מבוא למערכות לומדות
236756
אביב 2012
תרגיל 2

מגישים:
דולב רביב – 065961492
עמית גרוס – 036569598

תאריך:
5.08.2012

קוד מקור - http://code.google.com/p/ml-dolev-amit/ (הרשאות תינתן על פי דרישה)

הגדרת הבעיה

בתרגיל זה החלטנו לבחון האם נוכל להעריך מהי חוות הדעת של כותב ביקורת על מוצר לגביו, על סמך טקסט התגובה עצמה. למעשה ברצוננו לבצע ניתוח של סנטימנט התגובה - כלומר, האם חוות דעת הכותב חיובית, שלילית או ניטרלית ביחס למוצר. מאחר וביקורת על מוצר תלויה באופי המוצר (לדוגמה מחשב שמתחמם מהר זה דבר רע ותנור כזה דווקא מאוד רצוי), החלטנו לבנות את הפתרון שלנו מסביב למצלמות ואביזרים למצלמות.

פתרון הבעייה יכול להועיל במספר תסריטים שונים לדוגמת:
(1) המלצה על דרוג (חיובי/נטרלי/שלילי) לכותב הודעה בזמן כתיבת ההודעה.
(2) שימוש במסווג בפורומים בהם אין דרוג לזיהוי חוות דעת משתמשים, כדוגמת "טוקבקים".
(3) מציאת "מקרים חריגים" – מקרים בהם תוכן ההודעה מאוד תומך, אך ציונה נמוך, ולהתריע לכותב/מנהל האתר
(4) עם פיתוח כיוון זה, אולי ניתן יהיה לפקח על ניטרליות עיתונאים בצורה אלגוריתמית – כך שכתבים לא יורשו לפרסם מאמרים מוטים כמאמרים שאינם מאמרי דעה.

איסוף הדוגמאות

לצורך איסוף הדוגמאות ביצענו זחילה ב Amazon.com והורדנו מוצרים שעונים על השאילתה "lenses". לאחר מכן לקחנו כל מוצר ובעזרת הסקריפט מ http://www.esuli.it, הוצאנו את התגובות למוצר בתוספת הדירוג שהמשתמש נתן למוצר, הכותרת והתוכן. את הדוגמאות חילקנו לדוגמאות אימון ודוגמאות מבחן ביחס של 3:7 (2240 דוגמאות אימון ו 960 דוגמאות מבחן). האמת המוחלטת שלנו הינה דרוג המשתמש (כוכבים באמזון), כאשר 1-2 כוכבים מציינים דעה שלילית לגבי המוצר, 3 כוכבים מציינים דעה ניטרלית לגביו, ו4-5 כוכבים מציינים דעה חיובית על המוצר.

תיאור הפתרון

הגדרת Features

מאחר ומדובר בטקסט השתמשנו בפתרון המומלץ והידוע של Bag of Words. בחרנו מילים מהכותרת ומהתוכן של הביקורת וספרנו כמה פעמים הן הופיעו. את המילים לקחנו לאחר stemming, וכמו כן הורדנו stop words. בנוסף, הסרנו מילים שהופיעו בכותרת המוצר. לבסוף הוספנו את כל צמדי המלים (Bi-Grams) שהופיעו בטקסט בתור features נוספים. בסך הכל קיבלנו בצורה זו 12050 מאפיינים על פני 2240 דוגמאות האימון.
למעשה, כל מופע הכיל רשימת תכונות, שכל אחת מהן ציינה מילה. ציון המילה היה מספר המופעים (tf) של אותה מילה בטקסט, כאשר הפרדנו בין מילים שהופיעו בכותרת לבין מילים שהופיעו בתוכן. בחרנו לא להשתמש ב idf במחקרינו, מכיוון שלדעתנו המדד יותר חשוב עבור מובהקות של מסמך, ופחות בהקשר של מובהקות של סיווג. חשבנו על מספר וריאציות ל idf שניתן להוסיף – אך לבסוף החלטנו שדבר זה מחוץ ל scope הפרוייקט.

בחירת מאפיינים

כאשר יש בידינו למעלה מ12000 מאפיינים, נראה כי בידינו יותר מידי מאפיינים, לכן, בהתאם לדרישות התרגיל בחנו מספר דרכים לצמצום מרחב המאפיינים. מרחב המאפיינים הגדול הקשה עלינו בניסיון צמצום הבעיה, שכן מרבית האלגוריתמים המעניינים רצים בזמנים ריבועיים ואף יותר במספר המאפיינים, ומכיוון שרצינו לבחון את בחירת המאפיינים עבור פרמטרים שונים ועבור אפשרויות שונות למספר המאפיינים, אפשרויות אלה היו לא ריאליות מבחינת זמני ריצה. מספר פתרונות קלאסיים כגון חיפוש יוריסטי (חמדן וגנטי) או PCA, שרצינו לבחון נפלו בשלב זה. לבסוף החלטנו לבחון שיטות מבוססות ציון המחושב מהאנטרופיה של המאפיין, שיטת חיפוש סטוכסטית ושילוב בין דירוג תכונות להורדת מימדים בעזרת PCA. על אלה נפרט בהמשך.
בנוסף לאלו היה ניסיון לדרג תכונות בשיטה נוספת, בנינו עץ סיווג C4.5 (ללא שלב הגיזום) ונתנו ציון לתכונות על סמך העומק בו הן הופיעו. מאחר ובעץ ששואף להיות מאוזן בעל 2000 עלים יש סדר גודל של 200 צמתים שיטה זו לא אפשרה לנו לבחור מספיק מאפיינים על מנת לבצע סיווג מדויק בעזרת המסווגים שבחרנו.
לצורך בדיקת איכות כל פתרון, כפי שמוצג בגרפים בהמשך – ביצענו cross validation, עם 3 folds עבור ה feature selection, ו – 2 folds עבור ה parameter optimization.

דירוג תכונות

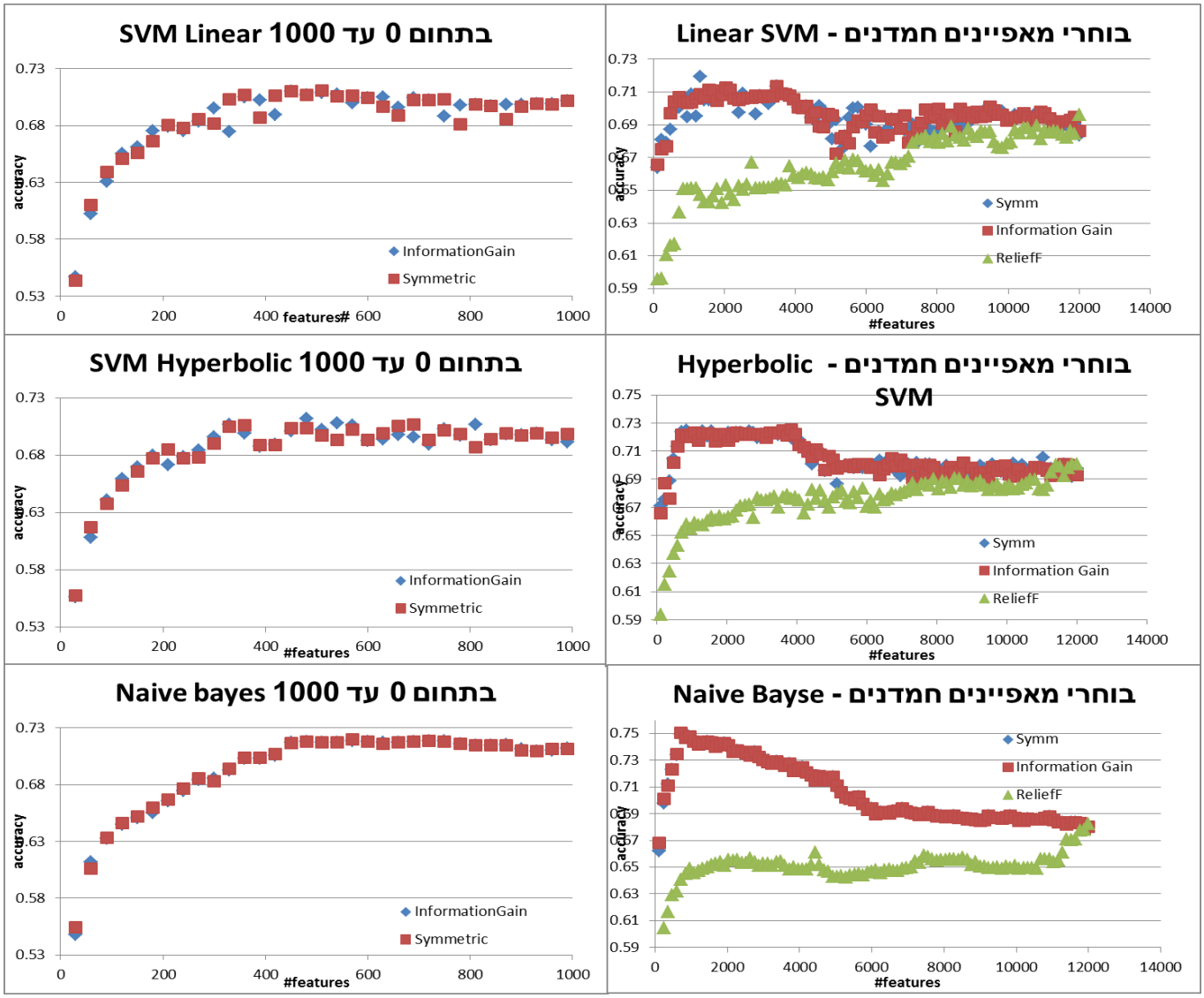
שיטה זו נותנת ציון לכל אחד מהמאפיינים, וכאשר בוחרים n מאפיינים מקבלים את n אלו שקיבלו ציון הכי גבוה. אנו השתמשנו בשלוש שיטות לקביעת ציונים אלו:

1. $InformationGain(att, Sempels) = H(Samples) - H(Samples|att)$
2. $SymmU(att, Sempels) = \frac{2\cdot(H(Samples)-H(Samples|att))}{H(Samples)} + H(Samples|att)$

3. אלגוריתם ReliefF המתואר במאמר[1] . רעיון האלגוריתם הוא עבור instance ממחלקה כלשהי c, למצוא את ה KNN שלו מאותו המחלקה (Nearest Hits), ואת ה KNN שלו ממחלקה אחרת (Nearest Misses), בהתאם לשונות של שתי הקבוצות – מתקבל ציון לכל מאפיין.

כאשר H מסמן את האנטרופיה, כפי שנלמד בהרצאה.

שיטות 1,2 דומות במהותן אך שונות בציון הסופי שהן נותנות לכל תכונה. מדד IG מחשבת את ההפרש בין האנטרופיה לאנטרופיה המותנית (מה מוסיפה תכונה זו לאנטרופיה). לעומתה SU, מחשבת את הפרש האנטרופיה מהאנטרופיה המותנית המנורמלת ומוסיפה לה את האנטרופיה היחסית של המאפיין. לשני החישובים השתמשנו בספריית הקוד הפתוח weka.



עבור השוואת Information Gain ו SymmU לעומת ReliefF, קיבלנו שהשניים הראשונים טובים משמעותית מהראשון – דבר זה נראה היטב בגרף, וכן לפי מבחן סטטיסטי ווילקוקסון[2], קיבלנו p_value שואף ל-0 (סדר גודל של 10^{-50}, לא ניתן להיות מדוייקים – שכן טווח הטעות הנומרית משמעותית במספרים נמוכים אלו).

עבור השוואת Information Gain לעומת SymmU: שמנו לב שעבור בחירת יותר מ 10% מהמאפיינים, השיטות בוחרות מאפיינים בצורה דומה אחת לשניה, ולכן התוצאות המתקבלות זהות כמעט לחלוטין. למעשה שבדקנו 100 דגימות בטווח [0,12000) – מקבלים גרפים מאוד קרובים החל מ10%, ולא ניתן להפריד סטטיסטית ביניהם (p_value=73.5% לפי מבחן ווילקוקסון)

ניסינו לבדוק האם המצב משתנה כאשר מתמקדים בטווח [0,1000), שכן זהו הטווח ה"מעניין" – בו קיבלנו את האחוז הגבוה ביותר עבור Naïve Bayes, שהיה רגיש במיוחד למספר המאפיינים, ותוצאות טובות גם עבור SVM. בדקנו 33 דגימות בטווח זה,

[1] An adaptation of Relief for attribute estimation in regression: Marko Robnik-Sikonja , Igor Kononenko

[2] כשהשתמשנו במבחן ווילקוקסון התחשבנו ברשימת זוגות, כאשר לכל זוג, יש את אותו המסווג ואותו מספר מאפיינים נבחר. הערה זו נכונה לכל מקום בו השתמשנו במבחן ווילקוקסון – אלא אם כתוב אחרת.

עבור כל מסווג (99 דגימות בסך הכל), וקיבלנו p_value=23%, ולכן לא נוכל להסיק מובהקות סטטיסטית לגבי יתרון של אחד המסווגים על פני השני.
ולכן, באופן שרירותי – בחרנו מעתה והלאה להמשיך לעבוד עם Information Gain.

חיפוש סטוכסטי

זוהי שיטת חיפוש חמדנית במרחב המאפיינים. השיטה מורכבת ממספר שלבים, תחילה אנו נותנים ציון לכל המאפיינים בעזרת IG, ציון זה מכתיב את ההסתברות להיבחר, זאת ביחס הפוך לציון שקיבלה. לאחר מכן בכל שלב אנו בוחרים כ 50 תתי קבוצות (נתון לשינוי על ידי פרמטר) בגודל 1% מכמות התכונות ההתחלתית (נתון לשינוי על ידי פרמטר). את התכונות שבחרנו אנו מנפים מהקבוצה שנותרה בסיום השלב הקודם ומבצעים הערכה לאיכות קבוצות התכונות החדשות. את השערוך הזה אנו מבצעם באמצעות מסווג מהיר (Naïve Bayes). לבסוף בוחרים את הקבוצה בעלת הציון המקסימלי. אנו חוזרים על התהליך עד להגעה למספר התכונות הרצוי. המימוש בשלב זה הוא מימוש שלנו פרט לדירוג התכונות. נשים לב שעבור מסווג זה אנו מצפים להטייה כלשהי לטובת Naïve Bayes, שכן אנחנו בוחרים את התכונה העדיפה לפי המסווג בו אנו משתמשים. החלטנו להשתמש ב Naïve Bayes בכל זאת, מכיוון שהוא מסווג מהיר – דבר שחשוב לנו באלגוריתם זה.
את החיפוש ביצענו יורד (הורדת תכונות מאוסף מלא במקום הוספת תכונות מגודל 0) מטעמי פשטות מימוש. בדיעבד, ייתכן ובחירה זו הייתה מוטעית, שכן לרוב מספר ה features ה"נכון" קרוב יותר ל-0, וייתכן מאוד שהתחלה מ-0 ועלייה כלפי מעלה הייתה משפרת את ביצועי השיטה, סוגיה זו מפורטת קצת יותר בסוף המסמך, בסעיף "רעיונות מחקר להמשך".
יתרון נוסף לשיטה הוא שהיא any time – על ידי הגדלת מספר תתי הקבוצות ו/או הקטנת ה"קפיצות" (גודל תתי הקבוצות), צפוי כי נשפר את התוצאות. בנוסף, מכיוון שהבחירה סטוכסטית – ניתן להפעיל שיטה זו מספר פעמים ועל ידי כך לקבל מרכיב נוסף לכוונון בהתאם לזמן הנתון.

שילוב IG עם PCA

כפי שציינו, מספר המאפיינים הרב מנע מאיתנו לבצע PCA על כלל המאפיינים. לכן, החלטנו לבצע אלגוריתם זה על מספר מצומצם של תכונות (2000). לקחנו את דירוג התכונות IG, כפי שתואר לעיל, חילקנו אותו לקבוצות של 2000 מאפיינים. את שיטת הורדת המימד PCA, הפעלנו רק על הקבוצה האחרונה (בעלת הציונים הנמוכים), שנבחרה עבור אותו מספר דרוש של features. על מנת לקבל מספרים שונים בתחום של הקבוצה האחרונה הגדלנו והקטנו את ערך השונות. ערך השונות הוא הסכום המנורמל של הערכים העצמיים של הווקטורים המתאימים, והגדלתו מביאה יותר מאפיינים של מאפיינים (מתוך ה 2000). את שני החלקים מימשנו בעזר ספריית weka.

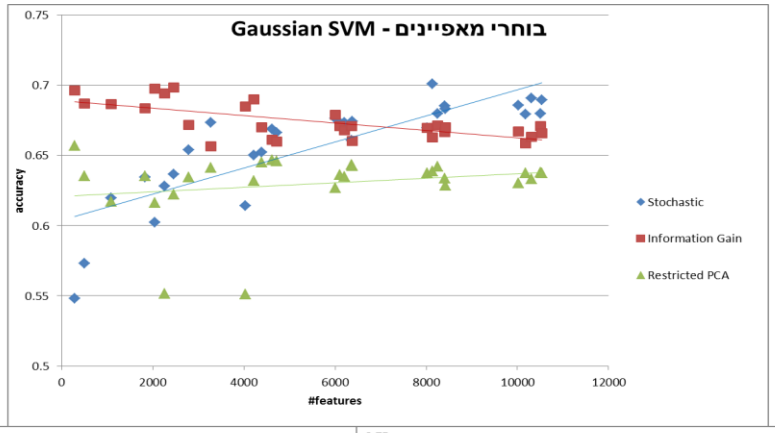
ניתוח התוצאות

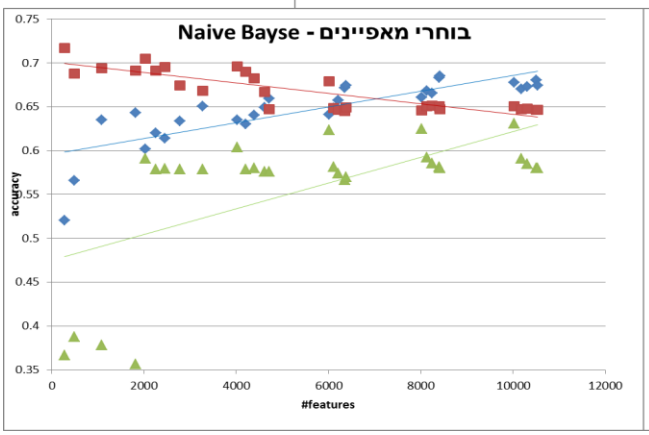
(*) הניתוח מתייחס לגרפים המופיעים בעמוד הבא.
עבור מספרי הפיצ'רים הגבוהים, ניתן לראות קיבוץ של דגימות לאחר כל כפולה שלמה של 2000. זאת אנו מסבירים על ידי כך שאנו בוחרים בשיטה המשלבת PCA. המאפיינים "מעל" ל k*2000 נבחרים על פי בחירת PCA – המשתמשת בחישוב וקטורים עצמיים של מטריצת השונויות. ככל שאנו עולים במספר המאפיינים, משתמשים במאפיינים "פחות אינפורמטיביים", ולכן יש יותר ערכים עצמיים עם ערך עצמי ששואף לאפס, ולכן חוזרים פחות וקטורים שעברו את הרף.
אם מתבוננים בגרף של Naive Bayes ניתן להבחין בדגימה שחורגת בתחילת כל חמישיה זאת מאחר ואנו מתקרבים לכפולה שלמה של 2000. כל המאפיינים עד תכונה זו נבחרים על פי Information Gain, ולכן דגימה זו תהיה קרובה יותר לדגימה המקבילה בשיטת Information Gain. כמו כן אנו רואים כי השיטה המשלבת PCA משמעותית נמוכה משאר השיטות (ב Naïve Bayes), זאת מאחר ואנו מקבלים מספר רב של וקטורים שהם צירוף לינארי של תכונות, עובדה זו כך נראה מקשה על המסווג ופוגעת בביצועיו, מכיוון ש Naïve Bayes מניח אי תלות בין המאפיינים והשיטה מגדילה משמעותית את התלות.
נקודה נוספת ראויה לציון היא החיתוך בין המגמה של השיטה הסטוכסטית ל Information Gain. צפוי שהשיטה הסטוכסטית תהיה יותר טובה בתחילת התהליך (עבור מספר תכונות גבוה), זאת מאחר והיא מורידה קבוצות מאפיינים שפוגעות הכי פחות, אבל, עם התקדמות התהליך וירידה גדולה במספר הפיצ'רים ישנה טעות הולכת ומצטברת. זאת מאחר ואנו לא בודקים את כל הצירופים האפשריים ולעיתים ישנן תכונות "טובות" שיורדות כי הן נבנו ביחד עם קבוצה של תכונות שאינן טובות. ניתן לראות עבור שלושת המסווגים, כי חיתוך זה מתבצע סביב 6000 תכונות.
מובהקות סטטיסטית: לפי מבחן ווילקוקסון, ניתן לראות מובהקות בין השיטות. הבוחר המבוסס על IG בלבד טוב מ PCA ומהסטוכסטי עם P_Value של $8 * 10^{-16}$ לעומת הראשון ו $2.8 * 10^{-3}$ לעומת השני.
לפי אותו מבחן, הבוחר הסטוכסטי עדיף על בוחר *PCA* עם *P_Value* 10^{-11}.

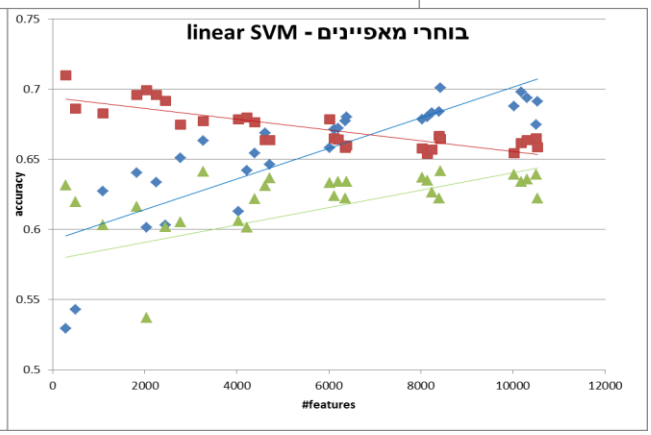
סיכום – בחירת מאפיינים:

ניתן לראות כי בניגוד למצופה – ראינו שדווקא המסווג החמדן, הפועל לפי Information Gain, ובוחר את התכונות בעלות ה IG הגבוה ביותר – דווקא הוא הטוב ביותר לפי הניסויים שלנו. הוא מהיר מאוד, ומביא תוצאות יפות. עם זאת, חסרונו הוא שהוא מאוד מוחלט, בהינתן הרבה זמן – לא נוכל לשפר את התוצאה על ידי שימוש בזמן הנוסף. גישת הבחירה הסטוכסטית לעומת

זאת, מאפשרת דבר זה. בניסוי לא גילינו יתרון כלשהו לשימוש בPCA משולב – הוא לוקח הרבה מאוד זמן, מבלי להצדיק זאת. מעניין לבדוק איך PCA על כל מרחב ו/או על קבוצות גדולות יותר של תת מרחב המאפיינים יעבוד, אך דבר זה צפוי לקחת הרבה מאוד זמן, בעיקר בבעיות מורכבות יותר בעלות יותר מאפיינים.







המסווגים

המסווגים אותם בחנו לפתרון הבעייה הם Naïve Bayes, Linear SVM, Gaussian SVM.

עבור מסווגי ה SVM מימשנו בנוסף גם parameter optimization באופן הבא:

ביצענו חיפוש על כל מרחב הפרמטרים שכולל פרמטר γ שנע על הערכים $2^3 \dots 2^{-15}$ כאשר הקפיצות הן של פי 4 (כלומר 2 באקספוננט) עבור הגרעין הגאוסייני. ופרמטר C על שני הגרעינים שנע על הערכים $2^{15} \dots 2^{-5}$ וגם הוא קופץ פי 4. בסך הכל חיפשנו על מטריצת הפרמטרים בגרעין הגאוסי, ווקטור פרמטרי C בגרעין הליניארי.

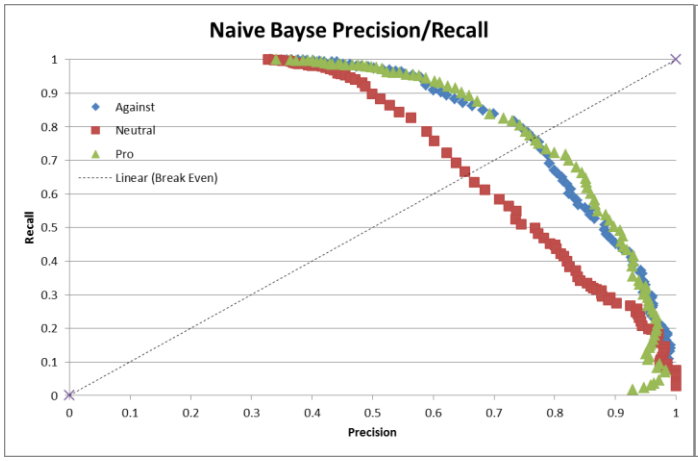
הערה: כאשר ביצענו חיפוש על מרחב המאפיינים הגדלנו את הקפיצות מפי 4 לפי 8, משיקולי ביצועים.

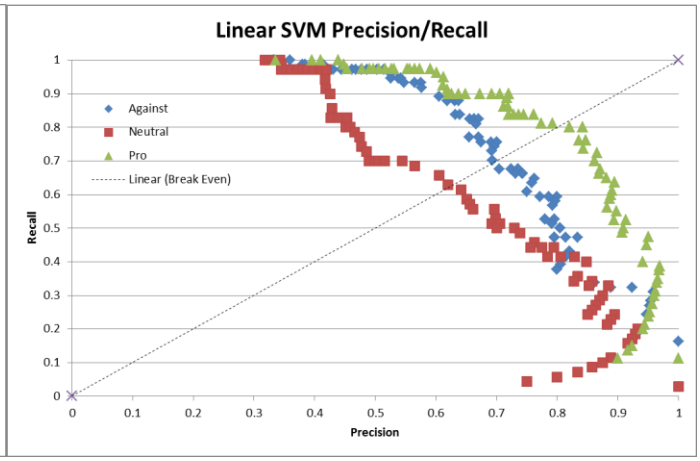
בהתאם לדרישות התרגיל, מעבר לשימוש במסווגי ה SVN בחרנו לבחון גם את הסווג ה Naïve Bayes. למסווג זה מספר יתרונות בפתרון "בעיית הסנטימנט" הנובעים באופן ישיר מהגדרת הבעייה. מאחר ובהגדרתה בעית סנטימנט היא אינה בעייה שמוגדרת היטב, שכן אנשים שונים בעלי סנטימנט שונה יכולים להגיב בצורה דומה וההפך. בעייה זו מצביעה על חוסר קונסיסטנטיות בסימון הדוגמאות, על כן לפתרון הבעייה בצורה המיטבית בתנאי רעש מסווג ה Naïve Bayes, הוא בחירה ראוייה. עם זאת, ראוי לציין שבעיקר בהקשר של טקסט – ההנחה הנאיבית של Naïve Bayes אינה ריאלית, שכן ידוע שיש קורולציה בשימוש במילים – דבר הבא לידי ביטוי בעיקר בשימוש בביטויים נפוצים.

הרצנו כל מסווג לפי מספר המאפיינים הטוב ביותר שהוא נתן, כולם עם בחירה לפי Information Gain, ועל התוצאות הרצנו מבחן מקנמר למובהקות סטטיסטית. קיבלנו כי naïve bayes עדיף על svm linear עם p_value = 0.07, ועל svm גאוסיאני עם p_value=0.085. אין מובהקות סטטיסטית בהשוואה בין שני סוגי SVM לפי מבחן זה.

ודאות הסיווג

מכיוון שעבור המקרה שלנו – מחלקת הסיווג אינה מחלקה בינארית, ולכן קשה להגיד מה אומר גרף precision recall במקרה זה. בשל כך, להצגת הדבר נקטנו בגישה טיפה שונה – הסתכלנו על כל מחלקה כמחלקה בינארית, כלומר – האם בעד או לא בעד? האם נטרלי או לא נטרלי? האם נגד או לא נגד? עבור שאלות אלה – ייצרנו את גרף ה precision recall כפי שמצורף:





עבור גרף Naïve Bayes, ניתן לראות, כצפוי – כי הגרף מתחיל מ precision = 1/3, recall=1 – זהו המקרה בו אנו מקבלים את כל הדוגמאות. הגרף יורד בהדרגתיות ומתכנס למצב בו precision->1, recall->0 עבור המצב בו דוחים את כל הדוגמאות, או מקבלים דוגמאות בהן הביטחון גבוה מאוד.

ניתן לראות כי באופן כללי, "מחלקת" הניטרליות קשה יותר – הגרף יותר נמוך מהמחלקות האחרות, וה break-even point נמצאת ב 0.65 לעומת 0.77 במחלקות האחרות. דבר זה נובע מכך שמחלקה זו קשה יותר לסיווג מהאחרות – נראה כי למסווג יותר קשה למצוא תכונות המקושרות למחלקה זו, מאשר תכונות למחלקות בעד ונגד. דבר זה צפוי, שכן אף לקורא אנושי – לעתים קשה לקבוע האם מדובר בכותב שדעתו בעניין ניטרלית. נקודה זו ברורה אף יותר כאשר מתבוננים בסטטיסטיקה של מבחני הסיווג, ניתן לראות שמספר הדוגמאות שמסווגות כניטרליות אך אינן כאלה, ולהפך הן השגיאות הנפוצות ביותר.

Naive Baye	Against	Natural	Pro	precision
Against	560	106	77	75.37%
Natural	119	454	160	61.94%
Pro	45	92	627	82.07%
recall	77.35%	69.63%	72.57%	

Linear SVM	Against	Natural	Pro	precision
Against	544	119	80	73.22%
Natural	120	449	164	61.26%
Pro	50	122	592	77.49%
recall	76.19%	65.07%	70.81%	

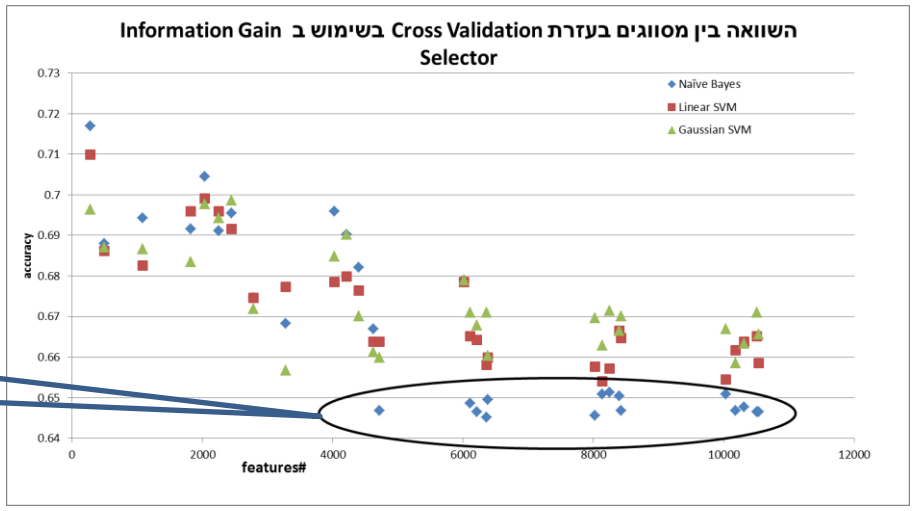
עבור Linear SVM, באזור בעל אחוזי ה Recall הנמוכים יש שינוי מגמה של הדגימות. ה Recall הנמוך נובע מכך שאנו דורשים בשלב זה ודאות גדולה (בערך 85%) ולכן חוזרות מעט דוגמאות. שינוי מגמת ה precision בחלק זה נובע מכך שיש מספר קטן של דוגמאות לא נכונות אך המסווג משוכנע בסיווגו (המוטעה) לגביהן (למעלה מ 95%), ולכן ככל שנעלה את רמת הודאות סך מספר הדוגמאות המסווגות יקטן, ושגיאה זו תהיה משמעותית יותר ויותר.

תוצאות ומסקנות

הרצת train test

בדקנו על סט ה test את ביצועי האלגוריתם בתחילת הניסוי (ללא בחירת מאפיינים, אלא שימוש בכולם וללא שימוש ב bi grams ו stemming) וקיבלנו שמסווגי SVM היו עדיפים על פני מסווג Naïve Bayes, עם accuracy=0.75 לעומת accuracy=0.66, ולפי מבחן הפרש פרופורציות (כפי שנלמד ב "מבוא לסטטיסטיקה") עדיפים עליו עם $P\_value=8*10^{-6}$

בדקנו את ריצת המסווגים – כל אחד לפני חילוץ מאפיינים ואחרי חילוץ מאפיינים. עבור Naïve_Bayes, שיפרנו את הביצועים מ 65.8% – ללא חילוץ מאפיינים וללא הוספת bi-grams ל 71.67%, לפי הפרש פרופורציות קיבלנו P_Value=0.0456 דבר זה מסתדר עם מה שראינו בשלב הקודם, שכן כבר עבור בדיקת Cross Validation, ראינו ש Naïve Bayes רגיש למספר התכונות ונפגע כאשר יש תכונות רבות מדי, כפי שניתן לראות בגרף הבא:



Naïve Bayes Fails for large #features

עבור linear SVM ועבור Gaussian SVM – קיבלנו תוצאות דומות אך לא מובהקות. עבור SVM היפרבולי, שיפרנו את אחוז הדיוק מ 0.753 ל 0.759 ועבור SVM לינארי שיפרנו את הדיוק מ 0.751 ל 0.767, שניהם – ללא מובהקות סטטיסטית לפי המבחן.

לאחר בחירת המאפיינים, SVM היפרבולי עדיף כאמור על Naïve Bayes, וזאת עם P_Value=0.012.

בנוסף – כפי שכבר ציינו עבור ה cross validation, גם כאן– קל לראות כי הבעיה של "נגד" לעומת "בעד" קלה הרבה יותר (רוב הטעויות נמצאות בעמודה/שורה המרכזית בטבלאות המצורפות) – דבר צפוי, שכן אף לקורא אנושי, לא תמיד קל להחליט האם מדובר בכותב ניטרלי.

Naive Bayes	Against	Natural	Pro	precision
Against	242	65	16	74.92%
Natural	48	245	40	73.57%
Pro	28	75	201	66.12%
recall	76.10%	63.64%	78.21%	

Linear SVM	Against	Natural	Pro	precision
Against	246	43	34	76.16%
Natural	44	232	57	69.67%
Pro	12	34	258	84.87%
recall	81.46%	75.08%	73.93%	

סיכום והמלצות

מהמחקר שלנו עולה בברור ש Naïve Bayes רגיש למספר המאפיינים, בעוד ש SVM יודע להתמודד היטב עם מאפיינים רבים, שחלקם אינם אינפורמטיביים.

למרות שהצלחנו להביא את Naïve Bayes לתוצאות עדיפות בבדיקת Cross-Validation, בבדיקה ה test, ניכר שבדיקות אלה מוטות במידת מה, והמסווג תלוי במידה רבה על מאפיינים בודדים שבחירתם או אי בחירתם תשפיע רבות על ביצועי המסווג. מסווג SVM לעומתו, אמין הרבה יותר, ואינו רגיש למספר המאפיינים. במידה ובפרוייקט זה היינו צריכים להחליט על מסווג לשימוש להמשך – היינו בוחרים להשתמש ב Linear SVM – המראה ביצועים עדיפים על פני האחרים, ואינו רגיש לבחירת/אי בחירת תכונות בודדות. היינו משתמשים במסווג זה ללא בחירת מאפיינים, ונותנים למסווג לתעדף מהי חשיבות כל מאפיין ומאפיין.

רעיונות מחקר להמשך

במחקרינו עלו מספר רעיונות שלא נכנסו ל scope הפרוייקט, אך ראויים לציון וייתכן וראוי יהיה לבצע מחקר נוסף לגבי נקודות אלה:

1. ראינו שהבעייה של "נגד" לעומת "בעד" קלה הרבה יותר מהבעייה בה טיפלנו. מעניין לנסות מסווג בינארי, המנסה לסווג "בעד" לעומת "נגד" בלבד, ונותן תוצאה "ניטרלית" עבור רמת מובהקות סיווג נמוכה יחסית. כמובן יש לחקור ולבדוק מהו סף זה.
2. כאמור, בוחר המאפיינים הסטוכסטי שמימשנו הוא למעשה backward search, ובדיעבד – זו הייתה כנראה טעות. מימשנו forward stochastic search והרצנו עליו ניסוי בסיסי עבור בחירת 500 מאפיינים. קיבלנו כי המסווג החדש (שמתחיל מ-0) עדיף על הקודם עם דיוק של 0.6 לעומת 0.54. לפי מבחן בדיקת השערות על הפרש פרופורציות כפי שנלמד בקורס "מבוא לסטטיסטיקה"[3], מקבלים שהדגימות שונות אחת מהשניה עם P_Value=0.067. מעניין לבדוק את ביצועי בוחר מאפיינים זה בתנאים שונים.
3. כאמור, בדיקת המאפיינים הסטוכסטי הינה any time, מעניין לראות מה גרף השיפור של שיטה זו כפונ' של הזמן.
4. רצינו לבדוק – אך לא הספקנו, את ביצועיו של מסווג שנבנה לפי תחום מסויים (למשל עדשות), ונבחן על data מעולם אחר (למשל – אופניים). דבר זה עשוי לשפוך אור עד כמה המסווג הנבנה עבור בעיית הסנטימנט תלוי ב data עליו הוא נבנה.

מקורות וקרדיט:

השתמשנו בספריות ומקורות קוד-פתוח הבאות לצורך מימוש הפרוייקט:

1. Weka לצורך מימוש Attributes Selection ו Naïve Bayes
2. LibSVM – לצורך מימוש Linear SVM, Gaussian SVM
3. Apache Lucene – לצורך stemming, ניפוי Stop Words ועיבוד הטקסט הגולמי
4. SnowBall – לצורך Stemming
5. סקריפט esuli.it ב Pearl לצורך שליפת הביקורות מ Amazon.com.

[3] http://moodle.technion.ac.il/mod/resource/view.php?id=27325