תרגיל בית 3: למידה וחיפוש לוקאלי

מטרות התרגיל

- הבנת התהליך של חיפוש לוקאלי באמצעות בעיה ספציפית
- הבנת ההשפעה של תכונות ושל דוגמאות על הביצועים של אלגוריתמי למידה
- הבנת תהליך כיוונון פרמטרים והערכת ביצועים של אלגוריתמי למידה
- ביצוע מחקר אמפירי מלא להשוואת שיפור עבור אלגוריתמי למידה

הערות

- תאריך הגשה: 22/01/15.
- את המטלה יש להגיש בזוגות בלבד.
- שאלות בנוגע לתרגיל יש לשלוח לליאור: liorf@cs.technion.ac.il.
- אנא עיינו ברשימת ה-FAQ המתעדכנת באתר לפני פנייה במייל.

מבוא

במטלה זו תשתמשו בחיפוש לוקאלי על מנת לשפר את ביצועיהם של מספר אלגוריתמי למידה שנלמדו במהלך הקורס.

עליכם לבנות אלגוריתם חיפוש לוקאלי, שבהינתן בעיית למידה, בוחר תת-קבוצה של התכונות ותת-קבוצה של הדוגמאות שתשפר את הדיוק של האלגוריתמים ביחס לבעיה המקורית.

תיאור הבעיה

מטרתנו היא לשפר את ביצועיהם של אלגוריתמי הלמידה K-Nearest Neighbours ו-Decision Tree, ע"י בחירת תכונות וכן ע"י בחירת דוגמאות.

לפיכך, נשתמש בחיפוש לוקאלי המתחיל בבעיה המקורית, ומאפשר פעולות של הסרת דוגמה והסרת תכונה מתוך הנתונים. נניח כי כל דוגמה נתונה בצורה של וקטור של ערכי התכונות עליה.

במילים אחרות, נתונה לנו מטריצה M המוגדרת $M(i,j) = f_j(o_i)$, כאשר f_j היא התכונה ה-j ו-o_i הוא הדוגמה ה-i. אנו רוצים להסיר שורות ועמודות מן המטריצה M על מנת לשפר את איכות האלגוריתם.

הסרת שורה:

$$\begin{bmatrix} 1 & 8 & 13 & 12 \\ 14 & 11 & 2 & 7 \\ \cancel{4} & \cancel{5} & \cancel{16} & \cancel{9} \\ 15 & 10 & 3 & 6 \end{bmatrix}$$

הסרת עמודה:

$$\begin{bmatrix} 1 & 8 & \cancel{13} & 12 \\ 14 & 11 & \cancel{2} & 7 \\ 4 & 5 & \cancel{16} & 9 \\ 15 & 10 & \cancel{3} & 6 \end{bmatrix}$$

מספר הדוגמאות גדול ממספר התכונות, וניתן לצפות שהסרת דוגמה בודדת מקבוצת האימון לא תשפיע מהותית על תהליך הסיווג. לפיכך, אלגוריתם חיפוש לוקאלי שלא מאפשר side steps עשוי לא להסיר דוגמאות. כדי להתגבר על כך, במקום מספר גדול של אופרטורים להסרת שורה בודדת, נבנה מספר אופרטורים כמספר העמודות (כדי לאפשר סיכוי שווה) של אופרטורים המסירים באקראי מספר שורות \ דוגמאות (נבחר למשל להסיר 5 דוגמאות באקראי).

על מנת להעריך את האיכות של מצב מסויים (שכאמור, מורכב מתת-קבוצה של דוגמאות ותת-קבוצה של תכונות), נשתמש בקבוצת הערכה (ולידציה) שתישאר קבועה במהלך החיפוש.

לאחר סיום החיפוש, נבצע בדיקה על קבוצת מבחן, שבה אנו לא נוגעים במהלך תהליך החיפוש כלל.

חלק א' [יבש]– הבנת הבעיה (20 נק')

א. מהו מרחב החיפוש של אלגוריתם החיפוש?
   1. מהי קבוצת המצבים?
   2. מהו המצב ההתחלתי?
   3. מהי קבוצת האופרטורים? יש להתייחס למקרה הכללי וגם למה שתממשו בפועל.
   4. מהי פונקציית ה-fitness?

חלק ב' [רטוב]– מימוש האלגוריתם וניסוי (25 נק')

א. ממשו את אלגוריתם החיפוש הלוקאלי (ראו טיפים והנחיות):
   1. ממשו את האלגוריתם First Choice Stochastic Hill Climbing על מצב כללי. על שם המחלקה להיות FirstChoiceLocalSearch ועליה לרשת מן LocalSearch בקוד המצורף.
   2. ממשו מצב חיפוש המייצג אלגוריתם למידה. על שם המחלקה להיות LearningState ועליה לרשת מהמחלקה SearchState בקוד המצורף.
   3. ממשו את פונקציית ההערכה למצב כזה באמצעות קבוצת ולידציה המתקבלת כקלט.
   4. ממשו אופרטורים המקבלים מצב חיפוש שכזה, ומחזירים מצב חדש. יש לממש אופרטורים עבור הסרת דוגמאות (כל אופרטור כזה יסיר 5 דוגמאות מקבוצת האימון שנבחרו באקראי), וכן עבור הסרת תכונה (לכל תכונה). שימו לב כי ברצוננו שמספר התכונות מכל סוג יהיה דומה לטובת אלגוריתם First Choice.

ב. הריצו את האלגוריתם על המידע המסופק עם הפרמטרים הבאים: אלגוריתם- K-Nearest Neighbours, מספר שכנים (K)-3.
   1. על מנת לטעון את המידע, השתמשו בפונקציה load_hw3_data_1 במודול data.py בקוד המצורף.
   2. שימו לב כי יש להפעיל את האופרטורים המסירים תכונות גם על קבוצת ההערכה, וכן גם על קבוצת המבחן בסוף תהליך החיפוש.
   3. הציגו את הדיוק המתקבל על קבוצת המבחן לפני ואחרי הפעלת אלגוריתם החיפוש. מי מהם גבוה יותר?
   4. הריצו פעם נוספת את האלגוריתם עם פרמטרים זהים. הציגו גם את התוצאה החדשה. האם היא זהה לתוצאה שקיבלתם קודם? מדוע?

בונוס (10 נקודות):

ממשו שניים או יותר מן השיפורים הבאים (שיפור יחיד יזכה ב-5 נקודות בונוס). על שם המחלקה להיות ImprovedFirstChoiceLocalSearch, ועליה לרשת מן FirstChoiceLocalSearch.

1. ממשו גם אופרטורים המאפשרים להחזיר תכונה או דוגמה שהוסרו בשלב קודם.
2. ממשו אלגוריתם חיפוש שונה, מבין אלו שנלמדו במהלך הקורס. הסבירו הסיבות לבחירת האלגוריתם.
3. ממשו מבחן סטטיסטי במקום או בנוסף להשוואת הדיוק. מבחן McNemar ([1], ראו גם ערך בויקיפדיה: http://en.wikipedia.org/wiki/McNemar%27s_test) מאפשר לנו להגיד האם אלגוריתם חדש שונה מהותית מאלגוריתם ישן, ע"י בדיקת מספר הדוגמאות עליהם הם לא מסכימים. במקרה זה, נגיד שמצב חדש טוב יותר ממצב ישן רק אם הוא טוב יותר כרגיל, וכן שונה מהותית לפי המבחן הנ"ל.
4. כל רעיון יצירתי או מעניין שעולה לכם בראש. יש להסביר היטב את הרציונל העומד מאחורי ההרחבה \ שיפור המוצע ומדוע הוא עשוי לסייע.

טיפים והנחיות

- לנוחיותכם מספר אופציות להורדה של הקוד המסופק:
  - כקובץ zip דרך האתר (Download ZIP): https://github.com/TechnionAI/Win14_15_HW3
  - על מנת ליצור עותק מקומי (clone) של הקוד, הריצו בסביבת Unix את הפקודה:
    ```
    git clone git://github.com/TechnionAI/Win14_15_HW3.git
    ```
    מספיק להריץ פקודה זו פעם אחת. על מנת למשוך שינויים עתידיים בקוד אם יהיו, פשוט הריצו:
    ```
    git pull
    ```
  - קיימים גם שלל כלים גראפיים לעבודה עם git בסביבת חלונות. פשוט חפשו "git for windows".

- ודאו כי בחירת האופרטור באלגוריתם החיפוש היא אכן אקראית.
- שימו לב שעל אופרטור לקבל מצב חוקי ולהחזיר מצב חוקי. במימוש המסופק, מצב כולל גם את קבוצת האופרטורים שניתן להפעיל עליו, כך שאם אתם משתמשים במימוש זה, עליכם לדאוג לכך.
- אינכם חייבים להשתמש במימוש לאלגוריתם הלמידה K-Nearest Neighbours המסופק בקוד. הוא נועד להיות יחסית קל להבנה ולא לדרוש ספריות חיצוניות, אך בעקבות זאת אינו יעיל במיוחד. ניתן להשתמש במימוש יעיל יותר, למשל זה המופיע בספריית Scikit-Learn.

[1] McNemar, Quinn (June 18, 1947). "Note on the sampling error of the difference between correlated proportions or percentages"

חלק ג' [רטוב] – הערכה אמפירית של השיפור (55 נק')

בחלק הקודם, ראינו כי יש אלמנט אקראי לאלגוריתם. מעבר לכך, במחקר אמפירי אין זה מספיק להראות שהאלגוריתם טוב יותר בהרצה אחת.

מהסיבות הנ"ל, כדי להיות משוכנעים שהאלגוריתם שהצענו טוב יותר, עלינו להראות שהוא טוב יותר בצורה שהיא **משמעותית סטטיסטית**, כלומר שלא סביר שהשיפור הוא עניין של מזל או וריאציות קלות בקבוצת האימון, אלא הוא נובע מהעובדה שהאלגוריתם עצמו טוב יותר.

כדי להראות זאת, נשתמש בחלק זה **במבחן סטטיסטי** המראה זאת, בשם Student's Paired t-test.

באופן כללי, המבחן נועד להשוות בין שתי מדידות שונות על אותם נבחנים (לכן Paired, מתייחסים לזוגות של לפני ואחרי) – למשל ציונים במבחן בקורס בינה מלאכותית של אותם סטודנטים במועד א' ובמועד ב'.

על מנת לעשות זאת, ננסה למדל את ההסתברות של תוצאות המדידות שלנו, תחת ההנחה (שאנו מקווים שתתברר כשגויה) כי השיפור אכן הגיע בצורה מקרית.

מטרת המבחן: להראות שיש שיפור משמעותי (לא כזה שנובע ממזל בלבד) בין שתי המדידות.

כדי לעשות זאת, נחשב:

$$t = \frac{\overline{X_D} - \mu_0}{S_D/\sqrt{n}}$$

כאשר:

n- מספר הדגימות \ הנבחנים

$\overline{X_D}$- ממוצע ההפרשים בין המדידות

S_D- סטיית התקן של ההפרשים בין המדידות

μ_0- קבוע. רוצים להראות כי ההפרש הממוצע בין המדידות הוא שונה מהותית מ-μ_0, לכן במקרה שלנו נבחר $\mu_0 = 0$.

את ערך t יש להציב בהתפלגות בשם Student's t-Distribution (עם פרמטר $n-1$) על מנת לקבל את ההסתברות שהמדידות הללו יתקבלו אם אכן ההפרש נובע ממזל בלבד. אם הסתברות זאת קטנה מסף מסויים (במקרה שלנו, נקבע סף של 0.05, כלומר 5%), נוכל להגיד די בביטחון שהשיפור אינו רק עניין של מזל, אלא הוא שיפור **מהותי סטטיסטית**. אם ההסתברות גבוהה מהסף, איננו יכולים להגיד בוודאות מספיקה כי זהו לא עניין של מזל ונאמר כי השיפור **איננו מהותי סטטיסטית**.

הערות:

- שימו לב כי ההפרדה למהותי\ לא מהותי היא בינארית בלבד: בעקרון, אין חשיבות **לכמה** ההסתברות גדולה או קטנה מהסף שנקבע מראש. קיים מושג נפרד של **רמת מהותיות**, שלא נדון עליו בקורס.
  - קיים קשר בין ההסתברות לבין רמת המהותיות, אך הוא מורכב.
  - כתלות במשימה ובדרישות מן המחקר, הסף הנדרש עשוי להיות פחות או יותר גדול.
  - מהסיבות הנ"ל, נפוץ לדווח את ההסתברות עצמה במחקר (המונח הטכני הוא p-value), ולא רק לציין את העובדה כי השיפור מהותי.
- אין צורך לממש את המבחן בעצמכם. קיים מימוש ב-Python המצורף בתוך הקוד. שימו לב שעל מנת לעשות בו שימוש, עליכם להתקין את המודול Scipy. ראו תרגיל התנסות בפייתון באתר.
- לחילופין, אתם רשאים להשתמש בגרסה של Microsoft Excel/Google Sheets (הפרמטרים הדרושים הם tails=2, type=1, ויש לחלק את התוצאה ב-2. שימו לב כי אם אין בממוצע **שיפור**, המספר המתקבל איננו נכון [למעשה המספר המתקבל יהיה הסיכוי שה**פגיעה** מהותית]).

א. ממשו גרסה של האלגוריתם הבודקת משמעות סטטיסטית:

1. בצעו חלוקת 10-Fold Cross Validation. לטובת קבוצת ההערכה, יש לשמור גם כן Fold אחד. במילים אחרות, בכל פעם ישנם 8 חלקים לטובת האימון, חלק אחד לטובת הערכה, וחלק אחד לטובת המבחן. יש לדאוג גם שהחלק בו משתמשים להערכה וגם החלק בו משתמשים למבחן משתנים כל פעם (כלומר לא לקחת את אותו Fold בתור קבוצת הערכה מספר פעמים).
2. לכל Fold, יש להריץ את האלגוריתם מחלק ב' ולשמור את שני הדיוקים (עם ובלי אלגוריתם החיפוש) ברשימות נפרדות
3. לבסוף, יש לבצע Student's Paired t-Test על שתי זוגות רשימות הדיוקים (לפני ואחרי חיפוש לוקאלי, לכל Fold, כלומר סה"כ 10 מדידות לפני ו-10 אחרי).
4. אם ההסתברות ששתי הזוגות שונים רק במזל קטנה מ-0.05, נגיד שההבדל **מהותי**. אם ממוצע הדיוקים עבור ההרצות בהן השתמשנו בחיפוש לוקאלי גבוה מזה של ההרצות ללא החיפוש, נגיד שישנו **שיפור מהותי**.

ב. הריצו את גרסה זו על המידע המסופק . יש לשמור את התוצאות לתוך קובץ (למשל ע"י ספריית pickle):

a. על מנת לטעון את המידע, השתמשו בפונקציה load_hw3_data_2 במודול data.py בקוד המצורף.

1. הריצו עבור אלגוריתם עץ החלטה ולפחות שני פרמטרים שונים עבור גודל עלה מינימלי.
2. הריצו עבור אלגוריתם K-Nearest Neighbours ולפחות שני פרמטרים שונים ל-K.
3. במידה ומימשתם את הבונוס, הריצו ריצה אחת לפחות עם אלגוריתם ופרמטרים לבחירתכם כאשר אתם משווים את הגרסה ללא חיפוש לוקאלי, את הגרסה ללא השיפורים שמימשתם, ואת הגרסה עם השיפורים שמימשתם.

סה"כ לפחות 4 הרצות ללא הבונוס, ולפחות 5 כולל הבונוס.

לכל הרצה יש להציג את הדיוק הממוצע לפני ואחרי חיפוש לוקאלי, וכן האם ההבדל הוא **מהותי סטטיסטית** והאם הוא מהווה שיפור מהותי. עבור הבונוס יש להציג דיוק ממוצע לשלושת המקרים, ולבצע בדיקת מהותיות סטטיסטית בין כל הזוגות. אין דרישה שהשיפורים המוצעים בבונוס יגררו שיפור מהותי על פני האלגוריתם הרגיל.

ג. סכמו את מסקנותיכם מן הניסויים. יש להתייחס הן לדיוק לפני ואחרי השינוי, והן למהותיות הסטטיסטית שלו. יש להציג תוצאות גם בעזרת גרפים ו\או טבלאות.

הוראות הגשה

- הגשת התרגיל תתבצע **אלקטרונית בלבד**.
- כהכנה להגשה, אתם מתבקשים ליצור בתוך התיקייה `search` שבקוד המסופק **תיקיה יחידה** בשם `AI3_<id1>_<id2>` (ללא הסוגריים המשולשים), שבתוכה הקבצים הבאים:
  - קובץ ריק בשם `__init__.py`.
  - קובץ המכיל מחלקה בשם FirstChoiceLocalSearch היורשת מן המחלקה LocalSearch בקוד המסופק. יש לכתוב מחלקה בשם LearningState היורשת מן SearchState.
  - במידה ומימשתם את הבונוס, קובץ עם מחלקה בשם ImprovedLocalSearch.
  - קובץ המממש את האלגוריתם לבדיקת מהותיות סטטיסטית.
  - כל קובץ עזר שכתבתם.
  - תיקייה בשם `part3results` ובה קובץ או אוסף קבצים המכילים את תוצאות הריצות בחלק ג'.
  - כל חבילה חיצונית בה השתמשתם (שלא אנחנו סיפקנו לכם), **אשר לא ניתן להתקין** באמצעות הפקודה `pip install`, זאת על מנת שיהיה אפשר להריץ את הקוד שלכם על כל מחשב.
  - קובץ `libs.txt`, המכיל את שמות כל החבילות בהן השתמשתם ואשר **ניתן להתקין** באמצעות הפקודה `pip install`. הקובץ יכול להיות ריק. הקובץ צריך להיות בפורמט הבא:

```
matplotlib
numpy
scipy
sklearn
```

  - קובץ בשם `AI_HW3.PDF`, המכיל את התשובות לחלק היבש והערות מיוחדות הנוגעות לקוד במידה ויש צורך בכך.
  - קובץ בשם `readme.txt` בפורמט הבא:

```
name1 id1 email1
name2 id2 email2
```

- את התיקייה `AI3_<id1>_<id2>` יש לקבץ לארכיון בשם `AI3_<id1>_<id2>.zip`, **שאותו (ואותו בלבד) עליכם להגיש.**
- אין צורך להעתיק את הקבצים המסופקים לכם אל תוך תיקיית ההגשה. הניחו כי קבצים אלו יהיו זמינים בעת בדיקת התרגיל.
- שימו לב שכל הפנייה למיקום קובץ/תיקייה כלשהם בקוד תהיה רלטיבית (relative path) ולא אבסולוטית, כך שהקוד יעבוד כפי שהוא על כל מחשב בכל מיקום שנבחר לתיקיית הפרויקט. הקפידו לבדוק זאת לפני ההגשה!
- הקפידו על קוד ברור, קריא ומתועד ברמה סבירה. עליכם לתעד כל חלק שאינו טריוויאלי בקוד שלכם.
- אתם רשאים לעשות שימוש בכל קוד שתמצאו ברשת, אך כל קוד חיצוני מצריך הצהרה מפורשת על המקור שלו בקובץ `AI_HW3.PDF`.

בהצלחה!