APML
שעור 3

Structured Prediction
עמרי אבנד

עוסקים במצבים בהם למרחב ה-labels שלנו יש מבנה.
- לפעמים מרחב לא חסום
- לפעמים אפילו התיוגים האפשריים משתנה בהתאם לקלט.

נתמקד במקרים supervised.

דוגמא: Natural Language Parsing
קלט: משפט
פלט: עץ שמתאר את האיברים במשפט, היחסים ביניהם, והתפקידים

למה זה קשה לאלגוריתם קלסיפיקציה רגיל?
1. מספר התיוגים האפשריים אקספוננציאלי.
2. התיוג תלוי במאפיינים של הקלט: למשל אורך משפט הקלט ואי אפשר לבנות את הכל ואז למצוא את כל המשפטים שמתאימים לו.

רעיון: נחלק את הבעיה לתתי בעיות הרבה יותר פשוטות
אבל זה לא אופטימלי! כי זה לא יאפשר לנו לאמוד חוקים שנוגעים ליחסים בין תתי הבעיות.

NER
POS
Sequence Prediction Problems: נעסוק פה ב-2 בעיות

I Named Entity Recognition:
הבעיה: בהנתן משפט, נרצה לסמן את חלקי המשפט שהם שמות ולתייג מאיזה סוג הם (Location, Company, ...)
איפה המבניות? תיוגים נמשכים יותר ממילה אחת במשפט אז יש תלות בין מילים שונות.

האתגרים בבעיה:
1. נצטרך המון דאטא כדי ללמוד את המודל כי אי אפשר להשתמש במה שלמדנו על משפט באורך 20 למשפט באורך 21.
2. גם החיפוש בשלב prediction יכול להיות קשה מאוד.

המשך: Sequence Prediction Problems

II Part of Speech Tagging : תיוג חלקי המשפט (noun, verb, ...)

את ההצלחה נמדוד בחלקיות (אחוז) המילים שתייגנו נכון.

אתגר: נרצה להצליח לתייג גם מילים שאף פעם לא ראינו.

⟸ נרצה להשתמש באינפורמציה של ההקשר.

איזה מידע יש לנו?

1. הסתברויות של מילים: באיזו הסתברות המילה saw מופיעה כפועל ובאיזו הסתברות כשם עצם.

2. מילים שכנות: יש רצפים של חלקי משפט שהם מאוד לא סבירים, אז ניתן לשלול תיוגים כך להסתמך על התוכן מסביב

3. מאפיינים מורפולוגיים: 'ly'$\rightarrow$adverb ← מה שהכי (עוזר)

אפשר ללמוד משהו על המילה מהמבנה שלה.

הערה: אפשר להגיע לתוצאות טובות למדי (~90%) אם מסתכלים רק על מילים בודדות. אבל אנחנו עוסקים פה ב-10% הנותרים...

עכשיו נרצה להגדיר את המודל ההסתברותי שלנו

הגדרה: סדרה (sequence) של משתנים מקריים $Y_1, ..., Y_n$ היא שרשרת מרקובית (Markov Chain, הומוגנית, מדרגה ראשונה) אם:

$$\forall i \quad P(Y_i | Y_1, \dots, Y_{i-1}) = P(Y_i | Y_{i-1})$$

$$(\Longleftrightarrow)\ Y_i \perp \{Y_1, \dots, Y_{i-2}\} | Y_{i-1}$$

$$(\Longleftrightarrow)\ P(Y_{1:n}) = \prod_{i=1}^{n} P(Y_i | Y_{i-1})$$

הסתברות משותפת של $Y_1, Y_2, \dots, Y_n$

הנחה נוספת: נסמן אצלנו הסתברות התחלתית $Y_0 = start$ כך שכשמתנים בה זה כמו לא להתנות בשום דבר - רק לנוחות.

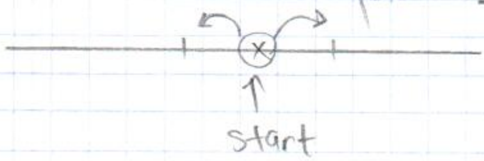
הגדרה 2: סדרות של מ"מ $Y_1, \dots, Y_n$, $X_1, \dots, X_n$ הן HMM (Hidden Markov Model) אם ההסתברות המשותפת שלהם מקיימת:

$$P(Y_{1:n}, X_{1:n}) = \underbrace{\prod_{i=1}^{n} P(Y_i | Y_{i-1})} \cdot \underbrace{\prod_{i=1}^{n} P(X_i | Y_i)}_{P(X_{1:n} | Y_{1:n})}$$

↑ כלומר מניחים שה-Y הם שרשרת מרקובית וזו בעצם ההתפלגות השולית $P(Y_{1:n})$

Sequence Prediction Problems: המשך

דוגמא: הילוך מקרי חד מימדי:



$$P(Y_0=0)=1 \qquad P(Y_i|Y_{i-1}) = \begin{cases} Y_{i-1}+1 & w.p.\ \frac{1}{2} \\ Y_{i-1}-1 & w.p.\ \frac{1}{2} \end{cases}$$

כלומר $\{Y_i\}_{i=1}^{n}$ היא שרשרת מרקובית.

נניח שנסתכל על $X_1,\dots,X_n$ שמודדים סיגנל שמשדר את המיקום, עם רעש

למשל:

$$P(X_i|Y_i) = \begin{cases} Y_i & w.p.\ \frac{1}{2} \\ Y_i+1 & w.p.\ \frac{1}{4} \\ Y_i-1 & w.p.\ \frac{1}{4} \end{cases}$$

הנחות: 1. $Y_{1:n}$ הם שרשרת מרקובית.

2. המ"מ X_i בלתי תלויים בכל דבר אחר חוץ מ-Y_i.

כלומר אם נדע את Y_i אז אין אינפורמציה נוספת על X_i במקום אחר.

הנחה זו הגיונית תחת חוסר זיכרון של התהליך של X.

כלומר למשל בדוגמא שלנו, הרעש נדגם בכל פעם מחדש בלי קשר לאיזה רעש או איזה סיגנל היה בצעד הקודם

אלו הנחות כלליות לכל HMM

בחזרה לתיוג POS:

$X_{1:n}$ = המילים במשפט

$Y_{1:n}$ - התיוגים POS של המילים במשפט ← אותם לא יודעים ונרצה לשערך

נניח שאכן $X_{1:n}, Y_{1:n}$ הם HMM.

האם ההנחות מתקיימות פה?

1. $Y_{1:n}$ הם שרשרת מרקובית: משמעות ההנחה פה היא שכדי לתת "ציון" לשרשרת תיוגים, מספיק לתת "ציון" לכל זוג תיוגים למילים עוקבות במשפט. זה די הגיוני כי התלויות באמת בד"כ מקומיות אבל זה לא תמיד מספיק ⟸ לפעמים נבחר שרשרת מרקובית מסדר שני או גבוה יותר (תלוי ב-2 מילים אחורה...) אבל כמובן שהסיבוכיות עולה.

Sequence Prediction Problems: המשך

POS: קיום ההנחות של HMM: המשך

2. X_i בלתי תלוי בשום דבר חוץ מ-Y_i: הנחה זו בבירור לא מתקיימת! כי אם אנחנו לא יודעים מה המילה, אם אם נדע מה התיוג POS שלה, נוכל ללמוד עליה הרבה משאר המילים.

פרמטרים של HMM:

1. transition probabilities: לכל (y,y') אפשריים, מה ההסתברות שיופיעו ככה? כלומר:
$$t(y,y') = P(Y_i = y \mid Y_{i-1} = y')$$
וחייב להתקיים:
$$\forall y' \quad \sum_y t(y,y') = 1$$

2. emission probabilities: לכל ערך y אפשרי ל-Y_i וערך w ל-X_i, מה ההסתברות?
$$e(w,y) = P(X_i = w \mid Y_i = y)$$
וגם כאן נדרוש:
$$\forall y \quad \sum_w e(w,y) = 1$$

דוגמא:

Bill	saw	that	man	yesterday
$=X_1$	$=X_2$	$=X_3$	$=X_4$	$=X_5$
Name	Verb	Conj	Noun	Adverb
$=Y_1$	$=Y_2$	$=Y_3$	$=Y_4$	$=Y_5$

$$P(X_{1:n}, Y_{1:n}) = t(Name, start) \cdot t(verb, Name) \cdot \ldots$$
$$\cdot\, e(Bill, Name) \cdot e(Saw, Verb) \cdot \ldots$$

וכל מה שנשאר לעשות זה ללמוד את e ו-t ולהבין איך לחפש תמונה מתוך המרחב הזה.

המשך: Sequence Prediction Problems

POS ב-HMM: Inference

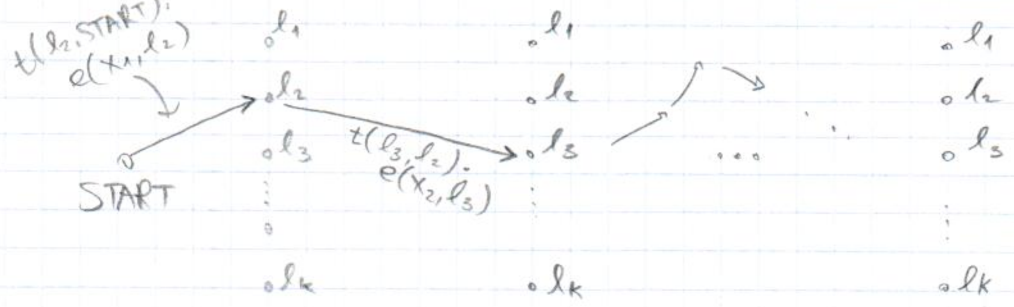
נ״ח כעת שכבר יש לנו מודל HMM: $\{t(i,j)\}_{i,j\in L}$ (POS tags), $\{e(w,i)\}_{w\in V,\, i\in L}$ (vocabulary)

ואנו בהינתן סדרה $X_{1:n}$ (משפט), נחפש:

$$y^* = \underset{y_{1:n}}{\operatorname{argmax}}\, P(y_{1:n}, X_{1:n})$$

$$= \underset{y_{1:n}}{\operatorname{argmax}}\, P(y_{1:n}|X_{1:n})\cdot P(X_{1:n})$$

אפשר להסתכל על זה כגרף של המסלולים האפשריים: $L=\{l_1,l_2,\dots,l_k\}$



כלומר נשקל את הצלעות ב: $w_{y_{i-1}\to y_i} = t(y_i,y_{i-1})\cdot e(x_i,y_i)$

ונחפש מסלול עם מכפלת משקולות מקסימלית

איך נמקסם? Viterbi Algorithm:

$$\pi(t,j) = \max_{\substack{y_1,\dots,y_t\\ y_t=j}} P(y_{1:t}, x_{1:t})$$

$$\pi(t,j) = \max_{j'}\left\{\pi(t-1,j')\cdot t(j,j')\cdot e(x_t,j)\right\}$$

כי מסלול אופטימלי באורך t חייב להתחיל במסלול אופטימלי באורך $t-1$ ועוד צעד אופטימלי.

$$\pi(1,j) = t(j,\text{START})\cdot e(x_1,j)$$

יש רק מסלול אחד באורך 1 שמסתיים ב-j

כעת בעזרת תכנון דינמי אפשר למצוא את זה.

POS ב-HMM: Learning

בעזרת MLE: נמקסם את

$$\ell(\{t(i,j)\},\{e(w,j)\}) = \sum_{r=1}^{M}\sum_{k=1}^{n_r}\left[\log t(y_k^{(r)}, y_{k-1}^{(r)}) + \log e(x_k^{(r)}, y_k^{(r)})\right]$$

עבור אוסף הדוגמאות $\{x_{1:n}^{(r)}, y_{1:n}^{(r)}\}_{r=1}^{M}$, תחת ההנחות שהנחנו קודם.

וזה פשוט יוצא: $t(i,j) = n_{ij}/\sum_k n_{kj}$, $e(w,i) = n_{wi}/\sum_{w'} n_{w'i}$

Sequence Prediction Problems : המשך

Maximum Entropy Markov Model : MEMM-ב PoS

זה מאפשר מספר שיפורים על פני HMM.

שיפור 1: ב-HMM לא יכולנו להשתמש בכל ה-features, כמו
למשל מורפולוגיה של מילים. אז על מילים שלא הופיעו או
הופיעו מעט מאוד - יש מעט מאוד מידע.
כי כל מידע נלקח מהשכיחות במאגר הנפרד מאשר המילים.

זה מודל discriminative. כלומר - אין אפשרות לדגום מה-X-ים אלא
רק בהינתן X-ים נחשב את ה-Y-ים.
=> לא צריך הנחות על X.

נניח:

$$P(Y_{1:n}|X_{1:n}) \underset{\text{conditional independance}}{=} \prod_{i=1}^{n} P(Y_i|Y_{i-1}, X_{1:n}) =$$

$$= \prod_{i=1}^{n} \frac{e^{<\Phi(y_i, y_{i-1}, x_{1:n}, \overset{\text{קבוע}}{i}), w>}}{z(x_{1:n}, Y_{i-1})}$$

כאשר Φ זו פונקציית ה-features שלנו ו-w יהיו הפרמטרים של המודל.
המכפלה הפנימית הזו $<\underset{\mathbb{R}^d}{\Phi}(y_i, y_{i-1}, x_{1:n}, i), \underset{\mathbb{R}^d}{w}>$ היא למעשה score. APML שיעור 4

אז מי היא Φ? כלומר איך נייצג את ה-features שלנו?

Φ היא וקטור ארוך ב-$\mathbb{R}^d$

$$\Phi(y_i, y_{i-1}, x_{1:n}, i) = \begin{pmatrix} \vdots \\ \mathbb{1}_{\{y_i, y_{i-1}\}} \\ \vdots \\ \mathbb{1}_{\{x_i, y_i\}} \\ \vdots \\ \mathbb{1}_{\{pref, y_i\}} \\ \vdots \end{pmatrix}$$

- transition features — כלומר אינדיקטור למי הם $\{y_i, y_{i-1}\}$ — $(\text{מספר POS})^2$
- emission features — כלומר אינדיקטורים ל-POS — מספר המילים $\times$ מספר POS
- prefixes $p_1, \dots, p_m$ — $\{pref(x_i)=p_j, y_i=y\}$ — מספר POS $\times m$

אז איברי w בחלק שמתאים ל-transition features יהיו בעלי
ערך גבוה אם הרצף y_i, y_{i-1} מופיע בהסתברות גבוהה
וערך שלילי (אולי אפילו שלילי גדול מאוד) אם הרצף בהסתברות נמוכה.
בחלק שמתאים ל-emission, נגיד למשל, אם ההסתברות לתיוג (man, verb)
גבוהה, יהיה ערך גבוה ואם נמוכה ערך נמוך...

המשך MEMM: Sequence Prediction Problem

תזכורת ממה שראינו בתרגול: Gradient Ascent

$$\nabla LL = \underbrace{\sum_{r=1}^{N}}_{\text{כל ה- instances}} \underbrace{\sum_{i=1}^{r}}_{\text{כל ה- transissions}} \Big[\Phi(y_i^{(r)}, y_{i-1}^{(r)}, x_{1:n}^{(r)}, i) - \underbrace{\sum_{y'} P(y' | y_{i-1}, x_{1:n}) \Phi(y', y_{i-1}, x_{1:n}, i)}_{\text{Expected value}} \Big]$$

Structured Perceptron:

במקום לחשב את התוחלת על פני כל ה-lables האפשריים, אולי יש הרבה כאלה ולא נרצה לחשב את הסכום הזה (אצלנו ב-POS זה אפשרי אבל לא תמיד)

מה ההבדל?

במקום לנרמל כל צעד בשרשרת, ננרמל את השרשרת כולה מאוסף כל השרשראות האפשריות.

ואז במקום להשתמש בשיטה מבוססת גרדיאנט שאולי קשה לחשב, נשתמש בפרספטרון.

נניח שיש לנו פונקציית score:

כדי לחשב ניקוד על סדרה, פשוט נסכום ניקוד על כל הצעדים הרצופים.

$$\text{score}(y_{1:n}, x_{1:n}) = \sum_{i=1}^{n} \text{score}(y_i, y_{i-1}, x_{1:n}, i) = \sum_{i=1}^{n} \Phi(y_i, y_{i-1}, x_{1:n}, i)^T \cdot w$$

נשים לב שזה מאוד דומה למה שעשינו ב-MEMM...

רוצים למצוא את w האופטימלי $\Longrightarrow$ פרספטרון יעזור לנו לעשות את זה בעזרת inference בלבד.

Perceptron for Sequence Labeling:

Input: $\{(x_{1:n}^{(r)}, y_{1:n}^{(r)})\}_{r=1}^{N}$

1. $w \leftarrow \vec{0}$ / random

אולי הרבה פעמים

2. for $r = 1 \ldots N$

   2.1. $\hat{y}_{1:n} \leftarrow$ find the seq of ys that maximizes the score = do inference

   2.2. $w \leftarrow w \cdot \eta \Big(\underbrace{\sum_{r=1}^{N} \Phi(y_i^{(r)}, y_{i-1}^{(r)}, x_{1:n}^{(r)}, i)}_{\text{the correct answer}} - \sum_{r=1}^{N} \Phi(\hat{y}_i^{(r)}, \hat{y}_{i-1}^{(r)}, x_{1:n}^{(r)}, i) \Big)$

ממקסמים את ההסתברות $P(y_{1:n}, x_{1:n})$

3. return w

לפעמים נרצה להחזיר את הממוצע על פני כל ה-w שראינו במהלך האימון

המשך: <u>Sequence Prediction Problem</u>

המשך: <u>Structured Perceptron</u>

<u>אינטואיציה על מה הפרספטרון עושה:</u>

נתחיל מניחוש כלשהו של w, ואז בכל פעם נעשה inference
ונשווה בין האמת לבין מה שיצא לנו מהמודל:

אם זה אותו דבר, אין עדכון
אם יש הבדל בין המודל לבין האמת, נעדכן את w.

זה דומה למה שעושים ב-GD, אבל במקום לקחת את התוחלת לעדכון,
לוקחים את המקסימום במובן מסוים.