Markov Networks as Prior Distribution for Multi-Label Classification (in Wikipedia)

#### By: Alon Wolf (318336930) and Aviv Hadar (208948885)

*code:* [*https://github.com/AlonWolfi/wikipedia-summary*](https://github.com/AlonWolfi/wikipedia-summary)

# תקציר - Abstract

נתקלנו בבעיית Multi Label, עבורה יצרנו מודל אשר מפריד כל תיוג בפי עצמו, אליו הוספנו שימוש ב-prior ע"י מודל מרקובי על המחלקות מבוסס אלגוריתם צ'או ליאו. רק שה-prior משפר את תוצאות המודלים, בפרט ובעיקר במודל מבוסס LightGBM. ניסינו ואריציות שונות של אלגוריתם צ'או-ליאו והשוונו אותם עבור מודל עם עץ רנדומלי. ראינו כי שימוש ב-M.I. נותן את התוצאות הטובות ביותר. לבסוף ניסינו להרכיב בין המודלים השונים בשביל להיעזר בשונות בין המודלים לצורך שיפור ה-prior, אמנם ניסיון זה העלה חרס.

# רקע תאורטי

## מודלים מרקוביים

כפי שציינו, במאמר נעסוק בשיפור ביצועי אלגוריתם בבעיית *multilabel* בעזרת יצירת prior על התיוגים בין מחלקות. זאת באמצעות מודל מרקובי מעל להתפלגות הלייבילים שנייצר עפ"י אלגוריתם *Chow Liu*[[1]](#footnote-1) (כפי שיפורט בהמשך).

עבור דוגמא ספיציפית, נסמן את התיוגים האמיתיים עבור המחלקה:

אלגוריתם *Chow Liu*, כפי שנראה בהמשך, מחזיר *Pairwise Markov Network* על המחלקות.

לכל *Classifier* נרצה לחשב את ה-*Marginals* כדי לדעת את הסבירות של כל לייבל להיות חיובי. מתקיים שהנוסחה ל-*Inference* זו:

אם נניח שמבנה הרשת הוא עץ (כפי שמתקבל לפי האלגוריתם) נקבל שהמשקלים האופטימליים לרשת הם:

כאשר מתקיים:

|  |  |
| --- | --- |
|  | (3) |

## Belief Propagation

ניתן לחשב Marginals של מודל מרקובי בעזרת שיטת ה-Belief Propagation בצורה הבאה:

כאשר מתקיים:

משוואה (4) בעצם נותנת לנו את ההסתברות לקיומו של תיוג מסוים לפי **התפלגות התיוגים בלבד**, זאת מפני שהמודל הנ"ל ממשקל את כל ערכי באופן זהה ו- מושפע מ- בלבד (בהינתן הצבת המשקלים האופטימליים). כלומר המודל מתעלם כרגע מהסבירות לתיוגים שונים בהינתן מאפייני הדוגמא, שזו יכולה באופן משמעותי להשתנות בין דוגמאות שונות בהתאם לערכי הפיצ'רים.

המטרה שלנו בפרויקט הייתה לבצע תיקון לחיזויים של המודל ע"י שימוש ב-prior () באמצעות רשתות מרקוביות כדי למדל את התפלגות התיוגים, הוספנו לנוסחת חישוב ה-Marginal מישקול ערכי בכל הודעה לפי החיזויים של המודל - .

כאשר מתקיים:

כלומר, מה שאנו רוצים לעשות הוא לשלב את הסבירות לקיומו של לייבל בהינתן הפיצ'רים () עבור מחלקה יחידה עם הסבירות שנובעת מהתפלגות התיוגים (), במטרה לשפר את ביצועי המודל.

*הנוסחה המשולבת היא להלן:*

|  |  |
| --- | --- |
|  | (6) |

*חשוב לציין ש- אינה פונקציית הסתברות שכן נדרש לנרמל אותה:*

*זהו החיזוי עבור כל תיוג שמוחזרת אחרי התיקון שלנו לפי ה-prior.*

## אלגוריתם צ'או ליאו

*נחזור לאלגוריתם Chow Liu. כאמור השתמשנו בו בשביל לבנות את המודל המרקובי לפי ה-prior. השתמשנו בו באופן הבא:*

1. *הגדרנו גרף קשרים בין המחלקות עם המשקלים הבאים:*
2. *לאחר מכן מצאנו עץ פורש מקסימאלי לגרף*
3. *והעץ שקיבלנו שימש אותנו לבניית המודל המרקובי*

*השתמשנו בואריציות של האלגוריתם כך שיתאימו יותר לשיפור הביצועים והשוואנו אותם לתהליך המקורי.*



*איור 1 – ערך ויקיפדיה של אלטון ג'ון. באדום – הערך (טקסט), בכחול – עובדות מהירות. מטרת הפרויקט היא לחזות את החלק הכחול בהינתן האדום.*

# מידע (דאטה)

את המידע חילצנו מאתר ויקיפדיה. השתמשנו בספריות wikipediaapi, wptools על מנת לחלץ את הטקסט וה-infobox של ערכי הויקיפדיה (דוגמה לערך ניתן לראות ב*איור 1*).

נשאלת השאלה אילו ערכים לקחת – לא היינו לקחת את כל ערכי ויקיפדיה כיוון שמספרם גדול מדי ודגימה רנדומלית מתוך ויקיפדיה אינה ממשית (וכן צריך דגימה גדולה מאוד בשביל שתספיק תייצג את כלל הערכים). על כן, דגמנו את הערכים מ-subset של ערכי ויקיפדיה והוא "מוזיקאים"[[2]](#footnote-2) (בכדי שהערכים ייבחרו על ידי התפלגות יחידה ומוגדרת היטב).

עבור כל ערך בהתפלגות זאת, לקחנו את הטקסט כמשתנה הבלתי-תלוי ואת השאלות של ה-infobox כמשתנה התלוי.

את הטקסט העברנו עיבוד פר מילה כך שלכל מילה ייצרנו ממד חדש על האם הטקסט הכיל את המילה. דוגמא לתהליך זה עבור המילה *'Birth\_date'* ניתן לראות ב*איור 2*. לכל מילה הכנסנו ציון לפי אלגוריתם TF-IDF כך שככל שמילה מופיעה יותר בטקסט כך היא מקבלת ציון גבוה יותר, מנורמל לפי מס' הפעמים שהמילה הופיעה בכלל הטקסטים. הורדנו מילים שחזרו במס' בודד של טקסטים כיוון שלא רצינו שהמודל ילמד קורלציה שרק קורלטיביות למילים בודדות.

*איור 2 – עיבוד המילה על מילה "Birth\_date". ניתן לראות שכל חלק ...*

Tokanization

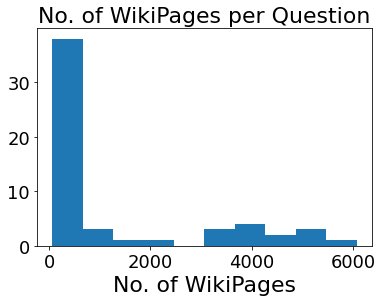
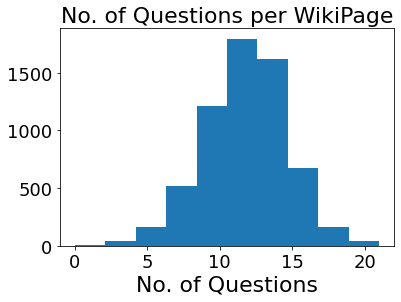
*'Birth\_date'*

Stop Words Deletion

*'Birthdate'*

Stemming

*'Birthdat'*



*איור 3 – היסטוגרמות של תיוגים ומחלקות. כמה שאלות (תיוגים) יש לכל דוגמא וכמה דוגמאות יש לכל שאלה (מחלקה)*

הסיבה שהשתמשנו בעיבוד זה היא כיוון שבשאלות ובטקסט היו עיוותים של מילים שמתארות את אותו הדבר - כן לדוג' *'Birth\_date', 'birth\_date', 'birthdate'* היו מילים שונות בטקסט שלאחר העיבוד שלנו הפכו למילה אחת *'birthdat'*.

כמו כן, מתוך השאלות גם כן סיננו שאלות לא נפוצות (מופיעות עבור מס' בודד של ערכים) שמספר מופען קטן ולכן אינן מייצגות את התפלגות המחלקה.

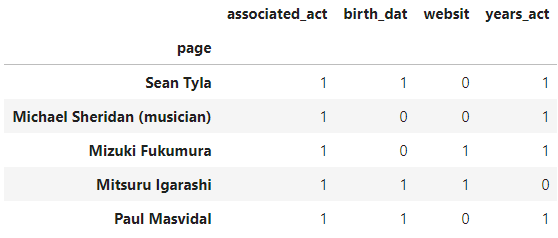
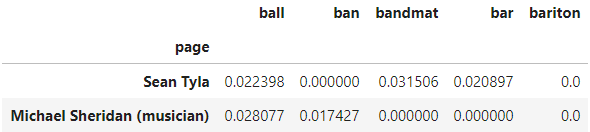
כפי שניתן לראות מהדוג', הבעיה הינה MultiLabel כיוון שלכל ערך ישנו מס' משתנה של שאלות.

את התפלגות מס' השאלות לכל ערך ניתן לראות באיור מס' 3. כפי שניתן לראות מהאיור, הופעת התיוגים מגוונת ואיננה טריוואלית – כלומר יש שאלות שמופיעות עבור עשרות דפים (כגון *'height'*) יש שאלות שמופיעות עבור מאות דפים (כגון *'record'*), ויש שאלות שמופיעות עבור רוב הדפים (כגון *'name'*).

## סיכום המידע

סה"כ קיבלנו 6221 דוגמאות עם 4328 פיצ'רים ו-56 סוגי מחלקות.

דוגמא למידע שקיבלנו ניתן לראות ב*טבלה 1*. ניתן לראות דוגמאות לפיצ'רים שקיבלנו וכן דוגמאות לשאלות עבור 2 ערכים – 'Sean Tyla'[[3]](#footnote-3), 'Michael Sheridan'[[4]](#footnote-4).



*טבלה 1 – דוגמא למידע שקיבלנו עבור 2 דפי ויקיפדיה. באיור העליון דוג' לשאלות (תיוגים) עבור הערכים. באיור התחתון דוג' לערכי פיצ'רים עבור הערכים.*

# תהליך

התחלנו בלייצר מודל לכל מחלקה (שאלה). השתמשנו במודל OneVsRest – כלומר, יצרנו מודל לכל מחלקה בנפרד. עשינו זאת כיוון שזהו common practice שמשתמשים בו כשרוצים למדל בעיות של multilabel (כיוון שיכול להיות יותר מתיוג אחד עבור כל מחלקה ולכן מודל מבוסס "*softmax*" אינו מתאים) . נשים לב שהמודל לא מתחשב בקשרים בין המחלקות כיוון שהוא מנתח כל מחלקה בנפרד.

את תיקון החיזויים ע"י ה-prior () חישבנו כפי שתיארנו בחלק התאורטי.

זאת בשלבים הבאים:

1. את חישבנו לפי סט האימון בלבד בשביל לא לייצר Target Leakage בבחינה של המודל.
2. *בחרנו את מבנה העץ של המודל המרקובי ע"י אלגוריתם Chow Liu (או ואריציה של האלגוריתם).*
3. השתמשנו בעץ וב- בשביל לחשב את על חיזוי המודל עפ"י האמור במשוואות (6) ו-(7) בעזרת *Belief Propagation* על המודל המרקובי.

*בהמשך בחנו את המודל תחת מבני עץ שנבנו בצורות שונות מזו, כפי שיוצג בחלק הבא.*

השתמשנו בסוגי מודלים שונים על מנת לבחון את עצמנו במס' תרחישים. איזנו את המשקלים של הדוגמאות על מנת שהחיזויים של המודל יהיו בעלי משקל הסתבורתי זהה עבור כל המחלקות.

\* יש לשים לב כי לפי הנוסחא ב-, משקול ההודעות החוזרות מתבצע לפי פרדיקציות המודל ו**מניח כי פרדיקציות אלו הן אכן הסתברויות** ולכן, בכדי שייצגו פונקציות הסתברות, אנחנו ממליצים לנרמל את משקלי המחלקות באימון המודל.

# תוצאות

## אופי בחינת מודל

השוואנו את תוצאות המודלים לפי 2 מדדים: , . החלטנו על מדדים אילו כיוון ש- הינו מדד אינטואיטיבי אשר מודד מה הסיכוי שהמודל ייתן לתיוג חיובי ציון גבוה מתיוג שלילי. אמנם, מקבל ציונים גבוהים עבור תיוגים לא מאוזנים וכיוון שיש לנו מס' שאלות עם מעט תיוגים, הוספנו את המדד *כיוון שאינו מתחשב ב- ולכן מייצג בצורה שונה את טיב המודל.*

*השתמשנו ב-2 מדדים לצורך בדיקת השיפור בין המודלים:*

*ROC\_AUC – במטרה לראות האם הסיכוי של תיוג חיובי להופיע לפני דוגמא שלילית גדל. כלומר האם הסדר בו המודל מדרג את התוצאות השתנה.*

*F1 – במטרה לראות האם החלוקה האופטימלית של המודל כמסווג בין חיוביים לשליליים השתפרה.*

כיוון ש- דורש *להקשיח את הפרדיקציות בחרנו את סף הציון (להלן ) אשר ממקסם את הציון על הסט הואלידציה (כך ש- ) (זאת כיוון שהבחנו שסף הציון האופטימלי עבור, במקרים רבים, אינו בהכרח 0.5 ולכן בחרנו אותו בשיטה זו ואנו ממליצים למי שמשתמש בשיטות דומות). למען הסר ספק, ביצענו את התהליך הזה עבור כל מודל, לפני ואחרי השימוש בprior.*

*כיוון שיש לנו מס' מחלקות, מיצענו בין המדדים בשיטת 'macro'. השתמשנו בשיטת 'macro' כיוון שיש מס' מחלקות שמרביתן חיוביות ויש מס' מחלקות שמרביתן שליליות, ומצאנו לנכון לבחון אותן ללא תלות במס' החיוביים.*

השתמשנו בשיטת K-Fold Cross Validation כאשר k=5. הציונים המוצגים בתוצאות הינם ממוצע החלוקות השונות.

## חלק I - מודל מרקובי

ראשית, בדקנו את הציון של מודל OneVsRest עבור מודלי בסיס שונים (*Logistic Regression, SVM, LightGBM) והשוונו אותם לציון של המודל לאחר הפעלת הפריור. ניתן לראות את התוצאות ב*טבלה 2*.*

|  |  |  |  |
| --- | --- | --- | --- |
| LGB | SVM | Logistic |  |
| 0.7084 | 0.7095 | **0.682** | OneVsRest |
| **0.7175** | **0.7109** | 0.673 | OneVsRest and Prior |
| 0.0091 | 0.0014 | -0.009 | Diffrence |

|  |  |  |  |
| --- | --- | --- | --- |
| LGB | SVM | Logistic |  |
| 0.9059 | 0.8981 | **0.904** | OneVsRest |
| **0.9068** | **0.9024** | 0.901 | OneVsRest and Prior |
| 0.0009 | 0.0043 | -0.003 | Diffrence |

*טבלה 2 – תוצאות עבור חלק 1*

ניתן לראות כי ה-prior משפר את התוצאות עבור המודלים ....

ניתן לראות כי השיפור הניכר ביותר קורה ב-LightGBM. אנחנו חושבים שכך כי .. (מודל יותר אקספרסיבי ולכן ...).

על כן האלגוריתם שהצענו שיפר ביצועים !!!! (לקוראים בעתיד ..)

## חלק II - סוגי עצים

לאחר מכן, רצינו לבדוק כיצד מבנים שונים של מודלים מרקוביים משפיעים על טיב התוצאות.

בנינו מודלים שונים ע"י ואריציות על אלגו' צאו לי .

הסתכלנו על:

1. עץ רנדומלי – בדיקת שפיות שנועדה לבדוק שהשיפורים אכן מגיעים מטיב האלגוריתם והמודל המרקובי.
2. M.I. – אלגוריתם *Chow Liu* הקלאסי. העץ נוצר לפי משקלים בין מחלקות המבוססים ב-MI בין התיוגים של המחלקות. (משוואה (8))
3. 'Semi-Positive' M.I. – נשים לב שנוסחת MI מורכבת מ-4 חלקים מארבעת האפשרויות לערכים של , שכן MI מתארת כמה "אינפורמציה" תיוג מכיל על תיוג ולהיפך, אמנם, לא בהכרח 4 האפשרויות משפיעות על תיקון הפרדיקציות כפי שהשתמשנו בו על העץ. כך לדוגמא, 2 לייבלים שלא מופיעים כמעט אף פעם כחיובים, , בעלי MI גבוה. כביכול, בעלי אינפורמציה משותפת גבוהה, כיוון שאם הראשון אינו חיובי סימן שגם השני ככל הנראה לא חיובי. אמנם, הם לא אמורים להשפיע על הפרדיקציות אחד של השני, ה-MI ביניהם גבוה אמנם הראשון לא אומר על השני הרבה על איכות החיזוי ולהיפך.

על כן, ניסינו לתקן את המשקלים כך שיתחשבו רק ב-.

1. 'Positive' M.I. – כעת ניסינו להמשיך קו מחשבה זה אף מעבר ולהתייחס רק למקרים בהם

תוצאות לחלק זה ניתן לראות ב*טבלה 3*.

|  |  |  |  |
| --- | --- | --- | --- |
| LGB | SVM | Logistic |  |
| 0.7084 | 0.7095 | **0.682** | baseline |
| 0.7086 | 0.7094 | 0.677 | Random |
| 0.7175 | **0.7109** | 0.673 | M.I. |
| **0.7177** | 0.7099 | 0.673 | 'Positive' M.I. |
| 0.7145 | 0.7084 | 0.665 | 'Semi-Positive' M.I. |

|  |  |  |  |
| --- | --- | --- | --- |
| LGB | SVM | Logistic |  |
| 0.9059 | 0.8981 | **0.904** | baseline |
| 0.9068 | 0.8996 | 0.902 | Random |
| **0.9097** | 0.9024 | 0.900 | M.I. |
| 0.9094 | **0.9041** | 0.900 | 'Positive' M.I. |
| 0.9085 | 0.8999 | 0.897 | 'Semi-Positive' M.I. |

*טבלה 3 – תוצאות עבור חלק 2*

כפי שניתן לראות ...

## חלק III - הרכבה של מודלים

לבסוף, בדקנו מה קורה כאשר מרכיבים מס' מודלים ..

ההנחה שלנו היא ..

תוצאות ניתן לראות ב*טבלה 4*.

|  |  |  |
| --- | --- | --- |
| LGB with normalization | LGB |  |
| - | 0.7084 | baseline |
| - | 0.7175 | M.I. |
| 0.7196 | 0.7156 | M.I. -> M.I. |
| **0.7220** | 0.7172 | M.I. -> 'Positive' M.I. |

|  |  |  |
| --- | --- | --- |
| LGB with normalization | LGB |  |
| - | 0.9059 | baseline |
| - | 0.9097 | M.I. |
| 0.9082 | 0.9076 | M.I. -> M.I. |
| **0.9101** | 0.9095 | M.I. -> 'Positive' M.I. |

*טבלה 4 – תוצאות עבור חלק 3*

ניסינו לנרמל כי משהו על הסתברות (כמו ממקודם \*) נורמליזציה משהו ..

ניתן לראות כי ...

# סיכום ומסקנות

נתקלנו בבעיית Multi Label, עבורה יצרנו מודל אשר מפריד כל תיוג בפי עצמו, אליו הוספנו שימוש ב-prior ע"י מודל מרקובי על המחלקות מבוסס אלגוריתם צ'או ליאו. ראינו שה-prior משפר את תוצאות המודלים, בפרט ובעיקר במודל מבוסס LightGBM. ניסינו ואריציות שונות של אלגוריתם צ'או-ליאו והשוונו אותם עבור מודל עם עץ רנדומלי. ראינו כי שימוש ב-M.I. נותן את התוצאות הטובות ביותר. לבסוף ניסינו להרכיב בין המודלים השונים בשביל להיעזר בשונות בין המודלים לצורך שיפור ה-prior, אמנם ניסיון זה העלה חרס.

1. <https://en.wikipedia.org/wiki/Chow%E2%80%93Liu_tree> [↑](#footnote-ref-1)
2. <https://en.wikipedia.org/wiki/Category:Musicians> [↑](#footnote-ref-2)
3. <https://en.wikipedia.org/wiki/Sean_Tyla> [↑](#footnote-ref-3)
4. <https://en.wikipedia.org/wiki/Michael_Sheridan_(musician)> [↑](#footnote-ref-4)