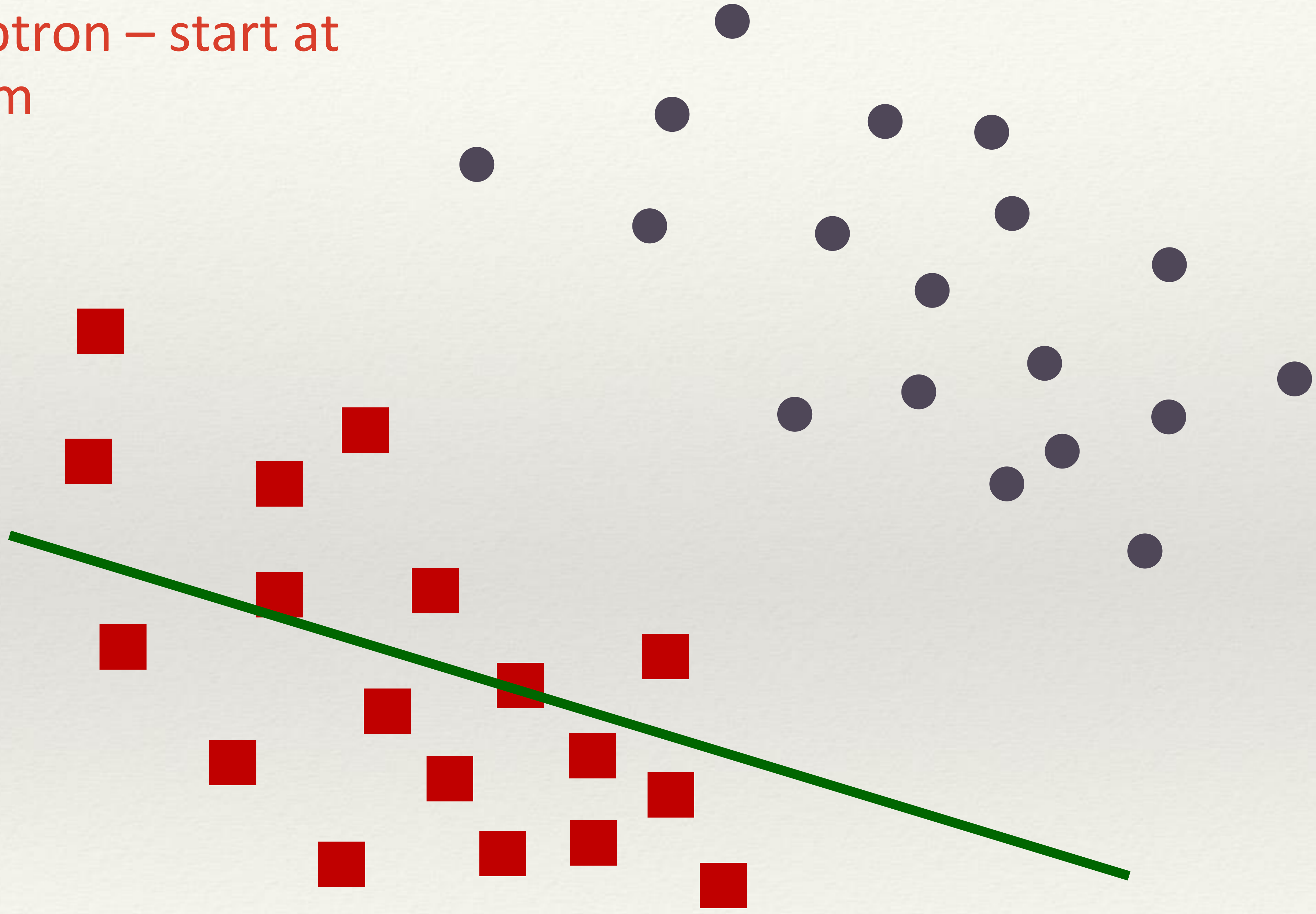
תזכורת: ייצוג מתמטי

Perceptron – find separation hyperplane



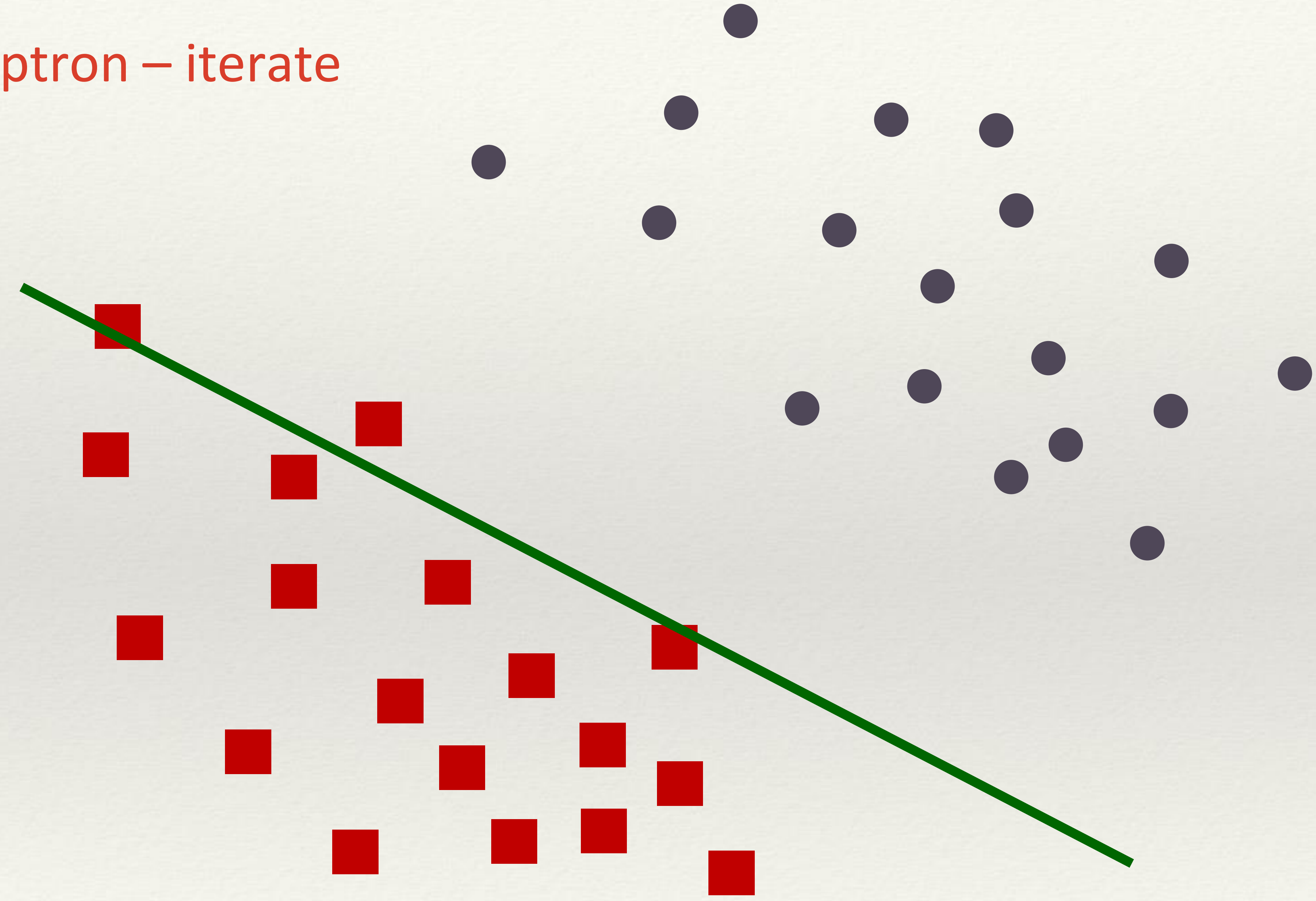
תזכורת: ייצוג מתמטי

Perceptron – start at random



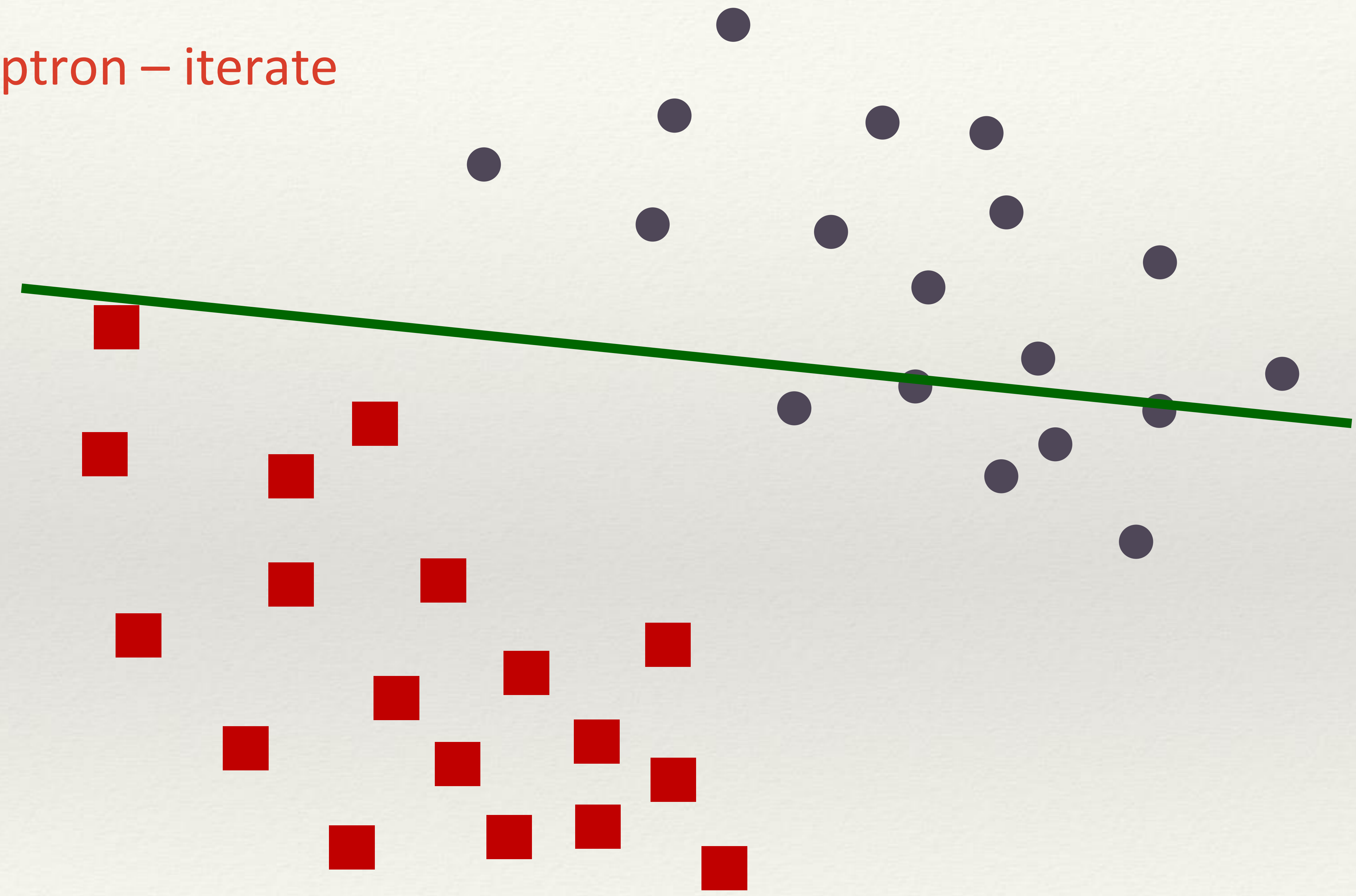
תזכורת: ייצוג מתמטי

Perceptron – iterate



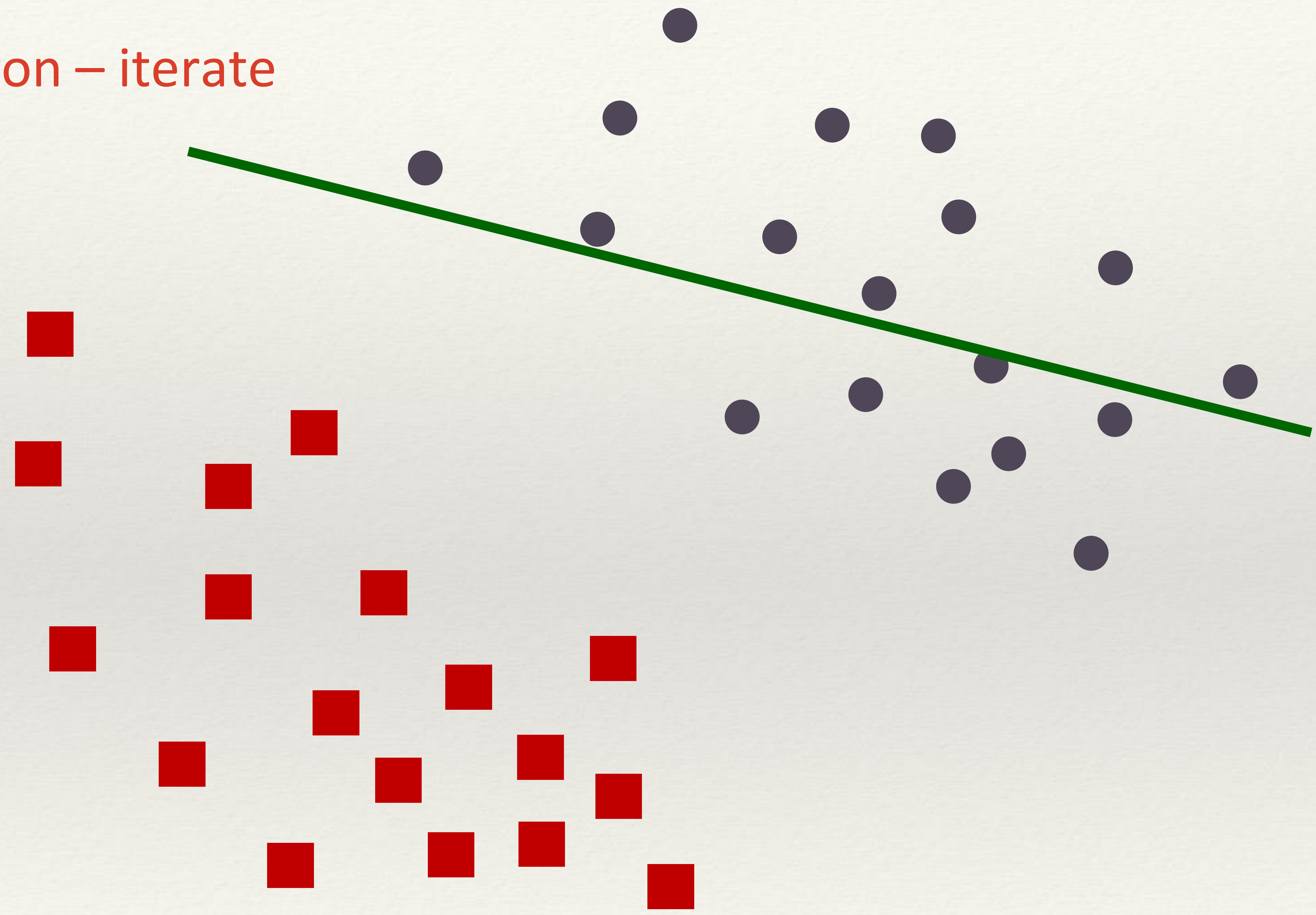
תזכורת: ייצוג מתמטי

Perceptron – iterate



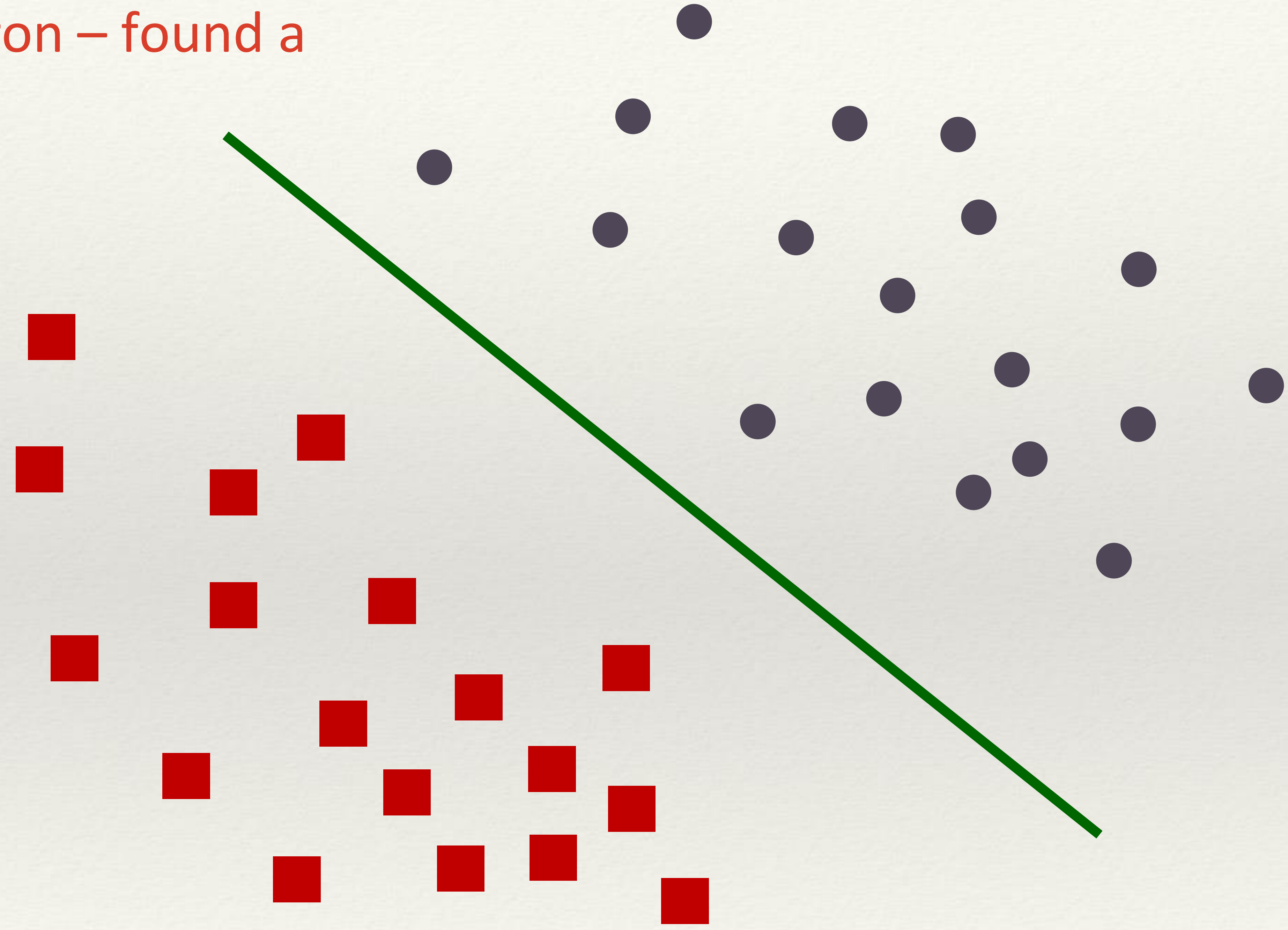
תזכורת: ייצוג מתמטי

Perceptron – iterate



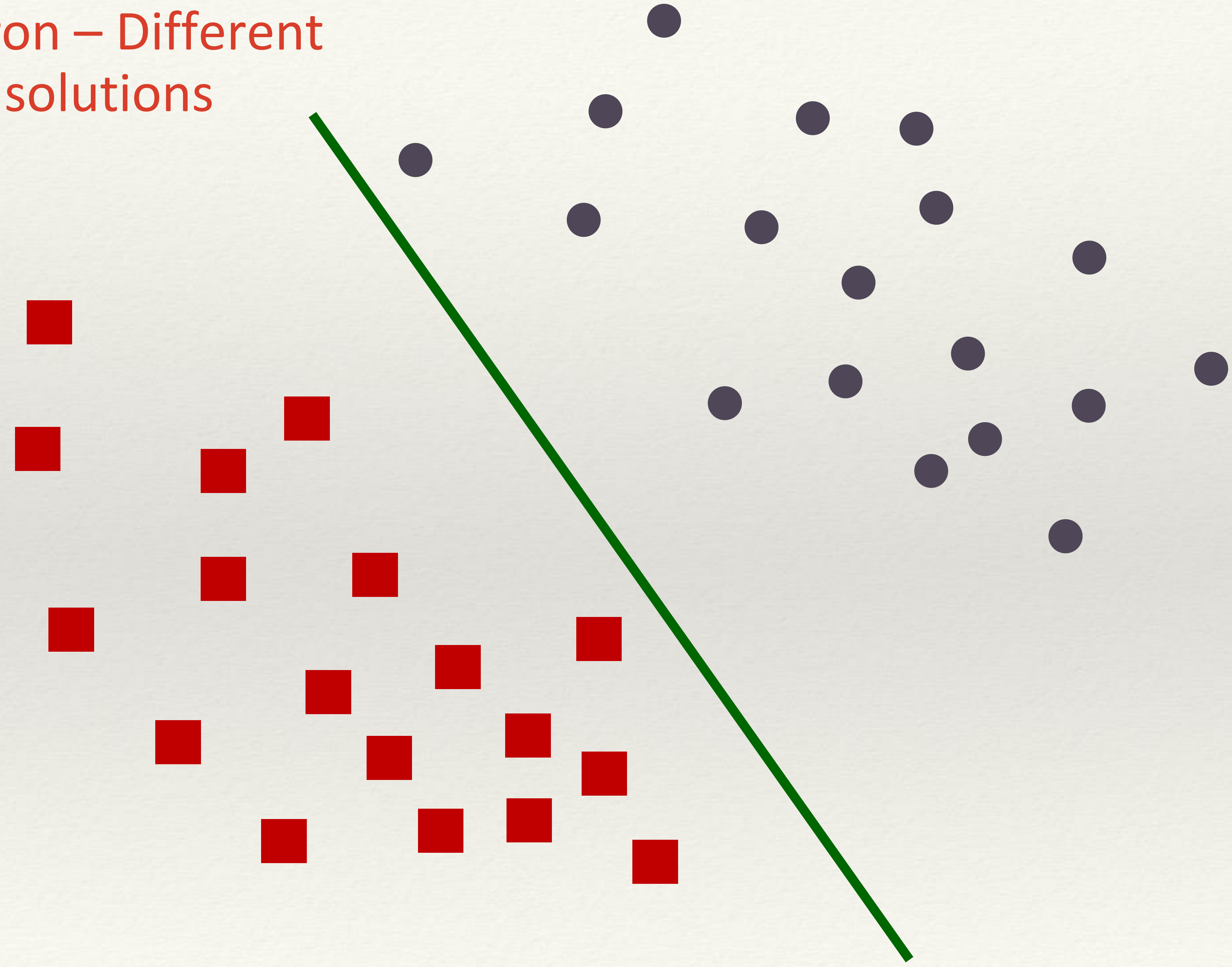
תזכורת: ייצוג מתמטי

Perceptron – found a solution

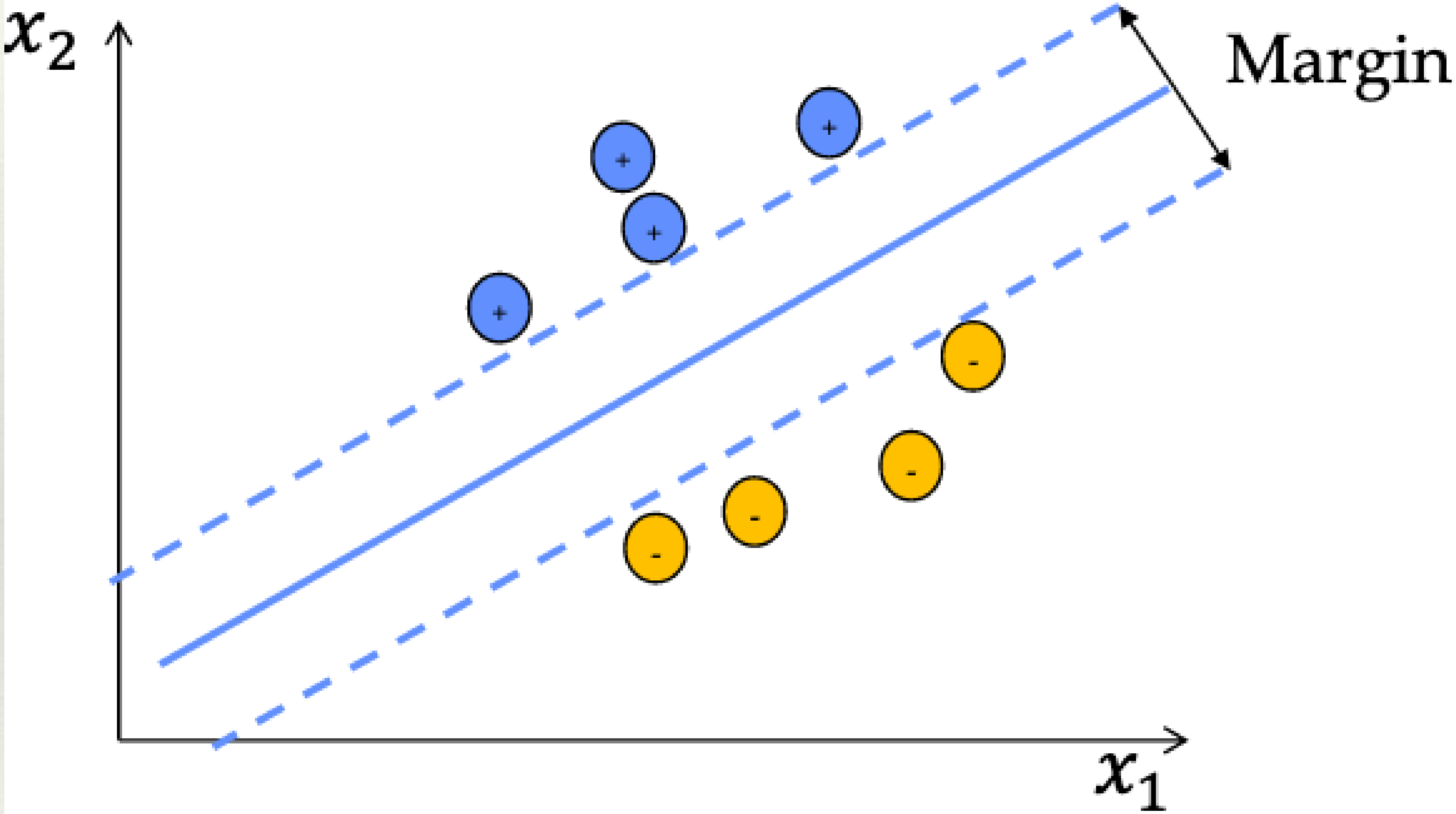


תזכורת: ייצוג מתמטי

Perceptron – Different possible solutions

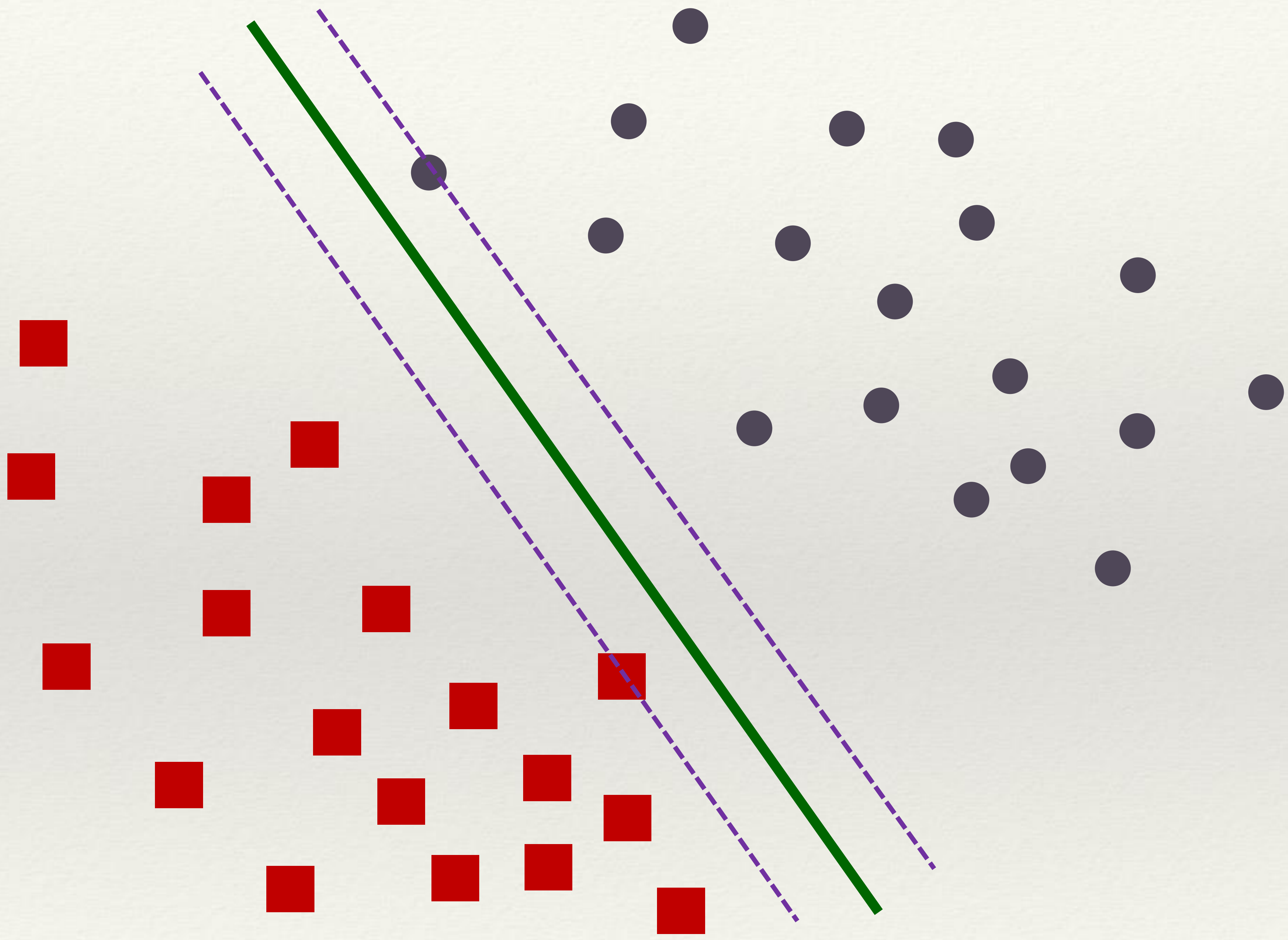


מהו ה-Margin

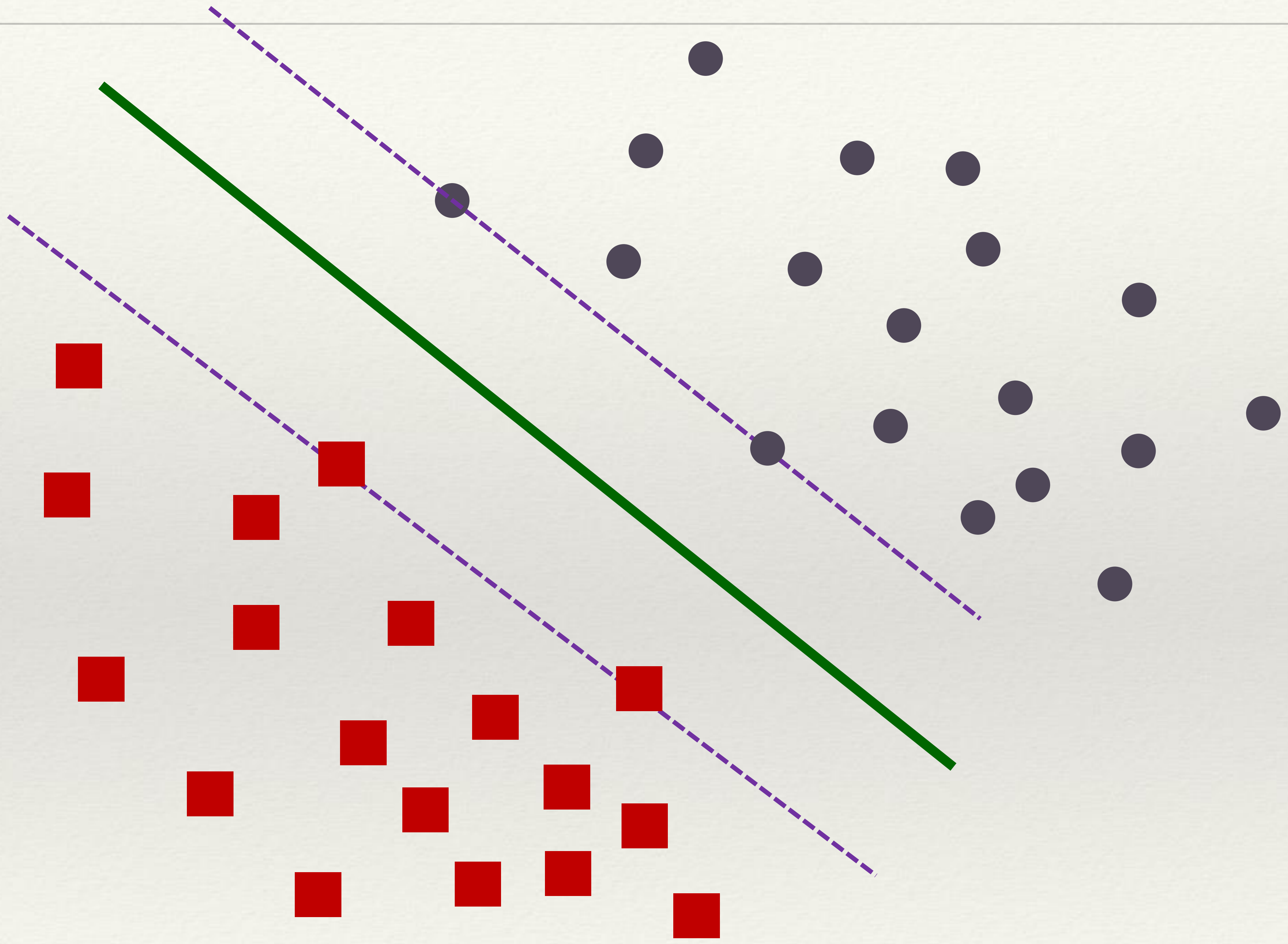


Margin - The minimal distance between positive an negative examples

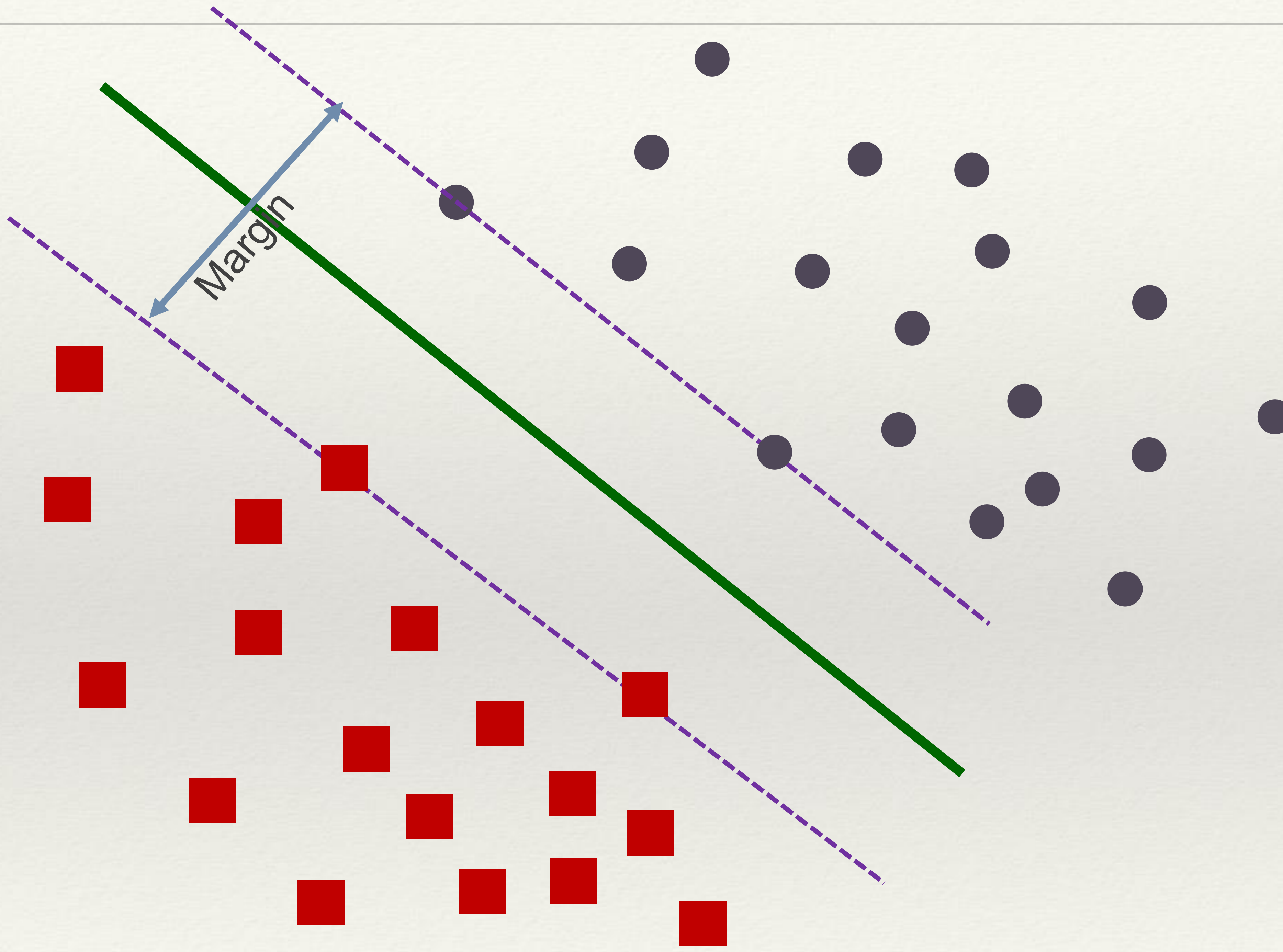
Different possible solutions for the margins



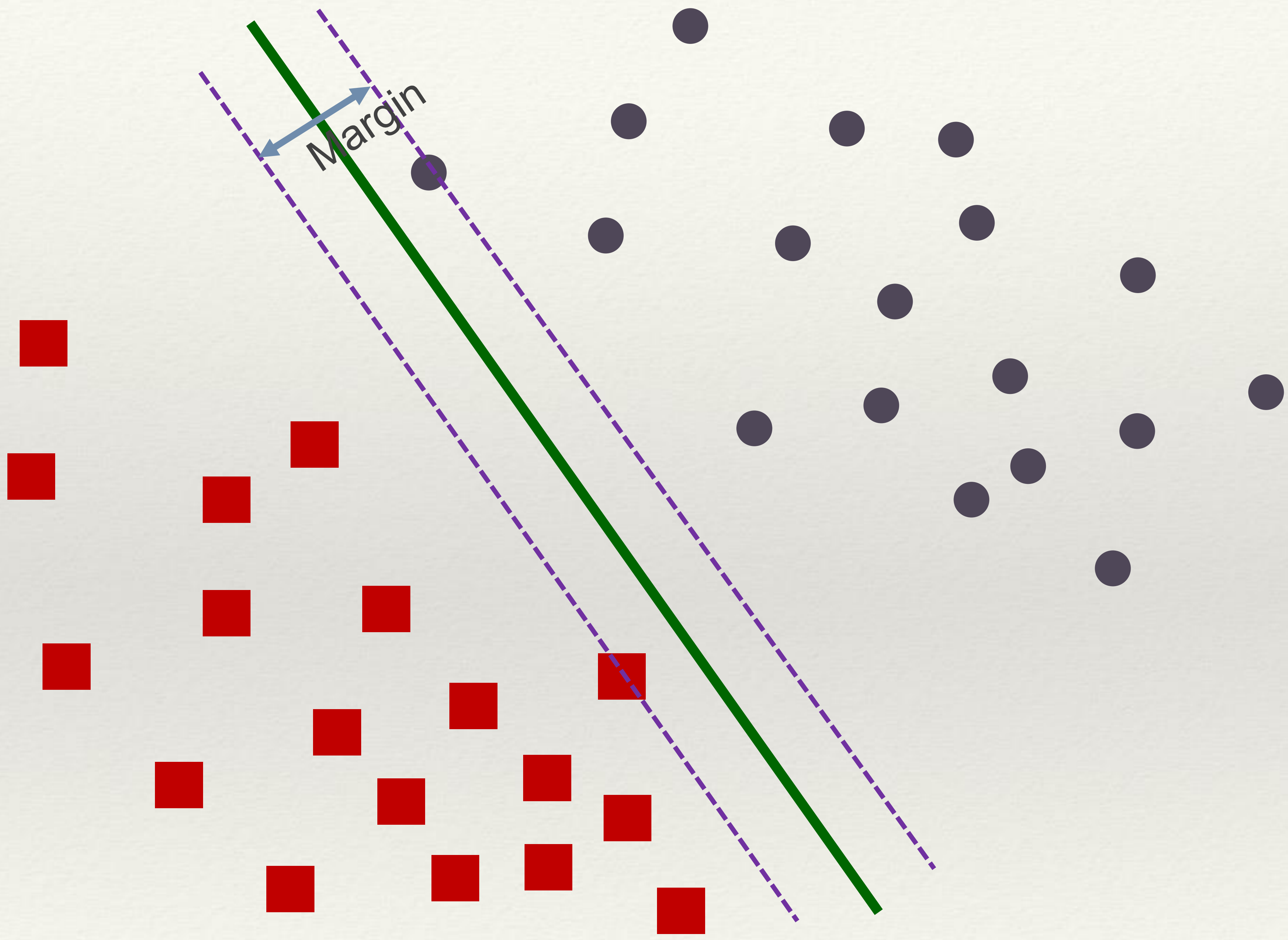
Different possible solutions for the margins



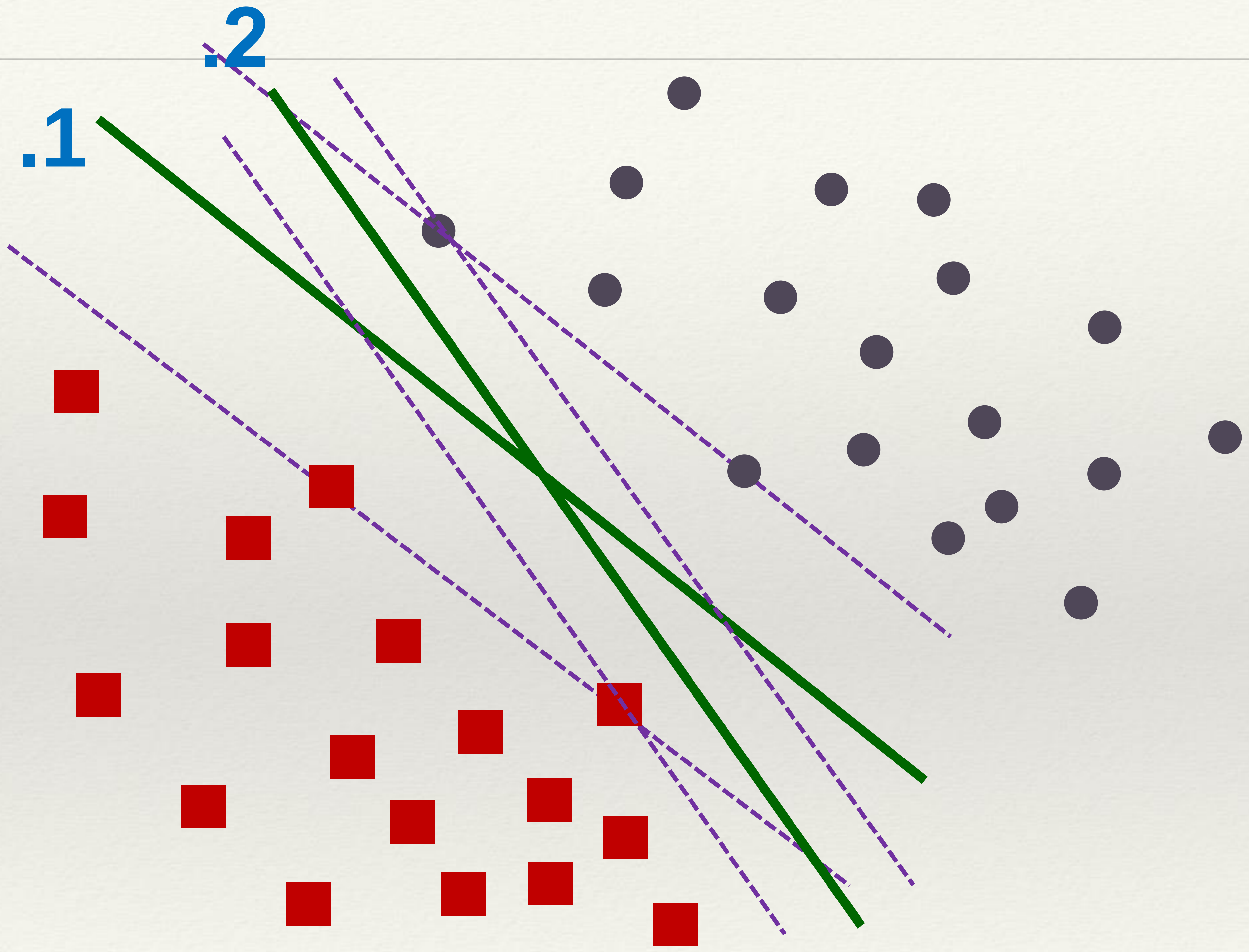
Different possible solutions



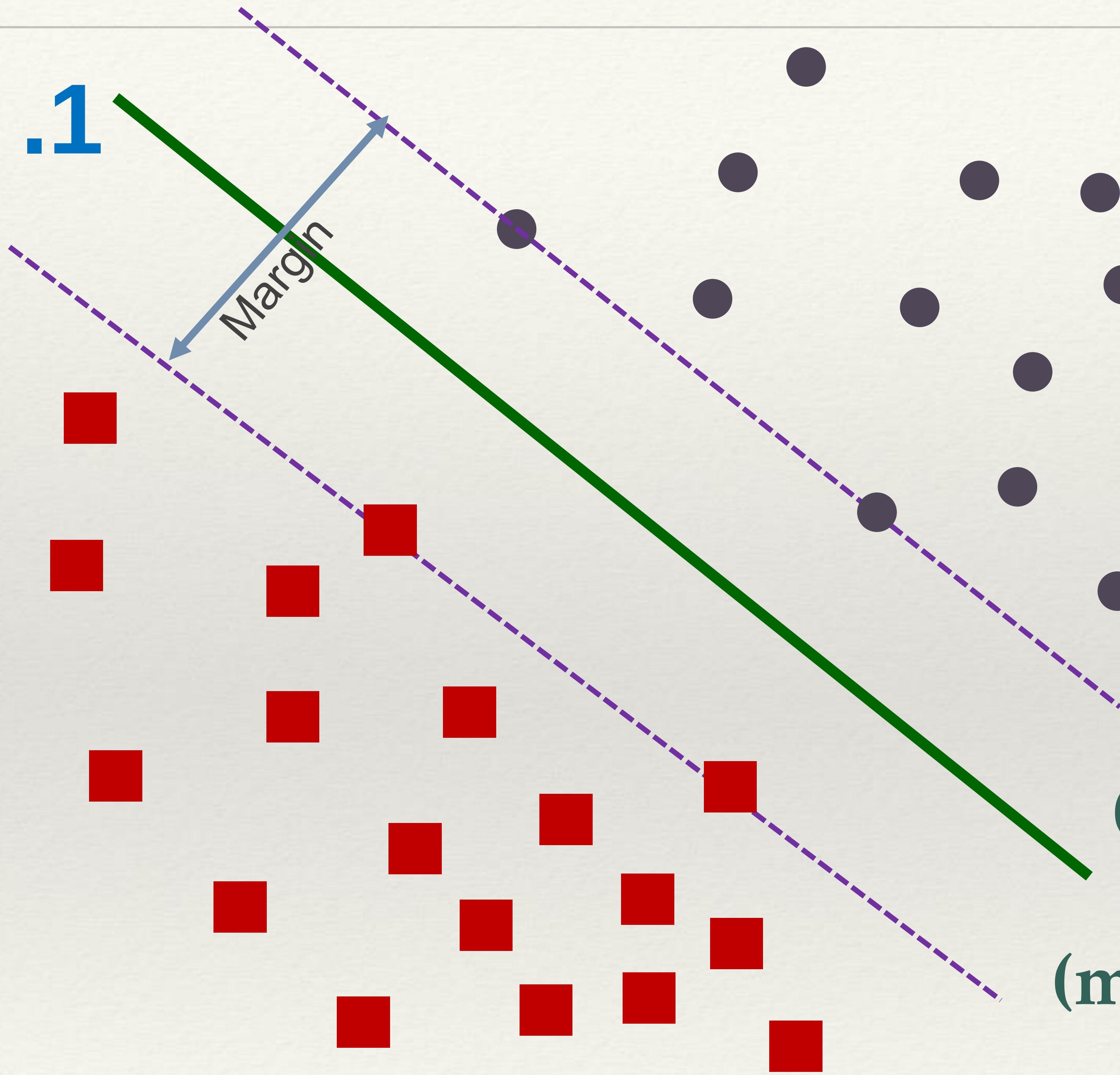
Different possible solutions



סקר – מה עדיף, 1 או 2?

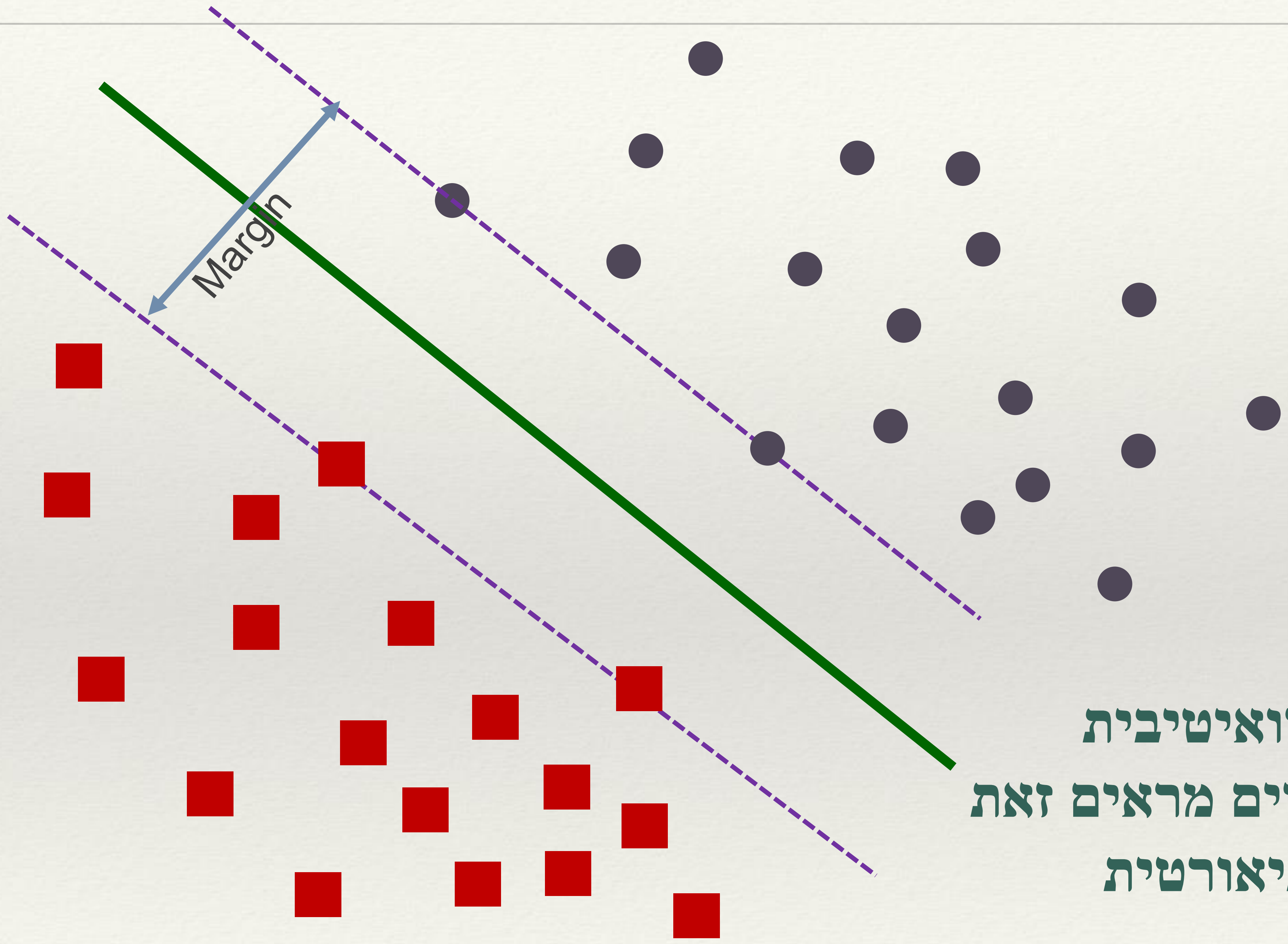


Different possible solutions



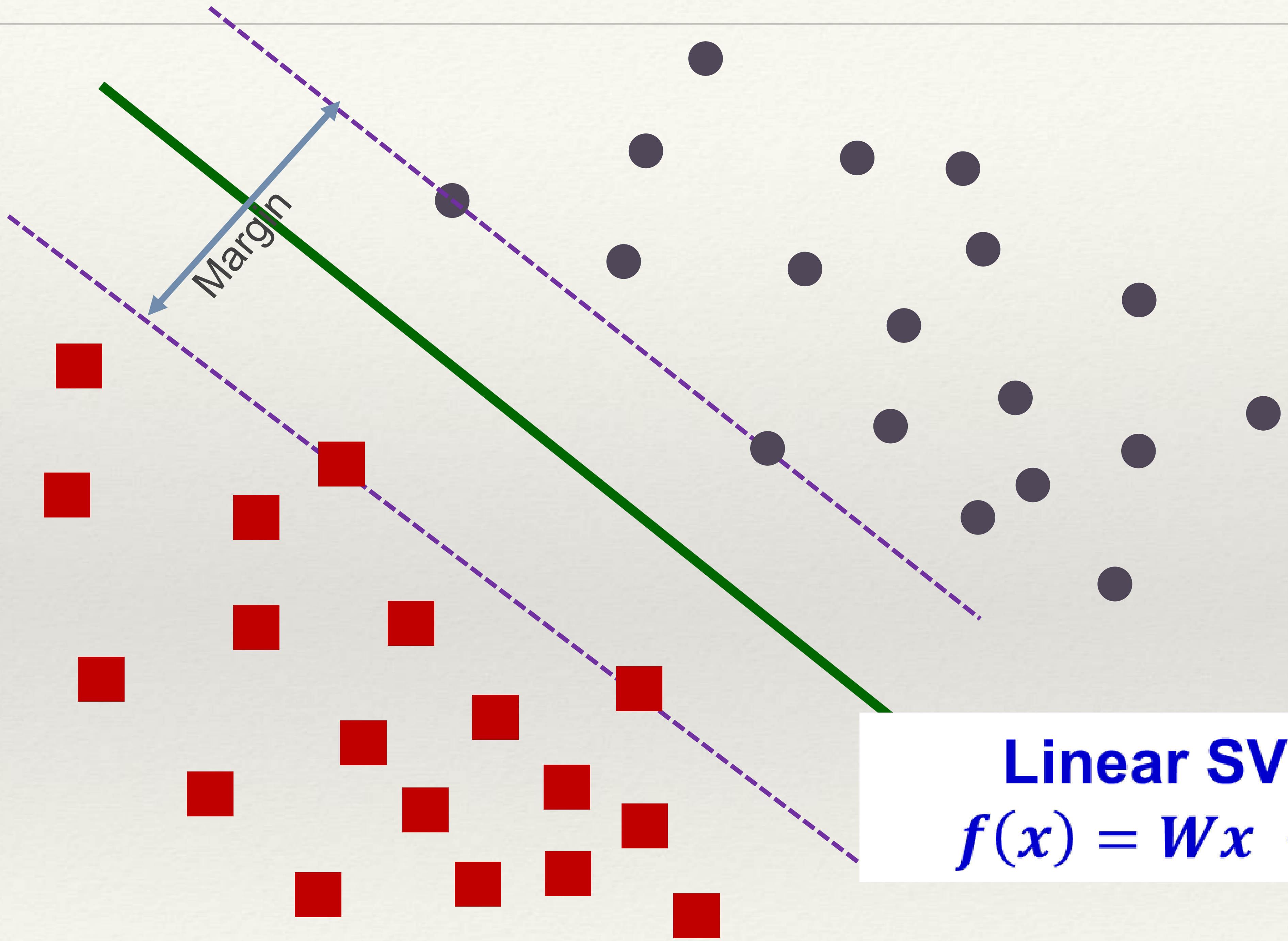
נחפש את הקו
(hyperplane)
שמייצר לנו את
ה"מרווח" (margin)
המקסימאלי

מדוע נחפש את ה- hyperplane שמייצר "מרווח" מקסימלי?



1. אינטואיטיבית
2. ניסויים מראים זאת
3. גם תיאורטית

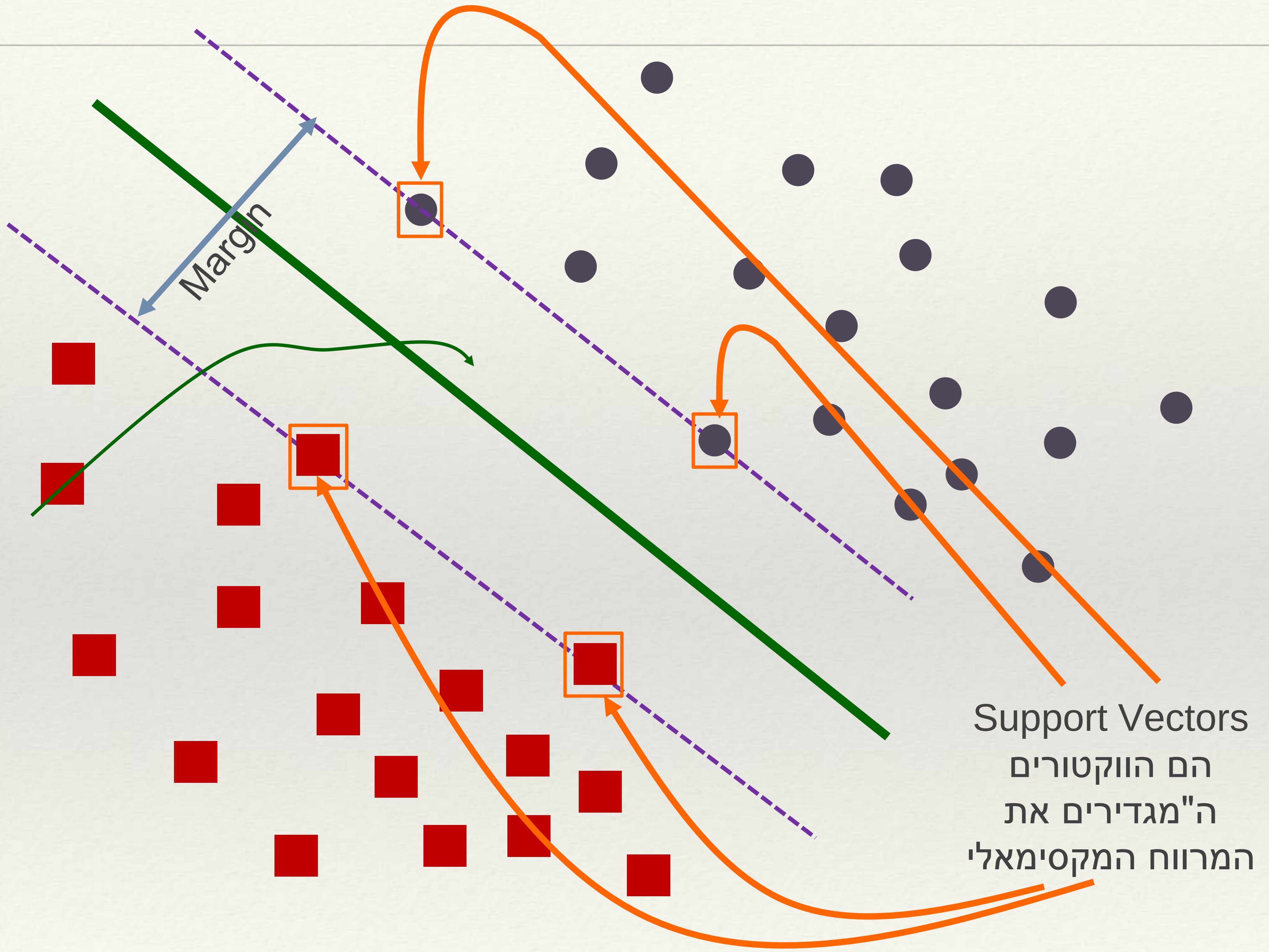
Different possible solutions



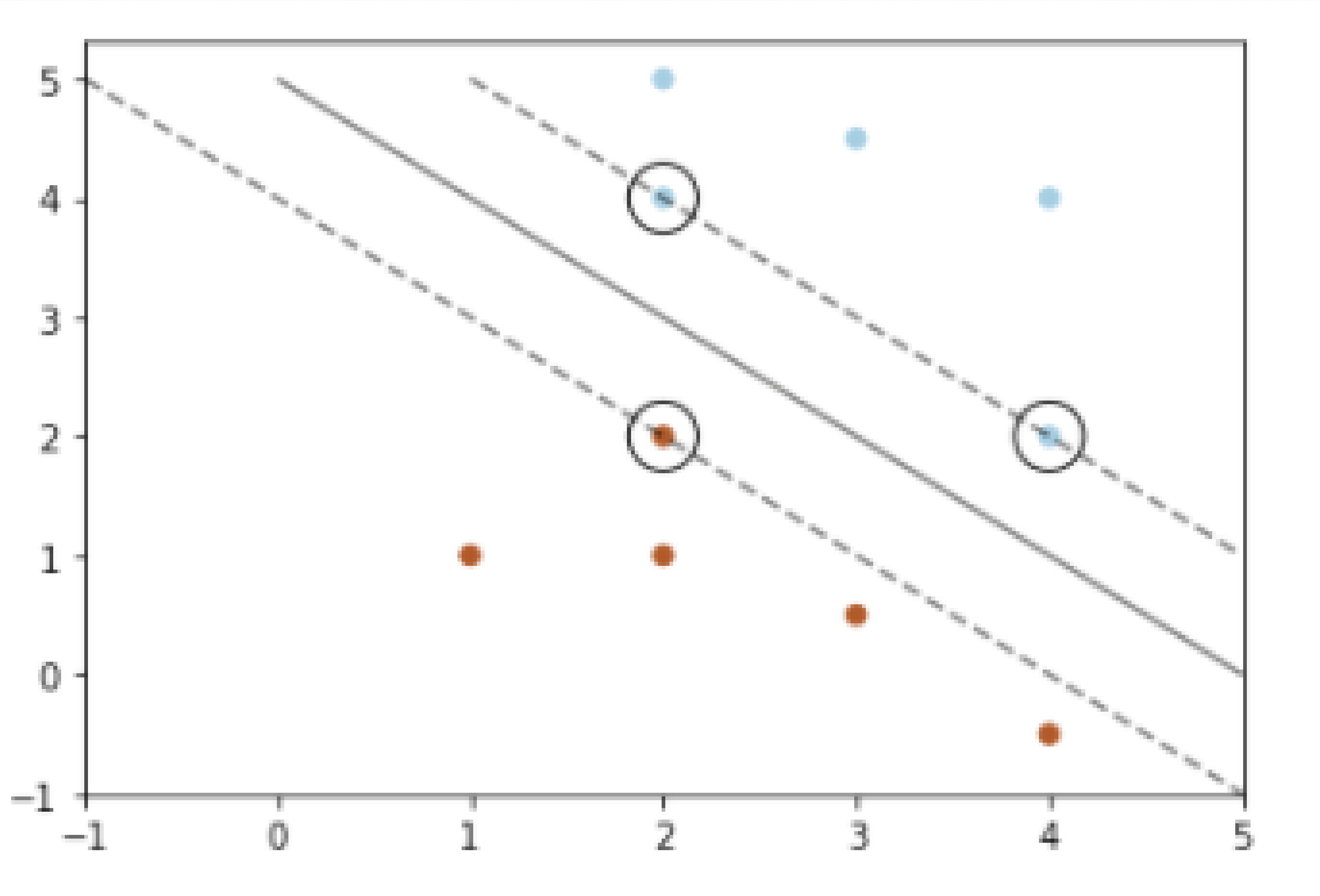
Linear SVM

$$f(x) = Wx - b$$

Different possible solutions



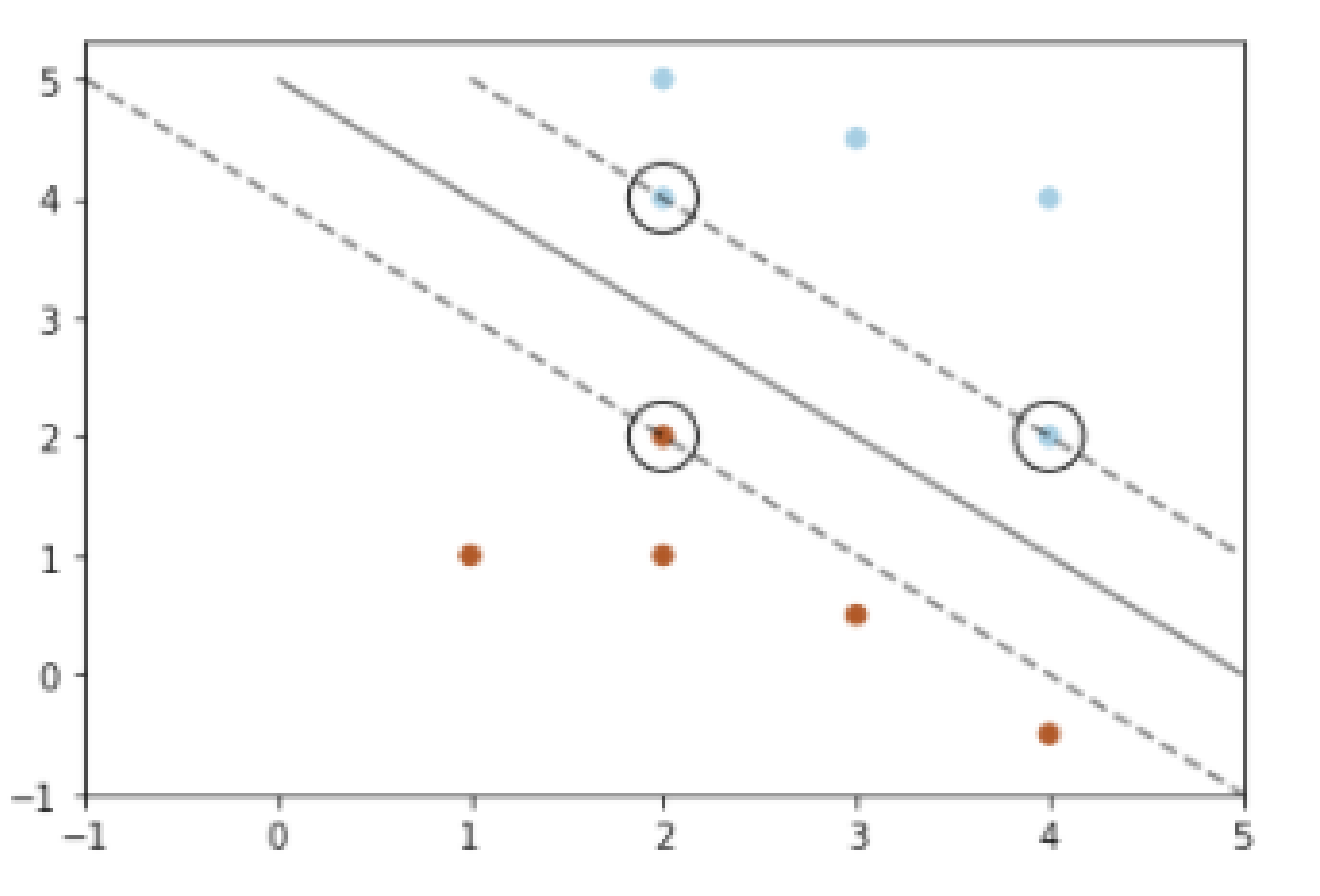
סקר



הישר $Ax + By + C = 0$

כמה וקטורים תומכים יש בדוגמא:
1
2
3
4

סקר



הישר $Ax + By + C = 0$

כמה וקטורים תומכים יש בדוגמא:

1

2

3 <– **התשובה הנכונה**

4

הדרישה מהווקטורים ב-training באלגוריתם SVM

$$(x_i \cdot w) + b - \Delta \geq 0 \text{ for } y_i = +1$$

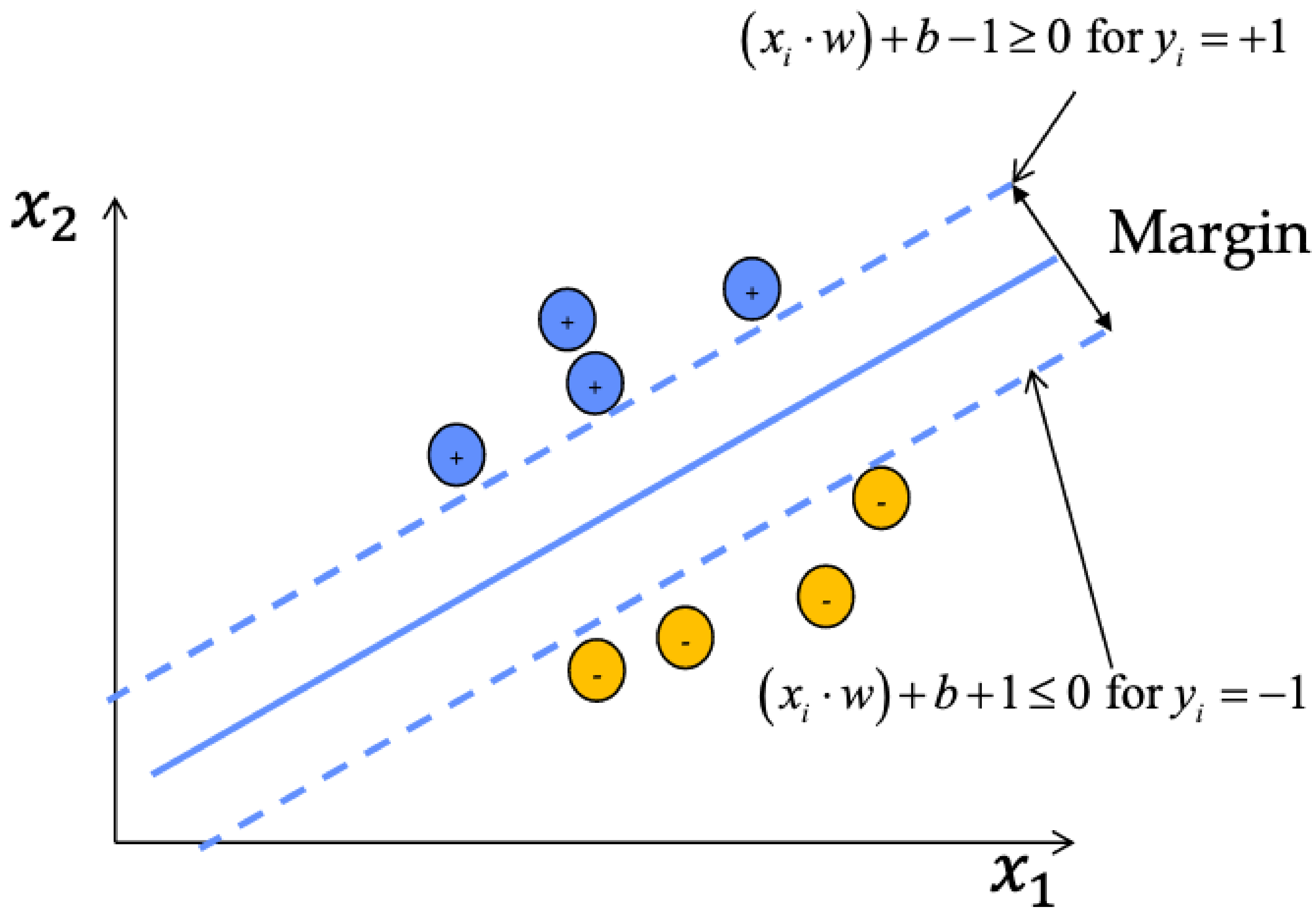
$$(x_i \cdot w) + b + \Delta \leq 0 \text{ for } y_i = -1$$

$$\Big\Downarrow \; \Delta = 1$$

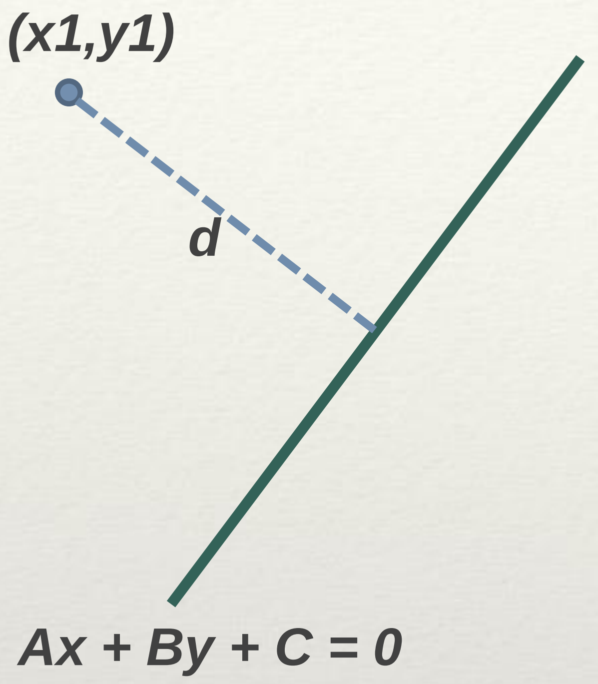
$$(x_i \cdot w) + b \geq +1 \text{ for } y_i = +1$$

$$(x_i \cdot w) + b \leq -1 \text{ for } y_i = -1$$

$$\Rightarrow y_i(x_i \cdot w + b) - 1 \geq 0 \;\; \forall i$$



מרחק נקודה מישר



- הישר Ax + By + C = 0

$$d = \frac{|Ax_1 + By_1 + C|}{\sqrt{A^2 + B^2}}$$

- במקרה הכללי – משוואת ה-hyperplane

$$\sum w_i x_i + b = 0$$

מרחק הנקודה x_s מה-hyperplane

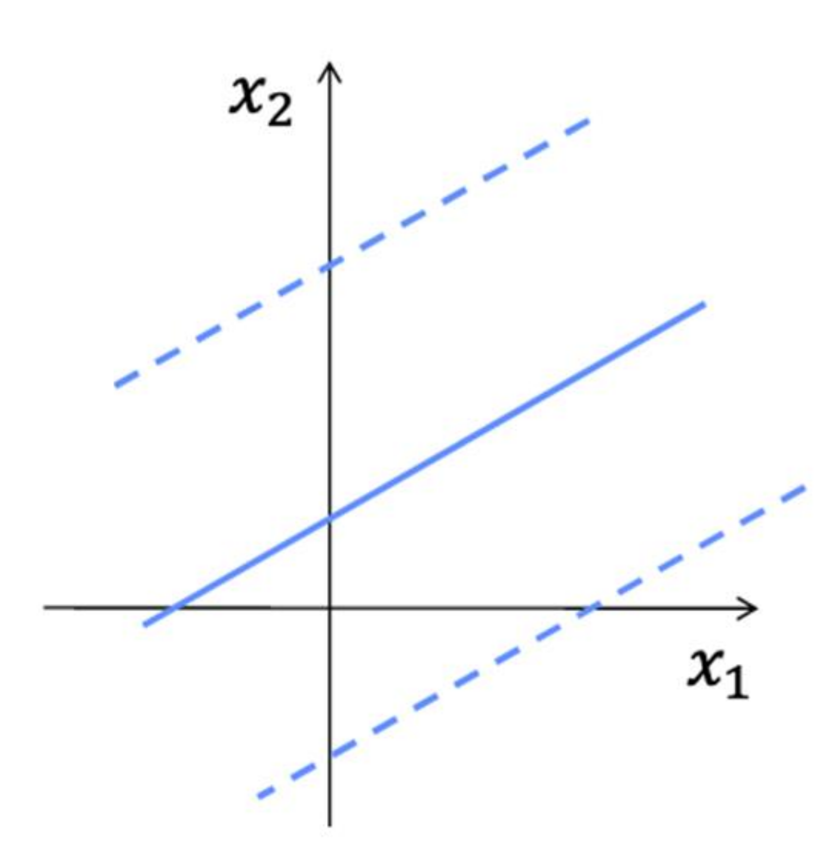
$$d = \frac{|wx_s + b|}{||w||}$$

?Margin-כיצד נחשב את גודל ה

Ax + By + C - 1 = 0

Ax + By + C + 1 = 0

$$d = \frac{|Ax_1 + By_1 + C|}{\sqrt{A^2 + B^2}}$$



$$\frac{|c-1|}{\sqrt{A^2+B^2}} + \frac{|c+1|}{\sqrt{A^2+B^2}} = \frac{(-(c-1)+c+1)}{\sqrt{A^2+B^2}} = \frac{2}{\sqrt{A^2+B^2}}$$

$$\boxed{\frac{2}{\sqrt{A^2+B^2}} = \frac{2}{\|w\|}}$$

= Margin-גודל ה

סוגי בעיות אופטימיזציה

<u>Convex programming</u> - בעיות אופטימיזציה של מזעור, בהם פונקצית המטרה הם קמורות (Convex). דוג' לכך נראה בשיעור הבא.

<u>Quadratic programming</u> – בבעיות אופטימיזציה – מאפשר לביטוי המופיע בפונקצית המטרה (אותה רוצים למקסם או למזער) להיות ביטוי ריבועי (quadratic)

- דוגמה לכך, נראה בפונקצית אופטימיזציה של SVM.
- תזכורת – ב-SVM המטרה ליצור מרווח (margin) מקסימלי בין שתי המחלקות

<u>הערה</u>: כמובן, ישנם סוגים נוספים שונים של פתרונות ובעיות אופטימיזציה

בעיית סיפוק אילוצים (constraint satisfaction problem)

<u>בעיות סיפוק אילוצים</u> הן בעיות של השמת ערכים למשתנים כך שיש אילוצים מסוימים בין ערכים

נתייחס ל-2 סוגי אילוצים אפשריים:

- אילוצי שיוויון (equality)
    - אילוצי השיוויון נראים כך:

$$g_i(\mathbf{x}) = c_i \quad \text{for } i = 1, \ldots, n$$

- אילוצי אי-שיוויון (inequality)
    - אילוצי אי-השיוויון נראים כך:

$$h_j(\mathbf{x}) \geqq d_j \quad \text{for } j = 1, \ldots, m$$

<u>*constraint optimization problems*</u> – תת סוג של בעיות סיפוק אילוצים, בהם האילוץ אינו קשיח, ובעצם המטרה, היא להוריד את מחיר האילוצים למינימום.

- אנחנו נתייחס *constraint optimization problems* ב-SVM

איך מוצאים את ה-Support Vectors

עבור ניחוש של w,b, נוכל:

- לחשב ולבדוק, האם כל ה-instances בצד הנכון של ה- -hyper plane
- לחשב את המרווח (margin) עבור ה-instances

<u>אופטימיזציה:</u>

מהו קריטיריון של ה-Quadratic optimization?

תשובה: ה-Margin המקסימלי, ששווה ערך למינימום של $(||w||^2)$

(מכיוון ש ||w|| בהגדרה הוא אי שלילי).

<u>סיפוק אילוצים:</u>

ניח שנתונים n דוגמאות אימון מסוג $\langle \vec{x_i}, y_i \rangle$, כאשר $y_i \in \{-1,1\}$

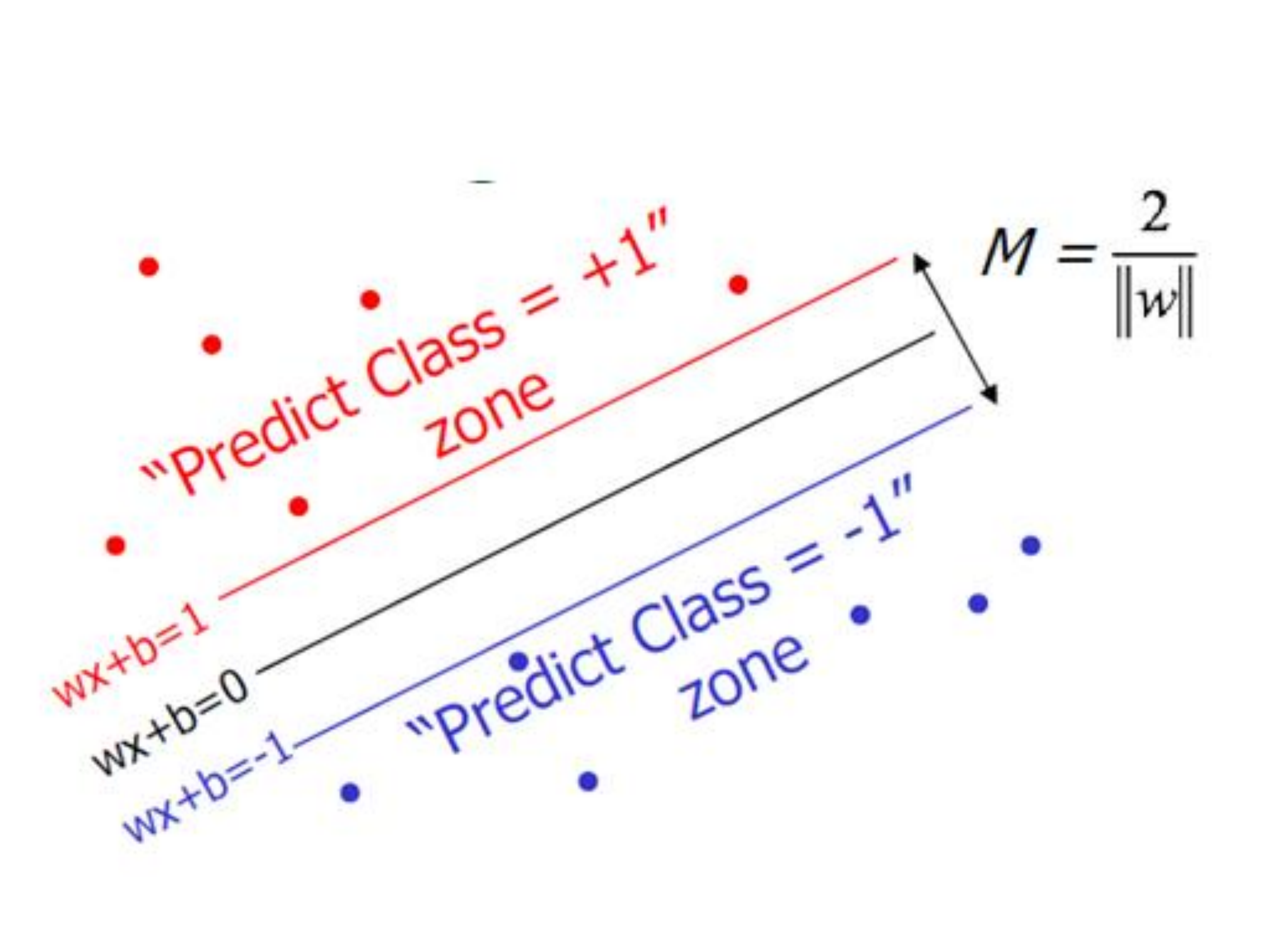
<u>שאלה</u>: כמה אילוצים נצטרך לספק?

<u>תשובה</u>: n אילוצים (כמס' הדוגמאות ב-train-set).

- $\vec{w}^T \cdot \vec{x_i} + b \geq 1$, if $y_i = 1$
- $\vec{w}^T \cdot \vec{x_i} + b \leq 1$, if $y_i = -1$ש

<u>ניתן להציג את האילוצים כך:</u>

- $y_i \cdot (\vec{w}^T \cdot \vec{x_i} + b) \geq 1$



Quadratic optimization for SVM

כיצד אנו מוצאים את הפיתרון לבעיית אופטימיזציה עם אילוצים?

תשובה: כופלי לגראנגז' (Lagrange Multipliers).

כופלי לגראנגז' - כופלי לגראנז' הם משתנים מלאכותיים אותם מוסיפים לפונקציה ממשית בת כמה משתנים, על-מנת לאפשר מציאת מינימום ומקסימום של הפונקציה בכפוף לאילוצים.

- מיועד לאילוצי שיוויון (equality)

נסמן L_p - כפונקציית המטרה הראשונית (primal) אותה רוצים למקסם בעזרת כופלי לגראנז'

נסמן L_d - עבור כופלי לגראנז' אי שליליים, ניתן להגדיר את הבעיה כבעיה דואלית (dual), ולמצוא מינימום עבור פונקציית מטרה נוספת L_d, השקולה לפונקציית המטרה הראשונית L_p

נסמן ב-α_i - את אותם כופלי לגראנגז'

בסוף האימון, לכל דוגמה ב-trainset, יוצמד α_i (כופל לגראנג') אי שלילי.

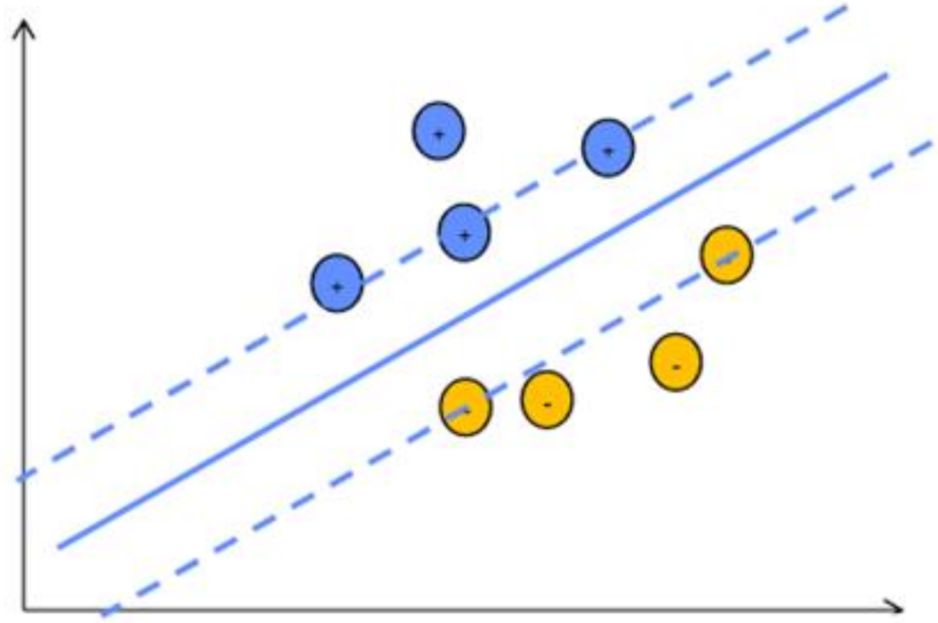
אם α_i>0 אותה דוגמה תחשב support vector (עבור שאר הדוגמאות הערך α_i=0)

סיווג ע"י המודל – Linear SVM
Step 1 – calculate *w*

Calculating w, b:

$$w = \sum_i \alpha_i y_i x_i$$

$$y_i\left(w \cdot x_i + b\right) - 1 \geq 0 \;\; \forall i$$



סיווג ע"י המודל – Linear SVM
Step 1 – calculate *w*

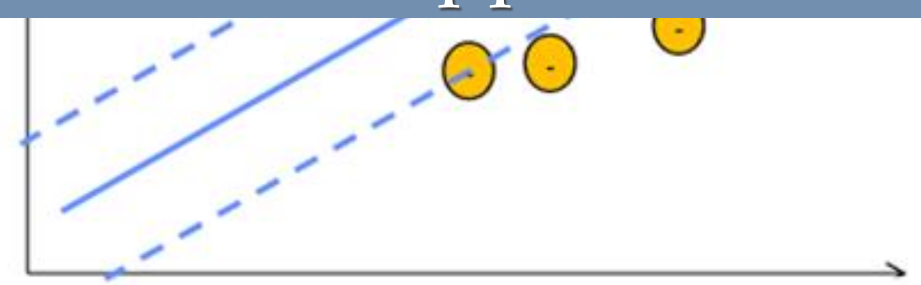
Calculating w, b:

$$w = \sum_i \alpha_i y_i x_i$$

$$y_i \left(w \cdot x_i + b \right) - 1 \geq 0 \ \forall i$$

Data points with α >0 will be the support vectors

So this sum only needs to. be over the support vectors



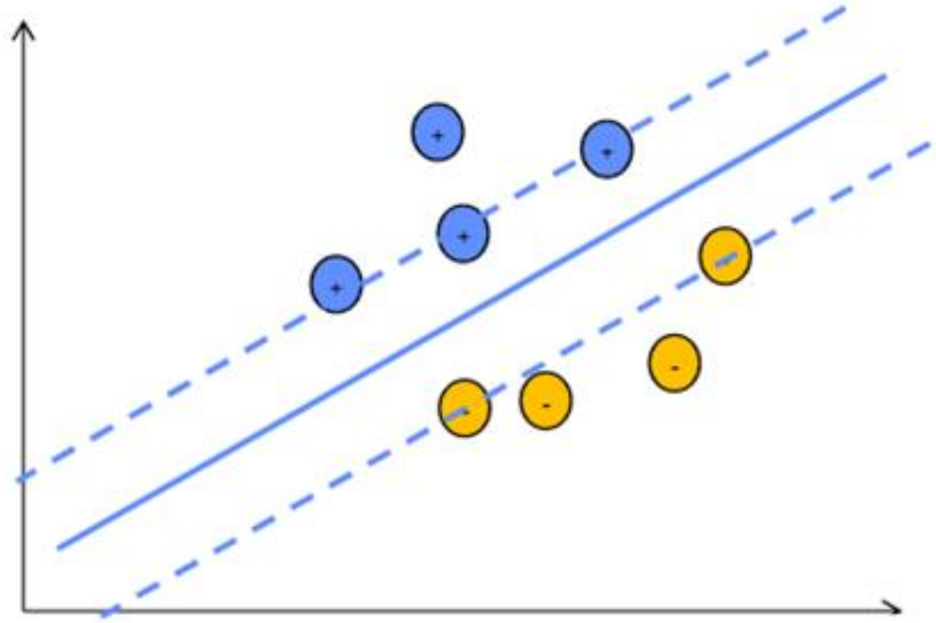
סיווג ע"י המודל – Linear SVM
Step 2 – calculate *b*

Calculating w, b:

$$w = \sum_i \alpha_i y_i x_i$$

$$y_i\left(w \cdot x_i + b\right) - 1 \geq 0 \;\; \forall i$$

The average of the nearest positive support vector and the nearest negative



$$b = -\frac{max_{y_i=-1}(\mathbf{w} \cdot \mathbf{x}_i) + min_{y_i=1}(\mathbf{w} \cdot \mathbf{x}_i)}{2}$$

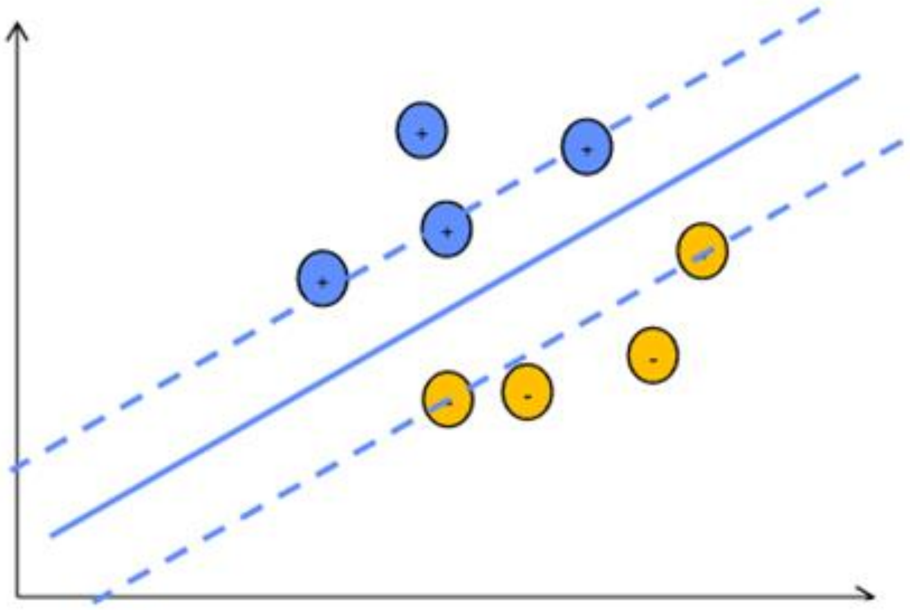
סיווג ע"י המודל – Linear SVM
Step 3 – final decision

Calculating w, b:

$$w = \sum_i \alpha_i y_i x_i$$

$$y_i (w \cdot x_i + b) - 1 \geq 0 \ \forall i$$

The average of the nearest positive support vector and the nearest negative



$$b = -\frac{max_{y_i=-1}(\mathbf{w} \cdot \mathbf{x}_i) + min_{y_i=1}(\mathbf{w} \cdot \mathbf{x}_i)}{2}$$

Final decision function:

$$f(x) = sign(w \cdot x + b) = sign\left(\sum_{i=1}^{l} \alpha_i y_i x_i \cdot x + b\right)$$

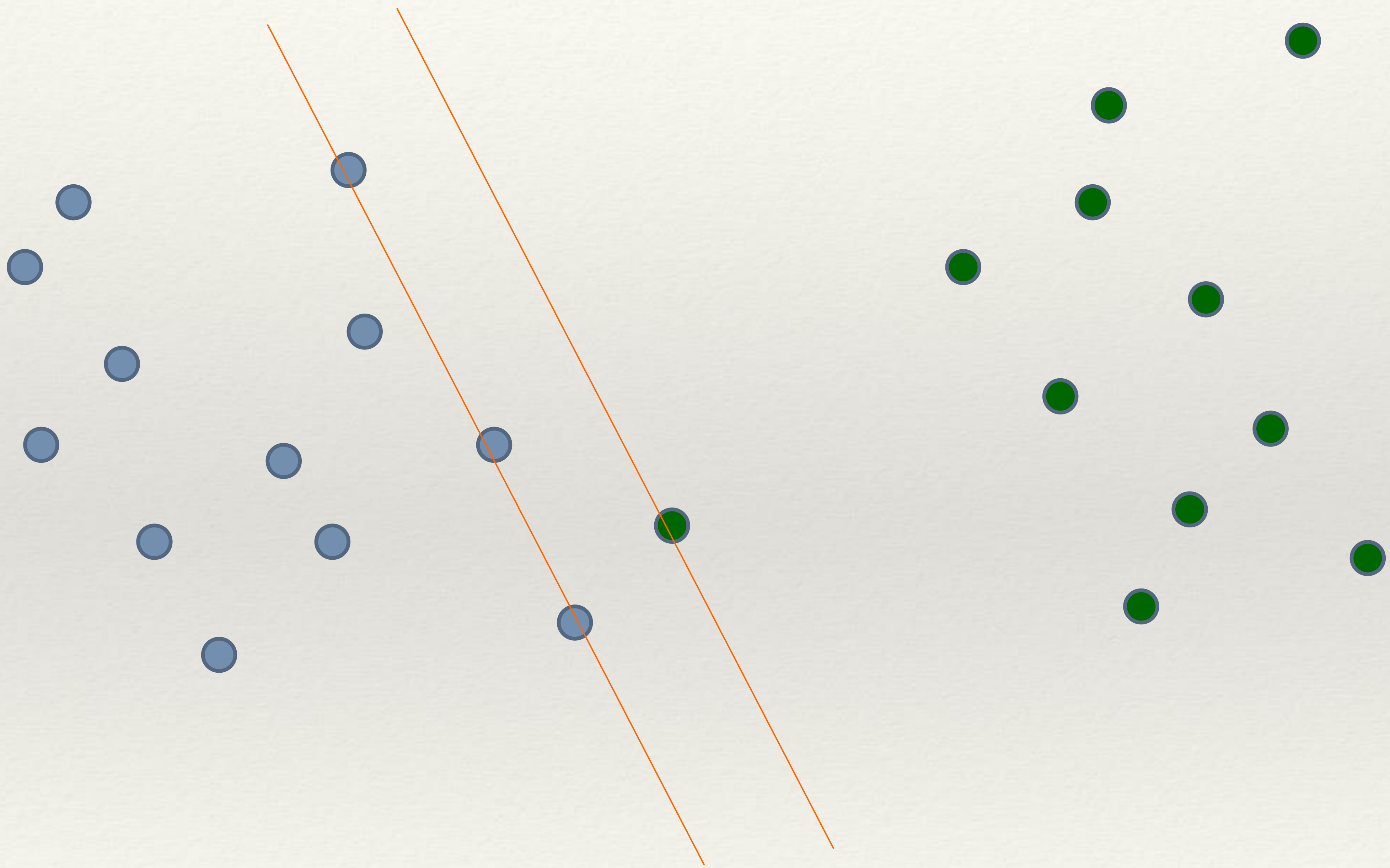
SVM ולינאריות

- **מקרה 1 - יש הפרדה לינארית מלאה וטובה (המקרה אותו כבר למדנו)**
    - נשתמש ב – maximize margin *(Hard Margin)*

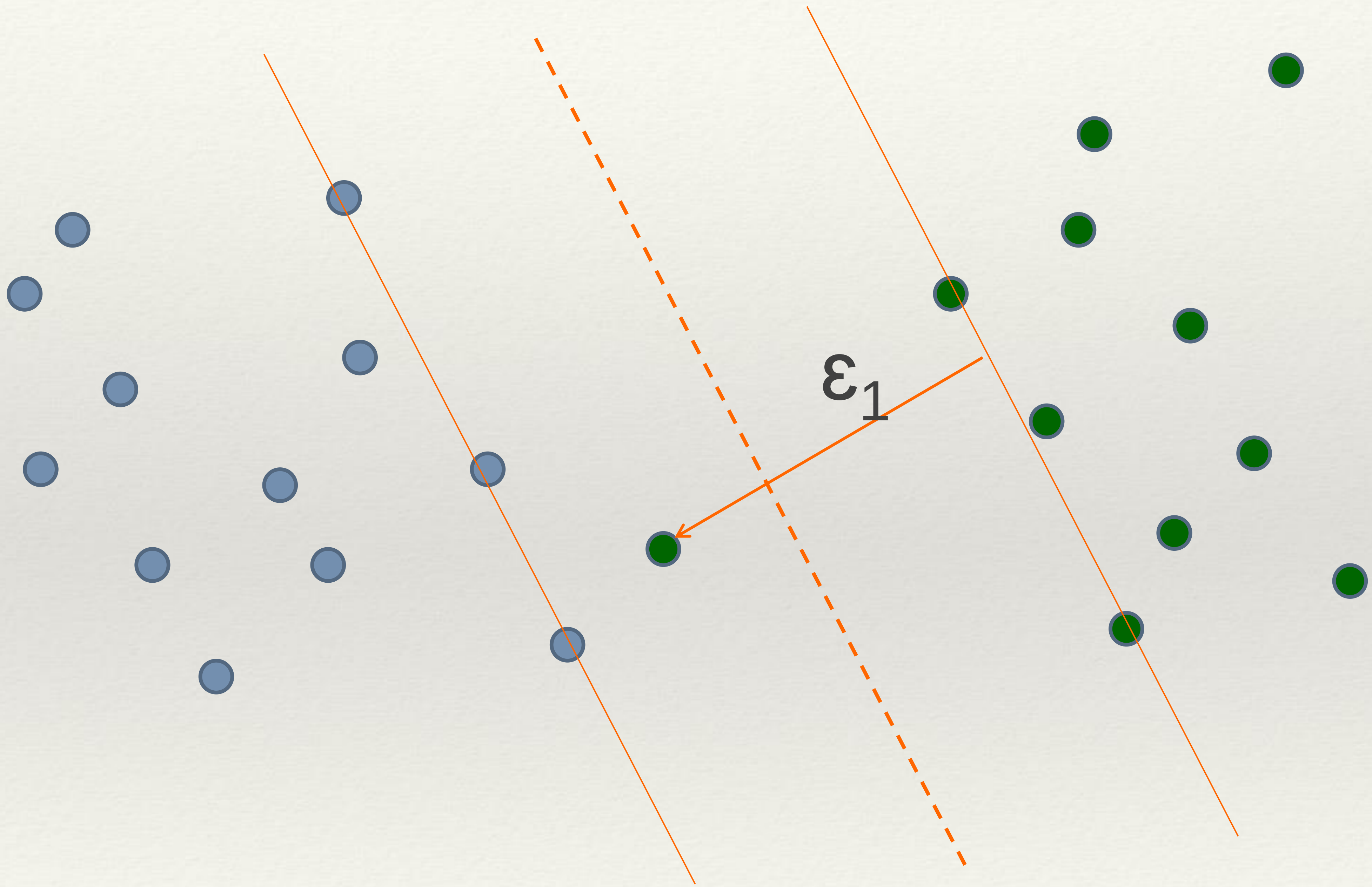
מקרים לא-לינאריים (אין הפרדה לינארית או שאין הפרדה לינארית טובה)

- **מקרה 2 – אין הפרדה לינארית (או שאין הפרדה טובה), אך עבור מספר קטן של דוגמאות מהאימון**
    - נשתמש ב-מדד "ענישה" עבור טעויות *(Soft Margin)*
- **מקרה 3 – אין הפרדה לינארית עבור מספר גדול של דוגמאות מהאימון**
    - "מיפוי" למימד גבוה יותר בו קיימת הפרדה לינארית *(Kernel function)*

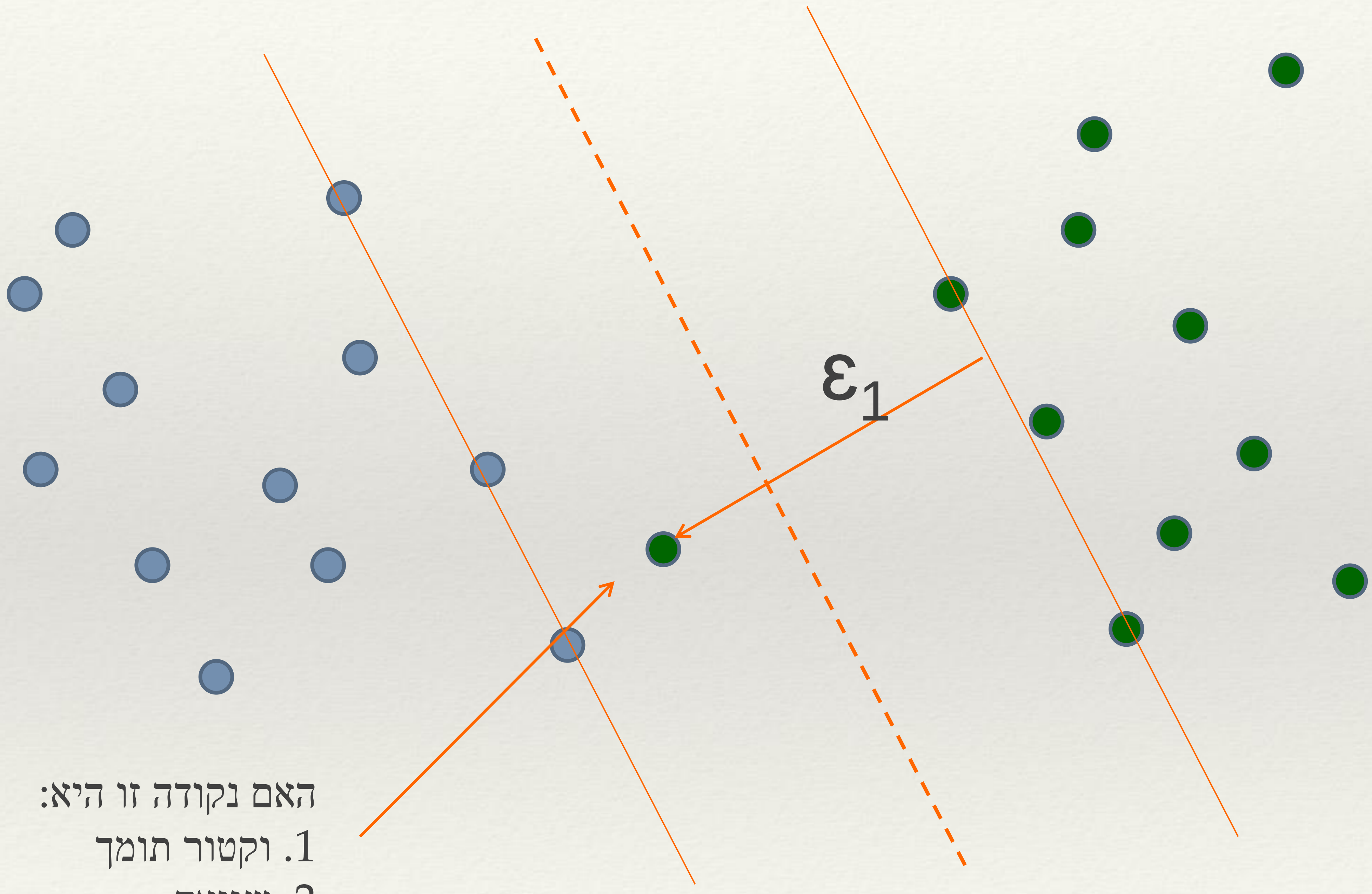
מקרה 2 – אין הפרדה לינארית (או שאינה מוצלחת)
השימוש ב- Hard Margin



מקרה 2 – אין הפרדה לינארית (או שאינה מוצלחת) השימוש ב- Soft Margin



סקר

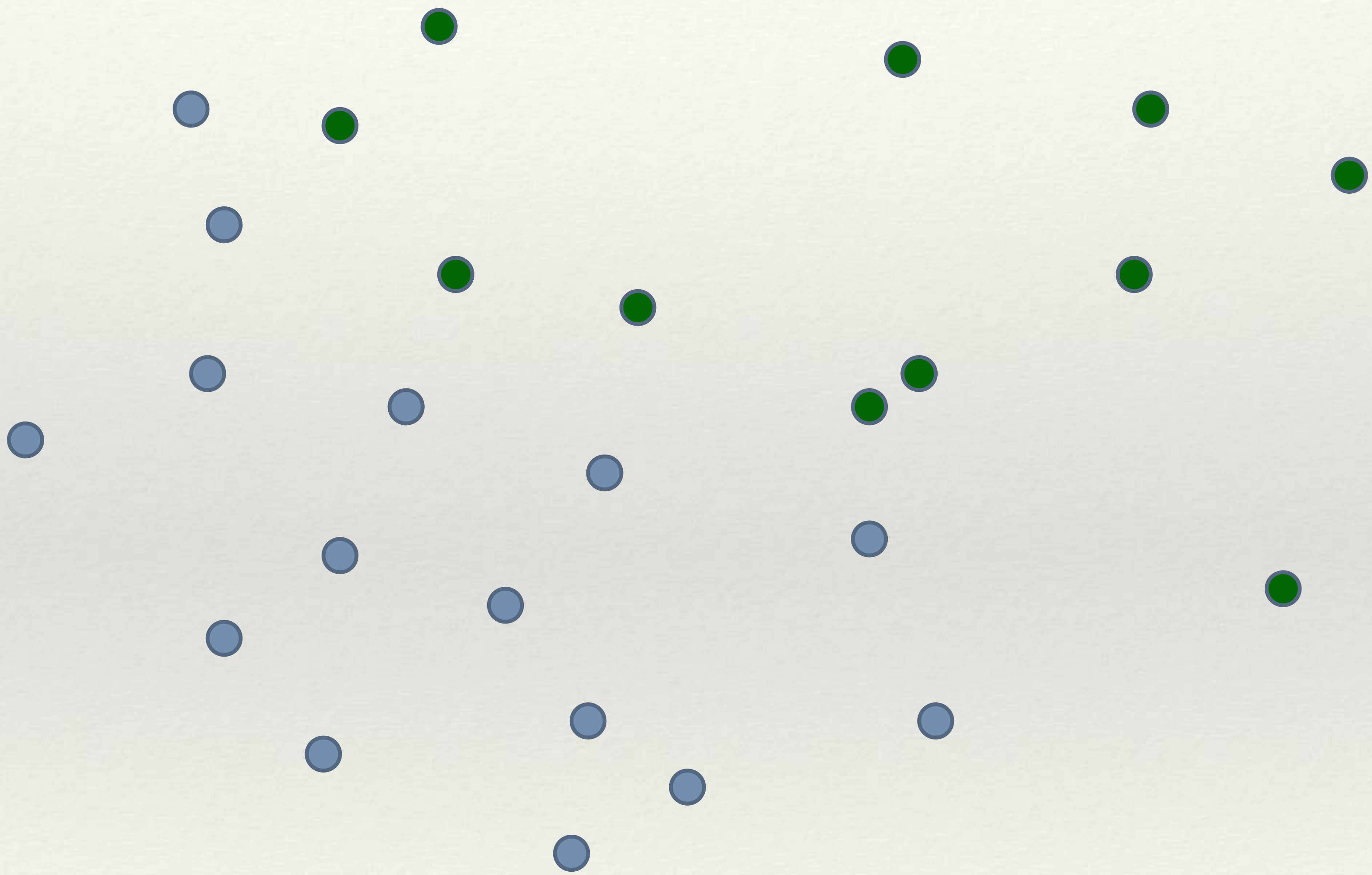


האם נקודה זו היא:
1. וקטור תומך
2. שגיאה

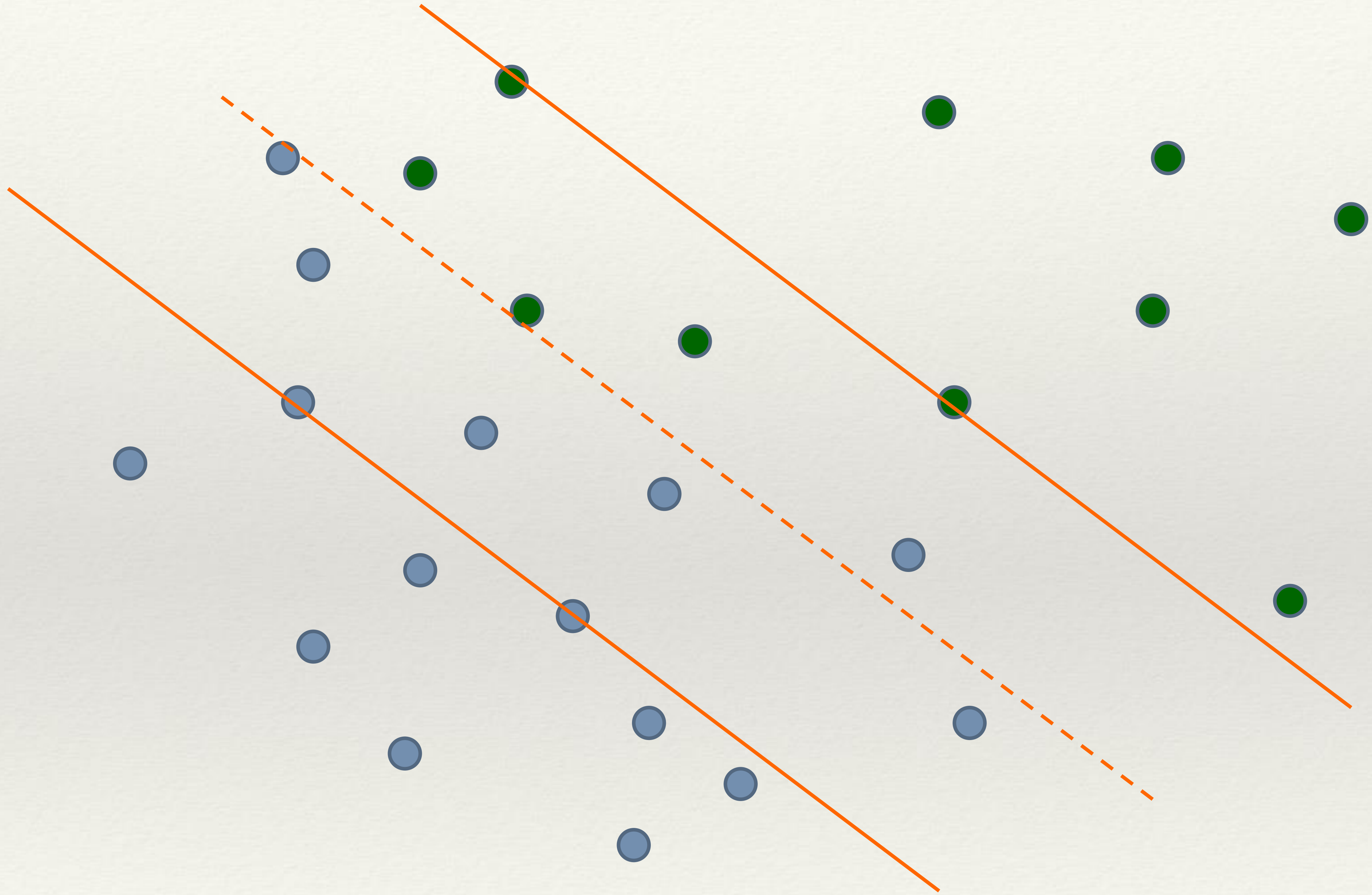
Soft Margin

- יתרונות:
  - תמיד יש פתרון (גם כשאין הפרדה לינארית)
  - עמידות בפני outliers

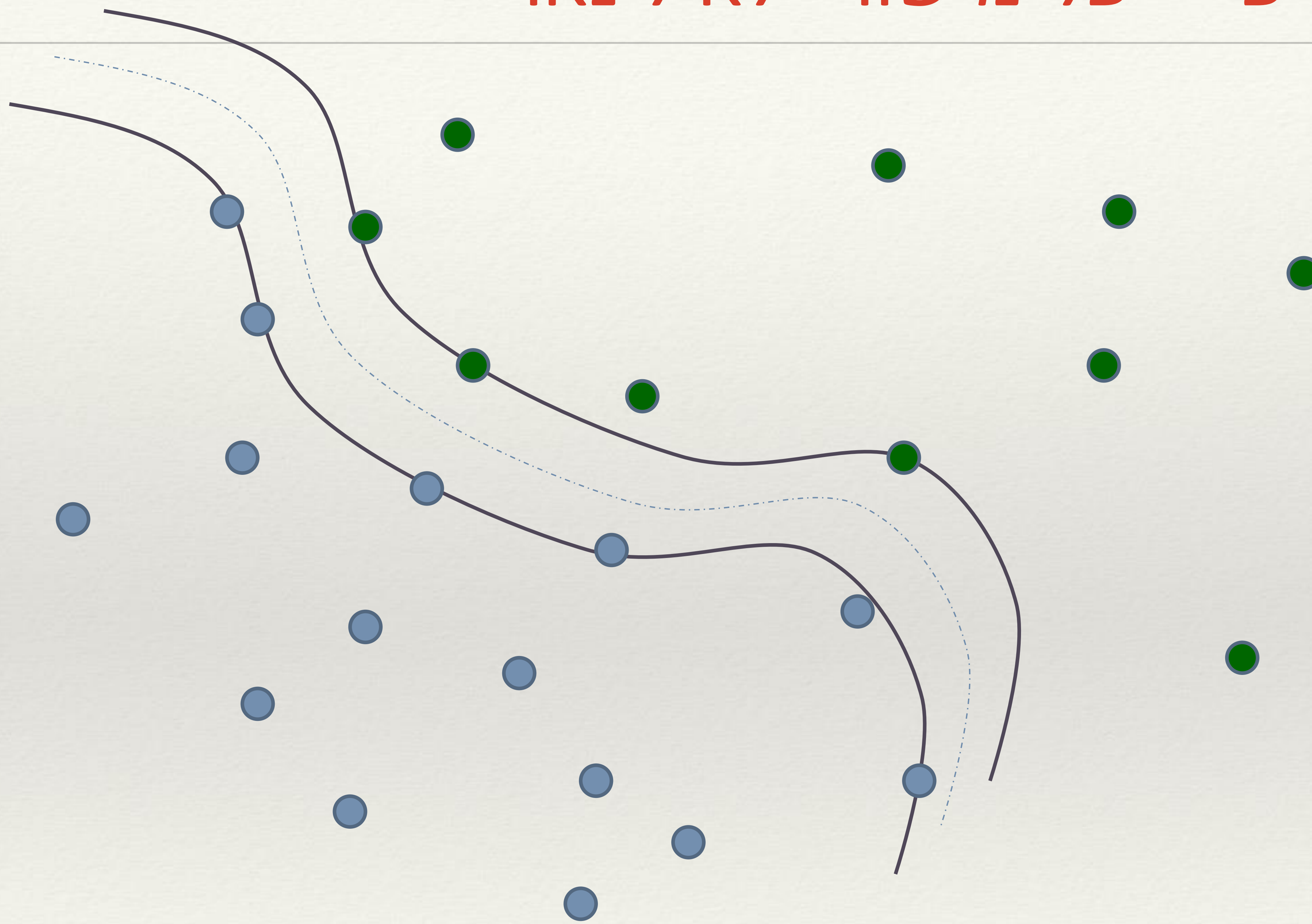
Non-Linear Problem – מקרה 3



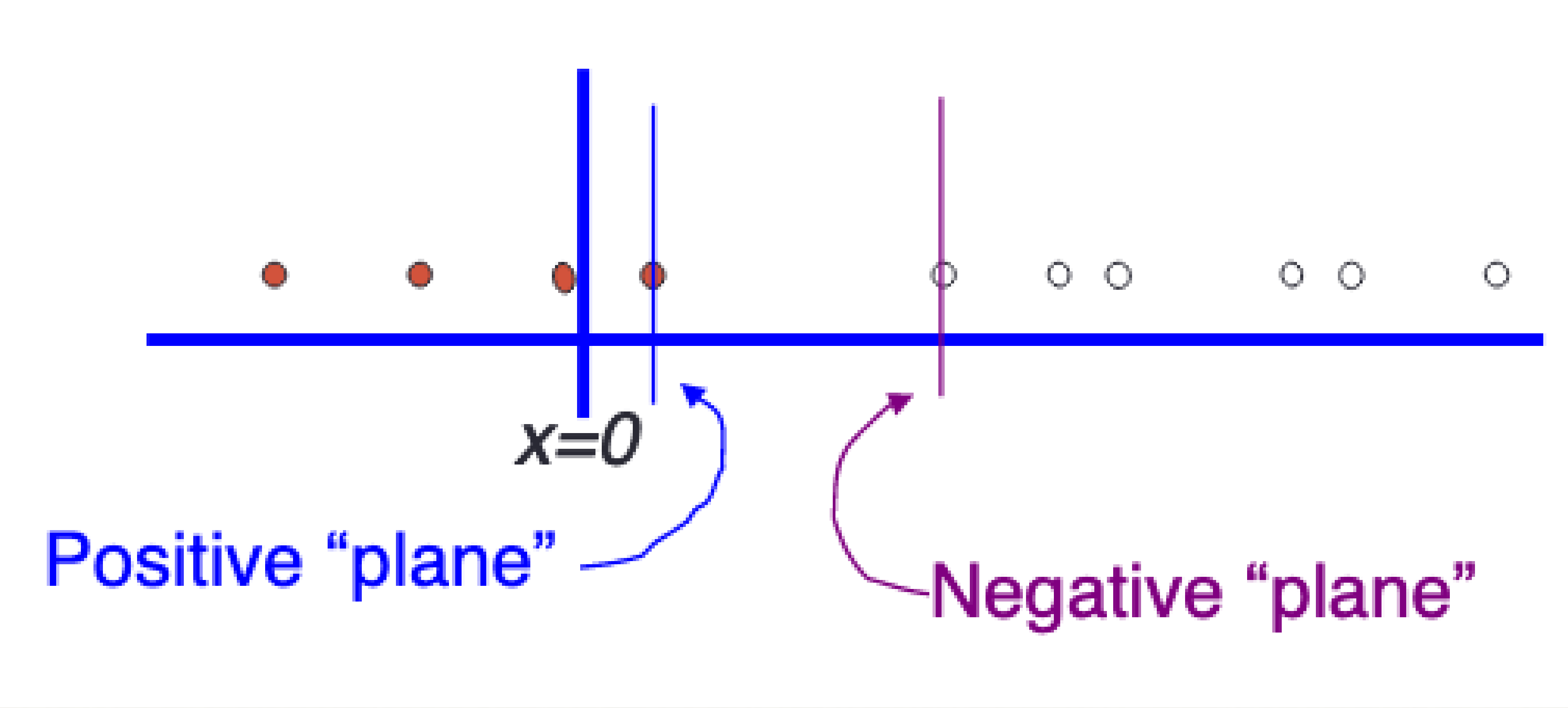
Non-Linear Problem – 3 מקרה
Soft Margin -השימוש ב



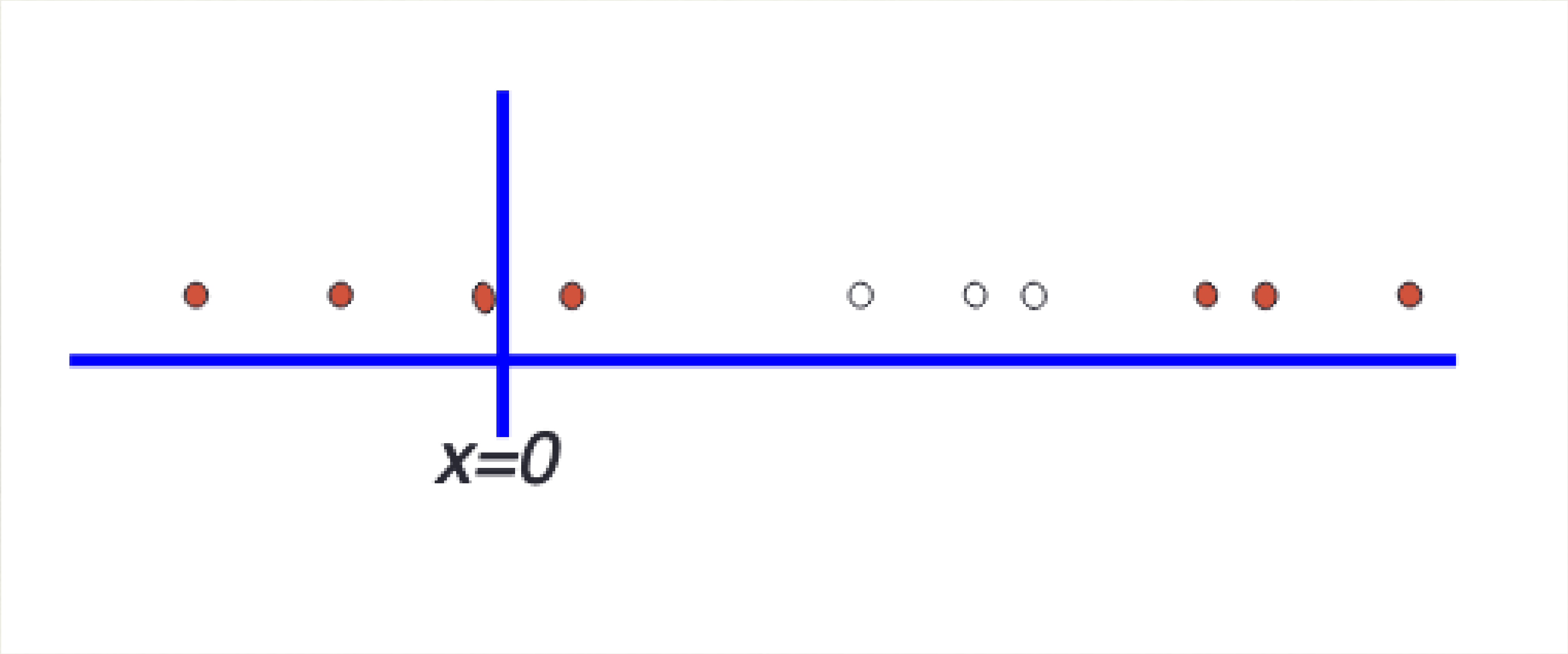
Non-Linear Problem – 3 מקרה
פתרון ע"י "על-מישור" לא לינארי



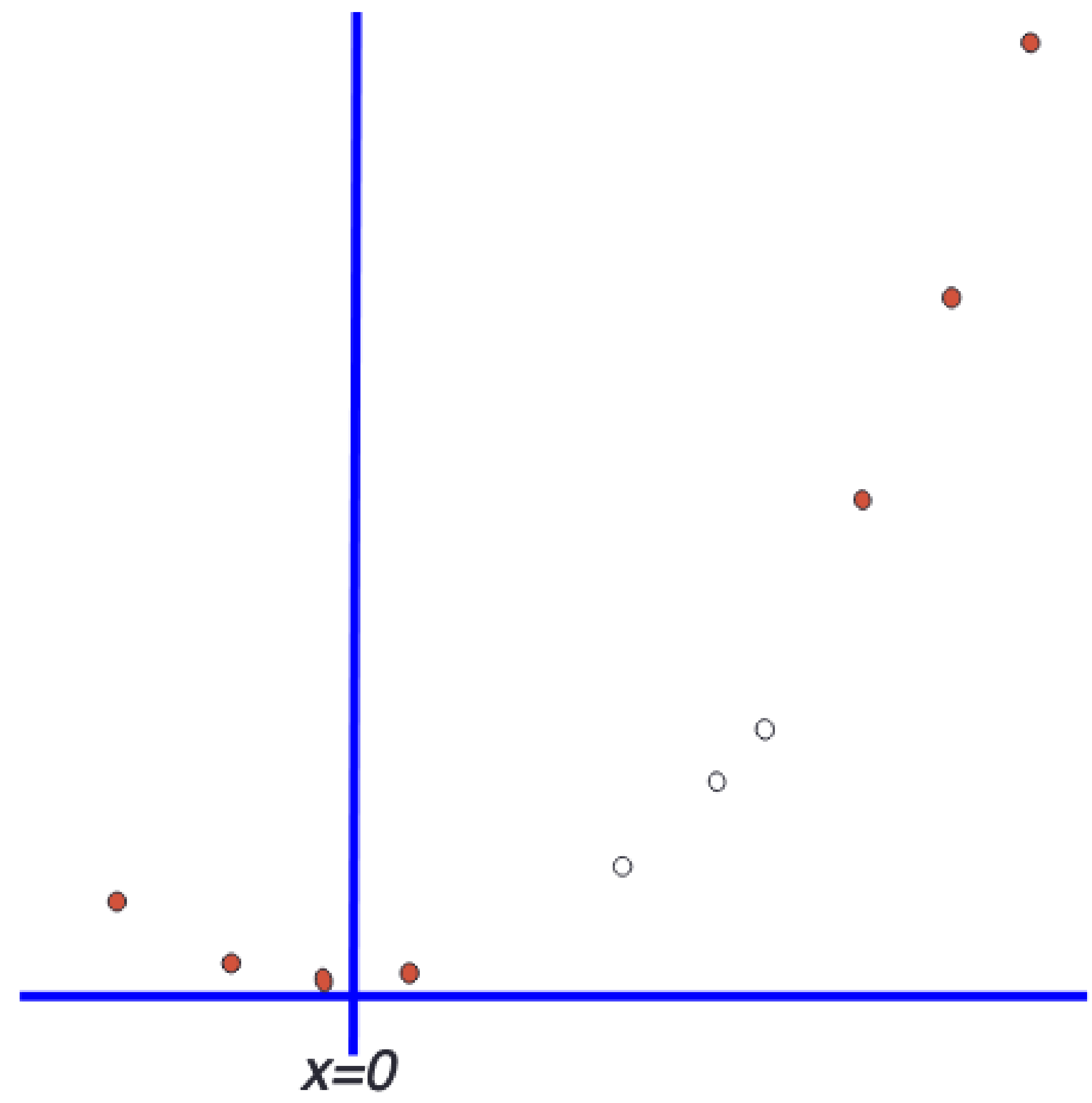
מקרה 3 – Non-Linear Problem
נניח שאנו במרחב חד ממדי – בעיה פשוטה



מקרה 3 – Non-Linear Problem
נניח שאנו במרחב חד ממדי – בעיה פשוטה

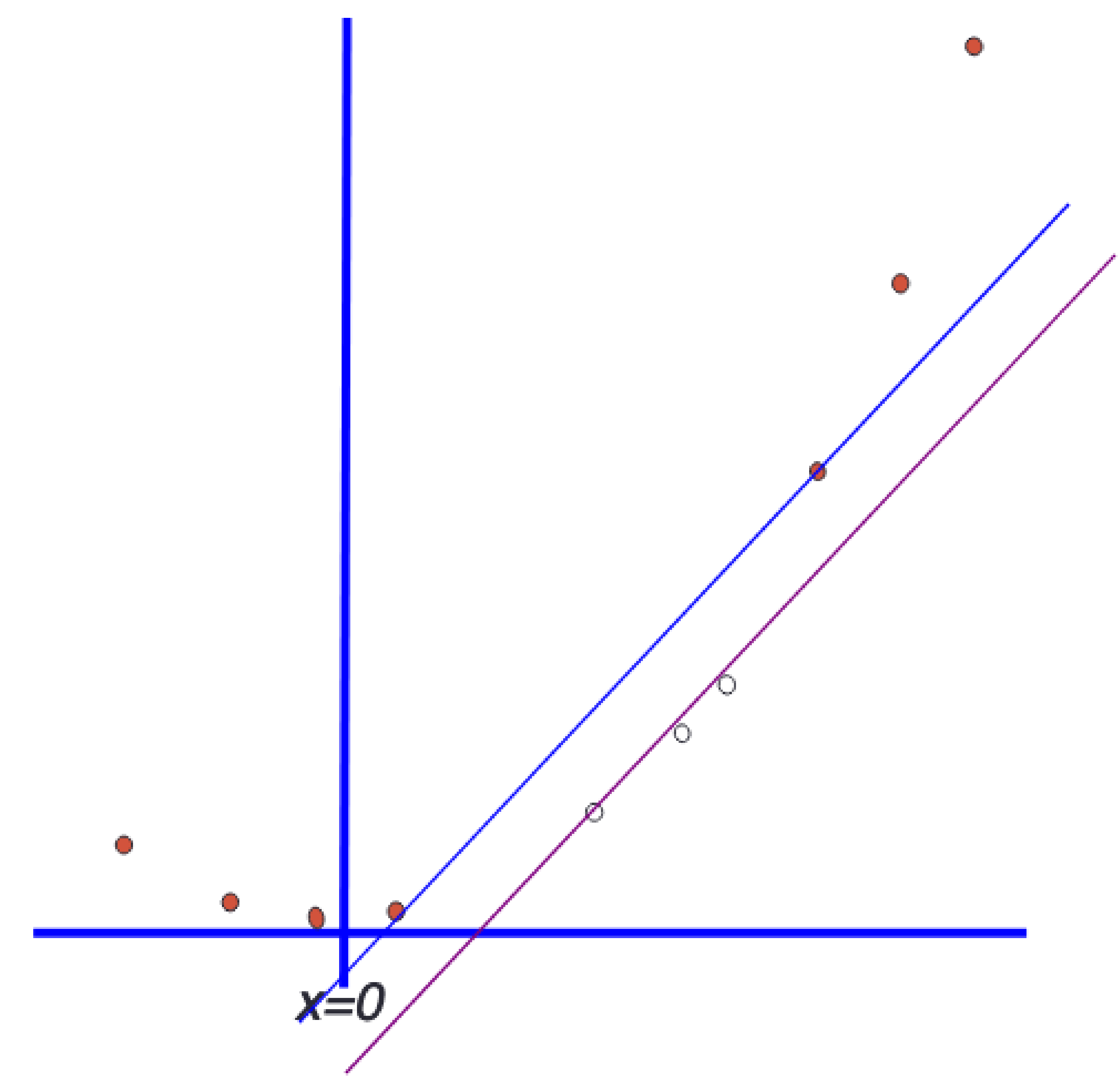


מקרה 3 – Non-Linear Problem
שימוש בפונקציית kernel - נמיר אותה לבעיה במרחב דו ממדי



$$x \mapsto (x, x^2)$$

מקרה 3 – Non-Linear Problem
פתרון SVM במרחב דו ממדי



$$x \mapsto (x, x^2)$$

סיכום SVM – יתרונות וחסרונות

יתרונות:

- אימון יעיל ומהיר (יחסית – במיוחד בממדים נמוכים)
- ביצועים ודיוק טובים מאד
- עובד טוב במקרים של ריבוי מאפיינים
- משמש במגוון בעיות: טקסט, OCR, ביו-אינפורמטיקה, שפה ועוד
- יחסית מעט פרמטרים לבחירה
- יסודות תיאורטיים חזקים

חסרונות:

- SVM עובד רק עם מאפיינים מספריים
- SVM פותר רק בעיות בינאריות
- לבחור kernel מתאים לא תמיד פשוט כ"כ..
- המודל "לא ניתן להסבר" בפשטות