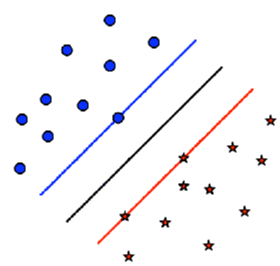
מבוא למערכות לומדות - הרצאה 6 - SVM, תחליפים קמורים ושיטות גרעין

21 במאי 2015

1 אלגוריתם ה־SVM

ניזכר שהאלגוריתם Hard-SVM, אותו ראיתם בתרגול, מוצא על מישור המפריד את הדוגמאות החיוביות מהשליליות עם שוליים (Margin) מקסימליים. בתמונה הבאה מופיעה דוגמא לעל מישור כזה:



אלגוריתמית, ראיתם שהמטרה היא למצוא פונקציונל לינארי[1] $h_w(x) = \langle w,x \rangle$ כך ש־ $h_w(x) \geq 1$ על כל הדוגמאות החיוביות ו־$h_w(x) \leq -1$ על כל הדוגמאות השליליות, ובנוסף w מנורמה קטנה ככל האפשר תחת האילוצים הללו. כלומר, המטרה היא לפתור את בעיית האופטימיזציה (הקמורה) הבאה:

[1] לשם פשטות, לאורך כל השיעור נצטמצם לעל מישורים הומוגניים. כלומר, לפונקציונאלים מהצורה $x \mapsto \langle w,x \rangle$ ולא $x \mapsto \langle w,x \rangle + b$.

Hard-SVM

קלט: $S=\{(x_1,y_1),\ldots,(x_m,y_m)\}\subset\mathbb{R}^n\times\{\pm1\}$
פלט: $w\in\mathbb{R}^n$ הממזער את $\|w\|^2$ תחת האילוצים

$$\forall 1\le i\le m,\quad 1-y_i\langle w,x_i\rangle\le 0 \tag{1}$$

האלגוריתם Hard-SVM הוא אלגוריתם נהדר, אך הוא פועל רק אם קיים על־מישור מפריד! למרבה הצער, רוב הפעמים זה לא המצב, ואז, כל וקטור $w\in\mathbb{R}^n$ יפר לפחות אחד מבין האילוצים (1). האלגוריתם SVM (או, Soft-SVM) מהווה אנלוג של Hard-SVM הפועל גם כאשר לא קיים על מישור מפריד. האלגוריתם מחפש h_w באופן המנסה מצד אחד למזער את $\|w\|^2$, ומצד שני להפר "כמה שפחות" את האילוצים (1).

אנו נכמת את "רמת ההפרה" של האילוץ

$$1-y_i\langle w,x_i\rangle\le 0$$

ע"י הביטוי[2]

$$l^{\text{hinge}}_{(x_i,y_i)}(w):=(1-y_i\langle w,x_i\rangle)_+$$

(בהמשך נסביר מאיפה בא הסימון $l^{\text{hinge}}_{(x_i,y_i)}(w)$. בינתיים, התייחסו אליו רק בתור סימון.) נשים לב שהביטוי הנ"ל שווה ל־0 כאשר האילוץ לא מופר, וגדל לינארית ככל האילוץ מופר יותר. רמת ההפרה של כלל האילוצים תכומת ע"י ממוצע ההפרות:

$$L^{\text{hinge}}_S(w):=\frac{1}{m}\sum_{i=1}^m l^{\text{hinge}}_{(x_i,y_i)}(w)$$

כעת, אלגוריתם ה־SVM ינסה למזער גם את $\|w\|^2$ וגם את $L^{\text{hinge}}_S(w)$. קונקרטית, האלגוריתם יקבל בתור פרמטר ערך $\lambda>0$ וימזער את הביטוי

$$L^{\text{hinge}}_S(w)+\lambda\|w\|^2$$

על פני $w\in\mathbb{R}^n$. נסכם:

(Soft-)SVM

קלט: $S=\{(x_1,y_1),\ldots,(x_m,y_m)\}\subset\mathbb{R}^n\times\{\pm1\}$, פרמטר רגולריזציה λ
פלט: $w\in\mathbb{R}^n$ הממזער את

$$L^{\text{hinge}}_S(w)+\lambda\|w\|^2=\frac{1}{m}\sum_{i=1}^m(1-y_i\langle w,x_i\rangle)_++\lambda\|w\|^2$$

[2] עבור $a\in\mathbb{R}$, נסמן ב־$(a)_+:=\max\{0,a\}$ את **החלק החיובי** של a.

1.1 הצגה כ־RLM וניתוח סיבוכיות מדגם וזמן ריצה.

על מנת לנתח את אלגוריתם ה־SVM ולהבין כיצד לממשו, נראה שניתן להציגו בתור אלגוריתם המממש את כל ה־RLM ביחס לבעיה קמורה מתאימה. לאחר שנעשה זאת, נוכל להשתמש בתיאוריה שפיתחנו בשיעור הקודם. אך ראשית, על מנת להקל על האנליזה, נכליל במעט את ההגדרה של בעיית למידה.

אלגוריתם ה־SVM (ואלגוריתמים נוספים) מחזיר היפותזה מהצורה $x \mapsto \text{sign}(h(x))$ עבור פונקציה $h : X \to \mathbb{R}$ (במקרה של SVM, $h(x) = h_w(x) = \langle w, x \rangle$). לעיתים יהיה נוח יותר לנתח ישירות את ההיפותזה h במקום את $x \mapsto \text{sign}(h(x))$. על מנת לטפל בזה במסגרת של PAC, נרשה להיפותזות להחזיר ערך ממשי כלשהו במקום ערך ב־$Y = \{\pm 1\}$. כפועל יוצא, בהינתן תיוג $y \in Y$, פונקציית ההפסד שנעבוד איתה תצטרך להיות מוגדרת לכל $\hat{y} \in \mathbb{R}$. לכן, נגדיר פונקציית הפסד להיות פונקציה $l : \mathbb{R} \times Y \to \mathbb{R}_+$. לדוגמא, בקלסיפיקציה, פונקצייה ההפסד הטבעית (ה־0-1-loss) תחזיר 0 אם הסימן של $\hat{y}$ שווה ל־y ו־1 אחרת. כלומר,

$$l_{0-1}(\hat{y}, y) = \begin{cases} 0 & \hat{y}y > 0 \\ 1 & \hat{y}y \leq 0 \end{cases}$$

נחוזר ל־SVM. נגדיר $l^{\text{hinge}} : \mathbb{R} \times Y \to \mathbb{R}_+$ באופן הבא:

$$l^{\text{hinge}}(\hat{y}, y) = (1 - \hat{y}y)_+$$

פונקציית ההפסד הנ"ל נקראית **ההינג'־לוס** (hinge loss).

נניח מעתה שכל הדוגמאות נמצאות בכדור $B_\rho = \{x \in \mathbb{R}^n : \|x\| \leq \rho\}$. נביט בבעיית הלמידה $(B_\rho, \{\pm 1\}, \mathcal{H}, l^{\text{hinge}})$ כאשר $\mathcal{H} = \{h_w : w \in \mathbb{R}^n\}$. נשים לב ש־

- הבעיה הזו הינה קמורה! יתר על כן, אנו נראה שהיא אף ρ־ליפשיצית, וכמו כן ניתן לחשב ביעילות את $l^{\text{hinge}}_{(x,y)}(w),\ \nabla l^{\text{hinge}}_{(x,y)}(w)$.
- לכן, ניתן לממש את כלל ה־RLM ביעילות.
- יתר על כן כלל ה־RLM הוא בדיוק אלגוריתם ה־SVM!

מהמשפט שהוכחנו בשיעור הקודם, אם נריץ את אלגוריתם ה־SVM עם $\lambda = \sqrt{\frac{2\rho^2}{R^2 m}}$ נקבל ש־

$$E_{S \sim \mathcal{D}^m}[L^{\text{hinge}}_{\mathcal{D}}(\mathcal{A}(S))] \leq L^{\text{hinge}}_{\mathcal{D}}(\mathcal{H}_R) + \rho R \sqrt{\frac{8}{m}} \tag{2}$$

כאשר $\mathcal{H}_R = \{h_w : \|w\| \leq R\}$. אי־שיוויון עדיין איננו מספק, שכל מעניין אותנו $L^{0-1}_{\mathcal{D}}(\mathcal{A}(S))$. הלמה הבאה מקשרת בין שתי פונקציות ההפסד.

למה 1.1 לכל $h : B_\rho \to \mathbb{R}$ מתקיים $L^{0-1}_{\mathcal{D}}(h) \leq L^{\text{hinge}}_{\mathcal{D}}(h)$

יחד עם אי־שיוויון (2) נקבל ש־

משפט 1.2 אם נריץ את אלגוריתם ה־SVM עם $\lambda = \sqrt{\frac{2\rho^2}{R^2 m}}$ נקבל ש־

$$E_{S\sim\mathcal{D}^m}[L_{\mathcal{D}}^{0-1}(\mathcal{A}(S))] \leq E_{S\sim\mathcal{D}^m}[L_{\mathcal{D}}^{\text{hinge}}(\mathcal{A}(S))] \leq L_{\mathcal{D}}^{\text{hinge}}(\mathcal{H}_R) + \rho R\sqrt{\frac{8}{m}}$$

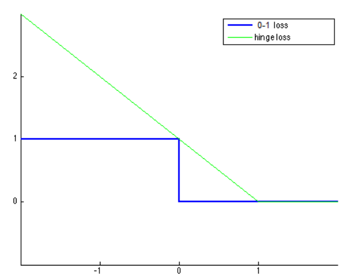
נשים לב שככל שאנו מגדילים את R (או, באופן שקול, מקטינים את λ) הגורם $L_{\mathcal{D}}^{\text{hinge}}(\mathcal{H}_R)$ קטן, שכן אנו נשתמש במחלקה יותר גדולה. לעומת זאת, הגורם השני, $\rho R\sqrt{\frac{8}{m}}$, יגדל.

ניגש להוכיח את הלמה.

הוכחה: נשים לב שמתקיים,

$$l_{0-1}(\hat{y}, y) \leq l^{\text{hinge}}(\hat{y}, y) \tag{3}$$

אכן, גם l^{hinge} וגם l_{0-1} הם מהצורה $f(\hat{y}y)$ עבור $f : \mathbb{R} \to \mathbb{R}_+$, כאשר עבור l^{hinge}, $f_{\text{hinge}}(x) = (1-x)_+$ ועבור l_{0-1}, $f_{0-1}(x) = \begin{cases} 1 & x < 0 \\ 0 & x \geq 0 \end{cases}$ (ראו בציור הבא).



כמו כן, לא קשה לוודא ש־$\forall x,\ f_{\text{hinge}}(x) \geq f_{0-1}(x)$. מאי־שיוויון (3) נסיק שמתקיים

$$L_{\mathcal{D}}^{0-1}(h) = E_{(x,y)\sim\mathcal{D}}[l_{0-1}(h(x), y)] \leq E_{(x,y)\sim\mathcal{D}}[l_{\text{hinge}}(h(x), y)] = L_{\mathcal{D}}^{\text{hinge}}(h)$$

■

1.2 תחליפים קמורים (Convex Surrogates) - מעבר ל־SVM

נאמץ את נקודת המבט הבאה בנוגע לאלגוריתם ה־SVM ולקשר שלו לבעיה של לימוד חצאי מרחבים. התחלנו עם בעיית הלמידה

$$(B_\rho, \{\pm 1\}, \mathcal{H}, l_{0-1})$$

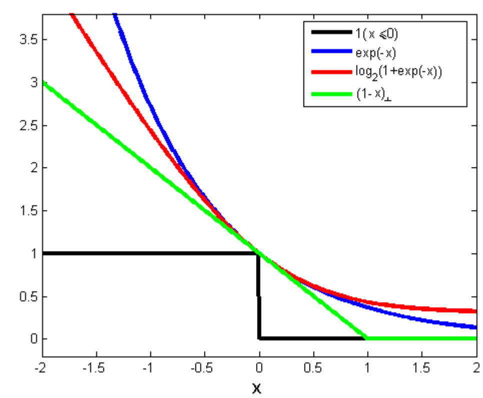
כאשר $\mathcal{H}$ היא אוסף הפונקציונאלים הליניאריים. לצערנו, הבעיה איננה קמורה ואף (ככל הנראה) קשה מאד חישובית. על מנת לתקוף אותה בכל זאת, **החלפנו** את פונקציית ההפסד

l_{0-1} ב־l_{hinge} וקיבלנו בעיה קמורה שניתן לפתור ביעילות. כמו כן, l_{hinge} היווה תחליף "טוב" ל־l_{0-1} במובן שמתקיים $L_{\mathcal{D}}^{0-1}(h) \leq L_{\mathcal{D}}^{\text{hinge}}(h)$. כפי שנראה, ניתן להשתמש בשיטה הנ"ל בבעיות נוספות, מעבר לקלסיפיקציה בינארית. הצעד הראשון יהיה להגדיר פורמאלית מתי פונקצייה הפסד היא תחליף קמור לפונקציית הפסד אחרת. על מנת שנוכל לטפל במגוון רחב של בעיות, נגדיר זאת עבור פונקציות $l : \mathbb{R}^k \times Y \to \mathbb{R}_+$.

הגדרה 1.3 תחליף קמור לפונ' הפסד $l : \mathbb{R}^k \times Y \to \mathbb{R}_+$ היא פונ' $l_s : \mathbb{R}^k \times Y \to \mathbb{R}_+$ המקיימת:

- לכל $y \in Y$, הפונקציה $\hat{y} \mapsto l_s(\hat{y}, y)$ הינה קמורה.
- לכל $\hat{y} \in \mathbb{R}^k$ ו־$y \in Y$ מתקיים $l(\hat{y}, y) \leq l_s(\hat{y}, y)$.

דוגמא. אם $f : \mathbb{R} \to \mathbb{R}$ קמורה המקיימת $f(x) \geq \begin{cases} 1 & x \leq 0 \\ 0 & x > 0 \end{cases}$ אז $l(\hat{y}, y) = f(\hat{y}y)$ הינה תחליף קמור ל־l_{0-1}. למשל, ההינג'־לוס מתאים ל־$f(x) = (1-x)_+$. בציור הבא מופיעות מספר פונקציות f המשתמשים בהן בפועל:

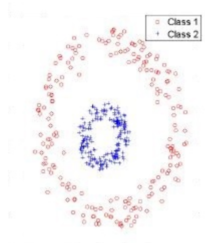


כפי שתראו בתרגול ובתרגיל, הרבה פעמים (גם בבעיות מעבר לקלסיפיקציה בינארית), החלפה של פונ' הפסד בתחליף קמור תניב בעיית למידה קמורה שניתן לפתור ביעילות.

2 שיכונים במרחבים ממימד גבוה ושיטות גרעין

2.1 שיכונים במימד גבוה

אלגוריתם ה־SVM מאפשר ללמוד חצאי־מרחבים (או, באופן שקול, פונקציונאלים לינאריים) ביחס להינג'־לוס. הרבה פעמים, פונקציונאלים לינארים היא מחלקה שאיננה עשירה דיה, ושגיאת הקירוב ביחס אליה הינה גבוהה, כמו למשל בתמונה הבאה:



שיכון במרחבים ממימד גבוה היא שיטה המאפשר להשתמש ב־SVM, ובעוד שורה של אלגוריתמים דומים כגון RLM, על מחלקות הרבה יותר עשירות. על מנת להיות קונקרטיים, נצמצם את הדיון ל־SVM.

הרעיון הבסיסי הוא למפות את מרחב הקלטים למרחב ממימד גבוה יותר, ואז להפעיל SVM. קונקרטית, יהא $\Psi : X \to \mathbb{R}^N$ מיפוי של מרחב הקלטים למרחב אחר, בד"כ ממימד גבוה יותר מהמימד של X (לפעמים אפילו $N = \infty$!).

(Soft-)SVM with a mapping Ψ:X$\to \mathbb{R}^N$

קלט: $S = \{(x_1, y_1), \ldots, (x_m, y_m)\} \subset X \times \{\pm 1\}$, פרמטר רגולריזציה λ.

1. הרץ SVM עם פרמטר λ על המדגם

$$\Psi(S) := \{(\Psi(x_1), y_1), \ldots, (\Psi(x_m), y_m)\}$$

וקבל $h_w : \mathbb{R}^N \to \mathbb{R}$ עבור $w \in \mathbb{R}^N$.

2. החזר את ההיפותזה

$$h_w^\Psi(x) := h_w(\Psi(x))$$

מהמשפט שהוכחנו עבור SVM (משפט 1.2) נובע שימוש בשיטה שתוארה ממזער את ההינג'־לוס ביחס למחלקת ההיפותזות

$$\mathcal{H}^\Psi := \{h_w^\Psi : w \in \mathbb{R}^N\}$$

קונקרטית,

משפט 2.1 נניח ש־$\Psi(X) \subset B_\rho$. אם נריץ SVM עם המיפוי Ψ ועם $\lambda = \sqrt{\frac{2\rho^2}{R^2 m}}$ נקבל

$$E_{S\sim\mathcal{D}^m}[L_\mathcal{D}^{0-1}(\mathcal{A}(S))] \le E_{S\sim\mathcal{D}^m}[L_\mathcal{D}^{\text{hinge}}(\mathcal{A}(S))] \le L_\mathcal{D}^{\text{hinge}}(\mathcal{H}_R^\Psi) + \rho R\sqrt{\frac{8}{m}}$$

הוכחת המשפט קלה ומושארת כתרגיל.

דוגמא

מדוע זה עוזר לנו? ובכן ע"י בחירת מיפוי מתאים ניתן לקבל מחלקות עשירות. לדוגמא אם $X = \mathbb{R}^2$ והמיפוי הינו

$$\Psi(x_1, x_2) = (1, x_1, x_2, x_1^2, x_2^2, x_1 x_2)$$

לא קשה לראות ש־$\mathcal{H}_\Psi$ היא מחלקת כל הפולינומים מדרגה ≤ 2. אכן, עבור $w \in \mathbb{R}^6$ מתקיים

$$\begin{aligned} h_w^\Psi(x) &= \langle (w_1, w_2, w_3, x_4, w_5, w_6), (1, x_1, x_2, x_1^2, x_2^2, x_1 x_2) \rangle \\ &= w_1 + w_2 x_1 + w_3 x_1 + w_4 x_1^2 + w_5 x_2^2 + +w_6 x_1 x_2 \end{aligned}$$

מכאן, h_w^Ψ הינו פולינום מדרגה ≤ 2 וע"י בחירה מתאימה של w ניתן לקבל כל פולינום. לסיכום, ע"י שימוש בשיכון Ψ ומעבר ל־$\mathcal{H}^\Psi$ הגדלנו את מחלקת ההיפותזות שלנו מפונקציונליים לינאריים (פולינומים מדרגה 1) לפולינומים מדרגה 2. למשל, כעת יש לנו במחלקה היפותזות הממפות פנים של אליפסה ל־1 ואת החוץ שלה ל־-1 ולכן שגיאת הקירוב בהתפלגות המתוארת בתמונה למעלה הינה טובה. כמובן, ניתן להרחיב את הדוגמא הנ"ל ולקחת $X = \mathbb{R}^n$ עבור $n \geq 2$ ופולינומים מדגה גבוהה יותר.

מכשולים בדרך

1. **הכללה.** לחצאי מרחבים במימד גבוה יש מימד VC גדול. לכן שימוש נאיבי בשיטה יצריך מספר דוגמאות גדול. למרבה המזל, SVM מאפשר לשלוט בשגיאת ההכללה גם במרחבים ממימד גבוה. אכן, ראינו שבאמצעות בחירה מתאימה של פרמטר הרגולריזציה ניתן לחסום את סיבוכיות המדגם ביחס לנורמה, ללא תלות במימד.
2. **חישוב.** כאשר N מאד גדול, יהיה יקר מאד מבחינה חישובית לעבוד ב־$\mathbb{R}^N$ (שלא לדבר על $N = \infty$...). לשם כך, אנו נלמד שיטה להתמודד עם הקושי הנ"ל באמצעות שימוש בפונקציות הנקראות **גרעינים (Kernels)**.

2.2 שיטות גרעין

נניח שאנו רוצים ליישם את הפרדיגמה שתוארה בפסקה הקודמת ביחס ל־$\Psi : X \to \mathbb{R}^N$, אך לצערנו N מאד גדול. מאחורי שיטות הגרעין עומדת האבחנה הפשוטה הבאה: הרבה פעמים אין צורך לעבוד "באמת" ב־$\mathbb{R}^N$, אלא, די לדעת לחשב את המכפלה הפנימית של שני מיפויים נתונים. כלומר, כל מה שצריך להיות מסוגלים לעשות ביעילות זה לחשב ביעילות את המכפלה הפנימית

$$k(x, x') := \langle \Psi(x), \Psi(x') \rangle$$

הפונקציה $k : X \times X \to \mathbb{R}$ הנ"ל נקראית **הגרעין** של Ψ.

קונקרטית, על מנת להשתמש ב־SVM צריך להיות מסוגלים לבצע ביעילות את הפעולות הבאות:

- **(ייצוג)** לייצג וקטורים $w \in \mathbb{R}^N$ באופן קומפקטי.

- **(הערכה)** בהינתן (ייצוג של) $w \in \mathbb{R}^N$ ו־$x \in X$, להעריך ביעילות את $h_w^\Psi(x)$.
- **(צעד גרדיינט)** בהינתן (ייצוג של) $w \in \mathbb{R}^N$, $\eta > 0$, ודוגמא (x, y), לחשב ביעילות את הייצוג של $w - \eta \nabla l^{\text{hinge}}_{(\Psi(x_i), y_i)}(w)$.

נסביר כיצד ניתן לבצע את הפעולות הללו בהינתן גרעין k כנ"ל.

ייצוג מפרידים לינארים ב־$\mathbb{R}^N$.

נייצג מפרידים לינאריים ב־$\mathbb{R}^N$ באמצעות צירוף לינארי של נקודות במדגם. נניח שנתון לנו מדגם $S \in (X \times Y)^m$. לכל (x_i, y_i) מתאים הוקטור $\Psi(x_i) \in \mathbb{R}^N$. אנו נעבוד רק עם וקטורים שהם צירוף לינארי של $\Psi(x_1), \ldots, \Psi(x_m)$. כלומר, עם וקטורים מהצורה

$$w = \sum_{i=1}^{m} \alpha_i \Psi(x_i)$$

לכן, על מנת לשמור ייצוג של וקטור כנ"ל, כל שעלינו לעשות זה לשמור בזיכרון את המקדמים $\alpha_1, \ldots, \alpha_m$ ואת הדוגמאות.

הערכת מפרידים לינאריים ב־$\mathbb{R}^N$.

בהינתן ייצוג כנ"ל של וקטור $w \in \mathbb{R}^N$, במהלך הלימוד ובודאי גם אח"כ נצטרך להיות מסוגלים להעריך, בהינתן $x \in X$, את $h_w^\Psi(x)$. נשים לב שמתקיים

$$h_w^\Psi(x) = \langle \sum_{i=1}^{m} \alpha_i \Psi(x_i), \Psi(x) \rangle = \sum_{i=1}^{m} \alpha_i k(x_i, x) \tag{4}$$

לכן, אם אנו מסוגלים להעריך את הגרעין $k(x, x')$ ביעילות נוכל גם להעריך את $h_w^\Psi(x)$.

צעדי גרדיינט

הפעולה האחרונה שעלינו לדעת לעשות על מנת לממש GD או SGD היא לחשב צעדי גרדיינט. כלומר, בהינתן ייצוג $(\alpha_1, \ldots, \alpha_m)$ של וקטור $w \in \mathbb{R}^N$ ודוגמא $(x_i, y_i) \in X \times Y$ עלינו לדעת לחשב את הייצוג של

$$\nabla l^{\text{hinge}}_{(\Psi(x_i), y_i)}(w)$$

כצירוף לינארי של $\Psi(x_1), \ldots, \Psi(x_m)$. אבל, ראיתם שמתקיים

$$\nabla l^{\text{hinge}}_{(\Psi(x_i), y_i)}(w) = \begin{cases} 0 & y_i \langle w, \Psi(x_i) \rangle \geq 1 \\ -y_i \Psi(x_i) & y_i \langle w, \Psi(x_i) \rangle < 1 \end{cases}$$

לכן, כל אשר עלינו לעשות הוא לחשב את $\langle w, \Psi(x_i) \rangle$ (שאת זה ראינו שניתן לעשות אם יודעים לחשב את הגרעין).

הוקטורים התומכים

לאחר שלב אימון, כבר אין לנו צורך לעשות צעדי גרדיינט. נשים לב שעל מנת להעריך את הביטוי (4), די לנו לשמור בזיכרון את המקדמים $\alpha_1, \ldots, \alpha_m$ **שאינם 0** ואת הדוגמאות המתאימות. הרבה פעמים ישנם מעט מקדמים כאלו, מה שמאפשר חיסכון משמעותי בזיכרון ובזמן שלוקח להעריך את ההיפותזות. הוקטורים המתאימים למקדמים הללו (כלומר, המיפוי של הדוגמאות המתאימות במרחב $\mathbb{R}^N$) נקראים **הוקטורים התומכים (Support Vectors)**.

2.3 דוגמאות לגרעינים

על מנת להשתמש בפרדיגמה שתיארנו עלינו להכיר מספר גרעינים. נצביע על שני גרעינים פופולאריים - הגרעין הפולינומי והגרעין הגאוסי (הנקרא גם RBF)

הגרעין הפולינומי

נקבע $d \geq 1$ ונביט במיפוי $\Psi : \mathbb{R}^n \to \mathbb{R}^N$ המוגדר ע"י

$$\Psi(x) = (x_{j_1} \cdot \ldots \cdot x_{j_d})_{(j_1,\ldots,j_d)\in\{0,\ldots,n\}^d}$$

כאשר x_0 פשוט מסמן 1. כלומר, $\Psi(x)$ הינו וקטור בין $N = (n+1)^d$ קואורדינטות המכיל את כל המכפלות מהצורה $x_{j_1} \cdot \ldots \cdot x_{j_d}$ עבור $(j_1, \ldots, j_d) \in \{0, \ldots, n\}^d$ (שימו לב שחלק מהמכפלות שוות אחת לשנייה). עבור המיפוי הנ"ל, $\mathcal{H}^\Psi$ הוא מרחב כל הפולינומים מדרגה $\geq d$.

נחשב את הגרעין המתאים. מתקיים

$$\begin{aligned}
(\langle x, x'\rangle + 1)^d &= \left(\sum_{i=1}^n x_i x_i' + 1\right)^d \\
&= \sum_{(j_1,\ldots,j_d)\in\{0,\ldots,n\}^d} (x_{j_1} x'_{j_1}) \cdot \ldots \cdot (x_{j_d} x'_{j_d}) \\
&= \sum_{(j_1,\ldots,j_d)\in\{0,\ldots,n\}^d} (x_{j_1} \cdot \ldots \cdot x_{j_d}) \cdot (x'_{j_1} \cdot \ldots \cdot x'_{j_d}) \\
&= \langle \Psi(x), \Psi(x')\rangle
\end{aligned}$$

לכן, הגרעין המתאים למיפוי הינו

$$k(x, x') = (\langle x, x'\rangle + 1)^d$$

נעיר שעבור d גדול, המרחב $\mathcal{H}^\Psi$ הוא מאד גדול. למשל, אם $d = n$ אז הוא מכיל את כל הפונקציות מ־$\{\pm 1\}^n$ ל־$\mathbb{R}$. מדוע זה לא מביא לסתירה לעובדה ש־"אין ארוחות חינם"? ובכן, אם נפעיל את Kernel-SVM עם הגרעין הנ"ל, גורם הרגולריזציה יאפשר לנו להשתמש בפועל רק בחלק קטן מאד מהמרחב.

הגרעין הגאוסי

נביט במיפוי $\Psi:\mathbb{R}^n\to\mathbb{R}^\infty$ המוגדר ע"י

$$\Psi(x)=\frac{e^{-\frac{1}{2}\|x\|^2}}{\sqrt{d!}}\left(x_{j_1}\cdot\ldots\cdot x_{j_d}\right)_{(j_1,\ldots,j_d)\in\{1,\ldots,n\}^d,d\in\{0,1,\ldots\}}$$

נתעלם מהעובדה שהטווח של Ψ הוא אינסוף מימדי (נעיר שניתן לטפל בזה בצורה פורמאלית ע"י באמצעות התורה של **מרחבי הילברט**). עבור המיפוי הנ"ל, $\mathcal{H}^\Psi$ מכילה את כל הפונקציות מהצורה $e^{-\frac{1}{2}\|x\|^2}p(x)$ עבור פולינום $p:\mathbb{R}^n\to\mathbb{R}$ מדרגה כלשהי (למעשה, $\mathcal{H}^\Psi$ מכילה עוד פונקציות). נחשב את הגרעין המתאים. מתקיים

$$\begin{aligned}
e^{-\frac{1}{2}\|x-x'\|^2} &= e^{-\frac{1}{2}\|x\|-\frac{1}{2}\|x\|^2}e^{\langle x,x'\rangle}\\
&= e^{-\frac{1}{2}\|x\|-\frac{1}{2}\|x\|^2}\sum_{d=0}^{\infty}\frac{(\langle x,x'\rangle)^d}{d!}\\
&= e^{-\frac{1}{2}\|x\|-\frac{1}{2}\|x\|^2}\sum_{d=0}^{\infty}\frac{1}{d!}\sum_{(j_1,\ldots,j_d)\in\{1,\ldots,n\}^d}(x_{j_1}x'_{j_1})\cdot\ldots\cdot(x_{j_d}x'_{j_d})\\
&= \sum_{d=0}^{\infty}\sum_{(j_1,\ldots,j_d)\in\{1,\ldots,n\}^d}\left(\frac{e^{-\frac{1}{2}\|x\|^2}}{\sqrt{d!}}x_{j_1}\cdot\ldots\cdot x_{j_d}\right)\left(\frac{e^{-\frac{1}{2}\|x'\|^2}}{\sqrt{d!}}x'_{j_1}\cdot\ldots\cdot x'_{j_d}\right)\\
&= \langle\Psi(x),\Psi(x')\rangle
\end{aligned}$$

לכן, הגרעין המתאים למיפוי הינו

$$k(x,x')=e^{-\frac{1}{2}\|x-x'\|^2}$$

הגרעין הנ"ל, נקרא **הגרעין הגאוסי (או, גרעין RBF)**. באופן כללי יותר, גרעין גאוסי הוא גרעין מהצורה

$$k(x,x')=e^{-\frac{1}{2\sigma}\|x-x'\|^2}$$

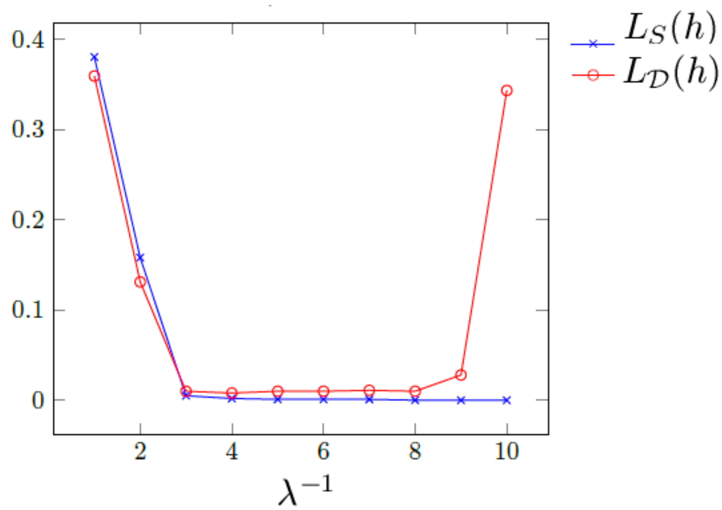
נעיר שהמרחב $\mathcal{H}^\Psi$ הוא מאד גדול. למעשה הוא **אוניברסלי** במובן שהוא יכול לקרב כל פונקציה רציפה $f:\mathbb{R}^n\to\mathbb{R}$ (במובן מתאים של קירוב). כמו שהערנו במקרה של הגרעין הפולינומי, זה לא מביא לסתירה לעובדה ש-"אין ארוחות חינם" מפני שגורם הרגולריזציה מאפשר להשתמש בפועל רק בחלק קטן מאד מהמרחב.

3 ולידציה ובחירת מודל

כאשר אנו לומדים עם SVM עלינו לבחור גורם רגולריזציה $\lambda>0$. לבחירה של λ יש שתי השפעות, עם השלכות שונות לגבי הבחירה:

- אם λ גדול, נצטמצם למחלקה קטנה של היפותזות ולכן שגיאת ההכללה תהיה קטנה.
- אם λ קטן, נעבוד עם מחלקה גדולה של היפותזות ולכן שגיאת הקירוב תהיה קטנה. זה נכון במיוחד כאשר משתמשים בגרעינים - למשל אם משתמשים בגרעין אוניברסלי (כמו למשל הגרעין הגאוסי), ע"י בחירה של $\lambda > 0$ מספיק קטנה נוכל להתקרב **לכל** פונקצייה.

הגרף הבא מתאר את יחסי הגומלין הללו:



עבור λ גדולה (צד שמאל של הגרף), אנו נקבל שגיאת קירוב גדולה. השגיאה תהיה גדולה גם על המדגם, שכן, שגיאת ההכללה תהיה מאד קטנה, ולכן השגיאה על המדגם תהיה קרובה לשגיאה האמיתית. עבור λ קטנה, שגיאת האימון תהיה קטנה מאד, שכן, אנו נעבוד עם מחלקה גדולה שסביר שתכיל היפותזה שכמעט ולא טועה על המדגם.

כיצד, אם כן, נבחר את λ? ובכן, אחת הדרכים הפופולאריות היא הדרך הבאה:

בחירת מודל

קלט: $S \in (X \times Y)^m$ ו־k, m_1, m_2 כך ש־$m = m_1 + m_2$.
1. חלק את המדגם לשני חלקים, $S_{\text{train}}, S_{\text{validation}}$ בגודל m_1 ו־m_2
2. לכל $\lambda \in \{1, 2^{-1}, \ldots, 2^{-k}\}$ הרץ SVM עם פרמטר λ על S_{train} וקבל היפותזה h_λ.
3. החזר את ההיפותזה h_λ עם השגיאה המינימאלית על $S_{\text{validation}}$

נעיר שהשיטה הזו טובה לא רק ל־SVM - היא מתאימה לכל סיטואציה בה יש לנו סדרה (או רצף) של מחלקות $\mathcal{H}_1 \subset \mathcal{H}_2 \subset, \ldots$. למשל בתרגיל תראו כיצד לעשות בחירת־מודל עבור רגרסייה כאשר $\mathcal{H}_d$ הוא מרחב הפולינומים מדרגה d.