

# מבוא לבינה מלאכותית – תרגול 13

## נושא:

גילוי חריגות – Anomaly Detection:

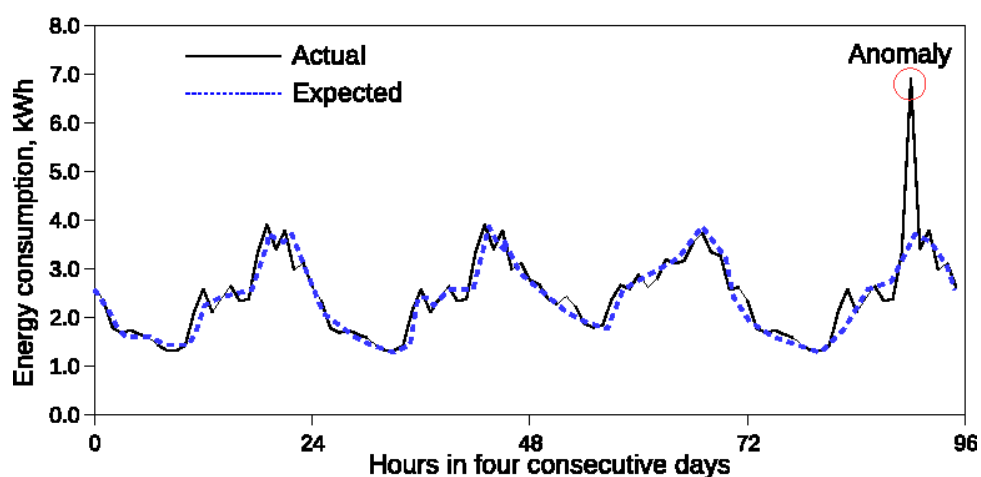
- בצורה מפוקחת
- בצורה חצי – מפוקחת (Semi-Supervised)
- בצורה לא מפוקחת: גישה סטטיסטית, אשכול, One class SVM, גישה מבוססת צפיפות

## רקע:

המון בעיות בחיים האמיתיים הן סוג של גילוי חריגות. למשל, בחברות פיננסיות כמו בנק ואחרות מנסים למצוא מתי מתקיימות הונאות, בחברות אבטחה וסייבר מנסים למצוא מתי מתבצע ניסיון פריצה, מפעלים מנסים לזהות באילו מוצרים נפלו פגמים בייצור, בבתי חולים מנסים לזהות מתי מדדים רפואיים של מאושפזים מצביעים על הידרדרות במצבם ועוד.

המשותף לכולם: בבעיות האלה יש דאטא המורכב מאוסף דגימות, שרובן נחשבות עבורנו "שגרתיות" (נתייג אותן 0) ויש חלק קטן מתוכן שנחשב נקודות "חריגות" (נתייג אותן 1). המטרה שלנו היא לנסות לזהות כמה שיותר חריגות (recall גבוה ככל האפשר), תוך ניחוש ממוקד ככל האפשר (precision גבוה ככל האפשר).

בתרגול הזה נסקור שיטות לגילוי חריגות בכל מיני מקרים ועם כל מיני גישות לגבי הדאטא.



## גילוי חריגות בבעיות מפוקחות:

המקרה הפשוט ביותר של הבעיה: יש לנו דאטא מתויג של דגימות, נבנה מודל לסיווג בינארי של הדאטא כדי שנוכל לסווג האם דגימות חדשות שנכנסות למודל הן חריגות או לא.

האתגר המרכזי שיש לנו כאן הוא חוסר האיזון – יש לנו כמות משמעותית של דגימות עם תיוג 0 לעומת כמות הדגימות המתויגות 1. אפשר לבצע כל מיני פעולות להתמודדות עם חוסר האיזון, למשל לתת משקל גבוה יותר לשגיאה על דגימה חריגה בפונקציית ה-loss, לבצע "דגימת יתר" (oversampling). זו שיטה שבה מנסים להתמודד עם חוסר איזון על ידי למידה של דגימות חריגות בתדירות גבוהה יותר מאשר התדירות הקיימת שלהם בתוך הדאטא) ועוד.

ניתן למדוד את הדיקו שלנו בבעיות מפוקחות באמצעות כלים שלמדנו בשיעור שעבר, כמו Recall, Precision, AUC (וכן F1-score). כאן חשובה לנו הרבה יותר טעות מסוג False Negative מאשר False Positive ולכן נייחס לה חשיבות גבוהה יותר.

הקושי העיקרי בבעיות כאלה הוא שבדרך כלל חוסר האיזון חמור מדי – יש כמות זניחה מאוד של דגימות חריגות בתוך הדאטא (למשל, 7 חריגות מתוך 900). במצב כזה, הלמידה על הדאטא קשה במיוחד וגם אם נלמד עליו, יכולת ההכללה שלנו לא תהיה גבוהה – נוכל לזהות דאטא חריג אך ורק מסוג דומה לדגימות חריגות שראינו.

## גילוי חריגות בבעיות חצי – מפוקחות (Semi-Supervised):

באופן כללי, בעיות למידה חצי – מפוקחות הן סוג של בעיות שבהן נתון אוסף של נתונים, אבל רק חלק מתוכם מתויגים, והמטרה היא להשתמש בתיוג הקיים לנקודות הידועות וגם בנקודות הלא ידועות כדי לסווג את כל הדאטא (ולהכליל גם לנקודות לא נתונות). בפרט, במשימות של גילוי חריגות, בדרך כלל נתון דאטא שידוע התיוג של חלק מהנקודות שבו, וכולן ידועות בתור שגרתיות. דרך התמודדות אפשרית עם צורה כזו של בעיה היא הכנסת prior לבעיה, כלומר רגולריזציה על פונקציית ה-loss, שתקבע כי יש סיכוי  $p$  כלשהו לנקודות לא מתויגות להיות שגרתיות וסיכוי  $1 - p$  להיות חריגות (לא נשייך תיוג אקראי ספציפי לנקודות כי זה יבלבל ממש את המודל. חכם יותר יהיה להסתכל על התפלגות התיוגים של כל הנקודות ולהתאים אותה למה שרוצים).

## גילוי חריגות בצורה לא מפוקחת:

הצורה הנפוצה ביותר לגילוי חריגות. נתונות לנו הדגימות עצמן, אוסף של וקטורי פיצ'רים. נשתמש בכל מיני רעיונות מרכזיים וגישות שבוחנות אותם במטרה לזהות את הדגימות החריגות מהדאטא.

### גישה סטטיסטית:

בגישה הזו, מחשבים סטטיסטי כלשהו על כל דגימה ומפתחים מבחן על הסטטיסטי הזה, כך שנקודות שלא עומדות במבחן תיחשבנה לחריגות. למשל, נבצע אשכול GMM (כך שידועים לנו כל המרכזים  $\mu_i$  ומטריצות השונות המשותפת שלהם  $\Sigma_i$ ), ונחשב לכל נקודה סטטיסטי שיהיה מרחק Mahalanobis של הנקודה ממרכז האשכול שאליו שייכנו אותה בסיכוי הגבוה ביותר:

$$S(x) = d(x, \mu) = \sqrt{(\vec{x} - \vec{\mu})^T \Sigma^{-1} (\vec{x} - \vec{\mu})}$$

כעת, נקבע  $z$  כלשהו וכל הנקודות שעבורן  $S(x) > z$  תהיינה הנקודות החריגות. לכל מרחק  $z$  קיים  $p$ -value מתאים של המודל, וניתן לבחור את  $z$  לפי ערך  $p$ -value שנרצה (לא נמוך מדי כדי שלא נצא עם מצב ללא נקודות, ולא גבוה מדי כדי שהנקודות החריגות תהיינה חריגות באופן מובהק מספיק).

יש גם דרכים חכמות יותר, מוזמנים לחפש למשל את מבחן Grubb. בקצרה, הוא מבצע בדיקת השערות בין ההשערה שאין בדאטא נקודות חריגות לבין ההשערה שקיימת נקודה חריגה אחת, מבצע GMM על הדאטא ובודק את מובהקות ההשערה שאין נקודות חריגות. אם ההשערה לא מובהקת מספיק, מוציאים את הנקודה החריגה ביותר ובוחנים את הדאטא ללא הנקודה. ניתן לבצע את המבחן הזה בצורה חמדנית ובהפעלה חוזרת של GMM (בזבזני), או להתייחס לבעיה כבעיית משתנים סמויים כאשר מניחים שכיחות יחסית  $\lambda$  של נקודות חריגות, לבצע אלגוריתם EM שמחשב את המשתנים הסמויים (אילו נקודות חריגות) בצורה לא חמדנית ומבצע GMM פעם אחת.

### אשכול:

גישה די פשטנית ודומה לגישה הסטטיסטית, אבל פחות חכמה: מבצעים אשכול לדאטא, מחשבים מדד איכות לכל נקודה בה (כמו silhouette score) ומגדירים את כל הנקודות עם score נמוך להיות אנומליות.

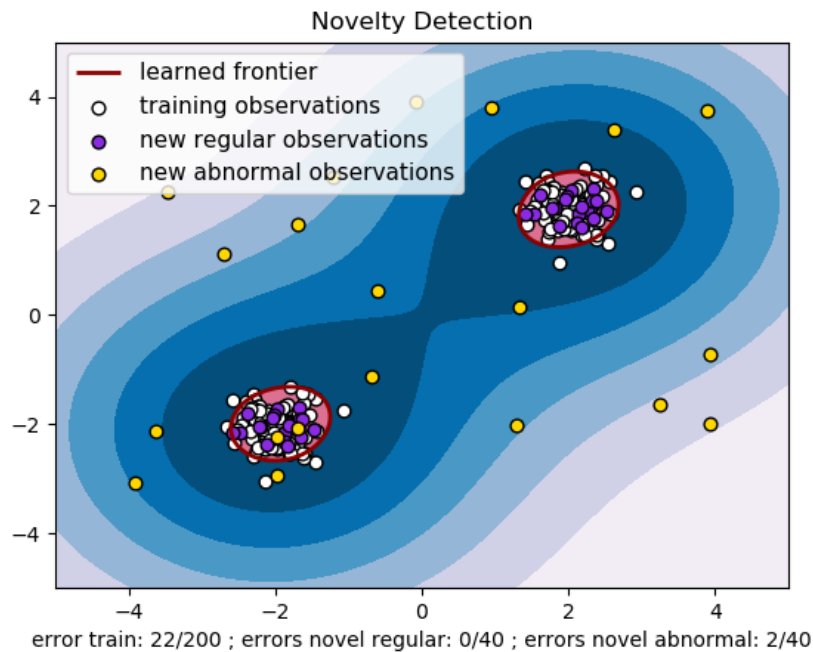
## :One class SVM

כמו שהשם מרמז, משתמשים כאן בפורמליזם של SVM לזיהוי חריגות. One class מתייחס לכך שאנחנו נפתור את הבעיה כך שנניח שהדאטא ה"שגרתי" שלנו מסודר כולו באשכול, והחריגות נמצאות בשוליים של האשכול הזה, מחוץ לאיזשהו גבול שנמצא. באופן כללי, נשתמש בהטלה כלשהי  $\phi$  על הפיצ'רים, כלומר Kernel, ובעיית האופטימיזציה תהיה:

$$\min_{w, \xi, \rho} \frac{1}{2} \|w\|^2 + \frac{1}{nv} \sum_i (\xi_i - \rho)$$
$$s. t. (w \cdot \phi(x_i)) \geq \rho - \xi_i, \quad \xi_i \geq 0$$

כאשר הפרמטרים  $\xi_i$  קובעים לנו כמה כל נקודה עונה על האילוץ שקבענו: אם  $\xi_i = 0$ , אז הנקודה בתוך ה-class שהגדרנו, ואם  $\xi_i > 0$  אז הנקודה מפרה את האילוץ ולכן חריגה. בנוסף, ההיפר – פרמטר  $v$  דומה בתפקידו ל- box constraint בבעיית SVM רגילה, ומשמש לקביעה של כמות החריגים שנרצה לקחת.

בתמונה מטה מוצג פתרון לבעיה Semi – supervised שפתרו באמצעות One class SVM כאשר הגרעין היה RBF.



נוח יותר להבין מה קורה כאן בעזרת מקרה פרטי:

$$\min_{R, z_0, \xi} R^2 + C \sum_i \xi_i$$

$$s. t. \quad \|x - z_0\|^2 \leq R^2 + \xi_i, \quad \xi_i \geq 0$$

בגרסה הזו למעשה בונים מעגל סביב המרכז  $z_0$  ומחפשים רדיוס  $R$  ופרמטרי התאמה  $\xi_i$  כך שמצד אחד המעגל יהיה קטן ככל האפשר ( $R$  מינימלי) ומצד שני יכיל כמה שיותר נקודות ( $\xi_i$  יהיו קטנים ככל האפשר).  $C$  הוא היפר – פרמטר שרירותי שהערך שניתן לו יקבע את כמות החריגות שתהיה – אם  $C$  גבוה אז נעדיף להכניס לעיגול כמה שיותר נקודות ולכן הרדיוס יהיה גבוה ותהיינה פחות חריגות, ואם  $C$  נמוך אז בדיוק להפך ותהיינה הרבה חריגות.

## גישה מבוססת צפיפות:

המחשבה שמאחורי גישה זו היא שנקודות חריגות תהיינה מצויות בסביבה דלילה במרחב, בבידוד מהשאר, בעוד שנקודות רגילות אמורות להימצא בקרבת נקודות רגילות אחרות ולכן בצפיפות.

צפיפות של גוף כלשהו מוגדרת להיות  $\rho = M/V$  לגוף הומוגני ו-  $\rho(\vec{r}) = \partial M / \partial V$  לגוף הטרוגני. במקרה שלנו, בו יש לנו דאטא בדיד במרחב לא אחיד ורציף, ולכן ישנן כל מיני דרכים לחשב את הצפיפות בצורה דומה לנוסחאות המופיעות מעלה (בפרט, המסה מתורגמת למספר הדגימות (בנפח), על נפחים שונים ולפי מטריקות שונות המציינות את המרחק שבין הדגימה לדגימות אחרות באיזשהו מרחב (שוב, Kernels משחקים פה תפקיד)).

אחת השיטות מהסוג הזה נקראת LOF – local outlier factor, והיא מתאימה לכל נקודה מדד בהתבסס על הצפיפות של הדאטא סביב הנקודה. ככל שמדד זה גבוה יותר עבור הנקודה, כך היא תיחשב חריגה יותר.

שיטה נפוצה אחרת היא Isolation Forest, שבה בונים יער, המורכב מעצי החלטה אקראיים לחלוטין. ככל שנקודה מסוימת מתגלה מתוך אוסף העצים בתור קלה להפרדה, אז היא נחשבת כסבירה יותר להיות אנומלית ומקבלת score גבוה יותר מהמודל.

