למידת מכונה הרצאה 1

הקדמה

מה זה למידת מכונה?

- הסקת מסקנות מתוך נתונים קיימים
- חיזוי של תכונות לא ידועות
- בדרך כלל מתוך דגימה

לדוגמה: בהינתן גובה של אדם, האם נוכל לנחש את המשקל? או בהינתן טמפרטורת גוף ודופק, האם נוכל לנחש אם זה זכר או נקבה?

נתבונן במערכי נתונים של Hope College: נראה שיש קורלציה בין הדברים. אז גם אם אנחנו כרגע לא יכולים לתת נוסחה מדויקת לחיזוי, נראה הגיוני שזה אפשרי.

עוד דוגמאות – זיהוי ספאם בג'ימייל. יש לגוגל מערך נתונים ענק של מיילים שמתוייגים כספאם. האם הוא יוכל לזהות מייל חדש בתור ספאם? בפועל אנחנו רואים שכן. באחוזי הצלחה גבוהים.

נשים לב שזו בעיה דומה לשאלת המגדר – הגדרה בינארית, כן או לא ספאם.

זיהוי פנים או חפצים בתמונה, לדוגמה לצורך רכב אוטונומי.

בעיה שנפתרה ע"י למידת מכונה – שאלת הEKG. יש צורה שדופק תקין נראה, והתקף לב נראה אחרת. רופא מומחה יכול מתוך המידע להסיק האם זה התקף לב או לא.

אנחנו יכולים לראות בעין שיש הבדל בין הדופק התקין והתקף לב (בדוגמה). אבל צריך הרבה ידע וניסיון כדי להסתכל על EKG חדש ולהגיד האם זה התקף לב או לא. אולי ננסה לבנות מודל חישובי שעושה את זה? נמצא מומחה שיודע להסביר איך נראה EKG תקין ואיך נראה התקף, ונבנה אלגוריתם. יש כמה בעיות:

האם יש מומחים ברמה מספקת? קודם כל, האם מומחים בכלל קיימים. ואם כן, האם יש לי גישה אליהם, והאם יש להם את הרצון והיכולת לעבוד על זה. וגם אם יש, האם הם יודעים להסביר לי איך לעשות את זה? כלומר, האם יש מישהו שיודע לתת רשימת "כללים" לדבר הזה? גם אם מישהו יודע לתת סיווג, לא תמיד יודעים להסביר **למה.** (דוגמה מתוך הספר "ממבט ראשון", רושם ראשוני, אינסטינקט). אז רופא מומחה שיודע לזהות התקף לב, לא בהכרח יידע לתת רשימה של חוקים לאלגוריתם.

זה לא "חסון" – המודל שאני אולי אצליח לייצר, מיועד למקרה מאד ספציפי. ולא עוזר לי לבעיות אחרות (או אם הבעיה משתנה קצת). כל פעם אני מתחיל מאפס.

המטרה שלנו היא לגרום למחשב ללמוד את זה לבד, באופן אוטומטי. זה פותר את הבעיות.

למידת מכונה: לייצר מודל שפותר בעיה מסוימת, בלי שנצטרך לכתוב את האלגוריתם במפורש. כאשר יש מידע שאפשר להסיק ממנו דברים.

עוד דוגמה – תרגום. הניסיון לייצר כלי תרגום אוטומטי. הגישה הישנה – Natural Language Processing. להגדיר למחשב כללים ומבנה של שפה. מאד מסובך ולא ממש עובד (יש כפל משמעות, ניואנסים, סלנג, השפעות תרבותיות...)

מה שעובד יותר טוב – לתת למחשב דאטאסט גדול של טקסטים מתורגמים. (דוגמה – אבן הרוזטה).

דוגמה אחרונה – נטפליקס. מתוך הדירוג של משתמש על סרטים, לנחש איזה סרטים הוא יאהב.

תחרות מי יצליח לפתח אלגוריתם. השיטה לבדוק את האלגוריתמים – לפרסם תת מטריצה של המידע, לתת לכל אלגוריתם לנסות לשחזר את שאר המטריצה, ולראות את אחוזי ההצלחה.

מה שעבד בסוף – Principal Component Analysis. "לדחוס" את המטריצה למימד קטן יותר (פחות עמודות), כך שהמידע החשוב נשמר. "מסיר רעש" מהמערכת. (לקראת סוף הסמסטר ניגע בפרקטיקה של התהליך). מאבדים קצת מידע אבל בגלל שהמידע החשוב נשמר, דווקא יותר קל להוציא מסקנות ממנו.

עד כאן הקדמה. עכשיו ניקח צעד אחורה. לא מובן מאליו שאם יש מאגר מתויג, נוכל להסיר מסקנות על נקודות לא מתויגות. נוכיח את זה באופן מתמטי.

מטבע מוטה

בהינתן מטבע, מה ההטיה שלו? (במטבע הוגן, חצי חצי). נטיל את המטבע n פעמים. כל הטלה היא משתנה מקרי שמתפלג **ברנולי** ($Ber(0,1)$) עם: $\mathbb{P}(1) = p, \mathbb{P}(0) = 1 - p$. אם יש n הטלות, אז הערך הממוצע של ההטלות יהיה: $X = (\sum_{i=1}^{n} x_i)/n$.

אנחנו ננחש שההטיה של המטבע היא הממוצע. ניקח מספר מסויים של הטלות ולפי זה ננחש – לפי הממוצע **האמפירי** (מה שראינו בפועל). אנחנו ננחש שהממוצע האמפירי הוא ההטיה בפועל.

נרצה שבהסתברות גבוהה, הממוצע האמפירי יהיה קרוב לסטייה האמיתית. נרצה בעצם להוכיח ש: ההסתברות שהממוצע האמפירי **שונה מאוד** מהסטייה האמיתית, היא נמוכה. נשתמש באי שוויון הופדינג (הסתברות 1, זוכרים?).

> תזכורת: אי"ש הופדינג. יהי B מ"מ המתפלג ברנולי על $\{0,1\}$ עם פרמטר p. נבצע n ניסויים בת"ל, ויהי X הממוצע האמפירי שלהם. אזי, לכל $0 < \epsilon < 1$, $\mathbb{P}(|X - p| > \epsilon) < 2e^{-2n\epsilon^2}$.

עבור כל $0 < \delta, \epsilon < 1$, נרצה לטעון ש: בהסתברות לפחות $1 - \delta$, הסטייה של המטבע היא $X \pm \epsilon$. מה גודל המדגם הנדרש?

לפי אי"ש הופדינג, ההסתברות לכישלון היא לכל היותר $2e^{-2n\epsilon^2}$, אז אנחנו צריכים ש: $2e^{-2n\epsilon^2} \leq \delta$. כלומר $n \geq \frac{\ln\frac{2}{\delta}}{2\epsilon^2}$.

$$2e^{-2n\epsilon^2} \leq \delta \Longrightarrow e^{-2n\epsilon^2} \leq \delta/2 \Longrightarrow \ln e^{-2n\epsilon^2} \leq \ln \delta/2 \Longrightarrow -2n\epsilon^2 \leq \ln \delta/2 \Longrightarrow n \geq \frac{\ln \delta/2}{-2\epsilon^2} \Longrightarrow n \geq \frac{\ln 2/\delta}{2\epsilon^2}$$

זה נותן לנו נוסחה לגודל של n, על מנת להקטין את δ, ϵ. נשים לב שזה לא סימטרי – הרבה יותר קל לקבל הסתברות טובה יותר (להקטין את δ) מאשר לשפר את ההערכה של הסטייה (להקטין את ϵ). בגלל ש- δ בתוך לן. שיפור ההערכה הרבה יותר "יקר" מאשר שיפור ההסתברות לניחוש נכון.

The memorizer

עובד בבית קפה שבו צריך לנחש האם כל לקוח רוצה תה (0) או קפה (1). יש שבוע שבו מותר לשאול (המדגם המתויג) ואח"כ צריך לנחש. מקבל תשלום רק אם צדק.

איזה אלגוריתם יכול לעבוד?

רעיון ראשון: בשבוע הראשון, לרשום את ההעדפות של כל לקוח. אח"כ, אם אותו לקוח נכנס שוב – יודעים מה לתת לו. וכל לקוח חדש מקבל משהו אקראי.

הבעיה בזה היא שאנחנו מתאימים **יותר מדי** בין הדאטא למדגם (*overfitting*). למדנו את המדגם, ולא את העיקרון מאחורי המדגם. אחוזי הצלחה – על המדגם 100%, בשאר העולם 50%.

רעיון שני: שמנו לב שמתוך המדגם, 95% מהגברים הזמינו תה, ו-95% מהנשים הזמינו קפה. אז אם הלקוח גבר, ניתן תה. אם זו אישה, ניתן קפה. הצלחה: 95% על המדגם, ויש סיכוי טוב שגם בשאר העולם 95%.

זה מביא אותנו לעיקרון חשוב:

בהינתן מדגם אקראי. אם חוק פשוט נותן הצלחה על המדגם – אז בהסתברות גבוהה ($1 - \delta$) לחוק יש הצלחה דומה ($\pm\epsilon$) בעולם.

3 נקודות חשובות:

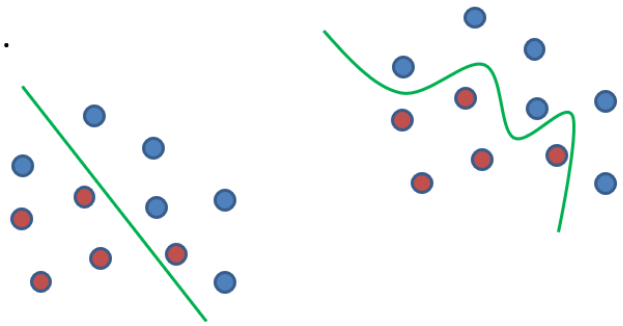
מדגם אקראי – כל זה עובד תחת ההנחה שהמדגם שלנו מתפלג כמו כל הדאטא. אי אפשר ללמוד ממטבע אחד על מטבע אחר.

כלל פשוט – שאין התאמת יתר (*overfitting*). כלומר כשהכלל שלנו ספציפי מדי ומדויק מאד לגבי הקבוצה המתויגת, אבל זה לא נותן לנו להבין את הגורמים מאחורי הדברים. נגיד בדוגמה של בית הקפה, אם ניתן חשיבות לצבע העיניים של הלקוחות. באיזשהו מובן, יש לנו אינטואיציה למה נחשב חוק פשוט ומה לא. אנחנו נפרמל את זה מתמטית. ככל שהמודל יכול למדל יותר דברים, הוא יותר מסובך ופחות יעיל. חוק פשוט יותר הוא "חלש" יותר אבל דווקא נותן לנו יכולת. אם מצאנו חוק פשוט שעובד עם הדאטא שלנו, זה אומר ששמנו את האצבע על העיקרון מאחורי הדאטא (בהנחה שיש).

הטיה מול שונות – (*bias – variance*). מה שראינו, האיזון בין דלתא (שונות) לאפסילון (הטיה).

התאמת יתר – *overfitting*

חוקים מסובכים שמתארים את המדגם בצורה מדויקת. כמו לזכור כל לקוח. עובד רק על המדגם ולא על שאר העולם.



זו גם אינטואיציה טובה למה נחשב כלל פשוט – אם זה "קו ישר" ולא מעוקל מדי. גם אם הוא מפספס דברים במדגם, בהסתברות גבוהה זה יעבוד בשאר העולם.

בעיות ומטרות:

סיווג – *classification*. הבעיה של להגדיר עבור נקודת דאטא חדשה, סיווג בינארי לאיזה קבוצה היא שייכת.

ריבוי קבוצות – *multiclass*. אין סיווג בינארי. לדוגמה שאלה של סגנון של אתר אינטרנט. הפתרון במקרים רבים הוא לקחת כלי של סיווג בינארי ולהכליל אותו למצב של ריבוי.

רגרסיה – מקרה של מספר רציף. בהינתן גובה לנחש משקל.

דירוג – *ranking*. מה שקורה בגוגל. דרך להחליט מה יותר טוב ממה.

מקבוץ – *clustering*. למידה "ללא השגחה" – *unsupervised learning*. להחליט על קבוצות של נקודות מתוך הדאטא, בלי תיוג. לדוגמה בפרסומות – להחליט על קבוצות של קהל יעד.

"הורדת מימד" – *dimensionality reduction, PCA*. המטריצה נשארת באותם המימדים, אבל בייצוג פשוט יותר.

חזרה קצרה על הסתברות 1 – התפלגויות, הסתברות מותנה, נוסחת בייס, חסם איחוד וחיתוך, אי תלות. משתנה מקרי – תוחלת, שונות, ליניאריות התוחלת, הגדרת אי תלות לפי תוחלת. אי-שוויונות.