

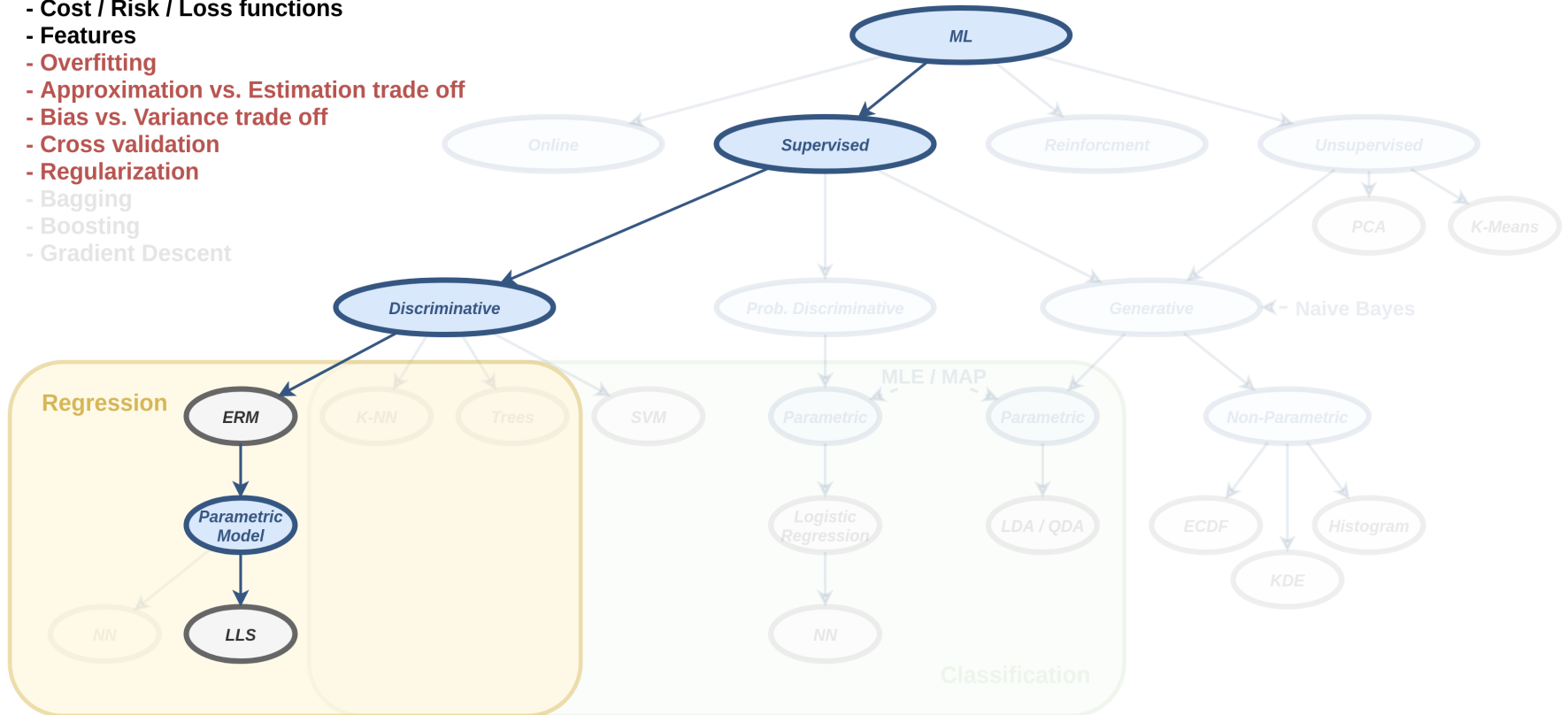
הרצאה 3

Generalization & overfitting

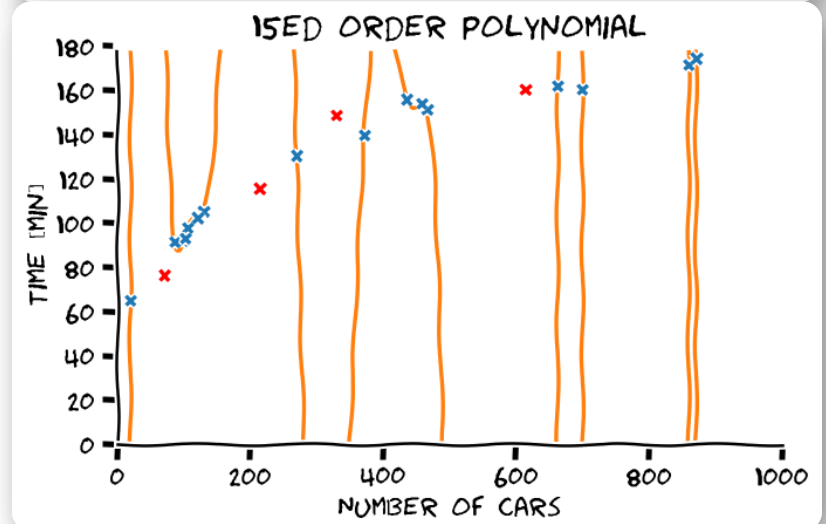
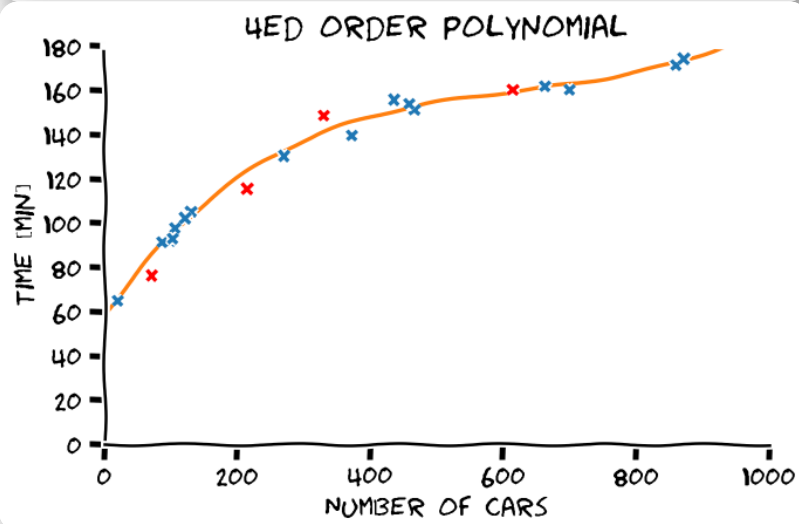
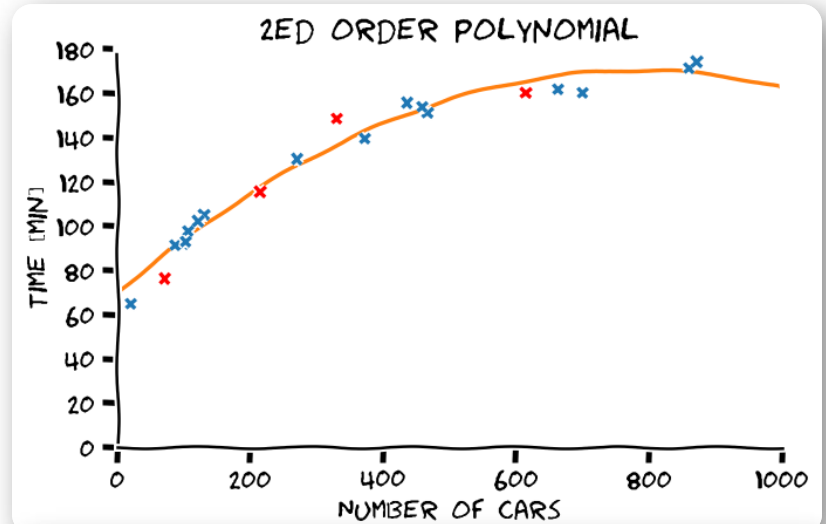
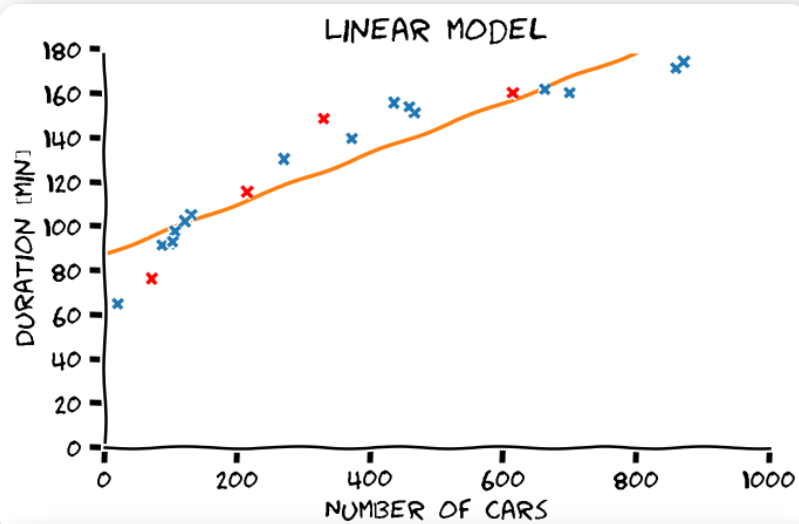
Subjects Covered in this Course

General concepts:

- Cost / Risk / Loss functions
- Features
- Overfitting
- Approximation vs. Estimation trade off
- Bias vs. Variance trade off
- Cross validation
- Regularization
- Bagging
- Boosting
- Gradient Descent



LLS בעבור פולינומים מסדרים שונים



הכללה (generalization)

בעיית הכללה בתחום של מערכות לומדות היא בעיית הכללה, שבה אנו מנסים על סמך דוגמאות להסיק מסקנות לגבי ההתנהגות הכללית של המערכת.

לדוגמא בבעיות supervised learning מטרה שלנו היא לבנות חזאי אשר יוכל לבצע חיזויים טובים על דגימות שלא ראינו לפני.

הערכת הביצועים / יכולת ההכללה של חזאי

- נרצה להעריך את יכולת ההכללה של החזאי שבנינו על דגימות שלא הופיעו בשלב הלימוד.
 - נצטרך מדגם נוסף המכיל דגימות שונות מהמדגם שבו השתמשנו בשלב הלימוד.
 - נקצה חלק מתוך המדגם לטובת הערכת הביצועים.
- נחלק את המדגם שלנו לשני חלקים:
- **Train set** - D_{train} - המדגם שעל פיו אנו נבנה את חזאי.
 - **Test set** - D_{test} - המדגם שבו נשתמש להעריכת ביצועים.

הערכת הביצועים של פונקציית risk

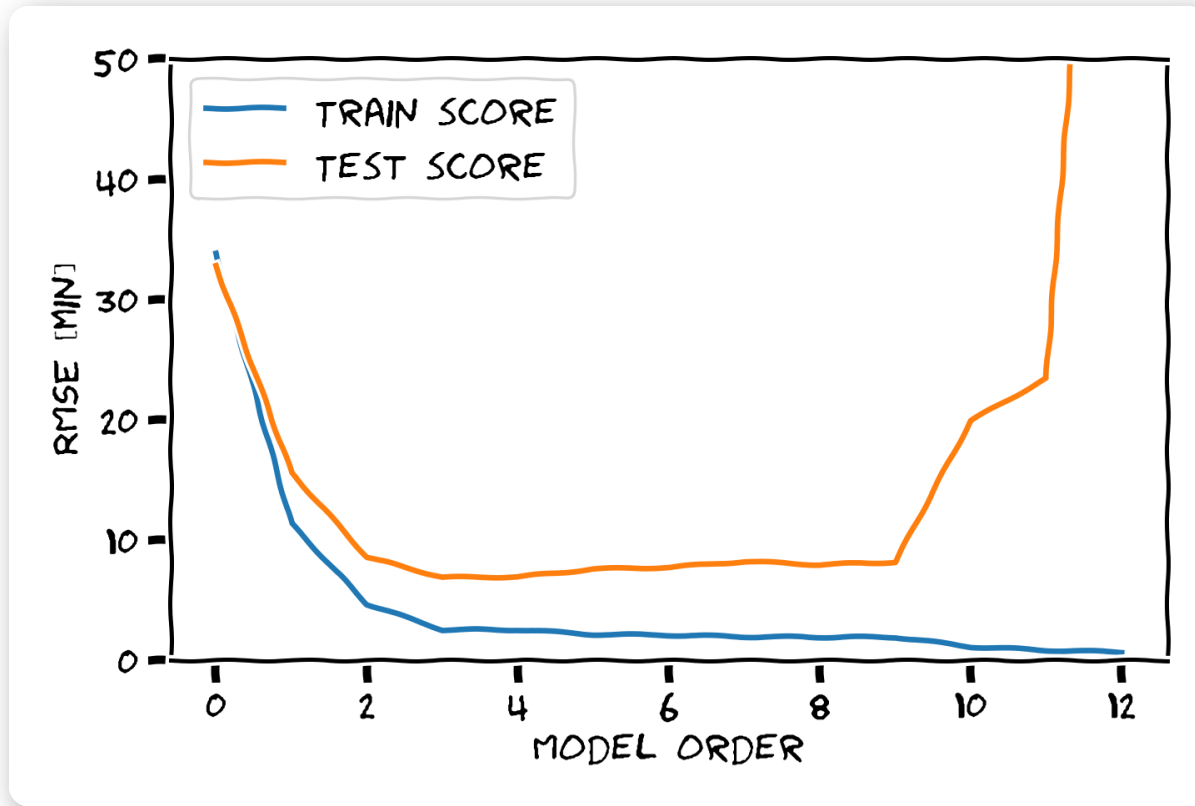
כאשר פונקציית המחיר שלנו היא מהצורה של פונקציית risk, הערכת הביצועים תעשה בעזרת תוחלת אמפירית על ה test set:

$$\text{test cost} = \frac{1}{N_{\text{test}}} \sum_{\mathbf{x}^{(i)}, y^{(i)} \in \mathcal{D}_{\text{test}}} l(h(\mathbf{x}^{(i)}), y^{(i)})$$

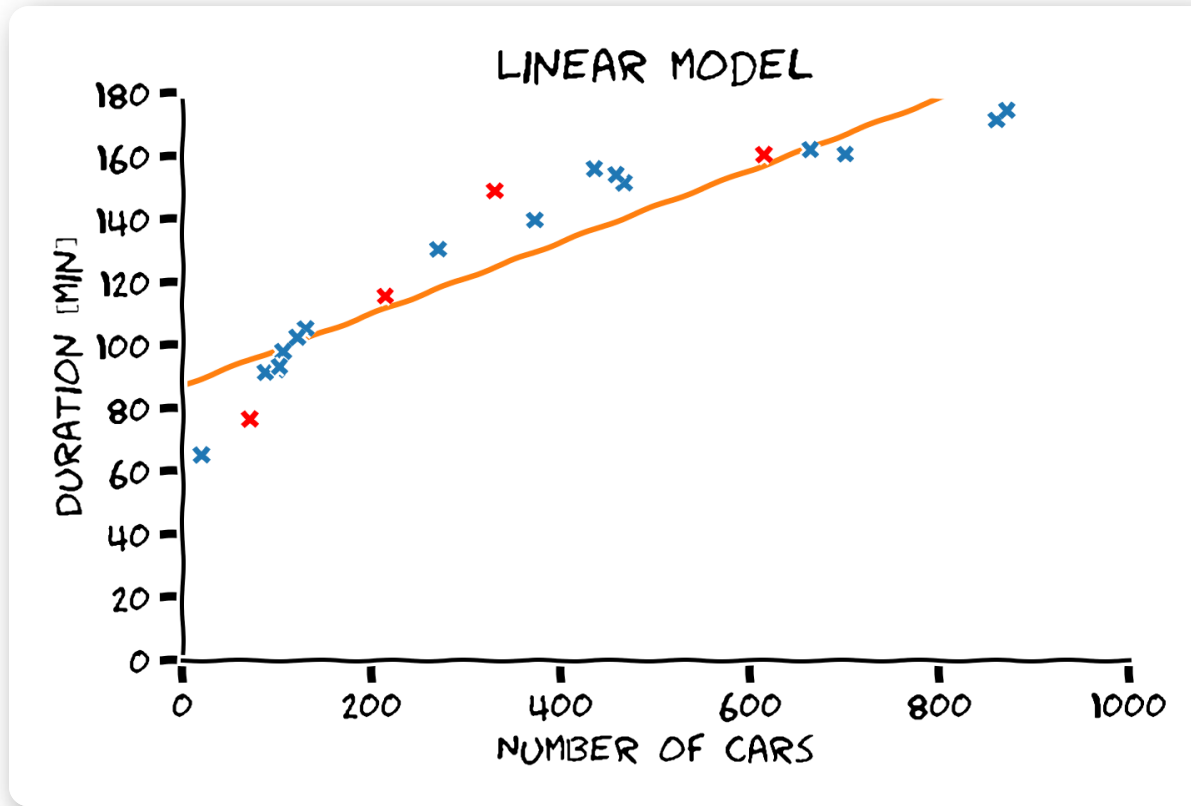
גודלו של ה test set

- אנו נרצה שיהיה גדול מספיק בכדי שההערכה תהיה מדוייקת.
- לא גדול מדי, בכדי לשמור את ה train set כמה שיותר גדול.
- כאשר המדגם לא מאד גדול מקובל לפצל את המדגם ל train 80% | test 20%.

דוגמא: פיצול train-test



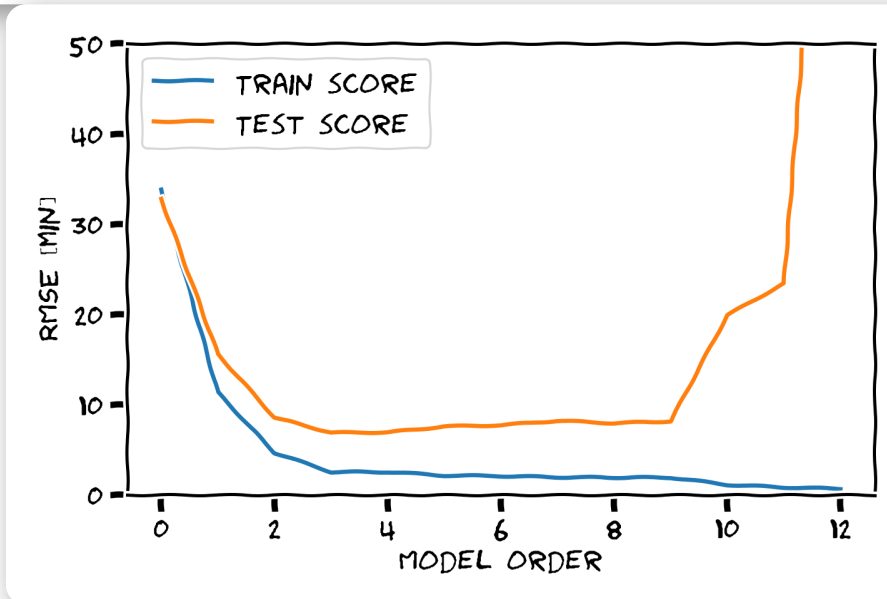
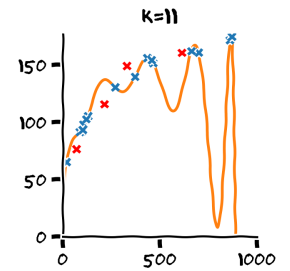
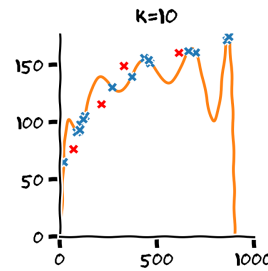
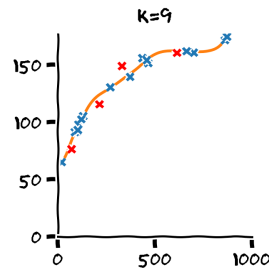
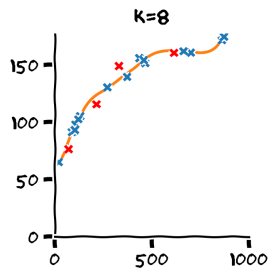
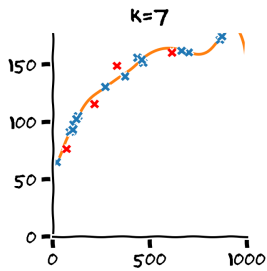
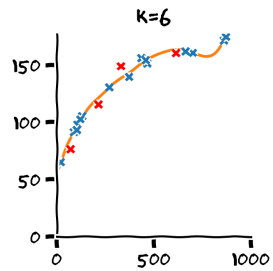
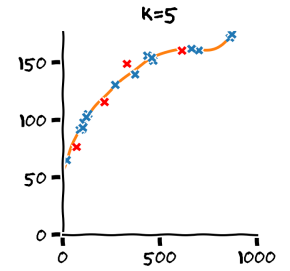
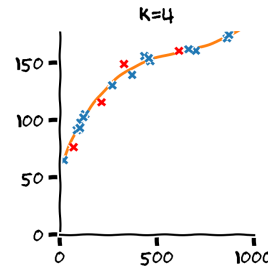
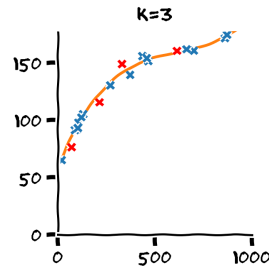
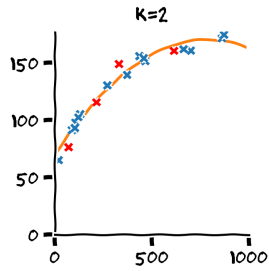
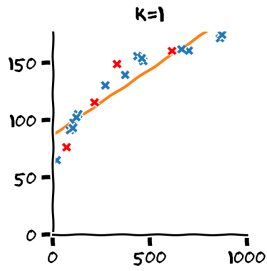
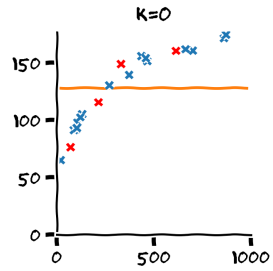
דוגמא: הערכת ביצועים



Train cost (RMSE): 11.34 min •

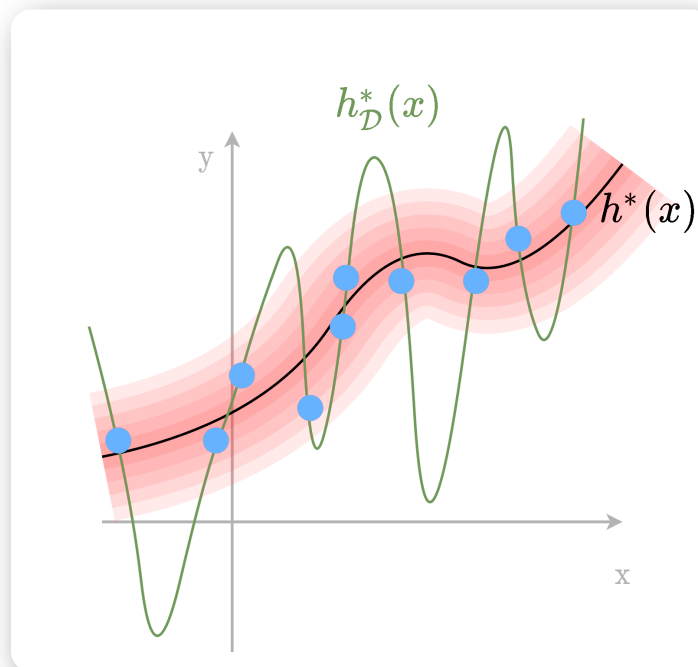
Test cost (RMSE): 15.58 min •

התלות בסדר הפולינום

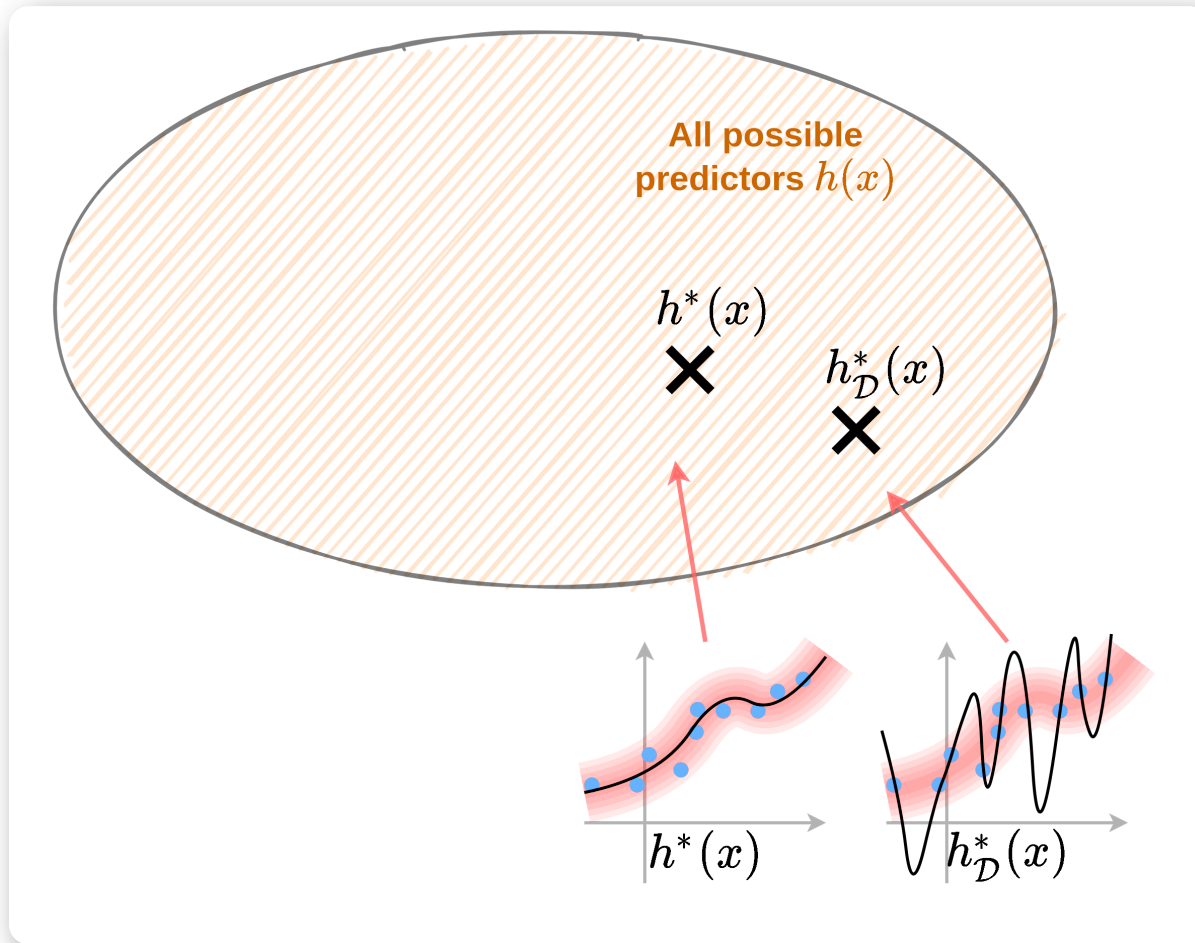


Overfitting (התאמת יתר)

תופעת ה **overfitting** מתארת את המצב שבו המודל הנלמד לומד מאפיינים מסויימים אשר מופיעים רק במדגם ואינם הם אינם מייצגים את התכונות של הפילוג האמיתי שלפיו מפולגים המשתנים האקראיים אשר מהם נוצר המדגם במדגם. תופעה זו פוגעת ביכולת ההכללה של המודל.



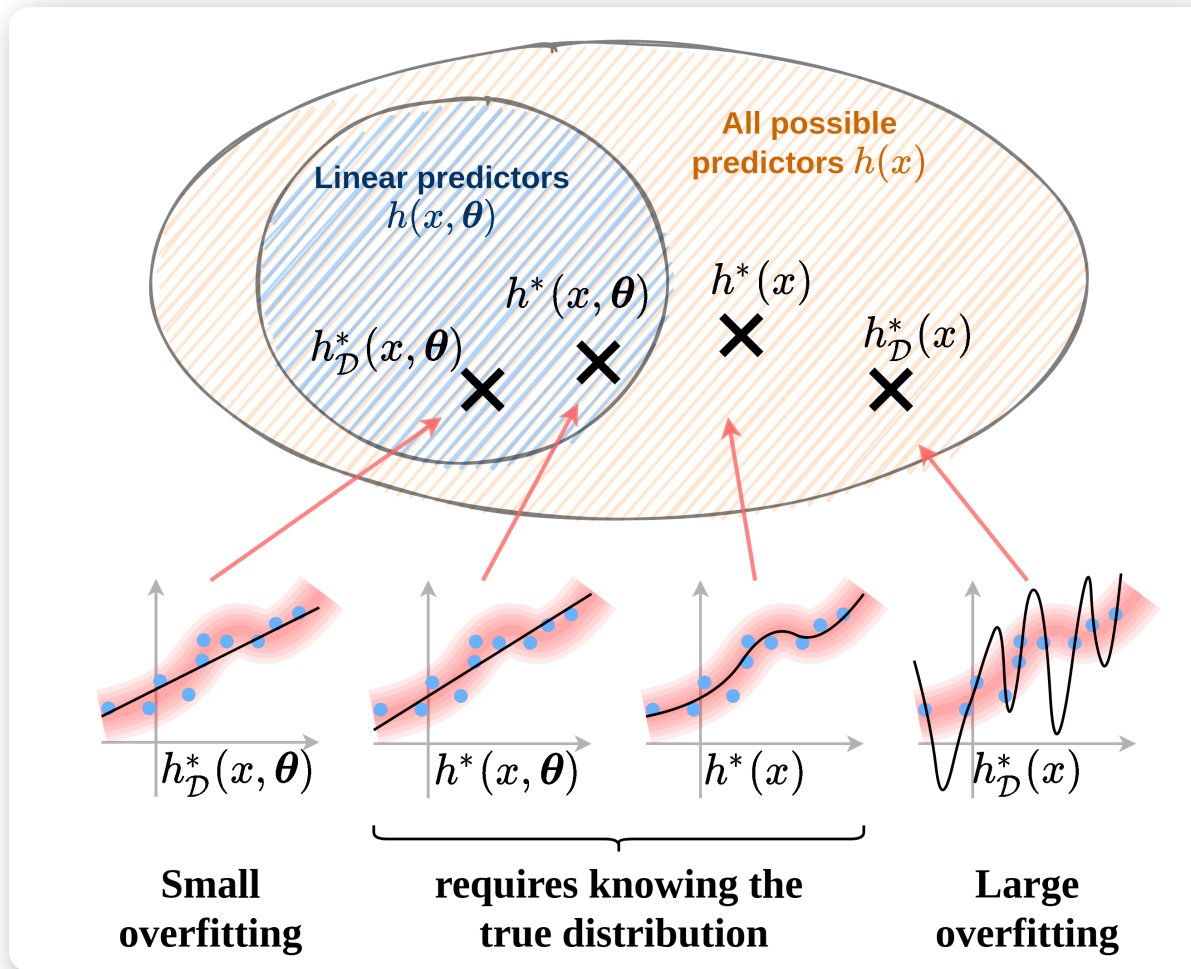
(התאמת יתר) Overfitting



- חזאי לא מוגבל יכול לקבל כל צורה כל עוד הוא עובר בין הנקודות של המדגם.
- בכדי לשלוט בצורה שבה הוא מתנהג נוכל להגביל את המרחב שבו אנו מחפשים.
- נעשה זאת על ידי שימוש במודל פרמטרי.

נסמן:

- $h(x; \theta^*)$: החזאי הפרמטרי האופטימאלי.
- $h_D(x; \theta^*)$: החזאי המשערך.



יכולת הביטוי של מודל פרמטרי

יכולת הביטוי (expressiveness) של מודל מתייחסת לגודל של מרחב הפונקציות שאותו יכול המודל פרמטרי מסויים לייצג.

- **יכולת ביטוי נמוכה** - < יודע לייצג משפחה מצומצמת. לדוגמא: מודל לינארי.
- **יכולת ביטוי גבוהה** - < יודע לייצג **או לקרב** משפחה רחבה. לדוגמא: פולינום מסדר גבוהה.

איזה יכולת ביטוי נעדיף?

- מצד אחד אנו נרצה מודל עם יכולת ביטוי גבוהה על מנת שיוכל לקרב את החזאי האידאלי.
- מצד שני יכולת יצוג גבוה תאפשר הרבה **overfitting**.

Hyper-parameters

Hyper parameters הינו שם כולל לכל הפרמטרים שמופיעים בשיטה או במודל הפרמטרי, אך הם אינם חלק ממשתני האופטימיזציה בשלב האופטימיזציה על ה-`train-set`.

דוגמאות:

- סדר הפולינום שבו אנו משתמשים.
- הפרמטר η אשר קובע את גודל הצעד באלגוריתם ה-`gradient descent`.
- פרמטרים אשר קובעים את המבנה של רשת נוירונים.

כאשר **hyper-parameter** מסויים שולט ביכולת הביטוי של המודל הפרמטרי, כדוגמאת המקרה של סדר הפולינום, נכנה פרמטר זה **הסדר של המודל**.

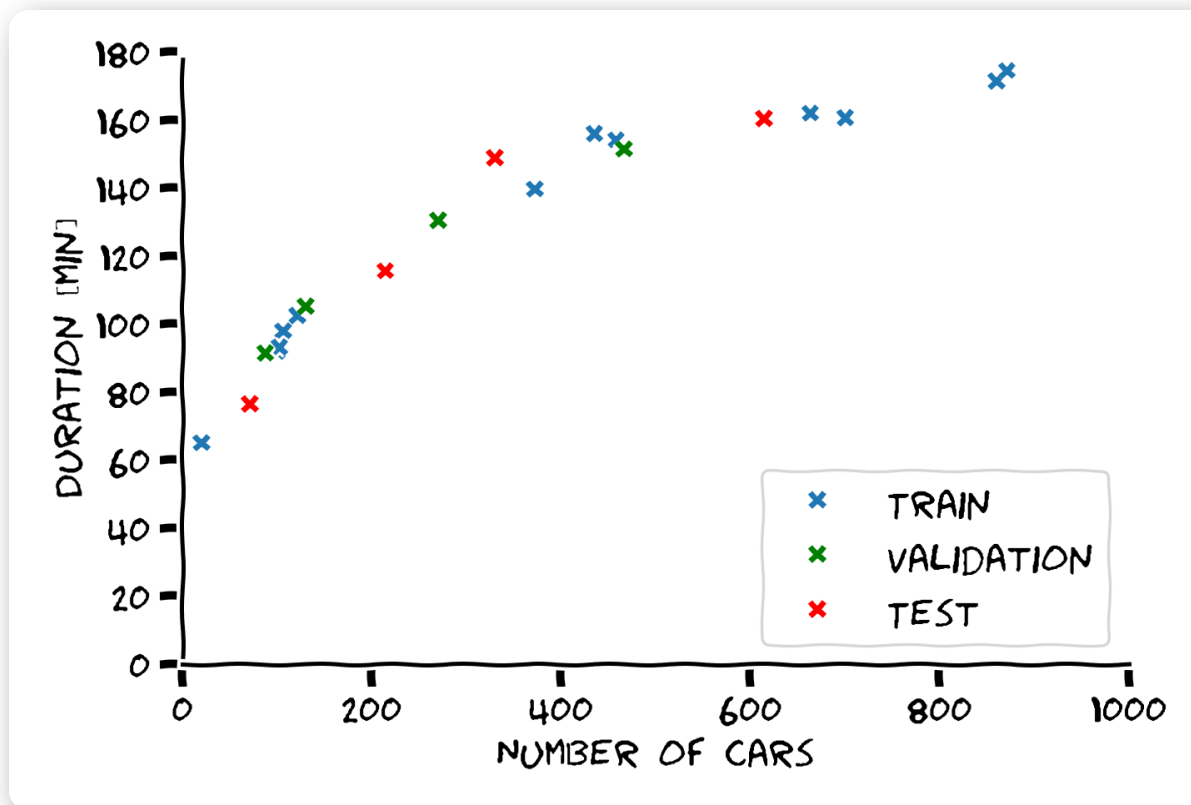
בחירת hyper-parameters בעזרת validation set

- **hyper-parameters אינם חלק מבעיית האופטימיזציה.**
- **אנו צריכים דרך אחרת לקבוע אותם.**
- **לרוב נאלץ לקבוע אותם בעזרת ניסוי וטעיה.**
- **לא נוכל להשתמש ב test set לצורך זה.**
- **נצטרך לייצר מדגם נפרד חדש.**
- **נפצל עוד את ה train set ל:**
 - **train set חדש.**
 - **validation set.**

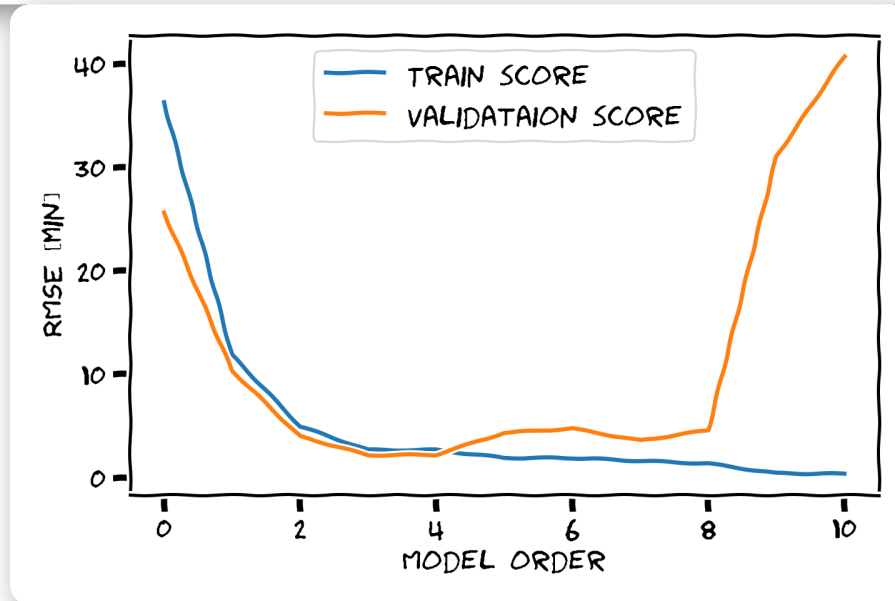
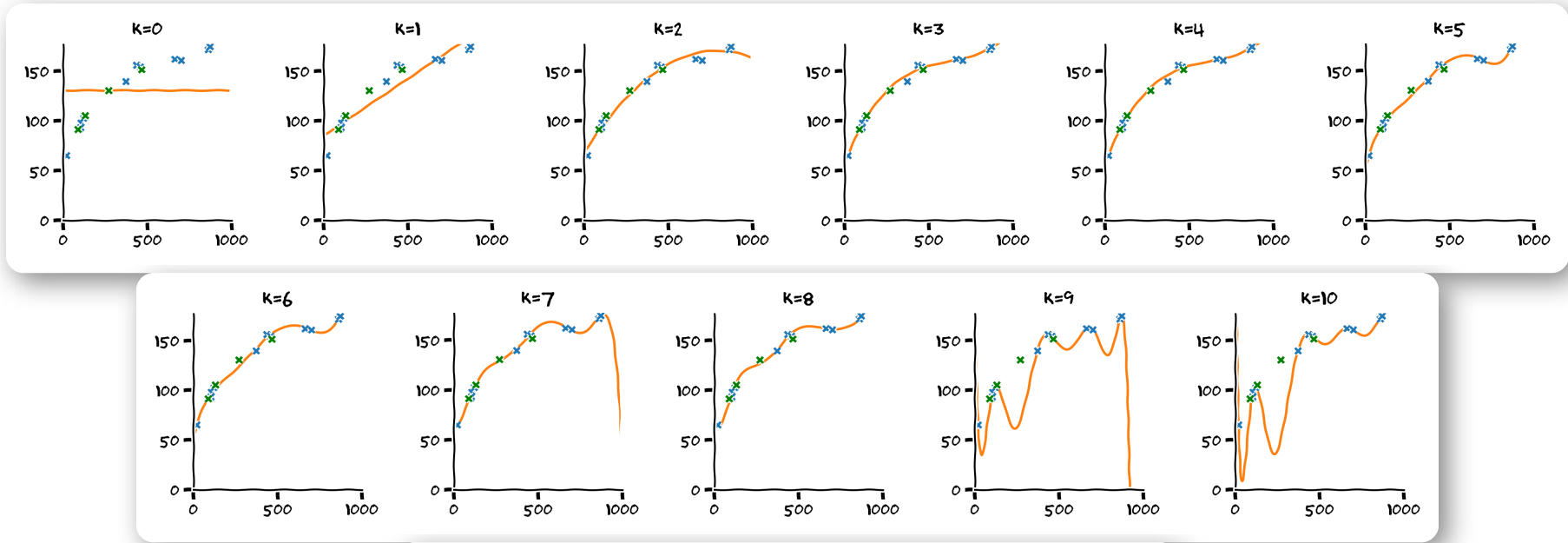
שלבי הבחירה של hyper-parameters

- נפצל את ה train set ל train ו validation.
- נחזור על הפעולות הבאות בעבור ערכים שונים של hyper-parameters:
 - נבנה את המודל על סמך ה train.
 - נשערך את ביצועי המודל על ה validation.
- נבחר את הפרמטרים עם הביצועים הטובים ביותר על ה validation.
- נאחד בחזרה את ה train וה validation.
- נבנה את המודל הסופי על סמך ה hyper-parameters שנבחרו.

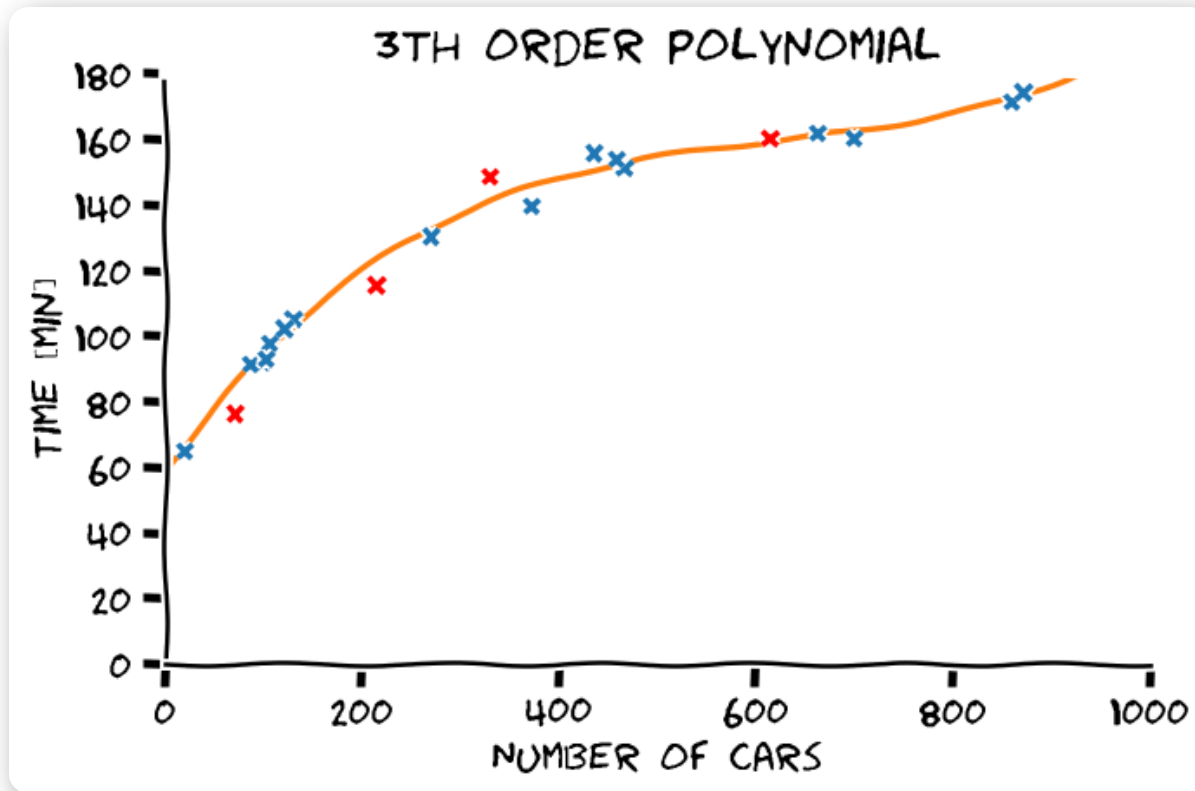
דוגמא: פיצול train-validation-test



התלות בסדר הפולינום



Retrain : דוגמא



Train cost (RMSE): 2.53 min •

Test cost (RMSE): 6.88 min •

Approximation-estimation decomposition

נתייחס לשני גורמים אשר מונעים מאיתנו למצוא את החזאי האופטימאלי $h^*(x)$:

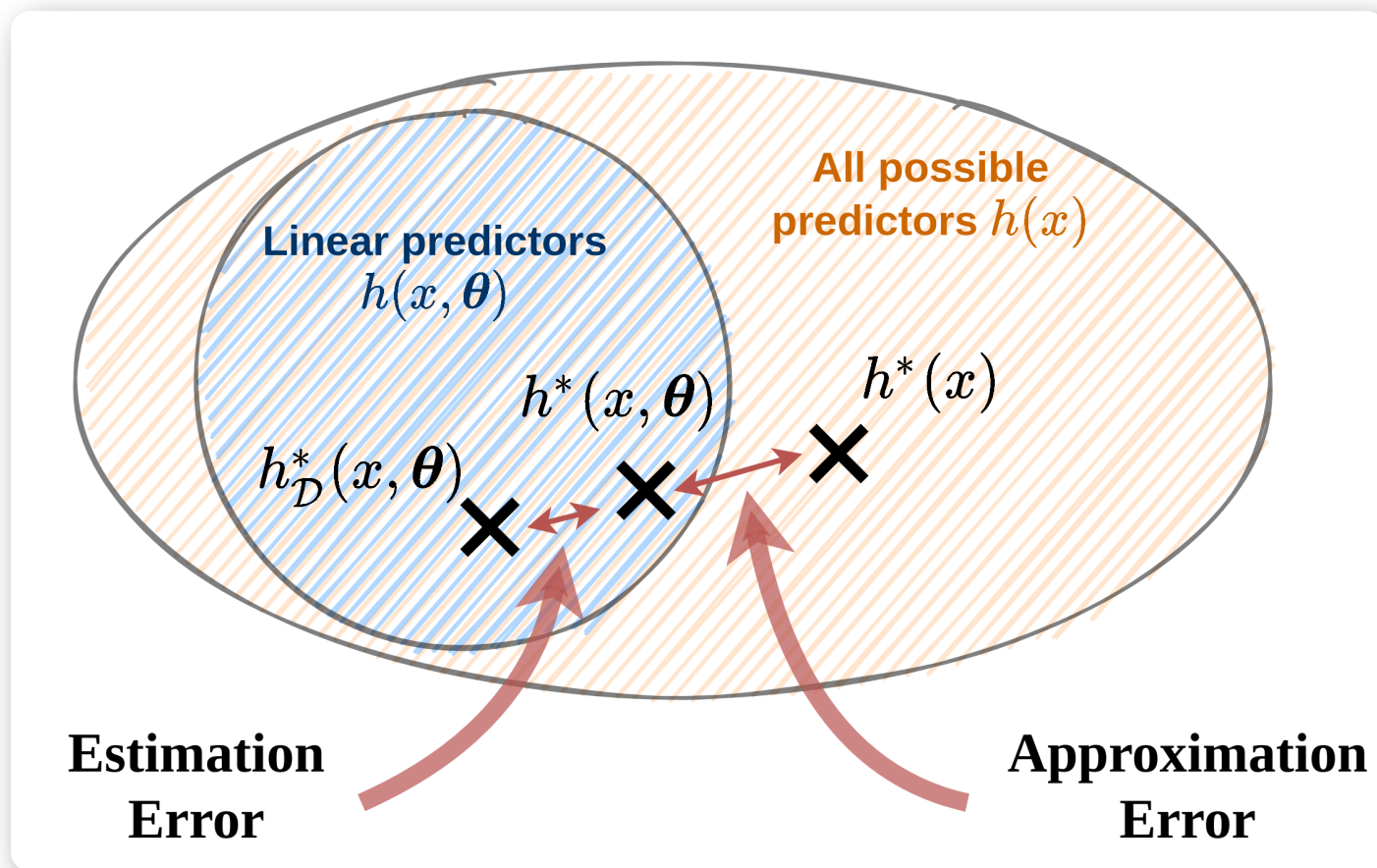
Approximation error - שגיאת קירוב

השגיאה עקב ההגבלה למשפחה פרמטרית מסוימת. נובעת מההבדל בין $h^*(x)$ לבין $h^*(x, \theta)$.

Estimation error - שגיאת השיערוך

השגיאה הנובעת מהשימוש במדגם כתחליף לפילוג האמיתי. נובעת מההבדל בין $h^*(x, \theta)$ לבין $h_{\mathcal{D}}^*(x, \theta)$.

Approximation-estimation decomposition



כאשר נרצה לדבר על השגיאה הכוללת נרצה להתייחס להבדל בין החיזוי של החזאי המשוערך $h_D^*(x; \theta)$ ו y .

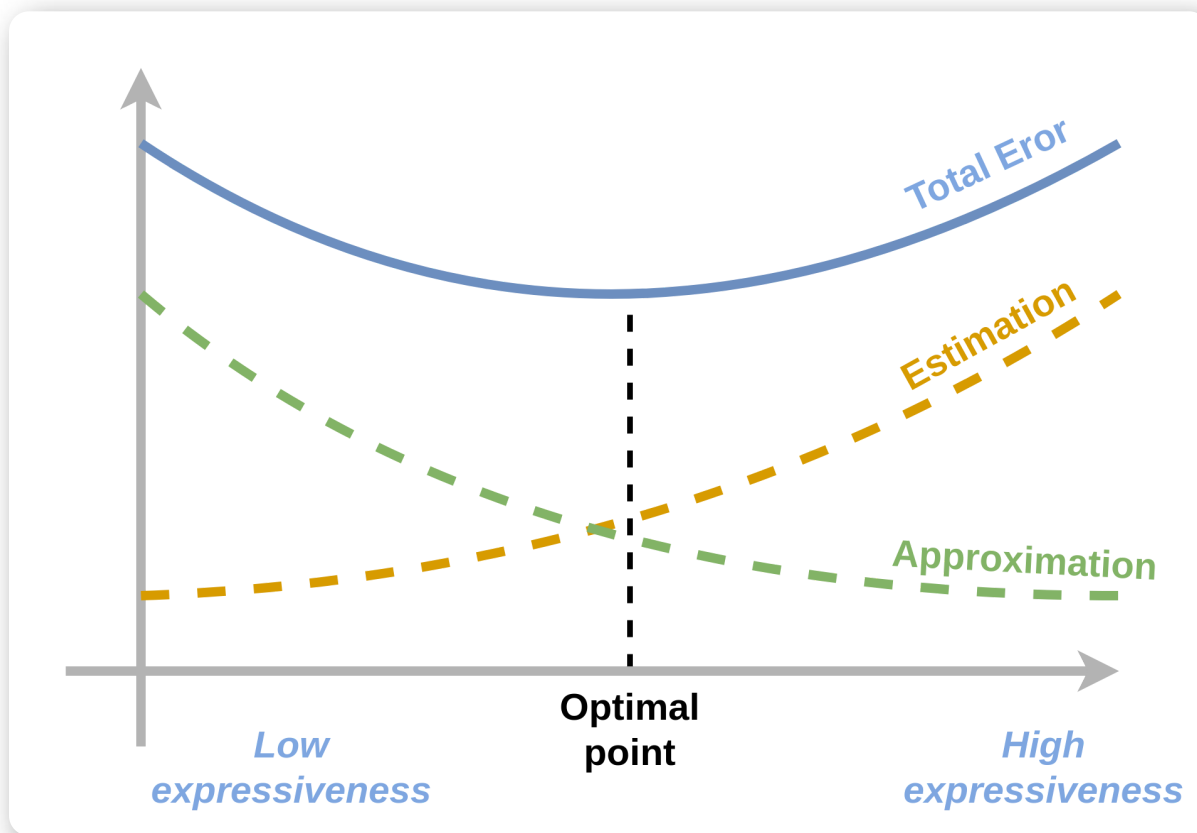
במקרים אלו נוסיף גורם שלישי:

Noise - ה"רעש" של התויות

השגיאה שהחזאי האופטימאלי צפוי לעשות. שגיאה זו נובעת מהאקראיות של התויות y .

Approximation-estimation Tradeoff

- ככל שיכולת הביטוי תגדל המרחק בין $h^*(x; \theta)$ לבין $h^*(x)$ יקטן ושגיאת הקירוב תקטן.
- בלא מעט מקרים ככל שיכולת הביטוי תגדל גם שגיאת השיערוך תגדל.



המדגם כמשתנה אקראי

- ביצועיו של חזאי כל שהוא תלויים לא רק בשיטה ובמודל הפרמטרי אלא גם במדגם שאיתו עבדנו.
- בעבור מדגמים שונים אנו מצפים לקבל ביצועים שונים.
- ניתן להסתכל על המדגם כמשתנה אקראי.
- בכדי לבטל את התלות במדגם נמצע את הביצועים על פני כל המדגמים האפשריים.

$$\text{average cost} = \mathbb{E}_{\mathcal{D}} [R(h_{\mathcal{D}})]$$

כאשר $\mathbb{E}_{\mathcal{D}}$ היא התוחלת על פני המדגמים האפשריים

לצורך הדיון התיאורטי על מרכיבי שגיאת החיזוי נגדיר את החזאי הממוצע באופן הבא:

החזאי אשר מחזיר את החיזוי הממוצע על פני כלל החזאים המתאימים למדגמים השונים:

$$\mathbb{E}_{\mathcal{D}} [h_{\mathcal{D}}(x)]$$

גודל זה אינו ניתן לחישוב!

Bias-variance decomposition

- פירוק יותר פרקטי.
- מתאים לפונקציית מחיר של MSE.

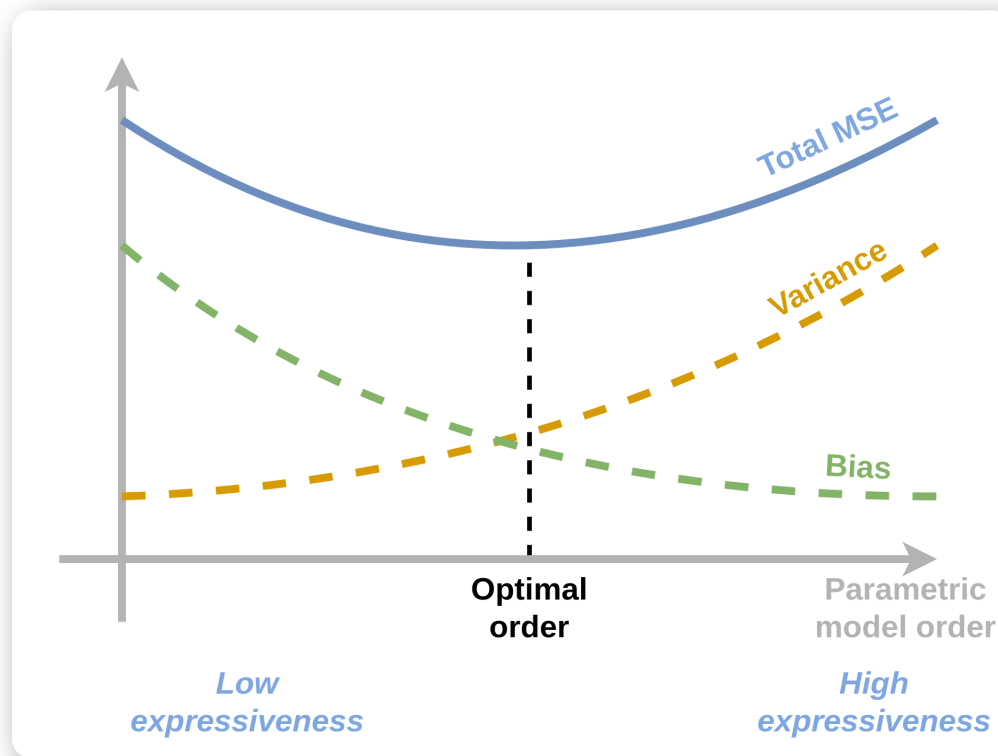
$$\begin{aligned} \mathbb{E}_{\mathcal{D}} [\mathbb{E} [(h_{\mathcal{D}}(\mathbf{x}) - y)^2]] \\ = \mathbb{E} \left[\underbrace{\mathbb{E}_{\mathcal{D}} [(h_{\mathcal{D}}(\mathbf{x}) - \mathbb{E}_{\mathcal{D}} [h_{\mathcal{D}}(x)])^2]}_{\text{Variance}} + \underbrace{(\mathbb{E}_{\mathcal{D}} [h_{\mathcal{D}}(x)] - h^*(\mathbf{x}))^2}_{\text{Bias}^2} \right. \\ \left. + \underbrace{(h^*(\mathbf{x}) - y)^2}_{\text{Noise}} \right] \end{aligned}$$

כאשר

$$\bullet h^*(x) = \mathbb{E}[y|x] \bullet$$

Bias-variance decomposition

$$\begin{aligned} & \mathbb{E}_{\mathcal{D}} \left[\mathbb{E} \left[(h_{\mathcal{D}}(\mathbf{x}) - y)^2 \right] \right] \\ &= \mathbb{E} \left[\underbrace{\mathbb{E}_{\mathcal{D}} \left[(h_{\mathcal{D}}(\mathbf{x}) - \mathbb{E}_{\mathcal{D}} [h_{\mathcal{D}}(x)])^2 \right]}_{\text{Variance}} + \underbrace{(\mathbb{E}_{\mathcal{D}} [h_{\mathcal{D}}(x)] - h^*(\mathbf{x}))^2}_{\text{Bias}^2} \right. \\ & \left. + \underbrace{(h^*(\mathbf{x}) - y)^2}_{\text{Noise}} \right] \end{aligned}$$



- דרך אלטרנטיבית להקטין את שגיאת השיערוך / variance .
- הרעיון: להתערב בבעית האופטימיזציה על מנת לגרום לה "להעדיף" מודלים מסויימים.
- זוהי הגבלה "רכה" של משפחת המודלים.
- מאפשר שימוש במודלים פרמטריים בעלי יכולת ביטוי גבוהה יותר.

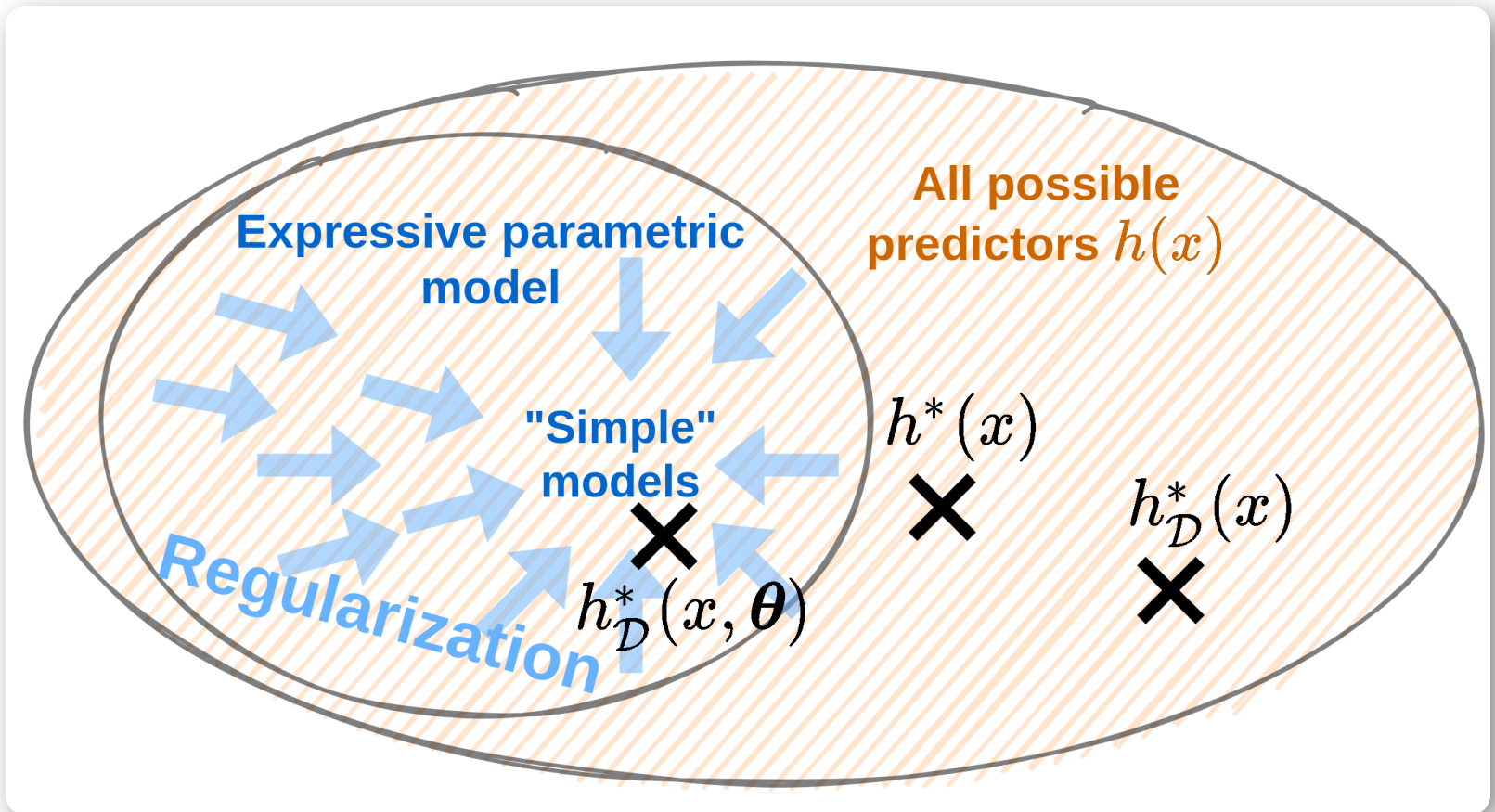
השיטה: נוסיף איבר אשר יתן "קנס" למודלים לא רצויים.

$$\theta = \left[\arg \min_{\theta} \underbrace{f(\theta)}_{\text{The regular objective function}} + \lambda \underbrace{g(\theta)}_{\text{The regularization term}} \right]$$

הפרמטר λ קובע את עוצמת (או משקל) הרגולריזציה.

הוא hyper-parameter שיש לקבוע בעזרת ה validation .set

רגולריזציה - אילוסטרציה



- באופן כללי, הבחירה של פונקציית הרגולריזציה $g(\theta)$ תלויה באופי הבעיה.
- לרוב הבחירה תהיה בשיטה של ניסוי וטעיה על פונקציות רגולריזציה נפוצות.
- פונקציות הרגולריזציה הנפוצות ביותר הינן:
 - l_1 - מוסיף $g(\theta) = \|\theta\|_1$
 - l_2 - מוסיף $g(\theta) = \|\theta\|_2^2$

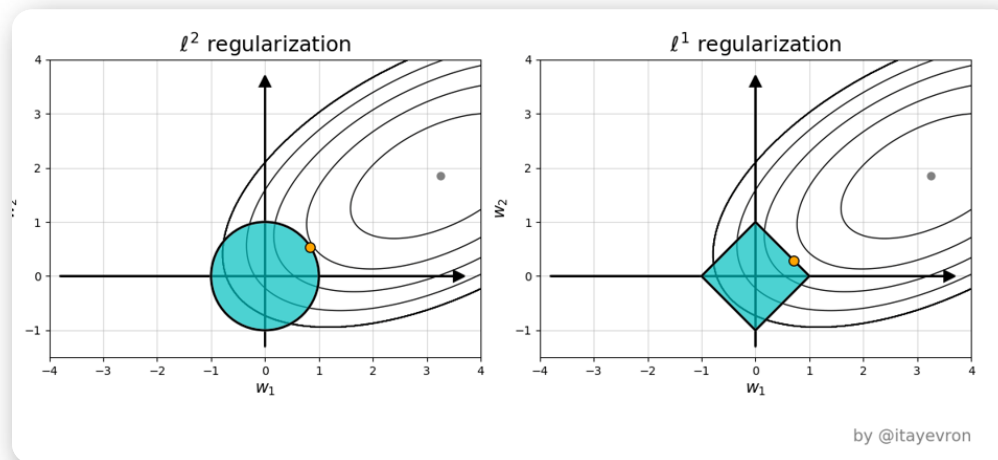
רגולריזציה l_2 מכונה גם Tikhonov regularization

- מנסות לשמור את הפרמטרים כמה שיותר קטנים.
- מוטיבציה: מודל בעל פרמטרים קטנים יותר יהיה לרוב בעל נגזרות קטנות יותר, ולכן הוא יהיה יותר "חלק".

l_2

- גדל בצורה ריבועית עם הפרמטרים
- ינסה להקטין בעיקר את הפרמטרים הגדולים ופחות את הקטנים.
- הרגולריזציה שואפת לפרמטרים בעלי גודל יותר אחיד.

- תפעל להקטין את כל האיברים כמה שיותר ללא קשר לגודלם.
- רגולריזציית l_1 תגרום לפרמטרים הפחות חשובים להתאפס.
- וקטור הפרמטרים שיתקבל יכול הרבה מאד אפסים - וקטור דליל (sparse).



Ridge regression: LLS + ℓ_2 regularization

$$\boldsymbol{\theta} = \left[\arg \min_{\boldsymbol{\theta}} \frac{1}{N} \sum_i (\mathbf{x}^{(i)\top} \boldsymbol{\theta} - y^{(i)})^2 + \lambda \|\boldsymbol{\theta}\|_2^2 \right]$$

גם לבעיה זו יש פתרון סגור והוא נתון על ידי:

$$\boldsymbol{\theta}^* = (X^\top X + \lambda)^{-1} X^\top \mathbf{y}$$

אנו נראה את הפתוח של פתרון זה בתרגיל 4.2.

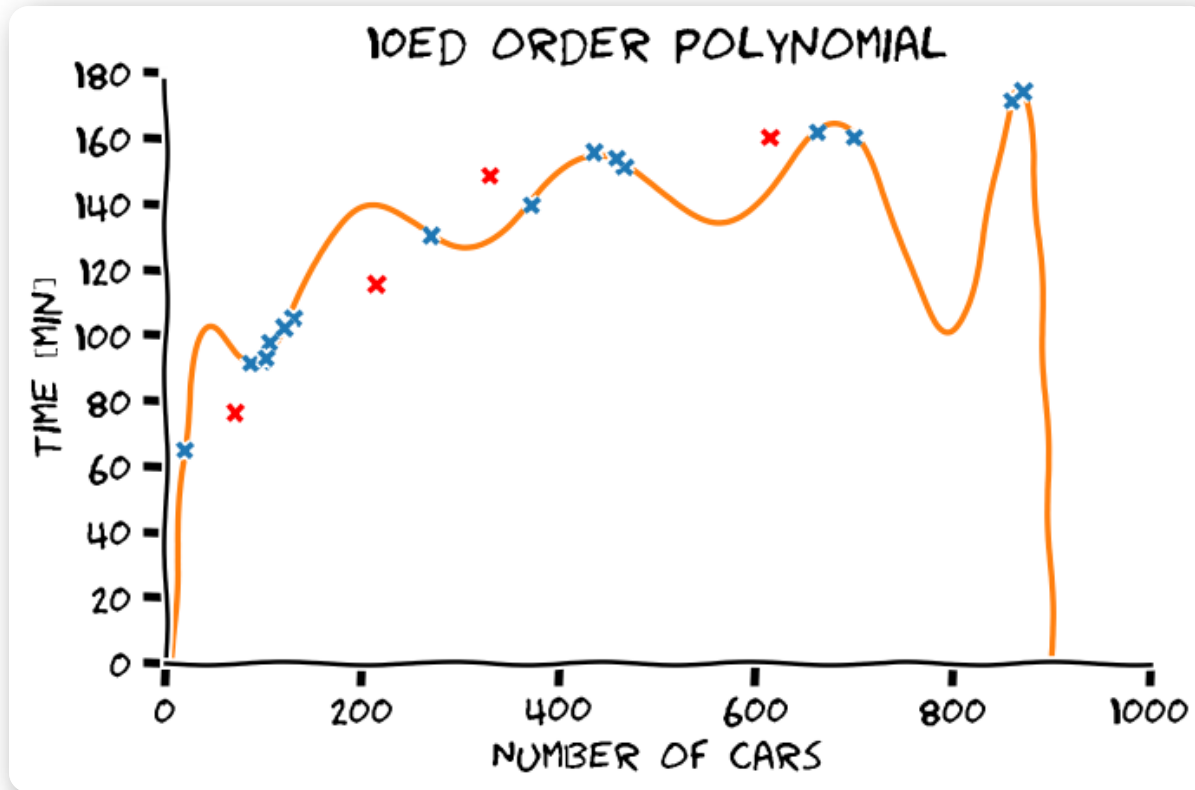
LASSO: LLS + l_1 regularization

$$\boldsymbol{\theta} = \left[\arg \min_{\boldsymbol{\theta}} \frac{1}{N} \sum_i (\mathbf{x}^{(i)\top} \boldsymbol{\theta} - y^{(i)})^2 + \lambda \|\boldsymbol{\theta}\|_1 \right]$$

לבעיה זו אין פתרון סגור ויש צורך להשתמש באלגוריתמים איטרטיביים אשר מבוססים על `gradient descent`.

LASSO = Linear Absolute Shrinkage and Selection Operator

Ridge regression : דוגמה



$$\lambda = 10^{-4} \bullet$$

Train cost (RMSE): 2.62 min •

Test cost (RMSE): 6.83 min •