מבוא למערכות לומדות - הרצאה 4 - מהכללה לאופטימיזציה

22 ביוני 2015

בשלושת ההרצאות הראשונות הגדרנו את מודל PAC ללמידה, ופיתחנו את התורת ההכללה, שמאפשרת לנו להבין כמה דוגמאות עלינו לראות על מנת ללמוד. על מנת להשתמש בפועל בלמידה חישובית עלינו לטפל בשתי נקודות נוספות בהן תורת ההכללה לא מטפלת:

- כיצד לבחור מחלקת היפותזות המתאימה לבעיה הקונקרטית אותה אנו רוצים לתקוף?
- אחרי שכבר בחרנו מחלקת היפותזות, כיצד נוכל למצוא בה היפותזה טובה? ניזכר שהאלגוריתם שעומד מאחורי תורת ההכללה הוא ה־ERM. מימוש נאיבי שלו, עבור רוב המחלקות הרלוונטיות, לוקח זמן אקספוננציאלי.

בחלק הראשון של השיעור נסכם את תורת ההכללה ונאמר כמה מילים על הנקודה הראשונה - כיצד לבחור מחלקת היפותזות. בחלק השני של השיעור, ולמעשה בחמשת השיעורים הקרובים, נטפל בנקודה השניה - ננסה להבין כיצד נוכל, בהינתן מחלקת היפותזות, למצוא בה היפותזה טובה.

1 סיכום ביניים - תורת ההכללה

1.1 אז מה בעצם למדנו עד כה?

ניזכר שהמטרה הבסיסית בלמידה חישובית היא ללמוד מיפוי

$$h^* : X \to Y$$

על סמך מדגם

$$S = \{(x_1, y_1), \ldots, (x_m, y_m)\} \subset X \times Y$$

במודל התיאורטי שהצגנו, הנחנו שקיימת התפלגות $\mathcal{D}$ שממנה נדגמות הדוגמאות. בהתאם לכך, הגדרנו את ההפסד של היפותזה $h : X \to Y$ ע"י

$$L_{\mathcal{D}}(h) = E_{(x,y)\sim\mathcal{D}} l(h(x), y)$$

כאשר $l : Y \times Y \to \mathbb{R}_+$ מודדת את המרחק בין זוג פלטים. האבחנה הראשונה שעשינו היא ש־"אין ארוחות חינם". כלומר, לא קיים אלגוריתם המסוגל ללמוד כל מיפוי כאשר כמות הדוגמאות מוגבלת. במילים אחרות, אנחנו צריכים איזשהו **ידע מוקדם** על הפונקציה h^* (או על ההתפלגות $\mathcal{D}$) על מנת להיות מסוגלים ללמוד.

הדרך שבה ניתן לבטא ידע מוקדם במודל PAC היא הצבעה על **מחלקת היפותזות** (כלומר, אוסף $\mathcal{H} \subset Y^X$) בה אנו יודעים (או, בד"כ, מניחים) שקיימת היפותזה טובה. תורת ההכללה שפיתחנו הראתה שכאשר $\mathcal{H}$ "קטנה" בהשוואה לכמות הדוגמאות שיש בידנו, השגיאה האמפירית ($L_S(h)$) של **כל ההיפותזות ב־$\mathcal{H}$** קרובה לשגיאה האמיתית ($L_{\mathcal{D}}(h)$). לכן, כל אלגוריתם המחזיר היפותזה ב־$\mathcal{H}$ עם שגיאה אמפירית טובה (בפרט, ERM), יחזיר בעצם היפותזה עם שגיאה אמיתית טובה. לכן, בהינתן אלגוריתם כזה, אם הידע המוקדם שלנו נכון ובאמת קיימת ב־$\mathcal{H}$ היפותזה טובה, נוכל ללמוד.

מהתיאוריה הנ"ל עולה השיטה הבאה ללמוד:

1. קבעו מחלקת היפותזות $\mathcal{H}$ בה אתם מעריכים ש־
   (א) יש היפותזה טובה.
   (ב) היא מספיק קטנה ביחס לכמות הנתונים שיש ברשותכם.
2. **(כלל ה־ERM)** מצאו ב־$\mathcal{H}$ היפותזה h עם שגיאה אמפירית מינימאלית ביחס לנתונים שאספתם.

שגיאת קירוב ושגיאת הכללה - יחסי גומלין

על מנת להעריך את ההשפעה של השלבים השונים, ואת השיקולים שיש לקחת בחשבון כאשר משתמשים בה, נוח לפרק את $L_{\mathcal{D}}(h)$ לשני רכיבים:

$$L_{\mathcal{D}}(h) = \underbrace{L_{\mathcal{D}}(h) - L_{\mathcal{D}}(\mathcal{H})}_{\text{Estimation Error}} + \underbrace{L_{\mathcal{D}}(\mathcal{H})}_{\text{Approximation Error}}$$

שגיאת ההכללה: שגיאת ההכללה היא ההפרש בין השגיאה (האמיתית) של ההיפותזה עם השגיאה האמפירית הטובה ביותר, לבין השגיאה (האמיתית) של ההיפותזה הטובה במחלקה. הרכיב הזה יהיה קטן יותר ככל שיהיו לנו יותר דוגמאות. עבור קלסיפיקציה בינארית, תורת ההכללה שפיתחנו מאפשרת לנו לחסום את שגיאת ההכללה באמצעות $\mathrm{VC}(\mathcal{H})$. נעיר שיש תורות דומות עבור בעיות מעבר לקלסיפיקציה בינארית (למשל, בתרגיל טיפלנו בקלסיפיקציה רב מחלקתית).

שגיאת הקירוב: שגיאת הקירוב היא השגיאה של ההיפותזה הטובה במחלקה. על מנת להקטין אותה, ניתן להקטין לנקוט בשתי דרכים:

- **ידע מוקדם.** אם אנו, או מומחה בבעיה הספיציפית שעל הפרק, מצליח להצביע על מחלקה בה באמת יש היפותזה טובה, שגיאת הקירוב תהיה קטנה.
- **שימוש במחלקה עשירה יותר.** ככל שהמחלקה אותה נבחר תכיל יותר היפותזות, שגיאת הקירוב תהיה קטנה יותר.

נעיר שקיימים יחסי גומלין בין שגיאת ההכללה ושגיאת הקירוב: שימוש בדרך הראשונה ישחק רק לטובתנו - ככל שנצליח להצביע על מחלקה המתאימה יותר לבעיה שאותה אנו רוצים לפתור, נקטין את שגיאת הקירוב. לכן, לפני שמשתמשים באלגוריתמי למידה, כדאי להבין טוב את הבעיה שעומדת לפנינו, ולחשוב באיזו משפחה של פונקציות נוכל למצוא היפותזה הקרובה להיפותזה אותה אנו רוצים ללמוד. אנו נדבר בקורס מעט על דרכים לעשות זאת. עם זאת, טיפול בשלב הזה שייך בעיקר לתחום אליו הבעיה הספציפית אותה אנו רוצים ללמוד שייכת (ראייה \ זיהוי דיבור\ הבנת שפה \ ביולוגיה \ פיננסים \ ...).

לעומת זאת, שימוש במחלקה עשירה יותר אמנם יקטין את שגיאת הקירוב - אך הוא עשוי לפגוע בשגיאה ההכללה! ככל שאנו עובדים עם מחלקה גדולה יותר, סביר ששגיאת ההכללה תהיה גדולה יותר. לכן, יש לנו כאן משחק עדין - עלינו למצוא מחלקה שתהיה מספיק גדולה ותכיל היפותזה טובה, ומצד שני לא תהיה גדולה מידי ותגרום לשגיאת הכללה. בהמשך, נראה שיטה המאפשר למצוא את האיזון הנכון בין הדרישות הסותרות הנ"ל.

1.2 מה הלאה? מהכללה לאופטימיזציה

על מנת להשתמש בפועל בשיטה שהצגנו עלינו לעבוד עם מחלקת היפותזות כך שבהינתן מדגם ניתן יהיה ניתן למצוא בה (בזמן סביר) היפותזה עם שגיאה אמפירית קטנה. לצורך הדיון, בהינתן בעיית למידה $(X, Y, \mathcal{H}, l)$, נסמן ב־$\mathrm{OPT}(\mathcal{H})$ את בעיית האופטימיזציה הבאה:

בהינתן מדגם

$$S = \{(x_1, y_1), \ldots, (x_m, y_m)\} \subset X \times Y$$

מצא $h \in \mathcal{H}$ הממזערת את

$$L_S(h) = \frac{1}{m} \sum_{i=1}^{m} l(h(x_i), y_i)$$

בשבועות הקרובים נתרכז בעיקר בדרכים לתקוף את הבעיה הנ"ל. אנו נתחיל (השבוע ושבוע הבא) בלהציג מחלקות $\mathcal{H}$ בהן ניתן לפתור את $\mathrm{OPT}(\mathcal{H})$. למרבה הצער, אוסף המחלקות בהן ניתן לפתור את $\mathrm{OPT}(\mathcal{H})$ איננו עשיר דיו. לכן, בשבועות שאח"כ נלמד על דרכים לעבוד עם מחלקות בהן $\mathrm{OPT}(\mathcal{H})$ הינה בעיה קשה חישובית.

1.3 כלים טכניים ומתמטיים שראינו

נזכיר בקצרה את המושגים והטיעונים המרכזיים שראינו במהלך פיתוח התיאוריה, וכמו כן את מבנה ההוכחה של המשפט היסודי.

- **חסם הופדינג והשיערוך השגיאה ע"י השגיאה האמפירית** - אנו השתמשנו על ימין ועל שמאל בחסם הופדינג על מנת להראות שעבור h בודדת, $L_S(h) \approx L_{\mathcal{D}}(h)$ כאשר המדגם גדול מספיק.
- **חסם האיחוד** - עבור מחלקות סופיות, השתמשנו בחסם האיחוד על מנת להסיק שכאשר המדגם גדול מספיק $L_S(h) \approx L_{\mathcal{D}}(h)$ **לכל** $h \in \mathcal{H}$. כמו כן, עשינו שימוש בחסם האיחוד גם בהוכחת למת המדגם הכפול.

- **למידה באמצעות התכנסות במידה שווה** - אחת האבחנות הבסיסית שעשינו היא שכאשר המדגם גדול דיו כך שלכל $h \in \mathcal{H}$ מתקיים $L_S(h) \approx L_{\mathcal{D}}(h)$, אלגוריתמי ERM יחזירו היפותזה טובה.

- **פונקציית הגידול ולמת המדגם הכפול** - על מנת לקבל את החסם העליון במשפט היסודי השתמשנו בלמת המדגם הכפול. למת המדגם הכפול איפשרה לנו "להחליף" את ההתפלגות במדגם גדול (בגודל כפול מהמדגם המקורי), וכפועל יוצא, "להחליף" את $\mathcal{H}$ בצמצום שלה למדגם הכפול. הקטנה זו צמצמה מאד את מספר ההיפותזות שעלינו להביא בחשבון ואיפשרה לנו להסיק (יחד עם למת סאור שלח) את החסם העליון.

- **למת סאור שלח** - למת סאור־שלח איפשרה לנו לחסום את פונקציית הגידול.

- **חישוב מימד** VC - למדנו מספר שיטות לחשב את מימד VC עבור מחלקות קונקרטיות. השיטות הללו מאפשרות לנו להפעיל את התורה על מחלקות שנפגוש בפועל.

2 קמירות על קצה המזלג

נקבע בעיית למידה $(X, Y, \mathcal{H}, l)$. כאמור, על מנת לממש את כלל ה־ERM עלינו לפתור את בעיית האופטימיזציה $\mathrm{OPT}(\mathcal{H})$. באופן כללי, (NP־)קשה לפתור בעיות אופטימיזציה. עם זאת, קיימות מספר משפחות של בעיות אופטימיזציה שניתן לפתור ביעילות. אחת מהמשפחות הללו היא **בעיות אופטימיזציה קמורות** (ליתר דיוק, תת משפחה גדולה של המשפחה הנ"ל). כלומר, בעיות בהן אנו מנסים למזער פונקציה קמורה. אנו נלמד כיצד ניתן לפתור בעיות אופטימיזציה קמורות. כמו כן, נצביע על בעיות למידה שניתן לעשות להן רדוקציה לבעיות אופטימיזציה קמורות. כלומר, על בעיות למידה שעבורן מימוש כלל ה־ERM שקול לפתרון בעיית אופטימיזציה קמורה. בהמשך, נראה איך ניתן להשתמש ביכולת לפתור בעיות למידה קמורות, על מנת לתקוף בעיות למידה שאינן קמורות.

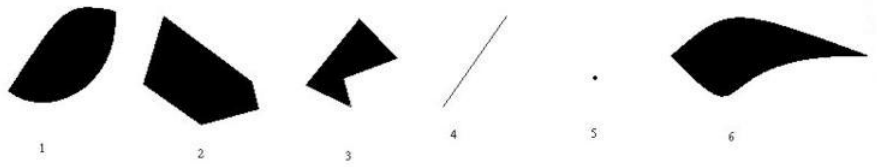
על מנת להוציא את התכנית הזו לפועל, נזדקק להבין מעט מושגים בסיסיים בקמירות.

2.1 קבוצות קמורות

הגדרה 2.1 קבוצה $C \subset \mathbb{R}^n$ תיקרא **קמורה** אם לכל $x, y \in C$ ולכל $0 \leq \lambda \leq 1$, מתקיים $\lambda x + (1-\lambda)y \in C$.

כלומר, קבוצה היא קמורה אם לכל שתי נקודות בקבוצה, הישר המחבר בין x ל־y מוכל כולו בקבוצה.

דוגמאות:



בציור הנ"ל, קבוצות 3 ו־6 **אינן** קמורות, ויתר הקבוצות כן קמורות. דוגמא נוספת לקבוצה קמורה שתשמש אותנו במהלך הקורס היא כדור:

טענה 2.2 עבור $r > 0$ הקבוצה $B = \{x \in \mathbb{R}^n : \|x\| \leq r\}$ הינה קמורה.

הוכחה: אם $x, y \in B$ ו־$\lambda \in [0,1]$ אז

$$\|\lambda x + (1-\lambda)y\| \leq |\lambda| \cdot \|x\| + |1-\lambda| \cdot \|y\| \leq |\lambda| \cdot r + |1-\lambda| \cdot r = r$$

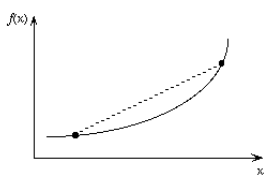
לכן $\lambda x + (1-\lambda)y \in B$ ■

2.2 פונקציות קמורות

הגדרה 2.3 תהא $C \subset \mathbb{R}^n$ קבוצה קמורה. פונקציה $f : C \to \mathbb{R}$ תיקרא **קמורה** אם מתקיים

$$\forall x, y \in C, 0 \leq \lambda \leq 1, \quad f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda) f(y)$$

במילים, עבור זוג נקודות $x, y \in C$, ערך הפונקציה בממוצע (משוקלל) של x ו־y קטן או שווה שלממוצע (המשוקלל) של ערכי הפונקציה ב־x ו־y. מבחינה גיאומטרית, ניתן לאפיין פונקציות קמורות באחד מהאופנים הבאים:

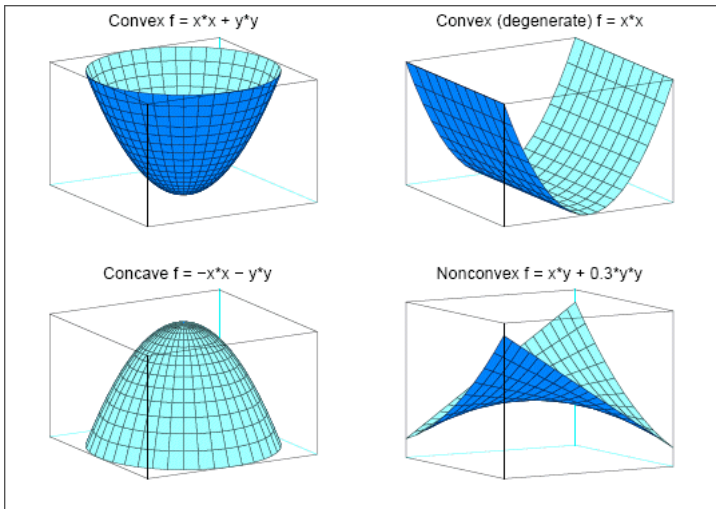


- המיתר מ־$(x, f(x))$ ל־$(y, f(y))$ נמצא מעל גרף הפונקציה:
- **האפיגרף** של f, כלומר, הקבוצה

$$\mathrm{epi}(f) := \{(x,y) \mid x \in C,\ y \geq f(x)\} \subset \mathbb{R}^{n+1}$$

הינה קמורה.

בתמונה הבאה, הפונקציות בשורה העליונה הינן קמורות, בעוד היתר אינן.



ארגז כלים קטן לזיהוי ובנייה של פונקציות קמורות

על מנת להשתמש בצורה אפקטיבית באופטימיזציה קמורה, עלינו לדעת לזהות ולבנות פונקציות קמורות. נציג מספר כלים בסיסיים שיאפשרו לנו לעשות זאת.

פונקציות קמורות במשתנה אחד. עבור פונקציות במשתנה יחיד קיים קריטריון פשוט למדי המאפשר לקבוע האם פונקציה נתונה $f : (a, b) \to \mathbb{R}$ הינה קמורה. מתברר שעבור f גזירה, מתקיים ש־f קמורה אמ"מ f' הינה מונטונית לא יורדת. בפרט, אם f גזירה פעמיים, אז f קמורה אמ"מ $f''(x) \geq 0$ לכל $x \in (a, b)$. בעזרת הקריטריון הנ"ל לא קשה להסיק שלמשל $x, |x|^p, e^x, -\log(x)$ הינן קמורות.

עוד עובדות שיעזרו לנו לבנות ולזהות פונקציות קמורות מסוכמות בלמה הבאה

למה 2.4

- אם $f : \mathbb{R} \to \mathbb{R}$ הינה קמורה ו־$F : \mathbb{R}^n \to \mathbb{R}$ נתונה ע"י

$$F(x) = f\left(a_1 x_1 + \ldots + a_n x_n\right)$$

  עבור קבועים $a_1, \ldots, a_n$ אז גם F קמורה.

- סכום של פוקציות קמורות הינו קמור, כמו גם מכפלה של פונקציה קמורה במספר חיובי.

- אם $f_1, \ldots, f_r : C \to \mathbb{R}$ הינה קמורות אז כך גם

$$f(x) := \max_{1 \leq i \leq r} f_i(x)$$

הוכחת הלמה אינה קשה ומושארת כתרגיל.

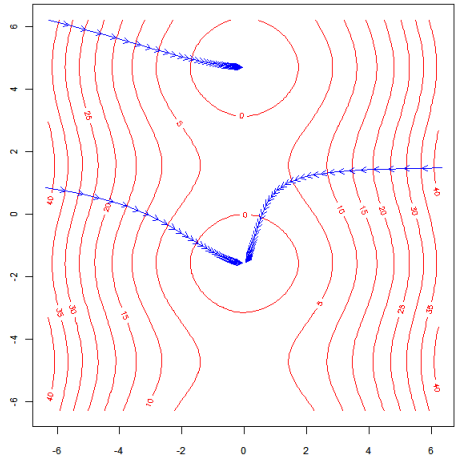
3 גרדיינט דסנט ואופטימיזציה קמורה

שאלה בסיסית באופטימיזציה היא כיצד ניתן למצוא מינימום של פונקציות $f : \mathbb{R}^n \to \mathbb{R}$. השאלה הנ"ל חולשת על מספר תחומים במדעי המחשב ובודאי לא נטפל בה כאן באופן מלא. עם זאת, נציג אלגוריתם פופולארי וטבעי לבעיה (ליתר דיוק, משפחה של אלגוריתמים). האלגוריתם נקרא **גרדייט דסנט (Gradient Descent)**. הרעיון מאחוריו פשוט: נתחיל מאיזשהי נקודה $x_0 \in \mathbb{R}^n$ (למשל, $x_0 = 0$) ונתקדם בכל שלב בכיוון שיקטין את הפונקציה מעט.

ניזכר שאם f גזירה, אז אם מתחילים מ־$x \in \mathbb{R}^n$, הכיוון בו f קטנה הכי הרבה הוא $(-\nabla f(x))$. לכן, טבעי להביט באלגוריתמים המתקדם בכל שלב לפי הכלל

$$x_{t+1} = x_t - \eta_t \nabla f(x_t) \tag{1}$$

כאשר $\eta_1, \eta_2, \ldots$ היא סדרה של מספרים חיוביים. אלגוריתם העונה לתיאור הנ"ל נקראת **גרדייט דסנט**. נשים לב שריצת האלגוריתם, כלומר, סדרת הנקודות x_t נקבעת ביחידות בהינתן x_0. בתמונה הבאה מתוארות שלוש ריצות של האלגוריתם (המתאימות לשלוש נקודות התחלה x_0 שונות) כאשר $f(x, y) = x^2 + 3\sin(y)$. באופן יותר מפורט, הקווים האדומים הם קווי הגובה של f והמסלולים הכחולים מתארים שלוש ריצות שונות של האלגוריתם. קונקרטית, מתוארות שלוש סדרות x_t המקיימות את תנאי (1) עם $\eta_t \equiv 0.05$.



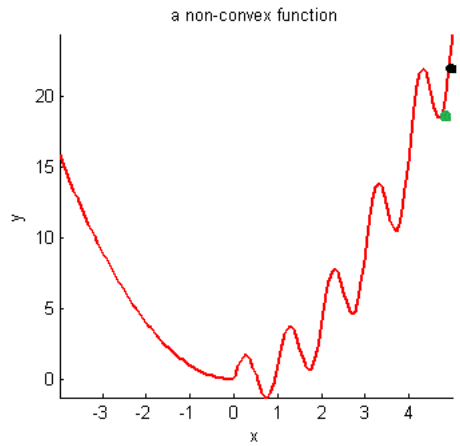
מספר שאלות בסיסיות עולות בהקשר הנ"ל: כיצד לבחור את הסדרה $\{\eta_t\}_{t=1}^{\infty}$? בהינתן שקבענו את הסדרה, עבור אילו פונקציות האלגוריתם מתכנס לפתרון? כלומר, עבור אילו פונקציות מתקיים $\lim_{t\to\infty} f(x_t) = \min_{x\in\mathbb{R}^n} f(x)$? ואם האלגוריתם מתכנס לפתרון, כמה מהר זה קורה? כלומר, כמה גדול צריך להיות t על מנת ש־$f(x_t)$ יהיה קרוב ל־$\min_{x\in\mathbb{R}^n} f(x)$? השאלות הנ"ל ושאלות אחרות מהוות בסיס לתחום הנקרא **"אופטימיזציה"**.

לא נטפל בשאלות הללו באופן מלא, אבל נראה שעבור משפחה עשירה של פונקציות קמורות ניתן לבחור סדרה η_t כך שמובטח לנו שהאלגוריתם יתכנס. רוב העבודה תיעשה בתירגול ובתרגיל. עם זאת, נצביע על שתי תכונות המייחדות פונקציות קמורות שאולי מבהירות מדוע נצפה ש־GD יתכנס עבור פונקציות קמורות.

3.1 תכונות של פונקציות קמורות

כל מינימום מקומי הוא גלובאלי

נביט בפונקציה הבאה:



אם נשתמש ב־GD ונתחיל בנקודה שחורה, ככל הנראה נתכנס לנקודה הירוקה. הנקודה הירוקה הינה מינימום מקומי אך רחוקה מלהיות מינימום גלובאלי! תכונה שימושית של פונקציות קמורות היא שכל מינימום מקומים הוא גלובאלי

למה 3.1 תהא $f : C \to \mathbb{R}$ פונקציה קמורה. ונניח ש־$x_0 \in C$ הינה מינימום מקומי (כלומר, קיים $r > 0$ כך שלכל $x \in C$ המקיים $\|x - x_0\| < r$ מתקיים $f(x) \geq f(x_0)$) אזי x_0 הינה מינימום גלובאלי (כלומר, $f(x_0) = \min_{x \in C} f(x)$).

הוכחת הלמה מושארת כתרגיל.

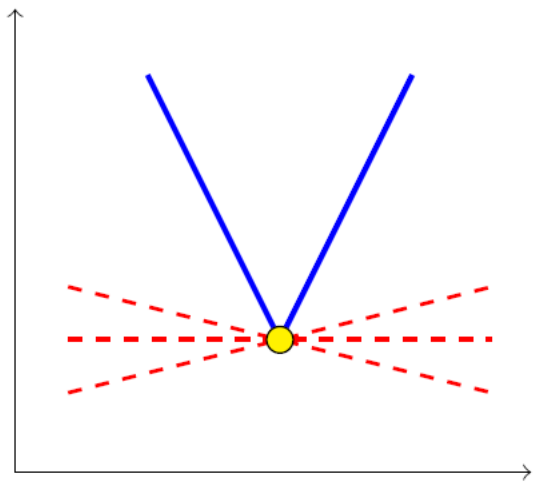
קיום סאב גרדיינטים

על מנת להשתמש בגרדיינט דסנט, אנו צריכים שלפונקציה יהיה גרדיינט בכל נקודה (במילים אחרות, שתהיה גזירה). זה אמנם לא נכון שכל פונקציה קמורה היא גזירה (למשל, $|x|$ היא פונקציה קמורה שאיננה גזירה). עם זאת, כל פונקציה קמורה היא "כמעט גזירה" במובן מסויים, שיאפשר לנו להפעיל אנלוג של גרדיינט־דסנט (הנקרא **סאב־גרדיינט־דסנט**) גם על פונקציות קמורות שאינן גזירות. קונקרטית, מתקיים שאם $f : C \to \mathbb{R}$ הינה קמורה, אז לכל $x_0 \in C$ קיים וקטור (לאו דווקא יחיד!) המסומן ב־$\nabla f(x_0)$ ונקרא **סאב־גרדיאנט** המקיים

$$\forall x \in C, \quad f(x) \geq f(x_0) + \langle x - x_0, \nabla f(x_0) \rangle$$

מבחינה גיאומטרית המישור שהוא הגרף של $x \mapsto f(x_0) + \langle x - x_0, \nabla f(x_0) \rangle$ נמצא מתחת ומשיק לגרף של f. למשל בתמונה הבאה מצויירים שלושה מישורים (במקרה הזה, קווים)

המתאימים לשלושה סאב־גרדיינטים שונים של הפונקציה הקמורה $f(x) = |x|$ בנקודה $x_0 = 0$.



כפי שתראו בתירגול, סאב גרדיינטים יכולים להוות תחליף לגרדיינט כאשר אנו רוצים להפעיל את האלגוריתם.

4 בעיות למידה קמורות

נקבע בעיית למידה $(X, Y, \mathcal{H}, l)$. כאמור, אנו נרצה להשתמש באופטימיזציה קמורה על מנת לפתור את בעיית האופטימיזציה $\mathrm{OPT}(\mathcal{H})$. לפני שנגדיר בצורה פורמאלית מתי ניתן לעשות זאת, נביט בדוגמא קונקרטית - **בבעיית הרגרסייה** הבאה. נניח ש־

$$X = \mathbb{R}^n,\ Y = \mathbb{R},\ l(\hat{y}, y) = (\hat{y} - y)^2$$

ו־$\mathcal{H}$ היא מחלקת הפונקציונלים הלינארים. כלומר המחלקה המכילה את כל הפונקציות מהצורה $h_w(x) = \langle w, x \rangle$ עבור $w \in \mathbb{R}^n$. נביט בבעיה $\mathrm{OPT}(\mathcal{H})$. בהינתן מדגם

$$S = \{(x_1, y_1), \ldots, (x_m, y_m)\} \subset \mathbb{R}^n \times \mathbb{R}$$

עלינו למצוא $w \in \mathbb{R}^n$ הממזער את

$$L_S(h_w) = \frac{1}{m} \sum_{i=1}^{m} (h_w(x_i) - y_i)^2 = \frac{1}{m} \sum_{i=1}^{m} (\langle w, x_i \rangle - y_i)^2$$

נדגיש שכאשר אנו מנסים למצוא w הממזער את $L_S(h_w)$ המדגם S הוא **קבוע**. כעת נשים שלכל $1 \leq i \leq m$ הפונקציה $l_{(x_i,y_i)}(w) := (\langle w, x_i\rangle - y_i)^2$ הינה קמורה (הוכיחו!). לכן, הפונקציה $L_S(w) := L_S(h_w)$ אף היא קמורה בתור ממוצע של פונקציות קמורות. לכן, ניתן להשתמש באופטימיזציה קמורה על מנת לפתור את בעיית האופטימיזציה $\mathrm{OPT}(\mathcal{H})$.

מה "הפך את $\mathrm{OPT}(\mathcal{H})$ לבעיה קמורה"? ובכן, שתי הנקודות הבאות היו מהותיות:

- הייתה לנו פרמטריזציה של $\mathcal{H}$ באמצעות קבוצה קמורה. כלומר הייתה לנו קבוצה קמורה W (במקרה שלנו $W = \mathbb{R}^n$) והעתקה $w \to h_w$ מ־W על $\mathcal{H}$.
- לכל $(x,y) \in X \times Y$ הפונקציה $l_{(x,y)}(w) = l(h_w(x), y)$ הייתה קמורה.

במובן מסוים, עשינו רדוקציה למקרה בו מחלקת ההיפותזות היא קבוצה קמורה W, ופונקציית ההפסד $L_S(w)$ הינה קמורה. נקודת המבט הזו מובילה להגדרה הבאה

הגדרה 4.1 בעיית למידה $(X, Y, \mathcal{H}, l)$ **תיקרא קמורה** אם קיימת העתקה $w \mapsto h_w$ מקבוצה קמורה W **על** $\mathcal{H}$ כך שלכל $(x,y) \in X \times Y$ הפונקציה $l_{(x,y)}(w) := l(h_w(x), y)$ הינה קמורה.

סימונים. כאשר נעבוד עם בעיות למידה קמורות יהיה נוח לסמן ב־$L_{\mathcal{D}}(w)$ ו־$L_S(w)$ את הפונקציות (הקמורות!) $L_{\mathcal{D}}(w) = L_{\mathcal{D}}(h_w)$ ו־$L_S(w) = L_S(h_w)$. כמו, יהיה נוח להשתמש בסימון $l_{(x,y)}(h)$ עבור $l(h(x), y)$.

דוגמא. משפחה בסיסית של בעיות למידה קמורות היא המשפחה הבאה, המכלילה את הדוגמא שהתחלנו ממנה. תהא $W \subset \mathbb{R}^n$ קבוצה קמורה, $l : \mathbb{R} \times \mathbb{R} \to [0, \infty)$ פונקציית הפסד כך שלכל $y \in \mathbb{R}$ הפונקציה $\hat{y} \mapsto l(\hat{y}, y)$ הינה קמורה, $X \subset \mathbb{R}^n$ ו־$Y = \mathbb{R}$. נביט בבעיית הלמידה $(X, Y, \mathcal{H}_W, l)$ כאשר $\mathcal{H}_W$ מכילה את כל הפונקציות מהצורה $h_w(x) = \langle w, x\rangle$ עבור $w \in W$. לא קשה להראות ש־$(X, Y, \mathcal{H}_W, l)$ הינה בעיית למידה קמורה (תרגיל).

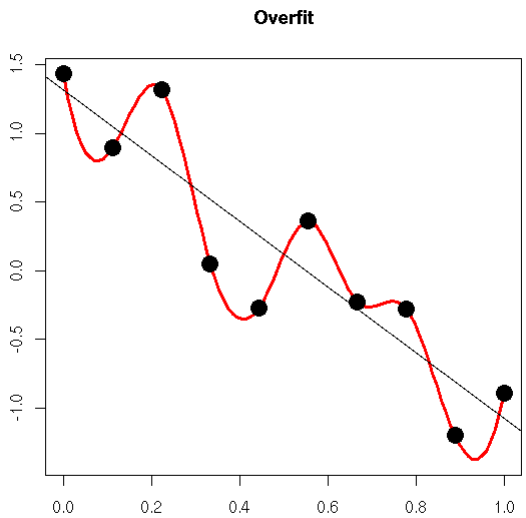
בשיעור הבא נדבר למידות של בעיות קמורות, ונראה שאם לכל (x,y), $l_{(x,y)}(w)$ היא ρ־ליפשיצית ו־W מוכלת בכדור ברדיוס R, אז קיים אלגוריתם למידה עם סיבוכיות מדגם $O\left(\frac{R\rho}{\epsilon^2}\right)$ עבור δ קבוע. יתר על כן, בתרגול תראו שכאשר לכל (x,y) ו־w ניתן לחשב ביעילות את $\nabla l_{(x,y)}(w)$ וכאשר W היא קבוצה "סבירה" (למשל, כדור), ניתן לממש את האלגוריתם הנ"ל ביעילות. העובדות הללו יתנו לנו משפחה עשירה של אלגוריתמי למידה יעילים. למשל, לבעיות רגרסייה. כאמור, בהמשך, נראה כיצד להשתמש באלגוריתמים הללו על מנת לטפל בבעיות למידה שאינן קמורות.

5 אלגוריתם יציבים ולמידות

במשפט היסודי חסמנו את סיבוכיות המדגם של אלגוריתמי ERM ע"י כך שהראנו שכאשר המדגם גדול מספיק, השגיאה האמפירית של כל ההיפותזות במחלקה קרובה לשגיאה האמיתית. ניתן לפתח תורה דומה גם עבור בעיות למידה קמורות, אך לא נעשה זאת. אלא, אנו נשתמש בטיעון אחר (פשוט הרבה יותר). קונקרטית, אנו נחסום את סיבוכיות המדגם של האלגוריתם שנציג ע"י זה שנראה שלושה דברים:

האלגוריתם הינו יציב. כלומר, אם משנים את המדגם מעט (למשל, מחליפים דוגמא אחת), הפלט לא משתנה הרבה.

אלגוריתמים יציבים לא עושים overfit. באופן לא מדוייק, נאמר שאלגוריתם למידה עושה overfit אם הוא מחזיר היפותזה ש"מתאימה מידי" למדגם. כלומר, יש לה שגיאה אמפירית מאד קטנה, אך שגיאה אמיתית גדולה. למשל אלגוריתם רגרסייה המחזיר את ההיפותזה האדומה על המדגם הנתון, כלל הנראה עושה אוברפיט, בשונה מאלגוריתם המחזיר את ההיפותזה האפורה.



נעיר שאוברפיט היא **תכונה של האלגוריתם**, ואיננה קשורה למחלקה! כמו כן, אלגוריתם שלא עושה אוברפיט לא בהכרח מחזיר היפותזה עם שגיאה נמוכה! למשל, אלגוריתם שמחזיר תמיד את אותה פונקציה איננו עושה אוברפיט - השגיאה האמפירית של ההיפותזה שהוא יחזיר לא תהיה רחוקה מהשגיאה האמיתית שלה. עם זאת, בודאי שאלגוריתם כזה בד"כ יחזיר היפותזה עם שגיאה גבוהה. ההערה הזו מובילה אותנו לתכונה הבאה:

האלגוריתם מחזיר היפותזה עם שגיאה אמפירית נמוכה, ביחס לשגיאה האמפירית הטובה ביותר של היפותזה במחלקה ($L_S(\mathcal{H})$). במילים אחרות, הוא "כמעט" ERM.

משלושת התכונות הללו, נסיק שלאלגוריתם שנציג יש סיבוכיות מדגם נמוכה, שכן, מהתכונה האחרונה נדע שהשגיאה האמפירית של ההיפותזה אותה הוא יחזיר תהיה קטנה, ומשתי התכונות הראשונות אנו נדע שגם השגיאה אמיתית תהיה קטנה.

בשיעור היום נטפל בנקודה השניה. כלומר, אנו נראה שאלגוריתמים יציבים לא עושים overfit. לשם כך, נצטרך להגדיר במדויק מתי אלגוריתם הוא יציב. נקבע, אם כן, בעיית למידה $(X, Y, \mathcal{H}, l)$ ואלגוריתם למידה $\mathcal{A}$ לבעיה. יהא

$$S = \{(x_1, y_1), \ldots, (x_m, y_m)\} \subset X \times Y$$

מדגם. כעת, נקבע דוגמא (x_i, y_i) במדגם, ונביט על השגיאה של $\mathcal{A}(S)$ על אותה דוגמא. כלומר, על $l_{(x_i,y_i)}(\mathcal{A}(S))$. נשאל את השאלה ההיפוטתית הבאה: מה היה קורה אם לא היינו מראים ל־$\mathcal{A}$ את (x_i, y_i)? קונקרטית, האם השגיאה על (x_i, y_i) תגדל כאשר נוציא את (x_i, y_i) מהמדגם (ואולי נחליף אותה בדוגמא אחרת)? במילים אחרות, האם האלגוריתם באמת "למד" את $h^*(x_i)$, או שהוא התאים את עצמו יותר מידי אליה? אנו נאמר ש־$\mathcal{A}$ הוא יציב, אם תמיד התשובה לשאלה הנ"ל האחרונה היא "כן, האלגוריתם באמת למד את $h^*(x_i)$". כלומר, אם החלפת (x_i, y_i) בדוגמא אחרת לא תגדיל בהרבה את $l_{(x_i,y_i)}(\mathcal{A}(S))$.

הגדרה 5.1 עבור פונקציה $\epsilon : \mathbb{N} \to \mathbb{R}_+$ אנו נאמר ש־$\mathcal{A}$ הוא **$\epsilon(m)$־יציב** אם לכל מדגם

$$S = \{(x_1, y_1), \ldots, (x_m, y_m)\} \subset X \times Y$$

ולכל $1 \leq i \leq m$ ו־$(x, y) \in X \times Y$ מתקיים

$$l_{(x_i,y_i)}(\mathcal{A}(S^i)) \leq l_{(x_i,y_i)}(\mathcal{A}(S)) + \epsilon(m)$$

כאשר S^i הוא המדגם המתקבל מ־S ע"י החלפת (x_i, y_i) ב־(x, y).

אינטואיטיבית, היינו מצפים שעבור אלגוריתם יציב, כאשר $S \sim \mathcal{D}^m$, השגיאה האמיתית של $\mathcal{A}(S)$ לא תהיה גדולה משמעותית מהשגיאה האמפירית. שכן, אינטואיטיבית, האלגוריתם לא "מרמה" ומקטין את השגיאה על דוגמה מסויימת בלי "באמת ללמוד אותה". הטענה הבאה נותנת אישוש פורמלי לאינטואיציה הזו

למה 5.2 (אלגוריתם יציבים לא עושים overfit) אם $\mathcal{A}$ הוא אלגוריתם **$\epsilon(m)$־יציב** אז מתקיים

$$E_{S\sim\mathcal{D}^m}\left[L_{\mathcal{D}}(\mathcal{A}(S)) - L_S(\mathcal{A}(S))\right] \leq \epsilon(m)$$

הוכחה: תהא $(x, y) \sim \mathcal{D}$ דוגמא אקראית שאיננה תלויה ב־S. יהא, כמו כן, $i \sim \mathrm{Uni}([m])$ (כלומר, i מתפלג אחיד ב־$\{1, 2, \ldots, m\}$). נסמן ב־S^i את המדגם המתקבל מ־S ע"י החלפת (x_i, y_i) ב־(x, y). מתקיים

$$\begin{aligned}
E_{S\sim\mathcal{D}^m}\left[L_{\mathcal{D}}(\mathcal{A}(S)) - L_S(\mathcal{A}(S))\right] &= E_{S\sim\mathcal{D}^m}\left[E_{(x,y)\sim\mathcal{D}} l_{(x,y)}(\mathcal{A}(S)) - L_S(\mathcal{A}(S))\right] \\
&= E_{S\sim\mathcal{D}^m,(x,y)\sim\mathcal{D}}\left[l_{(x,y)}(\mathcal{A}(S)) - L_S(\mathcal{A}(S))\right] \\
&= E_{S\sim\mathcal{D}^m,(x,y)\sim\mathcal{D},i\sim\mathrm{Uni}([m])}\left[l_{(x_i,y_i)}(\mathcal{A}(S^i)) - L_S(\mathcal{A}(S))\right] \\
&= E_{S\sim\mathcal{D}^m,(x,y)\sim\mathcal{D},i\sim\mathrm{Uni}([m])}\left[l_{(x_i,y_i)}(\mathcal{A}(S^i)) - l_{(x_i,y_i)}(\mathcal{A}(S))\right] \\
&\leq \epsilon(m)
\end{aligned}$$

כאן, השיוויון השלישי נובע מכך שההתפלגות של $((x, y), S)$ שווה לזו של $\big((x_i, y_i), S^i\big)$. השוויון הרביעי נובע מכך ש־$L_S(\mathcal{A}(S)) = \frac{1}{m}\sum_{i=1}^m l_{(x_i,y_i)}(\mathcal{A}(S)) = E_{i\sim\mathrm{Uni}([m])} l_{(x_i,y_i)}(\mathcal{A}(S))$.

■