

## חלק יבש – הבנת הבעיה

1. קבוצת המצבים היא כל האפשרויות לבחירת דוגמות ותכונות מתוך קבוצת האימון. כלומר כל מצב מכיל אוסף של דוגמות, ואוסף של תכונות, ואנו מחפשים את המצב כך שאלגוריתם הלמידה שהוא ייצר יהיה הטוב ביותר (אל מול קבוצת validation נוספת, אשר לא לומדים ממנה ומשמשת רק לבדיקת טיב המצב).
2. המצב ההתחלתי הוא המצב אשר מכיל את כל הדוגמות ואת כל התכונות הקיימות בקובץ המידע.
3. קבוצת האופרטורים הם כל האופרטורים הלוקחים מצב כלשהו מקבוצת המצבים ומחזירים מצב כלשהו מקבוצת המצבים. במקרה שלנו קבוצת האופרטורים תהיה כאלו שמורידים תכונה מסוימת (מהתכונות שנותרו במצב הנוכחי) או מורידים מספר דוגמות, במקרה שלנו 5.
4. פונקציית ה-fitness היא אחוז הדיוק בסיווג קבוצת הוולידציה (אשר כאמור, אנו לא לומדים ממנה). אנו נאמין כי מסווג בעל אחוז דיוק טוב יותר בקבוצת הוולידציה הינו טוב יותר.

## חלק רטוב – מימוש האלגוריתם וניסוי

3. לפני החיפוש הלוקלי קיבלנו דיוק של 0.75, ואחרי החיפוש הלוקלי קיבלנו דיוק של 0.75, שניהם יצאו לנו שווה.
  4. בהרצה אחרת קיבלנו שאחרי החיפוש הלוקלי הדיוק היה 0.8 ובהרצה אחרת קיבלנו שהדיוק היה 0.7. כפי שניתן לראות התוצאות אינן זהות לתוצאה הראשונה, והדבר כמובן נובע מכך שהחיפוש הלוקלי הוא סטוכסטי, כלומר הוא הסתברותי ולכן בכל הרצה נגיע לאופטימום לוקלי אחר, מה שישפיע על התכונות הנבדקות ולכן גם על הדיוק.
- בנוסף: בחרנו ליישם גם Random restarts עבור החיפוש הלוקלי שבנינו. ראינו כי התוצאה הסופית תלויה מאוד ברנדומיות של הפתרון ולכן נריץ מספר פעמים ונשמור את התוצאה הטובה ביותר (המצב ההתחלתי כל פעם זהה אבל הבחירות הרנדומיות שונות ולכן נקבל כל פעם דיוק אחר).
- לא היה לנו זמן לממש את הרעיון השני, אבל חשבנו על רעיון די מגניב, במקום להתחיל ממצב התחלתי עם כל הדוגמאות וכל הפיצרים, להתחיל עם פיצר אחד וכל הדוגמאות, ולהוסיף פיצרים עד שמגיעים לאופטימום לוקלי. חשבנו שאולי כך נגיע לשינויים יותר משמעותיים (כי ראינו ש-LR עושה גג 4 שלבים ולכן לא משנה הרבה את ה-training\_set). (לא מימשנו אבל הרעיון שווה ניקוד לא? ☺).

## חלק רטוב – הערכה אמפירית של השיפור

תוצאות ההרצות:

מס' מבחן	אלגוריתם	פרמטר	דיוק ללא LR	דיוק עם LR	האם דיוק עם LR משופר?	האם רגיל שיפר?	האם רגיל משמעותי?	האם LR משופר שיפר?	האם LR משופר משמעותי?
1	KNN	K=1	0.583	0.574	0.584	לא	לא	כן	לא
2	KNN	K=3	0.646	0.608	0.647	לא	כן	כן	לא
3	DT	L=1	0.579	0.506	0.526	לא	לא	לא	לא
4	DT	L=3	0.584	0.527	0.536	לא	לא	לא	לא

כפי שניתן לראות, התוצאות די מאכזבות. LR הרגיל לא שיפר בשום מקרה, והיה אף מקרה שהיה יותר גרוע באופן מהותי. האלגוריתם המשופר השיג תוצאות יותר טובות עבור המקרה של KNN אך גם זה היה בדיוק ממש זניח (0.001 הבדל).

מה שכן, ניתן לראות כי באופן עקבי האלגוריתם המשופר של LR משיג תוצאות יותר טובות מ-LR הרגיל. מה שהגיוני כי האלגוריתם המשופר פשוט מריץ את האלגוריתם הרגיל מספר פעמים ומחזיר את המסווג שנתן את התוצאה הגבוהה ביותר מבין ההרצות.

ההסברים שמצאנו לתופעות:

1. אולי הקלט לא מייצג מרחב למידה אמיתי, ולכן הפיצ'רים ש-LR מוריד לא באמת משנים כלום (או שלפחות חלק מהפיצ'רים סתם מוגרלים ולכן דופקים את LR כשהוא מגריל פיצ'רים אחרים, כי הוא יסתמך עליהם יותר לאחר שהוא יסיר פיצ'רים "אמיתיים")
2. האופטימום הלוקלי עבור סיווג של מספר דוגמאות קטן לא באמת מצביע על אופטימום לוקאלי עבור מספר דוגמאות יותר גדול, יכול להיות שאם ה-evaluation set היה יותר גדול היה פחות "רעש" והיינו מקבלים תוצאות שיותר קרובות למסווג אמיתי.
3. הטופולוגיה של מרחב המצבים לא מתאימה ל-LR, כלומר יש הרבה אופטימומים לוקליים בעלי ערך בערך אותו הדבר ולכן האלגוריתם לא באמת משנה כלומר (כלומר המרחב שלנו די "שטוח").
4. יכול להיות שהמצב ההתחלתי (בו משתמשים בכל הפיצ'רים וכל הדוגמאות) הוא לא מצב התחלתי טוב וצריך היה להגריל מצב התחלתי אחר כל פעם (למשל להתחיל מהורדת פיצ'ר כלשהו מההתחלה, או להתחיל ממצב ריק ולהוסיף פיצ'רים במקום להוריד).
5. יכול להיות שיש כמה folds ממש גרועים בקלט (כלומר ההתפלגות של הדוגמאות לא ממש אחידה) וזה דופק לנו את ההסתברות כשמחשיבים אותו עם שאר ה-folds.