דו"ח פרויקט: אנליזה של מידע

אמיר אברהמי – 203204367
אור לוי – 203518766

1. תיאור הניסיונות השונים לקבוצת התנאים C:

א) קבוצת התנאים הוגדרה כ- "האם ערך הפיקסל (x,y) גדול מ0?", כלומר הפכנו את התמונה למונוכרומטית (בינארית) ובדקנו אילו פיקסלים צבועים בתמונה. מבחנים אלו הניבו תוצאות יותר טובות מגרסה 1 – 11% שגיאה לעומת 14% שגיאה בגרסה הראשונה. קבוצה זו גם הניבה את השגיאה הנמוכה ביותר מבין כל התנאים שניסינו.

ב) חלוקת התמונה לFrames בגדלים שונים (2x2 , 4x4 , 7x7) כאשר קבוצת התנאים על כל חלק הוגדרה כ- "האם ממוצע הערכים בFrame זה גדול ממספר סף כלשהו" ובצענו מספר ניסיונות גם עבור מספר הסף. מבחנים אילו הוכחו כלא יעילים – קיבלנו שגיאה של 17% בממוצע.

ג) בדיקת ערכים ממוצעים עבור שורות, עמודות וFrames ובדיקה האם הם עומדים בסף מסוים, כאשר מספר הסף משתנה. מבחנים אלו נועדו לתת דגש על שורות / עמודות או Frames שיש בהם הרבה מידע בתמונות מסוימות לעומת כאלה שאין בהם הרבה מידע.

ד) מציאת צורות, קווים ועיקולים בתמונה בשילוב חלוקת התמונה לחלקים – גם קבוצת תנאים זו לא הניבה תוצאות טובות – בערך 15% שגיאה.

ה) שימוש בטכניקת PCA על מנת להקטין את מימד התמונה ולמצוא את הfeatures המשמעותיים ביותר בdata set – זמן הריצה של טכניקה זו גדול משמעותית מזמן בניית העץ (סביבות 30 דקות) ובסופו של דבר קיבלנו שגיאות במהלך ההרצה שלא הצלחנו למצוא את המקור שלהן.

ו) בנוסף, מצאנו מאמרים הקשורים לטכניקות של Features Extraction כגון Hough Transform (המשמש לזיהוי קווים וצורות בתוך תמונה) וכן מורפולוגיות נוספות המשמשות לזיהוי צורות בתוך התמונה אך אופן המימוש והיישום בתוך המודל של עץ החלטה לא עמדו בקנה אחד.
קישורים למאמרים בהם נעזרנו:
http://www.ias.ac.in/article/fulltext/sadh/035/04/0419-0426
http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.929.9088&rep=rep1&type=pdf

את קבוצות התנאים השונות בדקנו כמובן על MNIST Handwritten digits data set וכן על Data set נוסף – Fashion MNIST בעל פורמט זהה לMNIST Handwritten digits על מנת לוודא שיפור אחוז השגיאה באופן כללי ולא רק כזה המתאים לHandwritten digits. ניתן לקרוא על Fashion MNIST בקישור הבא:
https://github.com/zalandoresearch/fashion-mnist

לבסוף החלטנו לשלב מספר קבוצות תנאים אשר שילובם הניב את השגיאה הנמוכה ביותר מבין הניסיונות – 10% שגיאה על MNIST Handwritten digits וכן 17% שגיאה על Fashion MNIST.

<u>קבוצת התנאים בהם החלטנו להשתמש בגרסה השנייה:</u>

קבוצת התנאים מתחלקת ל-4 תתי קבוצות:

א. "האם ערך הפיקסל (x,y) גדול מ-0?"
ב. חלוקת התמונה לFrames בגודל 4x4 ובחנו "האם סכום הערכים בFrame זה גדול ממספר סף מסוים" – כאשר מספר הסף נע בין 0 ל16.
ג. "האם ממוצע הערכים בשורה ה-i גדול ממספר סף מסוים" – כאשר מספר הסף נע בין 0 ל-1 בקפיצות של 0.05.
ד. "האם ממוצע הערכים בעמודה ה-i גדול ממספר סף מסוים" – כאשר מספר הסף נע בין 0 ל-1 בקפיצות של 0.05.

2. <u>אופן מימוש הפרויקט:</u>

נחלק את אופן המימוש לשני חלקים עיקריים:

א) הטיפול ב-MNIST Data Set:

- חילוץ המידע מקבצי הCSV – על מנת לעבוד עם הData Set בצורה נוחה, יצרנו אובייקט המתאר תמונה בגודל 28x28 (תמונה מתוך הData Set).
- ייצוג הData Set – על מנת לשפר את נוחות העבודה עם הData Set, יצרנו אבסטרקציה שבעזרתה נוכל בצורה נוחה לממש את הפונקציונליות הנדרשת מהData Set, כגון: מעבר באיטרציה על כל הData Set, חישוב מספר הופעות של כל התוויות השונות וכד'.
- חלוקת המידע למדגם אימון ומדגם וולידציה – מימשנו את הפונקציונליות בצורה פשוטה אך יעילה: עבור p – אחוז התמונות מתוך הData Set שעלינו לשלוף לטובת וולידציה, חישבנו את מספר התמונות הכולל שעלינו לשלוף מתוך הData Set. לאחר מכן, באיטרציה, בכל שלב הגרלנו אינדקס רנדומלי לפי מספר התמונות שנותרו בData Set ושלפנו את התמונה באינדקס זה. בצורה זו, דימינו כמה שאפשר התפלגות אחידה על פני כל הData Set.

ב) מימוש אלגוריתם בניית העץ:

- **מבנה הנתונים של העץ:** מבנה הנתונים הוא עץ בינארי כאשר ישנם שני סוגים שונים של צמתים: צומת מסוג ראשון עבור צמתים פנימיים (מבחנים) וצומת מסוג שני עבור עלים (תוויות), אשר מחזיקים גם במבני נתונים של המבחנים בהם ניתן עוד להשתמש בבניית העץ וה-Data Set המורכב מהאובייקטים שהגיעו לעלה זה במסלול מהשורש. למבנה נתונים זה יש פונקציונליות של חיזוי אובייקט חדש כפי שנדרש ממודל החלטה זה.
- בחירת ערך T עם שגיאה מינימלית – במהלך המימוש הבחנו כי בצורה נאיבית על מנת לבדוק ערכי T שונים, נדרשת הרצה של אלגוריתם בניית העץ מספר פעמים, כאשר ישנם הרבה מאוד חישובים זהים לעצים שונים. דבר זה, יצר בעיית Performance. על מנת לפתור את הבעיה, החלטנו על המימוש הבא: במקום שאלגוריתם בניית העץ יחזיר עץ יחיד, נריץ את האלגוריתם כ-T* איטרציות (כאשר T* הוא הT המקסימלי) ועבור כל T נשמור את תמונת המצב הנוכחית של העץ. בסוף הריצה, אנו מחזירים את כל תמונות המצב. אופן מימוש זה שמר על הפונקציונליות הנדרשת אך הקטין משמעותית את זמן הריצה.
- מניעת חזרה על חישוב Information Gain – בכל איטרציה נדרש לחשב את הIG על כל העלים וכל ההחלפות האפשריות ולבחור את המקסימלי. נשים לב, כי דבר זה, בצורה נאיבית, גורר הרבה מאוד חישובים שחוזרים על עצמם. על מנת להתגבר על בעיה זו, חישבנו בכל איטרציה את הIG רק עבור שני העלים החדשים שנוצרו באיטרציה הקודמת.
- ביצוע ההחלפה בעץ – בעת חישוב הInformation Gain עבור כל עלה, אנו נדרשים לחשב את ההחלפה האופטימלית על עלה זה. לאחר מכן, בשלב ההחלפה, עלינו לבצע את ההחלפה בפועל על ההחלפה האופטימלית מבין כל העלים. על מנת להימנע מחישוב ההחלפה גם בשלב חישוב הIG וגם בשלב ההחלפה בפועל, מימשנו את פונקציית הIG לכל עלה כך שתחזיר שלשה המכילה את העלה עצמו, את ההחלפה האידיאלית עבורו (הצומת הפנימי ושני העלים) ואת הIG האופטימלי עבור עלה זה. בצורה זו, בשלב ההחלפה בפועל נמנענו מלחשב בשנית את הצומת הפנימי ושני העלים המחליפים.
- תנאי העצירה של האלגוריתם – כפי שהסברנו למעלה, האלגוריתם רץ בלולאה שאורכת T* איטרציות. יחד עם זאת, ישנו תנאי עצירה נוסף במטרה למנוע חישובים מיותרים: כאשר קיבלנו שהIG האופטימלי בעץ הוא 0, אנו מבינים שלא ניתן לשפר את העץ ולכן אנו שומרים את תמונת המצב הנוכחית (גם אם אין פרמטר T עבור גודל העץ הנוכחי) ומסיימים את הלולאה.

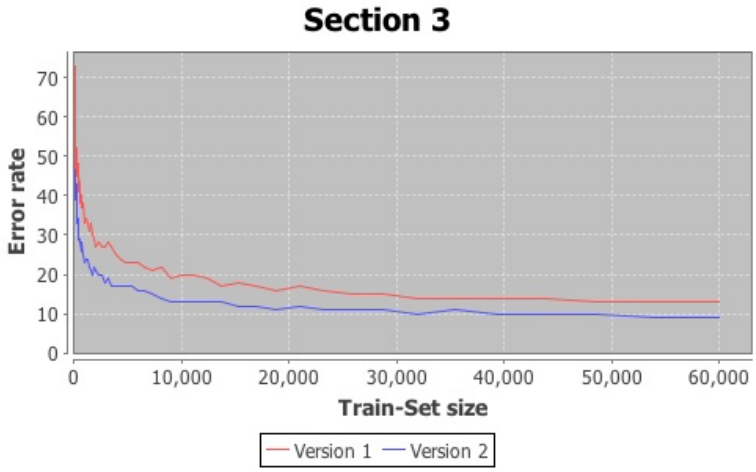
האתגרים בהם נתקלנו במהלך מימוש הפרויקט:

זמן ריצה וPerformance: במהלך בניית העץ נתקלנו בזמן ריצה ארוך מעבר לזמן ריצה סביר.
לאחר בדיקה של אופן המימוש זיהינו כי חישוב ה- Information Gain על כל עלה מתבצע בכל איטרציה למרות שנדרשים רק שני חישובים חדשים שלא בוצעו קודם. על מנת להתגבר על הבעיה, שמרנו את תוצאות החישוב על כל עלה שנוצר וביצוע חישוב רק על שני העלים החדשים שנוצרו באיטרציה הנוכחית. בנוסף, עלו בעיות Performance נוספות, בין היתר, ביצוע ההחלפה בעץ הדורש חישוב שכבר בוצע וכן בניית העץ מחדש עבור כל T שאנו נדרשים לבדוק.

מציאת גרסה 2 בעלת אחוז שגיאה נמוך: כאשר ניגשנו למציאת קבוצת תנאים C עבור גרסה 2, מצאנו כי סוג המבחנים הזהה לזה של גרסה 1 (סוג מבחן המשתמשים בו לעיתים תכופות בבניית עצי החלטה) עבור מספר סף שונה מזה של גרסה 1 מוריד משמעותית את אחוז השגיאה אך רצינו לבדוק האם ניתן לשפר את אחוז שגיאה זה.
מציאת קבוצת התנאים C שתשפר את השגיאה היוותה אתגר משמעותי בפרויקט.
על מנת לנסות ולהתגבר על אתגר זה, ביצענו מחקר מקיף הכולל קריאת מאמרים הנוגעים בנושאים שונים של Machine Learning בין היתר: Dimension Reduction, Feature Extraction, Image Processing על מנת לאתר את המידע החשוב ביותר בתמונה ועל פיו ליצור תנאים בהם העץ יכול להשתמש. במהלך המחקר, נתקלנו בקושי לחבר בין הדרכים השונות למציאת מידע בתוך התמונה לבין התאמת קבוצת תנאים C עבור עץ החלטה שכן, כפי שצוין קודם עץ החלטה קלאסי משתמש בקבוצת תנאים דומה לזו של גרסה 1. על תהליך ההתגברות על הבעיה הארכנו רבות בסעיף 1.

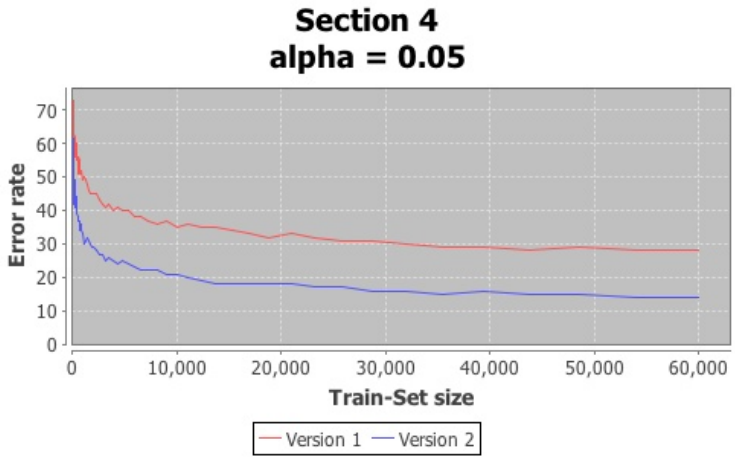
ניסויים ותוצאות

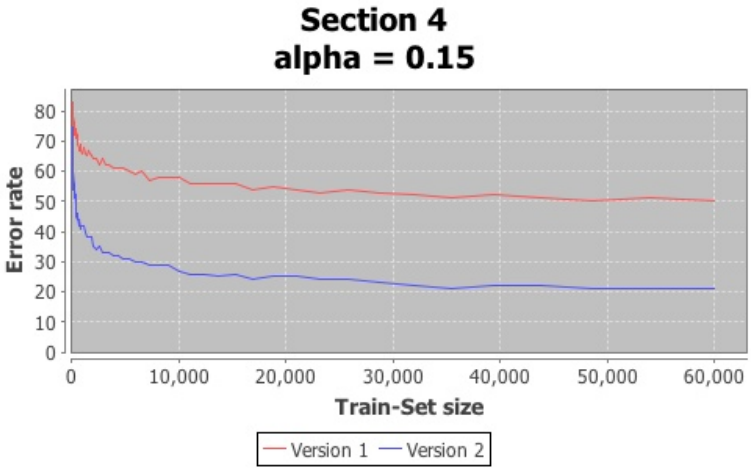
סעיף 3:

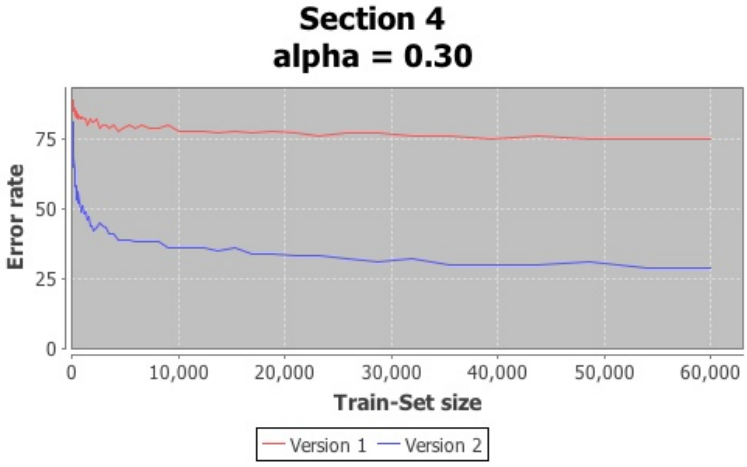


ניתוח תוצאות הגרף:
על פי הגרף, ניתן לראות כי ההבדלים בין בגרסאות באים לידי ביטוי בגודל השגיאה לכל גודל מדגם, אך שתי הגרסאות מתנהגות בצורה דומה: ככל שגודל מדגם האימון גדל השגיאה קטנה עד לנקודה מסוימת בה השגיאה מתייצבת ולא משתנה בצורה משמעותית (בסביבות גודל מדגם של 30,000).
ניתן להסביר את התנהגות זו באופן הבא: כאשר גודל המדגם הוא קטן, העץ מנסה להתאים את עצמו על פי מדגם זה אך כיוון שהמדגם מאוד קטן הוא לא משקף את ההתפלגות ולכן אנחנו מקבלים תופעה של overfitting.
כאשר המדגם גדול (30,000 ומעלה) אנו שמים לב כי השגיאה לא משתנה בצורה משמעותית, ככל הנראה בגלל שהגענו ל- Approximation Error של מחלקת ההיפותזות (קבוצת העצים שיכולים להבנות מכל קבוצת תנאים).
נשים לב כי App Error של גרסה 1 גבוהה יותר מזו של גרסה 2.

סעיף 4:



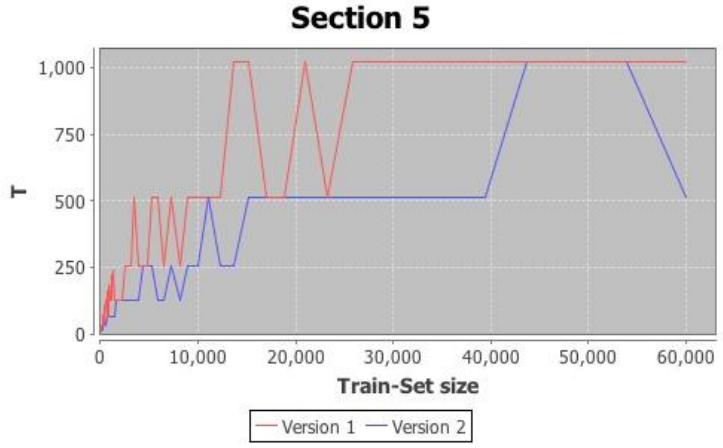


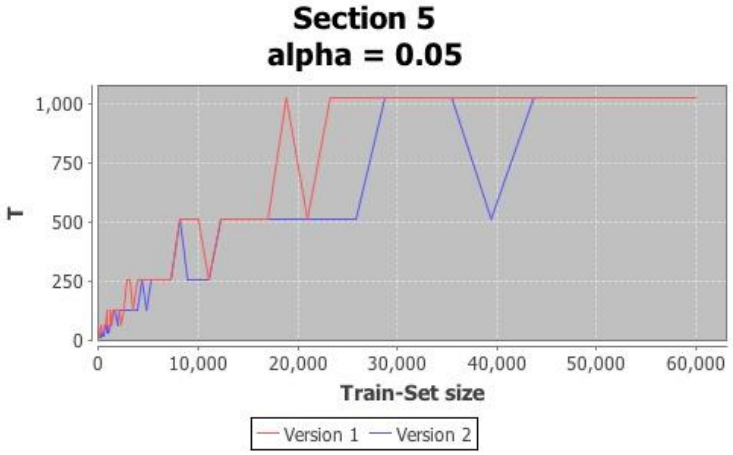


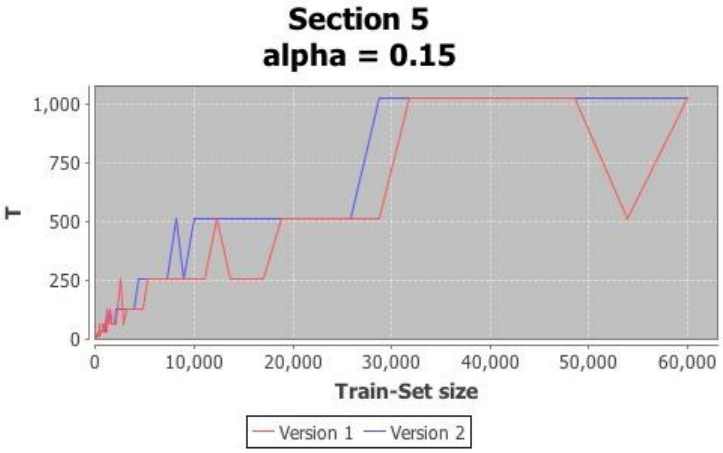
ניתוח תוצאות הגרפים:

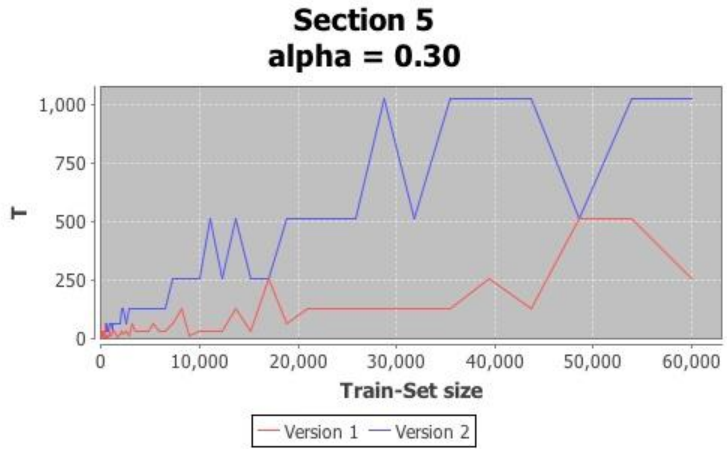
על פי הגרפים, ניתן להבחין כי ההתנהגות מסעיף 3 – ככל שגודל המדגם גדל השגיאה קטנה, נשמרת.
כמו כן, ניתן לראות כי ככל שרמת הרעש גוברת, השגיאה לכל גודל מדגם גדלה עבור שתי הגרסאות.
יחד עם זאת, ניתן להבחין כי השגיאה בגרסה 1 גדלה משמעותית כאשר רמת הרעש גדלה, לעומת גרסה 2 שיותר חסינה לרעשים. החסינות של גרסה 2 לרעשים ככל הנראה מושפעת מהגיוון בסוג המבחנים שמהם אנו בונים את העץ, שכן ישנם מבחנים שאינם מתבססים על פיקסלים בודדים אלא על frames / שורות / עמודות וממוצעים.

סעיף 5:


Section 5
1,000
750
500
250
0
T
0
10,000
20,000
30,000
40,000
50,000
60,000
Train-Set size
Version 1
Version 2


Section 5
alpha = 0.05
1,000
750
500
250
0
T
0
10,000
20,000
30,000
40,000
50,000
60,000
Train-Set size
Version 1
Version 2


Section 5
alpha = 0.15
1,000
750
500
250
0
T
0
10,000
20,000
30,000
40,000
50,000
60,000
Train-Set size
Version 1
Version 2



ניתוח תוצאות הגרפים:

עפ"י הגרפים ניתן להבחין כי בממוצע ככל שגודל המדגם גדל, כך גודל העץ T עם השגיאה המינימלית גדל.

- נתבונן בגרף הראשון (כאשר אין רעש כלל): נשים לב כי באופן גורף, העץ המתקבל בגרסה 2 קטן או שווה לגודל העץ המתקבל מגרסה 1 לכל גודל מדגם, אך על פי סעיף 3 לעץ המתקבל מגרסה 2 יש שגיאה נמוכה יותר מזו של העץ המתקבל מגרסה 1. ניתן להסביר זאת באופן הבא: העץ ה"קטן" יותר בגרסה 2 ככל הנראה משקף בצורה טובה יותר את ההתפלגות, או לחילופין, העץ מאותו גודל בגרסה 1 אינו משקף את ההתפלגות בצורה יעילה מספיק ולכן נדרש עץ גדול יותר על מנת לשפר את השגיאה.
- עבור הגרפים שהתקבלו מרמות רעש 0.05 ו-0.15 (5% ו15% רעש בהתאמה) ניתן לראות כי בממוצע גדלי העצים של גרסה 1 וגרסה 2 זהים.
- עבור הגרף שהתקבל מרמת רעש 0.30 (30% רעש) נשים לב כי באופן גורף גודל העץ המתקבל מגרסה 1 קטן ממש מגודל העץ המתקבל מגרסה 2. ננסה להסביר זאת באופן הבא: נשים לב מסעיף 4 כי השגיאה של העצים המתקבלים מגרסה 1 גבוהה מאוד עבור כל גודל מדגם. ככל הנראה, רמת הרעש הגבוהה יחסית, משבשת מאוד את מדגם האימון, כך שאינו מייצג בצורה טובה את ההתפלגות. כיוון שהעץ הנבנה מנסה להתאים את עצמו למדגם האימון, וככל שגודל העץ גדול יותר כך העץ מתאים יותר למדגם האימון, אנחנו מקבלים שעצים גדולים יותר מתאימים יותר למדגם האימון ופחות להתפלגות. לכן, העץ שייבחר יהיה מגודל קטן יותר, על מנת להתגבר על שיבוש זה. לעומת זאת, בגרסה 2 נקבל עצים בגודל יחסית דומה לרמות רעש נמוכות יותר, כלומר, העץ המתקבל חסין יותר לרמת רעש הולכת ומתגברת.