Co-Training Classifier

אופן מימוש האלגוריתם

API

את האלגוריתם מימשנו כ-scikit-learn classifier, כלומר יצרנו מחלקה בשם **CoTraining** אשר יורשת מהמסווג הבסיסי **BaseEstimator** של scikit-learn.

לשם בניית ה-API הנדרש מימשנו את הפונקציות הבאות כחלק ממימוש במסווג:

- **Fit(X_train, y)** – עבור סט תצפיות האימון X המתויגות והלא מתויגות בצירוף וקטור הסיווגים y (של התצפיות המתויגות). מאמן את המסווג בשיטת Co-Training.
- **Predict_proba(X_test)** - עבור סט תצפיות הבדיקה X_test. מעריך את ההסתברות לכל אחד מה-classes האפשריים (מדובר בבעיית סיווג בינארית), עבור כל תצפית בדיקה.
- **Predict(X_test)** - עבור סט תצפיות הבדיקה X_test. חוזה את ה-class של כל אחת מרשומות הבדיקה, על פי ממוצע ההסתברויות לכל class, כפי שמחזירה הפונקציה Predict_proba(X_test).

הכנת הנתונים

1. המימוש תומך רק בקבצי csv כ-dataset, ולכן קבצי אימון שהיו בפורמט txt המרנו לפורמט csv.
2. חלק מה-datasets לא מכילים חלוקה ברורה של התכונות לשני views, ולכן ב-datasets אלו הגרלנו קבוצת תכונות לפי פרופורציה נתונה (ברירת המחדל היא לחלק את התכונות בצורה שווה לשני views (V1_fraction=V2_fraction=0.5), אך המימוש תומך גם בביצוע חלוקה לפי פרופורציה נתונה.
3. לא מצאנו datasets המכילים דוגמאות מתויגות ולא מתויגות ולכן מחקנו (ע"י השמה של ערך NaN) 80% מהתיוגים, כאשר ביצענו את תהליך האימון של המסווג.

Evaluation

1. ביצענו כמה ניסויים על 5 קבצי אימון המותאמים לבעיית Classification, חלקם מכילים תכונות נומריות בלבד, וחלקם מכילים גם תכונות קטגוריאליות. רוב הקבצים הורדו מה-UCI Repository.

Dataset	Number of Instances	View 1 Features	View 2 Features
Adult	30,162	age, marital-status, sex, race, relationship, native-country	workclass, fnlwgt, education, education_num, occupation, capital-gain, capital-loss, hours-per-week
Airlines	18,272	Airline, DayOfWeek, Time, Length	Flight, AirportFrom, AirportTo
Electricity	18,531	Date, Day, Period	Nswprice, Nswdemand, Vicprice, Vicdemand, Transfer
Hyperplane	30,000	Attr1, Attr3, Attr5, Attr7, Attr9	Attr2, Attr4, Attr6, Attr8, Attr10
SEA	60,000	Attr1, Attr3	Attr2

- הסרנו רשומות שהכילו ערכים חסרים. מספר הרשמות המצוין בטבלה הוא לאחר הניקוי.
- תוך שימוש ב-**10-fold Cross Validation**, השווינו את **מדדי AUC ו- F1-score** עבור שני מסווגים בשיטת ה-Co-training מול אותם מסווגים בשיטתם הרגילה. בנוסף מדדנו את זמן החיזוי וזמן הלימוד בנפרד עבור כל שיטה, וזאת באמצעות הפונקציה cross_validate של הספרייה scikit-learn.

התוצאות מוצגות בטבלאות להלן (בעמוד הבא).

2. השוואה מול שיטות אחרות:

- את ה- CoTraining השווינו לשיטת הלימוד הרגילה (supervised) של שני מסווגים בסיסיים:
  - **Random Forest**
  - **Logistic Regression**
- את ביצועי האלגוריתמים מדדנו באמצעות שני מדדים:
  - **AUC** (השטח מתחת לעקומת ה-ROC).
  - **F1-score** (שהוא ממוצע הרמוני של מדדי precision ו-Recall).
- ניסינו מספר ערכים אפשריים ל-hyper parameters של k (מספר האיטרציות) ו-G (מספר הרשומות הלא מתויגות שנוסיף בשלב ה-cross-over.
- מכיוון שבבעיית ה-Co-Training יש תיוגים רק לחלק מהרשומות, כדי להשוות את הביצועים לאלגוריתמים ה-supervised, הגרלנו את ערכי ה-class של הרשומות הלא מתויגות, כדי לדמות מצב של רשומות לא מתויגות ביחד עם מתויגות (semi-supervised) וזאת כדי שההשוואה תהיה הוגנת.

3. תוצאות

1. בטבלאות הבאות מצוינות תוצאות ריצה של אלגוריתם CoTraining עם פרמטרים בסיסיים של 10 איטרציות ו-5 רשומות שנוספו בכל שלב של cross-over ועבור שני מסווגים. ניתן לראות כי ל-CoTraining יש יתרון קטן בבחינת הדיוק (במונחי AUC) עבור ה-Dataset. עבור יתר הקבצים, נראה כי הדיוק יוצא נמוך יותר מאשר לימוד של מודל רגיל. עבור F1-score, התוצאות דומות לתוצאות ה-AUC מבחינת יתרון למודל הרגיל.
2. מבחינת זמן הריצה, יש גידול משמעותי בזמן הלימוד עקב לימוד מודלים חלקיים עבור כל L1 ו-L2, וסיבוכיות זמן הריצה גדלה כפונקציה של מספר האיטרציות K. זמן הבדיקה עדיין נמוך מכיוון שבתהליך זה רק ממצעים את ההסתברויות שנתנה פונקציית ה-predict.

Parameters:	K (# of iterations)	G (Cross-over # records)	Base Learner	penalty	C	max_iter
	10	**5**	**Logistic Regression**	**L2**	**1**	**100**

	Standard training				Co-training			
DataSet Name	**AUC**	**F1-score**	**Training Time (Sec)**	**Testing Time (Sec)**	**AUC**	**F1-score**	**Training Time (Sec)**	**Testing Time (Sec)**
Adult	0.744	0.787	0.333	0.003	0.768	0.79	4.108	0.036
Airlines	0.605	0.581	0.126	0.003	0.603	0.579	2.39	0.026
Electricity	0.88	0.791	0.049	0.002	0.795	0.624	2.817	0.0317
Hyperplane	0.841	0.763	0.069	0.001	0.778	0.722	3.663	0.035
SEA	0.878	0.848	0.115	0.003	0.876	0.765	7.096	0.046

Parameters:	K (# of iterations)	G (Cross-over # records)	Base Learner	n_estimators	criterion	max_depth	min_samples_leaf	min_samples_split
	10	**5**	**Random Forest**	**100**	**gini**	**5**	**1**	**2**

	Standard training				Co-training			
DataSet Name	**AUC**	**F1-score**	**Avg. Training Time (Sec)**	**Avg. Testing Time (Sec)**	**AUC**	**F1-score**	**Training Time (Sec)**	**Testing Time (Sec)**
Adult	0.901	0.845	1.379	0.065	0.901	0.805	24.846	0.524
Airlines	0.67	0.631	1.3	0.052	0.649	0.625	13.414	0.16
Electricity	0.888	0.796	0.968	0.042	0.871	0.745	10.767	0.082
Hyperplane	0.83	0.742	4.758	0.079	0.758	0.7	24.952	0.146
SEA	0.875	0.842	4.4806	0.127	0.871	0.763	36.345	0.219

3. בשלב השני של הניסויים, בוצע ניסיון לבדוק את השפעת הפרמטרים K ו-G על הביצועים.
הטבלאות הבאות מראות לכל אחד מאלגוריתמי הסיווג הבסיסים, כיצד משפיעים הפרמטרים על הביצועים:

Random Forest	Co-training with K=10			
DataSet Name	**G=5**	**G=20**	**G=50**	**G=100**
Adult	0.901	0.902	0.902	0.902
Airlines	0.649	0.649	0.65	0.649
Electricity	0.871	0.866	0.867	0.868
Hyperplane	0.758	0.758	0.758	0.756
SEA	0.871	0.871	0.872	0.871

Logistic Regression	Co-training with G=20			
DataSet Name	**K=10**	**K=20**	**K=50**	**K=100**
Adult	0.902	0.902	0.901	0.901
Airlines	0.65	0.647	0.65	0.645
Electricity	0.867	0.864	0.87	0.867
Hyperplane	0.758	0.758	0.758	0.758
SEA	0.872	0.871	0.871	0.871

ציפינו לראות שיפור בדיוק של מסווג ה-CoTraining עבור מספר איטרציות (K) גדול יותר, כלומר מספר תצפיות מתויגות גדול יותר, אך **השינוי היה זניח** (כמה עשיריות האחוז) ולכן ניתן להסיק כי מספר איטרציות גבוה יותר לא יביא בהכרח לשיפור. **התוצאות הטובות ביותר התקבלו דווקא עבור K=10** (שוב, בהפרש מאוד קטן וכנראה לא מובהק סטטיסטית).

ציפינו לראות שיפור כלשהו כאשר מכניסים יותר תצפיות מתויגות למודל, כלומר כאשר מגדילים את ערכו של הפרמטר G, אך **גם פה לא ראינו שיפור משמעותי** (שיפור של כמה עשיריות).
השערתנו היא שהחולשה של האלגוריתם CoTraining מצויה בעובדה שתחילה מספר הרשומות המתויגות מאוד קטן ולכן האיטרציות הראשונות מהוות מעין "זמן חימום" של המודל, בו המודל אינו מדויק.

4. מסקנות:

- <u>זמן ריצה</u>: כפי שציפינו, זמן הריצה של אלגוריתם ה-Cotraining ארוך יותר ב<u>זמן הלימוד</u>, וזאת עקב ביצוע מספר לימודים עם מספר רשומות מתויגות שונה בכל איטרציה. בדיקת ההסתברויות של כל סווג של סט הבדיקה הגדיל אף יותר את זמן הלימוד (שלב ה--cross over). עם זאת, זמן הבדיקה נשאר נמוך, כמו בשיטות הרגילות.
- <u>המסווג הבסיסי</u>: שני המסווגים, Random Forest ורגרסיה לוגיסטית (עם פרמטרים דיפולטיים) נותנים תוצאות דומות, ולכן בניסויים שלנו לא הצלחנו לקבוע חד-משמעית כי קיימת השפעה של המסווג הבסיסי. אנחנו מניחים כי עבור כל מסווג פשוט (לא רשת נוירונים) אפשר להגיע לאחוזי דיוק דומים לתוצאות שהשגנו. עבור מודלים מורכבים, איננו מצפים לראות שיפור משמעותי בשימוש ב-CoTraining.
- <u>מספר האיטרציות ומספר הרשומות G</u>: ככל שהגדלנו את מספר האיטרציות ראינו שיפור מאוד נמוך בדיוק AUC ובמדד ה-F1-score שכנראה אינו מובהק סטטיסטית. יתכן כי datasets בעלי מספר רב של תכונות עדיפים על פני קבצים עם מעט תכונות. מספר הרשומות שהוספו לסט הרשומות המתויגות הראה תוצאה דומה, כלומר שיפור זניח מאוד.
- <u>באופן כללי</u>, לשיטת CoTraining יש חסרון בכך שקיים "זמן חימום" בו המודל מאומן על מספר רשומות מתויגות קטן מדי, ורק מגדיל את מספר הרשומות המתויגות באיטרציות הבאות. לדעתנו, עובדה זו יכולה לרמוז על חוסר השיפור שראינו בתוצאות שלנו. בנוסף, האלגוריתם תלוי בחלוקה של התכונות ל-V1 ו-V2. על אף שהשתדלנו לבצע חלוקה כמה שיותר נכונה (על פי סוג ה-dataset), לא הצלחנו להראות כי עובדה זו מביאה לשיפור בתוצאות. באותה מידה, יכולנו לבצע חלוקה רנדומלית של התכונות ולהשיג תוצאה דומה. לא מצאנו יתרון מובהק בשימוש בשיטת לימוד זו על ה-datasets שנבחרו.

יש לציין כי במקרים בהם מבוצעת השוואה בין תוצאות אלגוריתמים שונים, היה נכון לבצע מבחן סטטיסטי ולבדוק את המובהקות הסטטיסטית של ההבדלים, אך במסגרת התרגיל הזה לא נדרש.

הוראות להרצת הקוד

1. יש להוריד את ה-Datasets (מצורפים לתיקיית הפרויקט) אל תיקייה אחת, יחד עם קבצי קוד המקור.
2. במידה ורוצים להריץ את dataset חדש, יש לספק את הנתיב המלא שלו שכפרמטר ראשון של ה-script.
3. דרך שורת הפקודה (cmd) ניתן להריץ את הקוד ע"י כתיבת שם הקובץ יחד עם הפרמטרים של מספר האיטרציות (K) ומספר הרשומות הלא מתויגות להוספה (G). כמו כן יש לציין בדיוק איזה תכונות (האינדקסים שלהם) ירכיבו כל View.

לדוגמא: עבור אלגוריתם מסווג בסיסי של Random Forest עם שני views באינדקסים v1=[0,1] ו- v2=[2,3], ועבור הפרמטרים k=10, g=50 נרשום (**אין להכניס רווחים מיותרים בין אינדקסים בתוך הסוגריים המרובעות!**)

python CoTraining.py c:\Users\User\Adult1.csv [0, 5, 8, 9, 7, 13] [1, 2, 3, 4, 6, 10, 11, 12] RandomForestClassifier 10 50

פלט לדוגמא:

Dataset=Adult1.csv

K=10, **G**=50

Average training time= 58.016699934 seconds

Average testing time= 0.159700036049 seconds

Mean AUC= 0.758416354159

Mean F1-score= 0.692866666667