

תרגול 9 - Logistic Regression and Gradient Descent

[PDF](#)[Code](#)

תקציר התיאוריה

הגישה הדיסקרימינטיבית הסתברותית

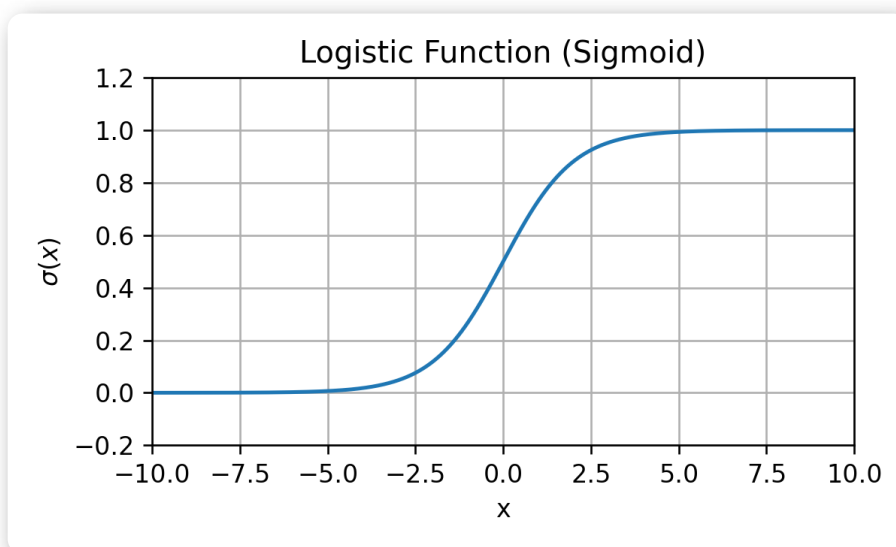
בגישה זו ננסה ללמוד מודל פרמטרי אשר ימדל ישירות את $p_{y|x}(y|\mathbf{x})$ (מבלי ללמוד את הפילוג של \mathbf{x}). גישה זו יעילה מאד לבעיות סיווג, בהם קל לבנות מודלים פרמטריים $p_{y|x}(y|\mathbf{x}; \theta)$ שהם פונקציות הסתברות חוקיות (חיובית שהסכום עליה הוא 1). את הפרמטרים של המודל הפרמטרי נלמד לרוב בעזרת MLE או MAP.

הפונקציה הלוגיסטית (סיגמואיד)

הפונקציה הלוגיסטית מקבלת מספר בתחום $[-\infty, \infty]$ ומחזירה מספר בין 0 ל 1. היא לרוב מסומנת ב σ :

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

והיא נראית כך:



פונקציה זו שימושית לצורך הגדרת מודלים הסתברותיים של משתנים בינאריים.

הערה: בתחום של מערכות לומדות מקובל לכנות את הפונקציה הזו **סיגמוייד (sigmoid)** למרות שמבחינה מתמטית השם הזה מתאר משפחה הרבה יותר רחבה של פונקציות בעלות צורה של S.

תכונות

$$\begin{aligned} \sigma(-z) &= 1 - \sigma(z) \\ \frac{\partial}{\partial z} \log(\sigma(z)) &= \sigma(z) - \sigma^2(z) \end{aligned}$$

פונקציית ה Softmax

פונקציית ה softmax היא הרחבה של הפונקציה הלוגיסטית, והיא יכולה לשמש למידול פונקציות הסתברות של משתנים דיסקרטיים לא בינאריים (אך סופיים). הפונקציה לוקחת וקטור כלשהו \mathbf{z} באורך C ומייצרת ממנו וקטור חדש חיובי שסכום האיברים שלו הוא 1. הפונקציה מוגדרת באופן הבא:

$$\text{softmax}(\mathbf{z}) = \frac{1}{\sum_{c=1}^C e^{z_c}} [e^{z_1}, e^{z_2}, \dots, e^{z_C}]^T$$

או לחילופין, הערך של האיבר ה i של הפונקציה הינו:

$$\text{softmax}(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{c=1}^C e^{z_c}}$$

תכונות

- אינווריאנטיות לתוספת של קבוע (לכל אברי הוקטור): $\forall i, \text{softmax}(\mathbf{z} + \mathbf{a})_i = \text{softmax}(\mathbf{z})_i$
- $\frac{\partial}{\partial z_j} \log(\text{softmax}(\mathbf{z})_i) = \delta_{i,j} - \text{softmax}(\mathbf{z})_j$ כאשר $\delta_{i,j} = \begin{cases} 1 & i=j \\ 0 & i \neq j \end{cases}$

Logistic Regression

בניגוד לשם, logistic regression היא שיטה לפתרון בעיות סיווג בגישה הדיסקרימינטיבית הסתברותית. בשיטה זו אנו נבחר פונקציות פרמטריות כלשהן $f_c(\mathbf{x}; \theta_c)$ ונשתמש בהן על מנת לבנות מודל פרמטרי. נסמן:

- את הוקטור θ כוקטור אשר כולל את כל C וקטורי הפרמטרים: $\theta = [\theta_1^T, \theta_2^T, \dots, \theta_C^T]^T$
- את הפונקציה \mathbf{f} כפונקציה המאגדת את כל C הפונקציות הפרמטריות: $\mathbf{f} = [f_1(\mathbf{x}; \theta_1), f_2(\mathbf{x}; \theta_2), \dots, f_C(\mathbf{x}; \theta_C)]^T$

את הפילוג המותנה נמדל באופן הבא:

$$p_{y|\mathbf{x}}(y|\mathbf{x}; \theta) = \text{softmax}(\mathbf{f}(\mathbf{x}; \theta))_y = \frac{e^{f_y(\mathbf{x}; \theta_y)}}{\sum_{c=1}^C e^{f_c(\mathbf{x}; \theta_c)}}$$

לבעיות ה MLE או MAP של מודל זה אין פתרון סגור ואנו נחפש את הפתרון לבעיית האופטימיזציה בעזרת gradient descent.

ביטול היתירות של המודל

בגלל האינווריאנטיות של פונקציית ה softmax המודל הפרמטרי המוגדר על ידי הפונקציות f_c יהיה אינווריאנטי לשינויים מהצורה של: $f_c(\mathbf{x}; \theta_c) \rightarrow f_c(\mathbf{x}; \theta_c) + g(\mathbf{x})$. דרך נפוצה לבטל יתירות זו הינה על ידי קיבוע של אחת הפונקציות הפרמטריות, לרוב הראשונה $c = 1$, להיות שווה זהותית ל 0: $f_1(\mathbf{x}; \theta_1) = 0$.

המקרה הבינארי

במקרה הבינארי ישנם רק שתי מחלקות ($C = 2$), אותן נסמן ב 0 ו 1. נקבע את הפונקציה הפרמטרית של המחלקה $y = 0$ להיות זהותית ל 0. נקבל את המודל הפרמטרי הבא:

$$p_{y|\mathbf{x}}(0|\mathbf{x}; \theta) = \frac{1}{1 + e^{f(\mathbf{x}; \theta)}} = 1 - \sigma(f(\mathbf{x}; \theta))$$

$$p_{y|x}(1|\mathbf{x};\boldsymbol{\theta}) = \frac{e^{f(\mathbf{x};\boldsymbol{\theta})}}{e^{f(\mathbf{x};\boldsymbol{\theta})} + 1} = \frac{1}{1 + e^{-f(\mathbf{x};\boldsymbol{\theta})}} = \sigma(f(\mathbf{x};\boldsymbol{\theta}))$$

רגרסיה לוגיסטית לינארית

הגרסא הלינארית של הרגרסיה הלוגיסטית היא המקרה שבו בוחרים את הפונקציות הפרמטריות להיות פונקציות לינאריות:

$$f_c(\mathbf{x};\boldsymbol{\theta}_c) = \boldsymbol{\theta}_c^\top \mathbf{x}$$

במקרה זה פונקציית ה objective שיש למזער היא קמורה (convex) ולכן מובטח ש gradient descent, במידה והוא מתכנס, יתכנס למינימום גלובלי.

Gradient descent (שיטת הגרדיאנט)

בעבור בעיית המינימיזציה:

$$\arg \min_{\boldsymbol{\theta}} g(\boldsymbol{\theta})$$

Gradient descent מנסה למצוא מינימום לוקאלי של $g(\boldsymbol{\theta})$ על ידי כך שהוא מתחיל בנקודה אקראית כלשהי במרחב ואז מתקדם בצעדים קטנים בכיוון ההפוך מהגרדיאנט, שהוא הכיוון שבו ה objective קטן בקצב המהיר ביותר. זהו אלגוריתם חמדן (greedy) אשר מנסה בכל צעד לשפר במעט את מצבו ביחס לשלב הקודם.

האלגוריתם

- מאתחלים את $\boldsymbol{\theta}^{(0)}$ בנקודה אקראית כלשהי.

- חוזרים על צעד העדכון הבא עד שמתקיים תנאי עצירה כל שהוא:

$$\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} - \eta \nabla_{\boldsymbol{\theta}} g(\boldsymbol{\theta}^{(t)})$$

הפרמטר η קובע את גודל הצעדים שהאלגוריתם יעשה.

תנאי עצירה

ישנם מספר דרכים להגדיר תנאי עצירה לאגוריתם:

- הגעה למספר צעדי עדכון שנקבע מראש: $t > \text{max-iter}$.
- כאשר הנורמה של הגרדיאנט קטנה מתחת לערך סף מסוים שנקבע מראש: $\|\nabla_{\boldsymbol{\theta}} g(\boldsymbol{\theta})\|_2 < \epsilon$.
- כאשר השיפור ב objective קטן מערך סף מסוים שנקבע מראש: $g(\boldsymbol{\theta}^{(t-1)}) - g(\boldsymbol{\theta}^{(t)}) < \epsilon$.
- שימוש בעצירה מוקדמת על מנת להתמודד עם overfitting (נרחיב על כך בהרצאה הבאה).

בעיית הבחירה של גודל הצעד

בכדי לגרום לאלגוריתם להתכנס (ולא להתבדר) אנו נאלץ לבחור גודל צעד שהוא לא גדול מידי. בפועל, בצורתו הפשוטה אלגוריתם ה gradient descent הוא מאד בעייתי משום שבכדי למנוע התבדרות גודל הצעד צריך להיות מאד קטן שידרוש מספר לא פרקטי של צעדים לצורך התכנסות.

תרגיל 9.1 - אופטימליות של כיוון גרדיאנט שלילי

תהי $f(x), x \in \mathbb{R}^d$ פונקציה ממשית גזירה פעמיים.

נתונה נקודה

$$x_\alpha = x + \alpha d, \quad d \in \mathbb{R}^d, \|d\| = 1, \alpha \geq 0.$$

מפיתוח טיילור לסדר שני עם שארית מתקיים

$$f(x_\alpha) = f(x) + \alpha d^\top \nabla f(x) + O(\alpha^2)$$

כאשר $O(\alpha^2)$ קטן בהרבה מ- $\alpha d^T \nabla f(x)$ עבור α קטנה מספיק.

שאלה מה הכיוון d שמוביל לירידה הכי גדולה בערך של $f(x)$ עבור ערך α קטן?

תזכורת אי שוויון קושי שזורץ קובע כי עבור מרחב מכפלה פנימית V מתקיים לכל $x, y \in V$

$$\langle x, y \rangle \leq \|x\| \cdot \|y\|$$

כאשר שוויון מתקיים כאשר שני הווקטורים באותו הכיוון.

פתרון 9.1

כדי לענות על שאלה זאת נרצה שהביטוי $\alpha d^T (-\nabla f(x))$ יהיה גדול ככל האפשר.

נשים לב כי מדובר במכפלה פנימית ולכן נשתמש באי שוויון קושי שזורץ

$$\alpha d^T (-\nabla f(x)) = \langle \alpha d, -\nabla f(x) \rangle \leq \|\alpha d\| \cdot \|-\nabla f(x)\| = \alpha \|\nabla f(x)\|$$

לכן הכיוון d עבורו נקבל את הירידה הכי גדולה בערך של $f(x)$ הוא

$$d = -\frac{\nabla f(x)}{\|\nabla f(x)\|}$$

מסקנה

הכיוון השלילי של הגרדיאנט, $d \propto -\nabla f(x)$, הוא הכיוון שמוביל לירידה המקסימלית בערך הפונקציה בסביבה מקומית של x .

הבחנה

כל כיוון d עבורו $d^T (-\nabla f(x)) > 0$ מוביל לירידה בערך הפונקציה $f(x)$.

תרגיל 9.2 - אלגוריתם הגרדיאנט

נתונה בעיית האופטימיזציה הבאה:

$$\arg \min_{\theta} \frac{1}{2}\theta^2 + 5 \sin(\theta)$$

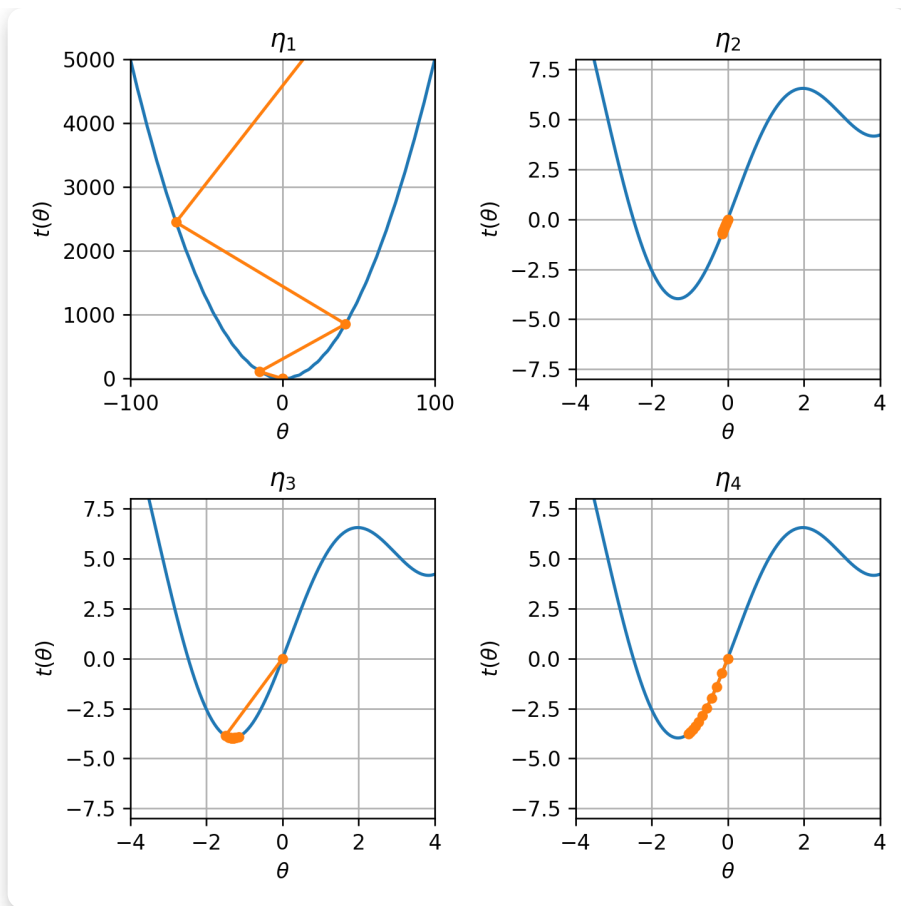
(1) נסו לפתור את הבעיה על ידי גזירה והשוואה ל-0. הגיעו למשוואה (סתומה) אשר מגדירה את נקודות המינימום האפשריות.

(2) רשמו את צעד העידכון של אלגוריתם הגרדיאנט.

(3) חשבו את שלושת צעדי העדכון הראשונים עבור אתחול של $\theta^{(0)} = 0$, וצעד לימוד של $\eta = 0.1$.

(4) חשבו את שלושת צעדי העדכון הראשונים עבור אתחול של $\theta^{(0)} = 2.5$, וצעד לימוד של $\eta = 0.1$. מודע האלגוריתם יתכנס כעת לפתרון אחר מבסעיף הקודם?

(5) הגרפים הבאים מציגים עשר איטרציות של gradient descent בעבור ארבעה ערכים שונים של גודל צעד: $\eta = \{0.003, 0.03, 0.3, 3\}$. התאם בין גודל הצעד לגרפים.

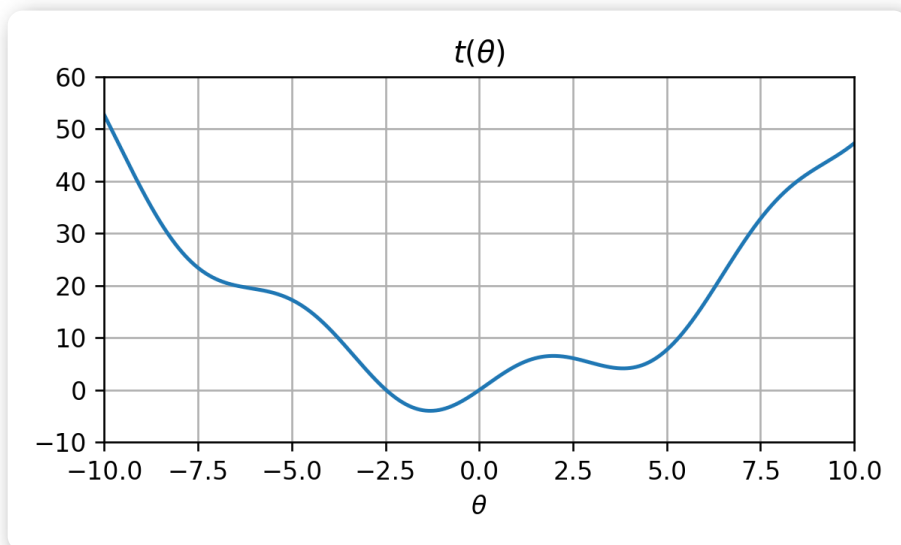


פתרון 9.2

(1)

נסמן את ה objective (פונקציית המטרה) של בעיית האופטימיזציה ב:

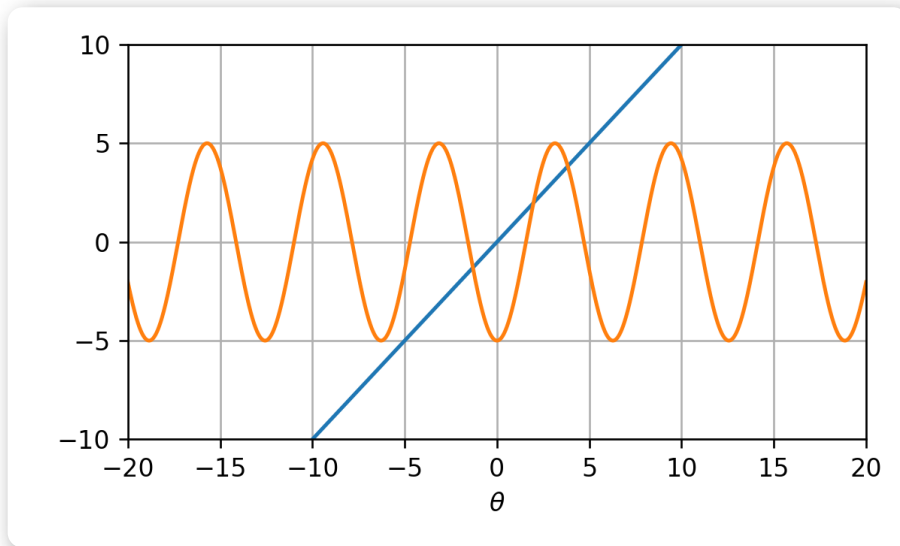
$$t(\theta) = \frac{1}{2}\theta^2 + 5 \sin(\theta)$$



נגזור אותו ונשווה אותו ל-0:

$$\begin{aligned} \frac{\partial}{\partial \theta} t(\theta) &= 0 \\ \Leftrightarrow \theta + 5 \cos(\theta) &= 0 \\ \Leftrightarrow \theta &= -5 \cos(\theta) \end{aligned}$$

בפועל זה אומר שעלינו למצוא את נקודות החיתוך של הפונקציות הבאות:



למשוואה זו אין פתרון אנליטי.

(2)

צעד העדכון של הגרדיאנט יהיה:

$$\theta^{(t+1)} = \theta^{(t)} - \eta \frac{\partial}{\partial \theta} t(\theta) = \theta^{(t)} - \eta (\theta^{(t)} + 5 \cos(\theta^{(t)}))$$

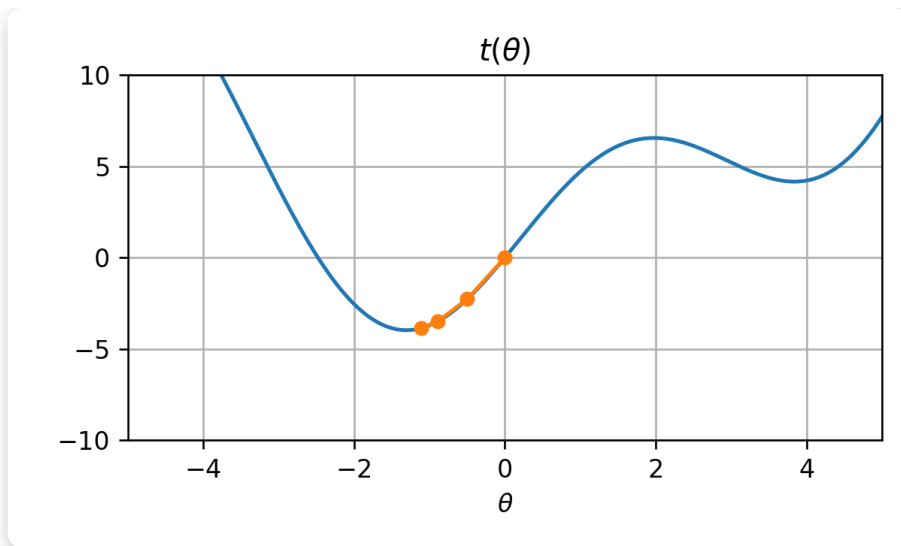
(3)

נאתחל את האלגוריתם עם $\theta^{(0)} = 0$ ונבצע שלושה צעדים (עם $\eta = 0.1$):

$$\theta^{(1)} = \theta^{(0)} - \eta (\theta^{(0)} + 5 \cos(\theta^{(0)})) = 0 - 0.1 (0 + 5 \cos(0)) = -0.5$$

$$\theta^{(2)} = \theta^{(1)} - \eta (\theta^{(1)} + 5 \cos(\theta^{(1)})) = -0.5 - 0.1 (-0.5 + 5 \cos(-0.5)) = -0.889$$

$$\theta^{(3)} = \theta^{(2)} - \eta (\theta^{(2)} + 5 \cos(\theta^{(2)})) = -0.889 - 0.1 (-0.889 + 5 \cos(-0.889)) = -1.115$$



(נקודת האופטימום האמיתי הינה $\theta = -1.30644$)

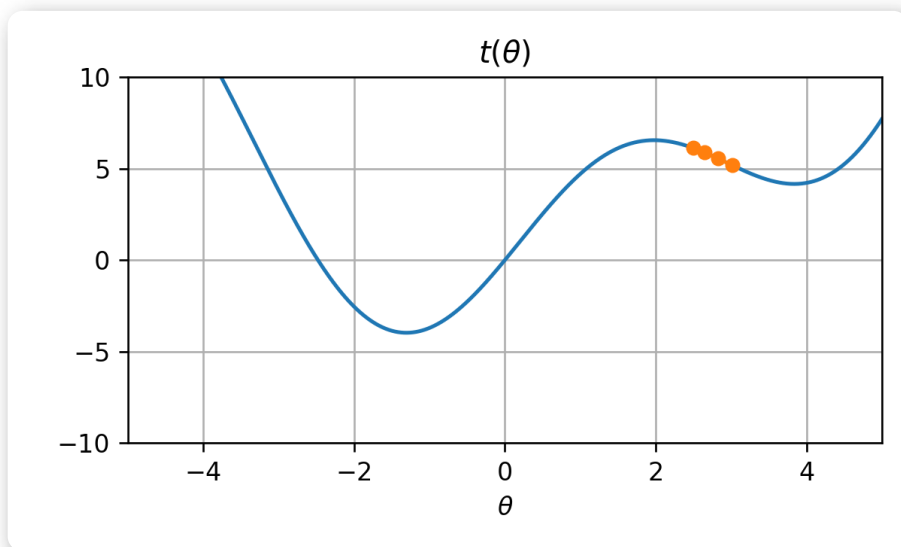
(4)

נחזור על הפתרון עם אתחול של $\theta^{(0)} = 2.5$ ונבצע שלושה צעדים:

$$\theta^{(1)} = 2.65$$

$$\theta^{(2)} = 2.83$$

$$\theta^{(2)} = 3.02$$



בעבור האתחול הזה אלגוריתם יתכנס לפתרון אחר מאשר הפתרון בסעיף הקודם. זאת כמובן משום ש gradient descent מתכנס למינימום לוקאלי, לכן בעבור איתחולים שונים האלגוריתם עלול להתכנס לפתרונות שונים.

(5)

הפרמטר η קובע כאמור את גודל הצעד.

- גודל צעד גדול מידי עשוי להרחיק בכל צעד את האלגוריתם מנקודת המינימום, כפי שקורה במקרה של η_1 . גודל הצעד שמתאים למקרה זה הינו הערך הגדול ביותר, זאת אומרת 3.
- גודל הצעד השני הכי גדול הינו 0.3 והוא מתאים ל η_3 . במקרה זה הצעדים עושים "over shoot" ועוברים במרבית הפעמים את המינימום אך עדיין מתקרבים אליו בכל צעד.
- גודל הצעד השלישי הכי גדול הינו 0.03 הוא מתאים ל η_4 . כאן האופטימיזציה מתקדמת לאט לאט באופן עקבי לכיוון המינימום.
- גודל הצעד הקטן ביותר הינו 0.003 והוא מתאים ל η_2 במקרה זה ההתקדמות היא מאד איטית ויקח לאלגוריתם מספר רב של צעדים על מנת להתקרב למינימום.

תרגיל 9.3 - צעד העדכון של logistic regression

1 בעבור המקרה של רגרסיה לוגיסטית בינארית, הראו כי ניתן לרשום את המודל של פונקציית ההסתברות המותנית באופן הבא:

$$p_{y|x}(y|\mathbf{x}; \boldsymbol{\theta}) = \sigma((-1)^{y+1} f(\mathbf{x}; \boldsymbol{\theta}))$$

2 נסתכל על אלגוריתם gradient descent אשר מנסה למצוא פיתרון לבעיית ה MLE בעבור רגרסיה לוגיסטית בינארית. הראו שניתן לרשום את צעד העדכון של האלגוריתם באופן הבא:

$$\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} - \eta \sum_{i=1}^N (1 - p_{y|x}(y^{(i)}|\mathbf{x}^{(i)}; \boldsymbol{\theta}^{(t)})) (-1)^{y^{(i)}} \nabla_{\boldsymbol{\theta}} f(\mathbf{x}^{(i)}; \boldsymbol{\theta}^{(t)})$$

3 ננסה לתת פרשנות אינטואיטיבית לתפקיד של האיברים השונים בצעד העדכון מהסעיף הקודם.

נתחיל בכך שנתעלם מהביטוי $(1 - p_{y|x}(y^{(i)}|\mathbf{x}^{(i)}; \boldsymbol{\theta}))$. ונקבל את צעד העדכון הבא:

$$\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} - \eta \sum_{i=1}^N (-1)^{y^{(i)}} \nabla_{\boldsymbol{\theta}} f(\mathbf{x}^{(i)}; \boldsymbol{\theta}^{(t)})$$

הסבירו כיצד ישפיע כל צעד עידכון על הפונקציה $f(\mathbf{x}; \boldsymbol{\theta})$. ספציפית הסבירו מה יקרה לערך של הפונקציה בנקודות $\mathbf{x}^{(i)}$? התייחסו להשפעה השונה יש לדגימות עם $y = 1$ ולדגימות עם $y = 0$ מהמדגם.

4 נחזיר כעת את האיבר $(1 - p_{y|x}(y^{(i)}|\mathbf{x}^{(i)}; \boldsymbol{\theta}))$. למה שווה איבר זה במקרים בהם המודל הפרמטרי נותן הסתברות גבוהה לדגימה כלשהי $\{\mathbf{x}^{(i)}, y^{(i)}\}$ ולמה הוא שווה במקרים בהם המודל נותן הסתברות נמוכה לדגימה כלשהי?

התייחסו לאיבר זה כאל איבר מישקול, אשר נותן משקל שונה לכל דגימה מהמדגם. הסבירו מה תהיה ההשפעה של משקול זה על צעד העדכון.

5 (לקריאה עצמית) נרחיב את הדוגמא למקרה הלא בינארי. הראו שניתן לכתוב את צעד העדכון של אלגוריתם ה gradient descent במקרה הלא בינארי באופן הבא:

$$\boldsymbol{\theta}_c^{(t+1)} = \boldsymbol{\theta}_c^{(t)} + \eta \sum_{i=1}^N (\delta_{y^{(i)},c} - p_{y|x}(c|\mathbf{x}^{(i)}; \boldsymbol{\theta}^{(t)})) \nabla_{\boldsymbol{\theta}_c} f_c(\mathbf{x}^{(i)}; \boldsymbol{\theta}_c^{(t)}) \quad \forall c$$

הסבירו את התפקיד של $\nabla_{\boldsymbol{\theta}_c} f_c(\mathbf{x}; \boldsymbol{\theta}_c^{(t)})$ ושל $(\delta_{y,c} - p_{y|x}(c|\mathbf{x}^{(i)}; \boldsymbol{\theta}^{(t)}))$ בצעד העדכון.

פתרון 9.3

1

נשחק מעט עם הצורה של המודל הפרמטרי בכדי להגיע לצורה אותה בקשו בשאלה:

$$\begin{aligned}
p_{y|x}(y|\mathbf{x};\boldsymbol{\theta}) &= \begin{cases} \sigma(f(\mathbf{x};\boldsymbol{\theta})) & y = 1 \\ 1 - \sigma(f(\mathbf{x};\boldsymbol{\theta})) & y = 0 \end{cases} \\
&= \begin{cases} \sigma(f(\mathbf{x};\boldsymbol{\theta})) & y = 1 \\ \sigma(-f(\mathbf{x};\boldsymbol{\theta})) & y = 0 \end{cases} \\
&= \sigma((-1)^{y+1} f(\mathbf{x};\boldsymbol{\theta}))
\end{aligned}$$

(2)

נציב את המודל הפרמטרי כפי שרשמנו אותו בסעיף הקודם:

$$\begin{aligned}
\boldsymbol{\theta}^* &= \arg \min_{\boldsymbol{\theta}} - \sum_{i=1}^N \log(p_{y|x}(y^{(i)}|\mathbf{x}^{(i)};\boldsymbol{\theta})) \\
&= \arg \min_{\boldsymbol{\theta}} - \underbrace{\sum_{i=1}^N \log(\sigma((-1)^{y^{(i)}+1} f(\mathbf{x}^{(i)};\boldsymbol{\theta})))}_{\triangleq t(\boldsymbol{\theta})}
\end{aligned}$$

לשם הנוחות, סימנו את ה objective של בעיית האופטימיזציה ב $t(\boldsymbol{\theta})$. נחשב את הגרדיאנט של ה objective ונסה להביא אותו לצורה דומה לזו שבקשו בשאלה:

$$\begin{aligned}
\nabla_{\boldsymbol{\theta}} t(\boldsymbol{\theta}) &= - \sum_{i=1}^N \nabla_{\boldsymbol{\theta}} \log(\sigma((-1)^{y^{(i)}+1} f(\mathbf{x}^{(i)};\boldsymbol{\theta}))) \\
&= - \sum_{i=1}^N (1 - \sigma((-1)^{y^{(i)}+1} f(\mathbf{x}^{(i)};\boldsymbol{\theta}))) \nabla_{\boldsymbol{\theta}}((-1)^{y^{(i)}+1} f(\mathbf{x}^{(i)};\boldsymbol{\theta})) \\
&= \sum_{i=1}^N (1 - p_{y|x}(y^{(i)}|\mathbf{x}^{(i)};\boldsymbol{\theta})) (-1)^{y^{(i)}} \nabla_{\boldsymbol{\theta}} f(\mathbf{x}^{(i)};\boldsymbol{\theta})
\end{aligned}$$

צעד העדכון של אלגוריתם ה gradient descent יהיה אם כן:

$$\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} - \eta \sum_{i=1}^N (1 - p_{y|x}(y^{(i)}|\mathbf{x}^{(i)};\boldsymbol{\theta}^{(t)})) (-1)^{y^{(i)}} \nabla_{\boldsymbol{\theta}} f(\mathbf{x}^{(i)};\boldsymbol{\theta}^{(t)})$$

(3)

נתייחס לצעד עדכון מהצורה של:

$$\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} - \eta \sum_{i=1}^N (-1)^{y^{(i)}} \nabla_{\boldsymbol{\theta}} f(\mathbf{x}^{(i)};\boldsymbol{\theta}^{(t)})$$

נסתכל על התרומה של הדגימות מהמדגם ששייכים למחלקה $y = 1$ (אשר גורר: $(-1)^y = -1$). איברים אלו ינסו לשנות את $\boldsymbol{\theta}$ בכיוון הגרדיאנט בנקודות $\mathbf{x}^{(i)}$ ששייכות למחלקה. זאת אומרת, שהם ינסו לגרום לשינוי של הפרמטרים כך שהערך של הפונקציה הפרמטרית $f(\mathbf{x};\boldsymbol{\theta})$ בנקודות $\mathbf{x}^{(i)}$ יהיה גדול יותר.

באופן הפוך, התרומה של הדגימות מהמחלקה $y = 0$ (ו $(-1)^y = 1$) תהיה לנסות ולעדכן את $\boldsymbol{\theta}$ בכיוון ההפוך מהגרדיאנט. זאת אומרת, שהם ינסו להקטין את הערך של $f(\mathbf{x};\boldsymbol{\theta})$ בנקודות $\mathbf{x}^{(i)}$ מהמחלקה $y = 0$.

בסה"כ הכל נקבל שהאלגוריתם ינסה בכל צעד לשנות את $f(\mathbf{x};\boldsymbol{\theta})$ כך שיניב ערכים גבוהים על הנקודות \mathbf{x} שמתאימות ל $y = 1$ וערכים נמוכים על הנקודות שמתאימות ל $y = 0$. התנהגות זו הגיונית משום שזה בדיוק מה שאנחנו רוצים מהמודל שלנו, שאמור לחזות את ההסתברות ש $y = 1$ בהינתן \mathbf{x} . משום שהסתברות זו שווה ל $\sigma(f(\mathbf{x};\boldsymbol{\theta}))$ אנו רוצים רוצים ש f יניב ערכים גבוהים באיזורים שבהם $y = 1$ בסבירות גבוהה בהינתן \mathbf{x} וערכים נמוכים בשאר המקומות.

(4)

נסתכל על הביטוי $(1 - p_{y|x}(y^{(i)}|\mathbf{x}^{(i)}; \theta))$. נזכור שההסתברות היא מספר בין 0 ל 1. האיבר כולו יהיה לכן קרוב ל-0 כאשר הסתברות של y מסויים בהינתן \mathbf{x} היא גבוהה והוא יהיה קרוב ל-1 כאשר ההסתברות נמוכה.

נזכור גם כי $p_{y|x}(y^{(i)}|\mathbf{x}^{(i)}; \theta)$ היא אינה ההסתברות האמיתית של \mathbf{x} ו y אלא ההסתברות שהמודל שלנו נותן לדגימה כלשהי מהמדגם. (אנו רוצים שבסופו של דבר שמודל זה יהיה קרוב להסתברות האמיתית). היינו מעוניינים שכל שנעשה יותר צעדי עדכון המודל יגדיל לאט לאט את ההסתברות שהוא נותן לדגימות במדגם (זו בעצם המטרה של MLE ו MAP).

נסתכל על איבר זה כעל משקל בין 0 ל 1 שמשוייך לכל דגימה במדגם. לדגימות שהמודל חושב הם סבירות הוא נותן משקל קרוב ל 0 ולדגימות שהמודל נותן להם סבירות נמוכה הוא נותן משקל 1. מה שאיבר זה עושה לצעד העידכון הוא לגרום לו יחסית להתעלם מדגימות שכבר מקבלות סבירות גבוהה ולהתמקד בדגימות שהוא עדיין "טועה" עליהם, זאת אומרת, שהוא נותן להם הסתברות נמוכה.

5

בעיית ה MLE הינה:

$$\begin{aligned} \theta^* &= \arg \min_{\theta} - \sum_{i=1}^N \log \left(p_{y|x}(y^{(i)}|\mathbf{x}^{(i)}; \theta) \right) \\ &= \arg \min_{\theta} - \underbrace{\sum_{i=1}^N \log \left(\text{softmax}(\mathbf{f}(\mathbf{x}^{(i)}; \theta))_{y^{(i)}} \right)}_{\triangleq t(\theta)} \end{aligned}$$

נחשב את הגרדיאנט של ה objective:

$$\begin{aligned} \nabla_{\theta_c} t(\theta) &= - \sum_{i=1}^N \nabla_{\theta_c} \log \left(\text{softmax}(\mathbf{f}(\mathbf{x}^{(i)}; \theta))_{y^{(i)}} \right) \\ &= - \sum_{i=1}^N \left(\delta_{y^{(i)},c} - \text{softmax}(\mathbf{f}(\mathbf{x}^{(i)}; \theta))_c \right) \nabla_{\theta_c} f_c(\mathbf{x}^{(i)}; \theta) \\ &= - \sum_{i=1}^N \left(\delta_{y^{(i)},c} - p_{y|x}(c|\mathbf{x}^{(i)}; \theta) \right) \nabla_{\theta_c} f_c(\mathbf{x}^{(i)}; \theta) \end{aligned}$$

צעד העדכון יהיה:

$$\theta_c^{(t+1)} = \theta_c^{(t)} + \eta \sum_{i=1}^N \left(\delta_{y^{(i)},c} - p_{y|x}(c|\mathbf{x}^{(i)}; \theta^{(t)}) \right) \nabla_{\theta_c} f_c(\mathbf{x}^{(i)}; \theta^{(t)}) \quad \forall c$$

האיבר $\left(\delta_{y^{(i)},c} - p_{y|x}(c|\mathbf{x}^{(i)}; \theta^{(t)}) \right)$ הוא חיובי כאשר $c = y^{(i)}$ ושילולי אחרת. איבר זה גורם לכך שבעבור כל דגימה מהמדגם צעד העדכון ינסה לגדיל את הפונקציה הפרמטרית $f_c(\mathbf{x}^{(i)}; \theta_c^{(t)})$ שבה $c = y^{(i)}$ ויקטין את הפונקציות הפרמטריות שבהם $c \neq y^{(i)}$.

בדומה למקרה הבינארי ככל שהסבירות של הדגימה במדגם $p_{y|x}(y^{(i)}|\mathbf{x}^{(i)}; \theta)$ כך ההשפעה של הדגימה על העדכון יהיה קטן יותר.

תרגיל 9.4 - MLE and KL divergence

בתרגיל זה נציג דרך אחרת לפתח את בעיית האופטימיזציה של משערך ה MLE.

נתון לנו מדגם של N דגימות i.i.d. של משתנה אקראי כלשהו x :

$$\mathcal{D} = \{\mathbf{x}^{(i)}\}_{i=1}^N$$

ומודל פרמטרי כל שהוא $p_x(x; \theta)$. נרצה לבחור את הפרמטרים של המודל θ כך שהמודל יתאר בצורה טובה את הדגימות במדגם.

לשם כך נשתמש במדד הבא אשר מודד עד כמה פונקציית צפיפות הסתברות כלשהי $q_x(x)$ תהיה טובה בכדי לתאר דגימות המגיעות מצפיפות הסתברות אחרת $p_x(x)$. המדד נקרא Kullback-Leibler divergence והוא מוגד באופן הבא:

$$D_{KL}(p_x(x)||q_x(x)) = \int p_x(x) \log \left(\frac{p_x(x)}{q_x(x)} \right) = \mathbb{E}_{(p)} \left[\log \left(\frac{p_x(x)}{q_x(x)} \right) \right]$$

הסימון $\mathbb{E}_{(p)}$ הוא תוחלת לפי הפילוג p_x . מדד זה מגיע מתורת האינפורמציה ואנו לא ניכנס למשמעות ולמקור של מדד זה. ככל שהמדד נמוך יותר כך הפילוגים קרובים יותר.

השתמשו במדד זה על מנת להגדיר בעיית אופטימיזציה שבוחרת את הפרמטרים של המודל כפרמטרים כך שהם ממזערים את ה Kullback-Leibler divergence בין המודל הפרמטרי לפילוג האמיתי. בכדי להיפטר מהתוחלת על הפילוג הלא ידוע החליפו אותו בתוחלת אמפירית על המדגם. הראו כי בעיית האופטימיזציה המתקבלת זהה לזו של משעריך ה MLE.

פתרון 9.4

נסמן את הפילוג האמיתי (הלא ידוע) של x ב $p_x(x)$ (בלי θ). בעיית האופטימיזציה שהיינו רוצים לפתור הינה:

$$\begin{aligned} \theta^* &= \arg \min_{\theta} D_{KL}(p_x(x)||p_x(x; \theta)) \\ &= \arg \min_{\theta} \mathbb{E}_{(p)} \left[\log \left(\frac{p_x(x)}{p_x(x; \theta)} \right) \right] \\ &= \arg \min_{\theta} \mathbb{E}_{(p)} [\log(p_x(x))] - \mathbb{E}_{(p)} [\log(p_x(x; \theta))] \\ &= \arg \min_{\theta} - \mathbb{E}_{(p)} [\log(p_x(x; \theta))] \end{aligned}$$

נחליף את התוחלת בתוחלת אמפירית על המדגם:

$$\theta^* = \arg \min_{\theta} - \frac{1}{N} \sum_{i=1}^N \log(p_x(x^{(i)}; \theta))$$

שזה בדיוק המינימיזציה של ה log-likelihood עד כדי החלוקה ב N שלא משנה את בעיית האופטימיזציה.

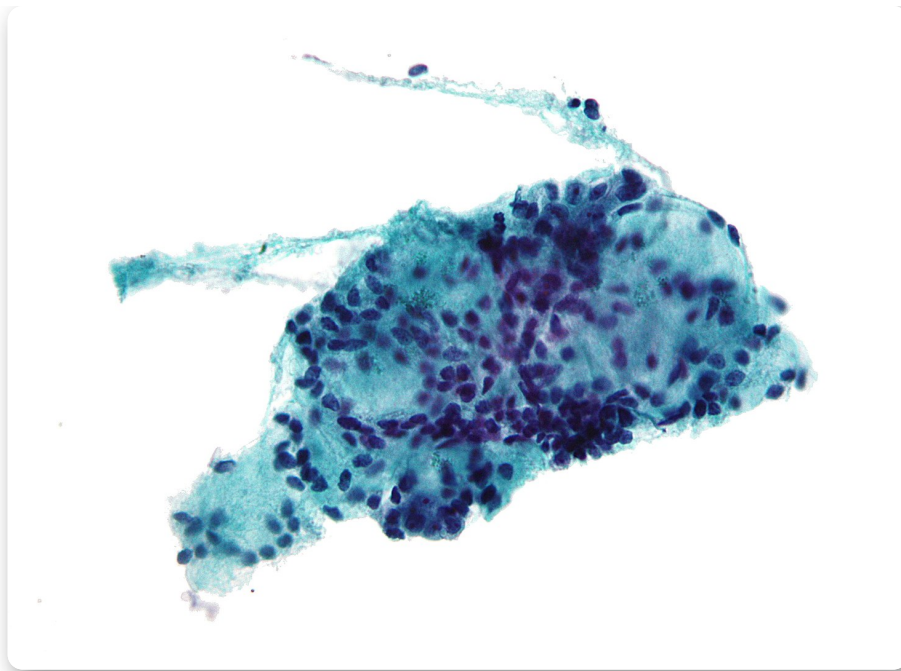
תרגיל מעשי - איבחון סרטן שד

Code

שיטה נפוצה כיום לאבחון של סרטן הינה בשיטת Fine-needle aspiration. בשיטה זו נלקחת דגימה של רקמה בעזרת מחט ומבוצעת אנליזה בעזרת מיקרוסקופ על מנת לאבחן שני מקרים:

- Malignant - רקמה סרטנית
- or Benign - רקמה בריאה

להלן דוגמא לתמונת מיקרוסקופ של דגימה שכזו:



בתרגול זה נעבוד עם מדגם בשם **Breast Cancer Wisconsin Diagnostic** אשר נאסף על ידי חוקרים מאוניברסיטת ויסקונסין. הוא כולל 30 ערכים מספריים, כגון שטח התא הממוצע והרדיוס הממוצע, אשר חושבו בעבור 569 דגימות שונות. בנוסף יש לכלל דגימה במדגם תווית של האם הדגימה הינה סרטנית או לא.

את המדגם המקורי ניתן למצוא פה: [Breast Cancer Wisconsin \(Diagnostic\) Data Set](#), אנחנו נשתמש בגרסה מעט מעובדת שלו הנמצאת פה.

נציג כמה עמודות ושורות מייצגות מהמדגם:

	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness
0	M	17.99	10.38	122.8	1001	0.1184	
1	M	20.57	17.77	132.9	1326	0.08474	
2	M	19.69	21.25	130	1203	0.1096	
3	M	11.42	20.38	77.58	386.1	0.1425	
4	M	20.29	14.34	135.1	1297	0.1003	
5	M	12.45	15.7	82.57	477.1	0.1278	
6	M	18.25	19.98	119.6	1040	0.09463	
7	M	13.71	20.83	90.2	577.9	0.1189	
8	M	13	21.82	87.5	519.8	0.1273	
9	M	12.46	24.04	83.97	475.9	0.1186	

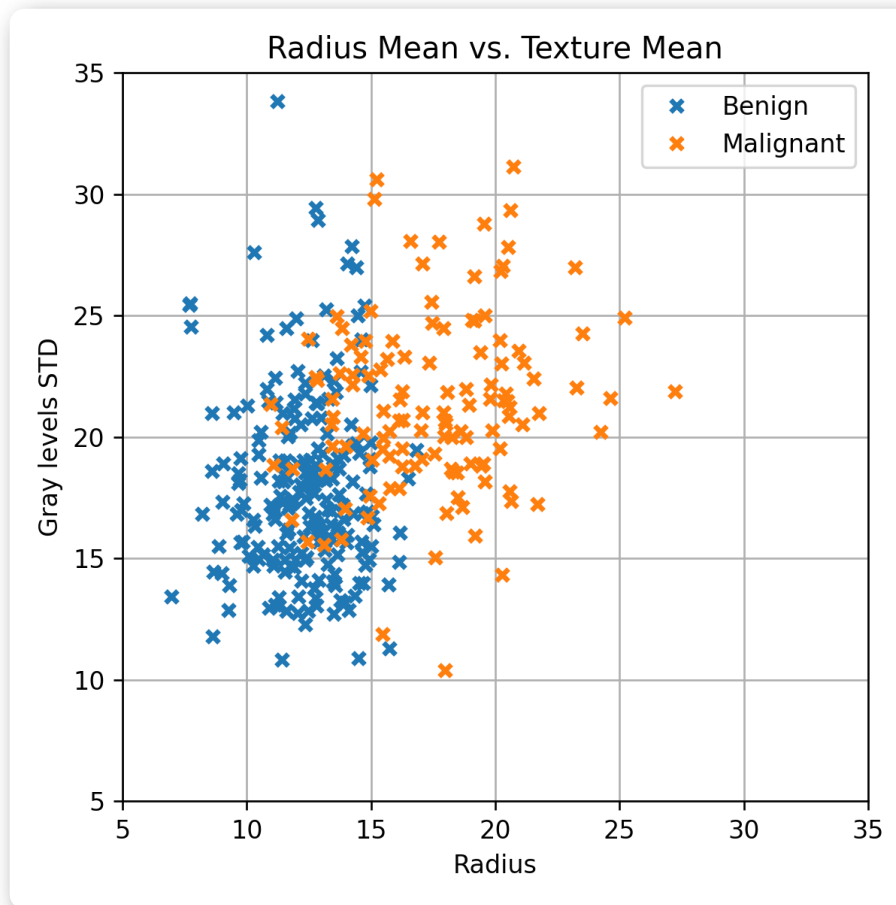
רק לשם המחשה נתחיל בניסיון לחזות האם הרקמה סרטנית או לא רק על פי שני השדות הראשונים:

- **radius_mean** - רדיוס התא הממוצע בדגימה.
- **texture_mean** - סטיית התקן הממוצעת של רמת האפור בצבע של כל תא בדגימה.

השדה של התוויות y הינו:

• **diagnosis** - התוויות של הדגימה: M = malignant (סרטני), B = benign (בריא)

(בחרנו להתחיל עם 2 שדות משום שמעבר לכך כבר לא נוכל לשרטט את הפילוג של הדגימות ואת החיזוי).



נרצה למצוא חזאי אשר יפריד בין הנקודות הכתומות לנקודות הכחולות. לשם כך נפצל את המדגם ל 60% train / 20% validation / 20% test. נתאים שלושה מודלים: LDA, QDA, linear logistic regression.

LDA

נחשב את פרמטרים של המודל:

$$p_y(0) = \frac{|\mathcal{I}_0|}{N} = 0.37$$

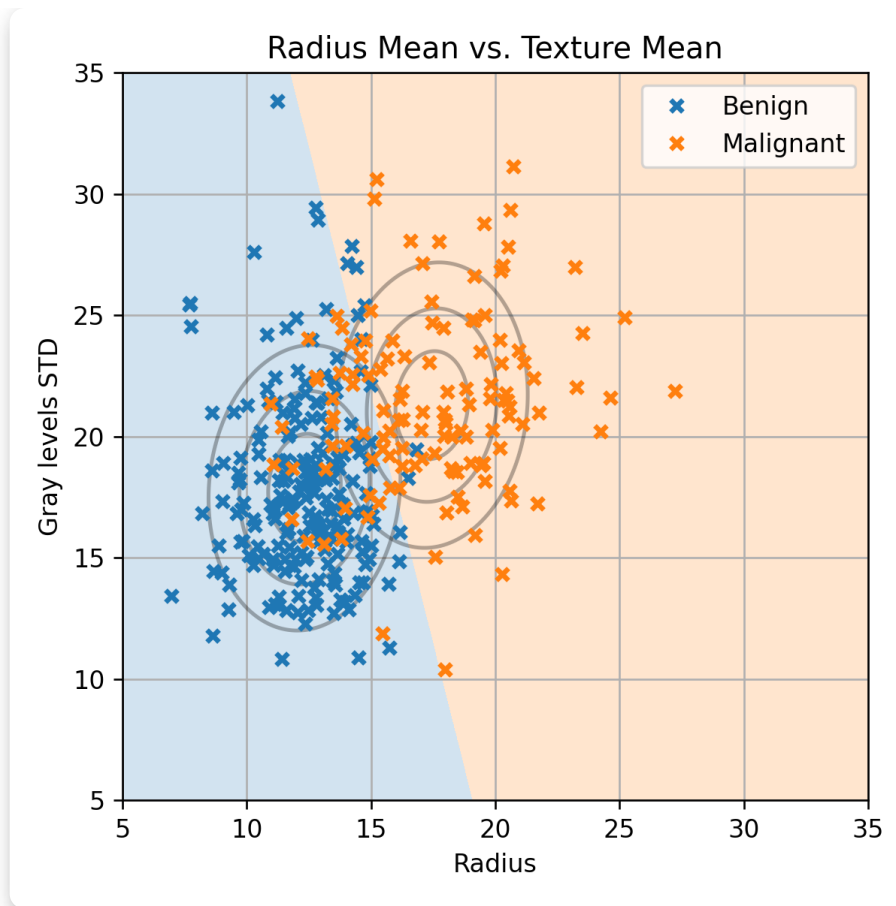
$$p_y(1) = \frac{|\mathcal{I}_1|}{N} = 0.63$$

$$\boldsymbol{\mu}_0 = \frac{1}{|\mathcal{I}_0|} \sum_{i \in \mathcal{I}_0} \mathbf{x}^{(i)} = [12.3, 17.9]^T$$

$$\boldsymbol{\mu}_1 = \frac{1}{|\mathcal{I}_1|} \sum_{i \in \mathcal{I}_1} \mathbf{x}^{(i)} = [17.5, 21.3]^T$$

$$\Sigma = \frac{1}{N} \sum_i (\mathbf{x}^{(i)} - \boldsymbol{\mu}_{y^{(i)}}) (\mathbf{x}^{(i)} - \boldsymbol{\mu}_{y^{(i)}})^T = \begin{bmatrix} 5.8 & 0.67 \\ 0.67 & 13.5 \end{bmatrix}$$

פרמטרים אלו יתנו את החיזוי הבא:



נזכיר כי החזאי המתקבל ממודל ה LDA הינו חזאי אשר מחלק את המרחב לשני חלקים על ידי משטח לינארי (במקרה זה קו ישר).

ביצועי חזאי זה על ה validation set (במובן של misclassification rate) הינם: 0.09. זאת אומרת שאנו צפויים לצדוק באבחון ב 91% מהמקרים.

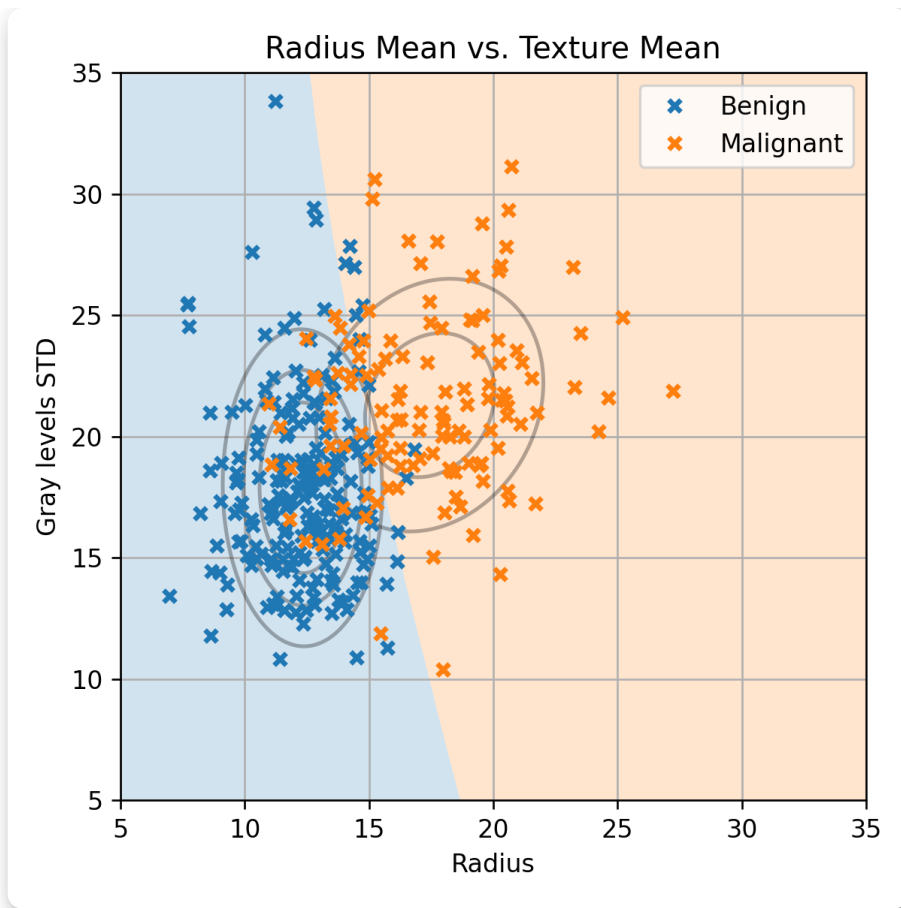
QDA

נחשב את פרמטרים של המודל. הפרמטרים של $p_y(y)$ ו μ_c לא ישתנו. נחשב לכן רק את מטריצות הקווריאנס:

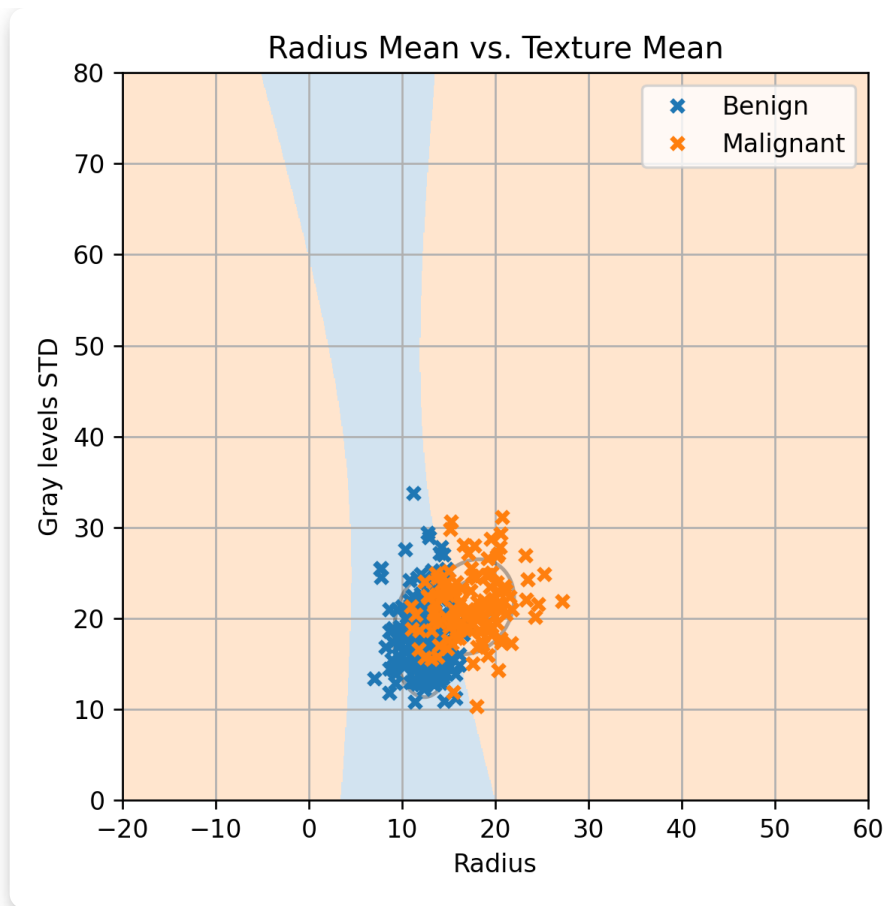
$$\Sigma_0 = \frac{1}{|\mathcal{I}_0|} \sum_{i \in \mathcal{I}_0} (\mathbf{x}^{(i)} - \boldsymbol{\mu}_0) (\mathbf{x}^{(i)} - \boldsymbol{\mu}_0)^T = \begin{bmatrix} 3.3 & -0.13 \\ -0.13 & 13.8 \end{bmatrix}$$

$$\Sigma_1 = \frac{1}{|\mathcal{I}_1|} \sum_{i \in \mathcal{I}_1} (\mathbf{x}^{(i)} - \boldsymbol{\mu}_1) (\mathbf{x}^{(i)} - \boldsymbol{\mu}_1)^T = \begin{bmatrix} 10.2 & 2 \\ 2 & 13.2 \end{bmatrix}$$

פרמטרים אלו יתנו את החיזוי הבא:



החזאי המתקבל ממודל ה QDA מחלק את המרב על ידי משטח ריבועי. בשרטוט זה המשטח אומנם נראה כמעט ישר אך אם נגדיל טיפה את השרטוט נראה שהוא אכן ריבועי:



ביצועי חזאי זה על ה validation set הינם: 0.08. זהו שיפור של 1% מביצועיו של מודל ה LDA.

שימוש בכל 30 העמודות במדגם

אם נחזור על החישוב של מודל ה QDA רק עם כל 30 העמודות שבמדגם נקבל misclassification rate של 0.02.

Linear Logistic Regression

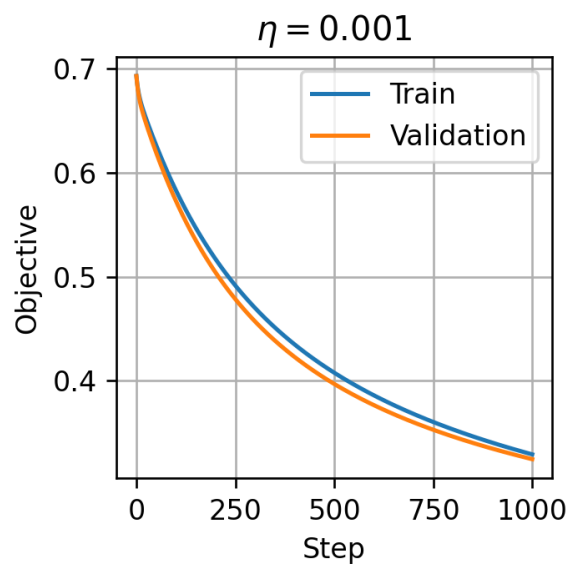
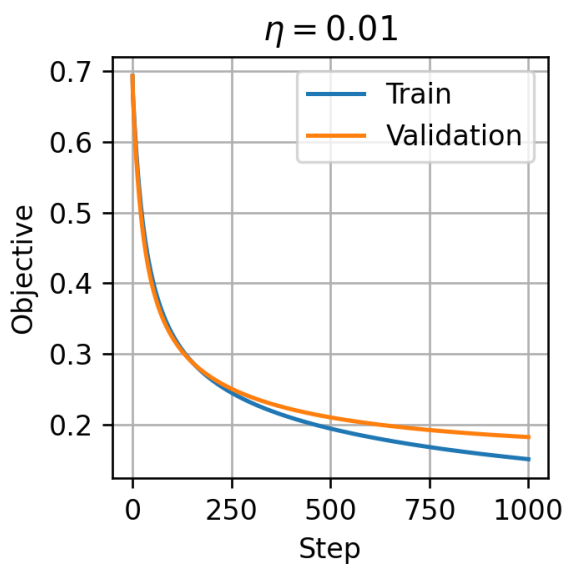
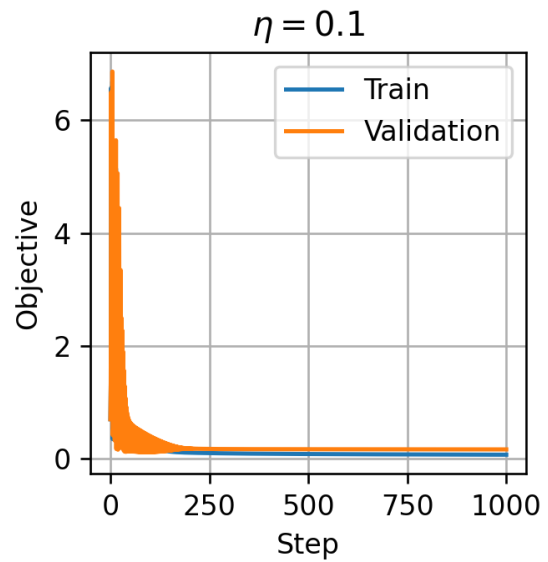
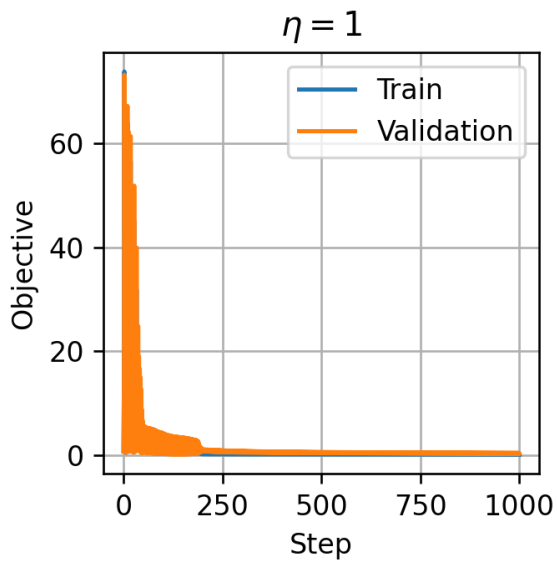
ננסה כעת להתאים מודל של linear logistic regression מהצורה:

$$p_{y|x}(1|\mathbf{x}; \boldsymbol{\theta}) = \sigma(\mathbf{x}^\top \boldsymbol{\theta})$$

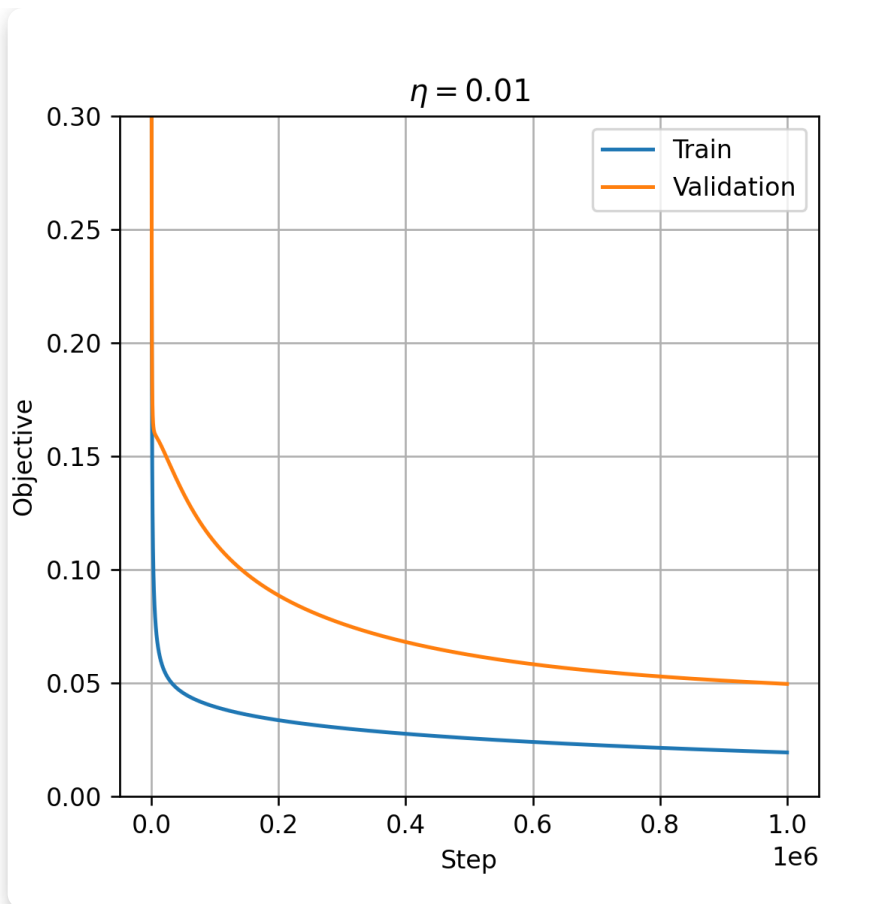
בעיית האופטימיזציה של MLE תהיה:

$$\boldsymbol{\theta}^* = \arg \min_{\boldsymbol{\theta}} - \sum_{i=1}^N I\{y^{(i)} = 1\} \log(\sigma(\mathbf{x}^{(i)\top} \boldsymbol{\theta})) + I\{y^{(i)} = 0\} \log(1 - \sigma(\mathbf{x}^{(i)\top} \boldsymbol{\theta}))$$

נשתמש ב gradient descent על מנת למצוא את הפרמטרים של המודל. בכדי לבחור את גודל הצעד ננסה כמה ערכים שונים ונריץ את האלגוריתם מספר קטן של צעדים (1000) ונבחר את גודל הצעד הגדול ביותר אשר גורם למודל להתכנס. בדוגמא זו נציג את התוצאות בעבור 4 ערכים של גודל הצעד:



בגרפים האלה רואים את החישוב של ה objective על ה train set ועל ה validation set כפונקציה של מספר הצעדים. נשים לב שבעבור בחירה של $\eta = 1$ או $\eta = 0.1$ המודל מתבדר לערכים מאד גדולים וזה ימנע ממנו להתכנס למינימום של הפונקציית המטרה. נבחר אם כן את גודל הצעד להיות $\eta = 0.01$ ונריץ את האלגוריתם מספר רב של צעדים (1000000):



נראה אם כן שגם אחרי מיליון צעדים האלגוריתם עדיין לא התכנס. כפי שציינו זוהי אחת הבעיות העיקריות של אלגוריתם הגרדיאנט בצורתו הפשוטה. למזלנו, ישנן מספר שיטות פשוטות לשפר את האלגוריתם בכדי לפתור בעיה זו אך אנו לא נפרט עליהן בקורס זה.

הביצועים של המודל עם הפרמטרים המתקבלים אחרי מיליון צעדים נותנים misclassification rate של 0.02 שזה דומה לתוצאה שקיבלנו על ידי שימוש ב QDA.

ביצועי המודל על ה test set הינם: 0.04.