

## מבוא לבינה מלאכותית – פתרון תרגיל 2:

1. ראשית, דרך הכתיבה של בעיית SVM דואלי לפתרון בעיית אופטימיזציה ריבועית: לשם תזכורת, הבעיה הדואלית (hard-margin) מנוסחת כך:

$$\begin{aligned} \max_a \quad & \sum_i a_i - \frac{1}{2} \sum_{i,j} a_i a_j y_i y_j \langle x_i, x_j \rangle \\ \text{s. t.} \quad & \sum_i a_i y_i = 0, \\ & a_i \geq 0 \quad \forall i \end{aligned}$$

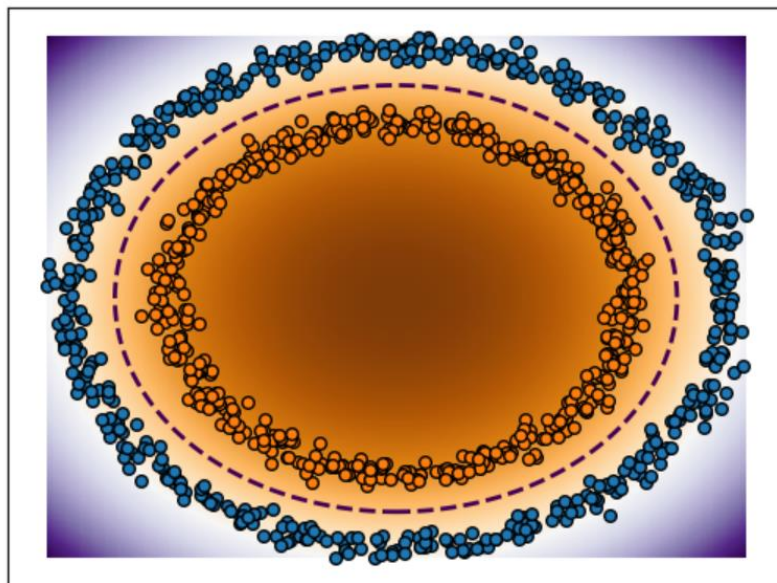
כאשר  $y_i \in \{-1, 1\}$  התיוג של הנקודה  $x_i$ , וההגעה לפתרון המקורי מהדואלי נעשית ע"י:  $w = \sum_i a_i y_i x_i$  והערך של  $b$  (המקדם החופשי במפריד הלינארי) הוא ערך הפתרון של כופל לגראנז' המתאים לאילוץ השוויון.

כדי להשתמש באלגוריתם לאופטימיזציה של בעיה ריבועית (כאן ספציפית OSQP), נצטרך להשתמש בכתיב מעט שונה לאותה הבעיה. בנוסף, כדי להשתמש ב-kernel, נחליף בניסוח את השימוש במכפלה הפנימית  $\langle x_i, x_j \rangle$  במטריצה  $K_{ij} = \psi(x_i) \cdot \psi(x_j)$ .  
נקבל:

$$\begin{aligned} \min_a \quad & \frac{1}{2} a^T \begin{pmatrix} y_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & y_N \end{pmatrix} K \begin{pmatrix} y_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & y_N \end{pmatrix} a - (1 \dots 1) \cdot a \\ \text{s. t.} \quad & \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \leq \begin{pmatrix} 1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & 1 \\ y_1 & \dots & y_N \end{pmatrix} a \leq \begin{pmatrix} \infty \\ \vdots \\ \infty \\ 0 \end{pmatrix} \end{aligned}$$

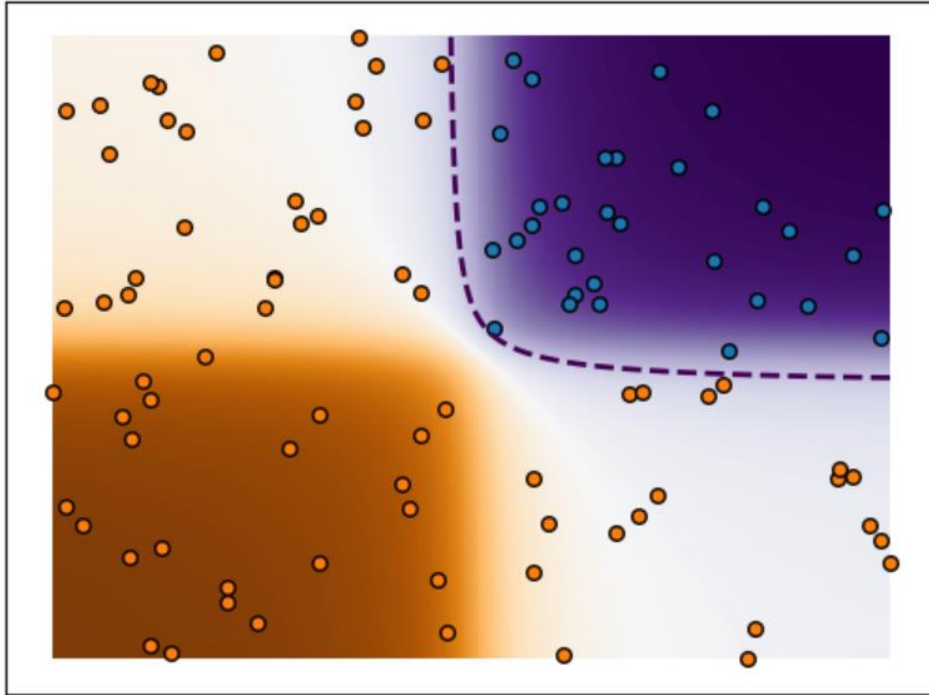
א. ניתן לשרטט ולראות שהדאטא הנתון בסעיף זה מחולק למעגלים, מעגל פנימי בצבע אחד ומעגל חיצוני בצבע שני.

נשתמש בהעתקה הבאה:  $\psi((x, y)) = (x, y, x^2 + y^2)$  (למעשה מספיק רק הרכיב האחרון) ונקבל דאטא במרחב חדש שבו ניתן להפריד לינארית (במרחב החדש, הנקודות מתיישבות על פרבולואיד והצבעים מתחלקים יפה על תחומי ערכי  $z$  שבעזרתם נוכל להפריד. ההפרדה במרחב המקורי נראית ככה:



ב. הפעם, הדאטא מחולק ככה שכל נקודה ברביע הראשון שייכת לתיוג 1 וכל נקודה ששייכת לאחד משלושת האחרים מתיגת 0.

נשתמש בהעתקה כזו:  $\psi((x, y)) = (\arctan(10 * x), \arctan(10 * y))$  (אפשר לחשוב על העתקות דומות, למשל להחליף את ה- $\arctan(x)$  בסיגמואיד או בפונקציות אחרות שעבור ערכים חיוביים נותנות בקירוב ערך מסוים ועבור שליליים ערך אחר לגמרי). התוצאה:



שימו לב שבשני הסעיפים הגדרתי ממש העתקה  $\psi(x)$ , אבל זה ממש לא חובה כדי לפתור את הבעיה (אפשר למצוא את ערך פונקציית המסווג לגרעין כלשהו, גם בידיעה של פונקציית הגרעין  $K$  וללא שימוש ב- $\psi(x)$ ).

2. כפי שיצא לי, טעיתי בכתיבת הנקודות.

מבדיקה קצרה שעשיתי על טווח התנאים ההתחלתיים  $(0.45, 0.6)$  עד  $(0.55, 0.6)$  בקפיצות של 0.01 בציר  $x$ , קיבלתי שנקודת המינימום אליה מתכנס האלגוריתם היא:  $(-1.79896, 1.47916)$  עבור כל הערכים מ- $(0.45, 0.6)$  עד  $(0.509, 0.6)$  כולל, ואילו עבור שאר הערכים האלגוריתם מתכנס ל- $(1.52278, 1.32360)$ .

המסקנה מכאן היא שאלגוריתם gradient descent רגיש מאוד לתנאי ההתחלה, ושההליכה אל מורד הגראדיינט בצורה חמדנית לא תמיד מביאה אותנו אל נקודת מינימום גלובלית.