

מבוא לבינה מלאכותית – תרגול 9

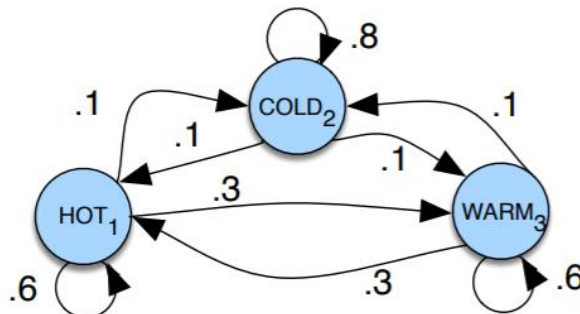
נושא:

HMMs - (Hidden Markov Models) מודלים מרקוביים חבויים

- שרשרות מרקוב
- מודלים מרקוביים חבויים
- אימון HMMs: חישוב הנראות (Forward Algorithm), מציאת המצבים (Decoding – Viterbi Algorithm), אימון HMM (Forward-Backward Algorithm).

שרשרות מרקוב:

לפני שנבין מהו HMM, נסביר (נזכיר?) מהי שרשרת מרקוב. שרשרת מרקוב היא מודל הסתברותי שמורכב מאוסף מצבים שיש ביניהם סיכויי מעבר. לשימושינו היום, שרשרת מרקוב מסדר ראשון (לא תלויה בזמן) היא סדרה של משתנים מקריים $\{q_t\}$ שכל אחד מהם שייך לאחד מתוך אוסף ערכים $\{s_1, \dots, s_n\}$ (מצבים), כך שהסיכוי לעבור למצב s_j בזמן $t + 1$ תלוי אך ורק במצב הקודם: $P(q_{t+1} = s_j | q_t, \dots, q_0) = P(q_{t+1} = s_j | q_t)$ (ללא תלות ב- t). נשתמש במטריצת מעבר בין מצבים A כך ש- $A_{ij} = P(q_{t+1} = s_j | q_t = s_i)$, ובהתפלגות מצבים התחלתית π , המתארת את ההסתברות להיות בכל מצבים כאשר נתחיל בתהליך. נשים לב: וקטור ההסתברות להיות בכל מצב בזמן k שווה ל- $A^k \pi$ (שימו גם לב שאצלו הזמן הוא בדיד). דוגמה קלאסית לשרשרת מרקוב היא הילוך מקרי פשוט – נתחיל בנקודה $q_0 = 0$, ובכל שלב נתקדם ימינה או שמאלה בסיכויים שווים (q_1 יהיה 1 בסיכוי $\frac{1}{2}$ ו-1 בסיכוי $\frac{1}{2}$). סדרת המיקומים שבהם נהיה מהווה שרשרת מרקוב, כי הסיכוי להיות בכל מקום תלוי אך ורק במקום שהיינו בו בזמן הקודם. גם הציור מטה מתאר שרשרת מרקוב על מצבים שהם מזג האוויר בכל יום, והקשתות שבין כל מצב למצב מתארות את סיכויי המעברים שבין המצבים.



Hidden Markov Model (HMM)

זהו מודל עם רעיונות דומים לשרשרת מרקוב, אבל מסובך יותר ומכיל קצת יותר רכיבים. המודל מתאר בעיה עם הנחות דומות לקודם (למשל, מעברים בין מזג אוויר בימים שונים), אבל כאן אנחנו לא יודעים מה המצבים שלנו בכל זמן נתון – הם חבויים לנו. מה שכן, לכל מצב סמוי בזמן מתאימה תצפית. בהסתמך על רצף התצפיות, ננסה להעריך את רצף המצבים של המודל ואת סיכויי המעברים שביניהם. שימו לב שהבעיה שעומדת לפנינו היא בעיה לא מפוקחת – אין לנו תיגוים למצבים האמיתיים.

דוגמה: מחפירות ארכיאולוגיות שנעשו לאחרונה (במדינת "שקר כלשהו" על ידי מדענים מקולג' קהילתי "סיפור מומצא סטייט"), מצאו לוחות שעליהם נכתב בכתב קדום ומוכר, אבל למרבה הצער הבלאי לאורך השנים השאיר אותם עם לוחות שבהם הכתב לא ברור. הם לא יאמרו נואש – הטכנולוגיה של השנים האחרונות מאפשרת להוציא פיצ'רים לכל תו על הלוח באמצעות מיקרוסקופ אלקטרוני סורק. כעת, יש להם רצף מצבים חבויים – התווים שהיו על הלוח במקור, ורצף תצפיות – הפיצ'רים ממיקרוסקופ האלקטרוני.

מתמטית, ב-HMM מופיעים הרכיבים הבאים:

- רצף המשתנים המקריים המתארים את המצבים – q_1, \dots, q_T (לא ידועים). כל מצב הוא מתוך קבוצה של N מצבים אפשריים.
- מטריצת המעברים שבין המצבים – A (לא ידועה).
- רצף תצפיות – o_1, \dots, o_T (ידועות), מתוך קבוצת תצפיות $\{v_k\}_k$.
- ההסתברויות לתצפית o_t בהינתן מצב i , $B = b_i(o_t)$ (מוכרות גם בתור "observation likelihoods" או "emission probabilities", לא ידועות).
- התפלגות מצבים התחלתית π (לא ידועה, פחות נתמקד בה).

במודל מרקובי חבוי מסדר ראשון קיימות הנחות המודל הבאות (והן לא טריוויאליות ולא תמיד נכונות, אבל בכל זאת משתמשים בהן):

- ההנחה המרקובית – כמו קודם, $P(q_{t+1}|q_t, \dots, q_1) = P(q_{t+1}|q_t)$.
- הנחת אי תלות של התצפיות – $P(o_t|q_1, \dots, q_T, o_1, \dots, o_{t-1}, o_{t+1}, \dots, o_T) = P(o_t|q_t)$.

כעת, לאחר שהגדרנו את המודל שלנו לפתרון הבעיה, נגדיר מה צריכים לעשות כדי לפתור אותה. עומדות בפנינו 3 בעיות:

1. Likelihood - בהינתן HMM ידוע המתואר ע"י $\lambda = (A, B)$ וסדרת תצפיות ידועה O , מהו $P(O|\lambda)$, הסיכוי לרצף התצפיות שקיבלנו?

2. Decoding - בהינתן HMM ידוע המתואר ע"י $\lambda = (A, B)$ וסדרת תצפיות ידועה O , מה הסיכוי לרצף המצבים הסביר ביותר $Q^* = \arg \max_Q P(Q, O|\lambda)$?

3. Learning - בהינתן סדרת תצפיות ידועה O , מה המטריצות האופטימליות לתיאור המודל A, B ?

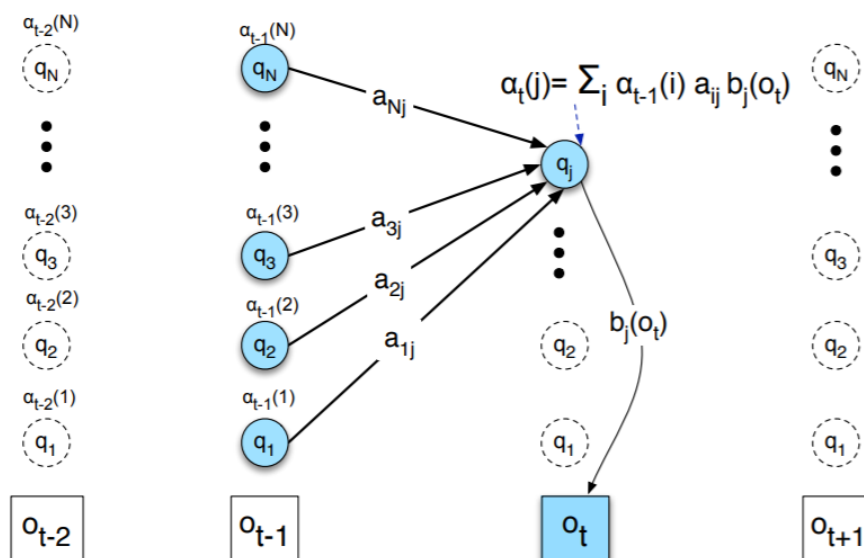
לכל בעיה נראה את האלגוריתם לפתרון שלה. בכולם מופיעים רעיונות של תכנון דינמי, במטרה לבצע את האלגוריתמים בזמן הגיוני.

1. Likelihood – Forward Algorithm

כאן, ידועים לנו A, B (וגם π) ונשתמש בהם, ידוע שיש לנו סדרה של מצבים באורך T כשכל מצב אפשרי יכול להיות אחד מבין N אפשרויות, ולכן לא נוכל לבצע את חישוב ההסתברות בצורה נאיבית כי הוא יעלה לנו לפחות N^T פעולות.

כדי לפתור את הבעיה בצורה חכמה, נחזיק משתני עזר $\alpha_t(j) = P(o_1, \dots, o_t, q_t = j|\lambda)$, ונחשב אותם רקורסיבית (ראו ציור). הרעיון המרכזי של החישוב הוא שימוש בנוסחת ההסתברות השלמה

$$P(O|\lambda) = \sum_Q P(O, Q|\lambda) = \sum_Q P(O|\lambda, Q)P(Q|\lambda)$$



אלגוריתם:

- a. אתחול: לפי ההגדרה ומה שאנחנו יודעים: $\alpha_1(j) = \pi_j b_j(o_1) \forall j$.
- b. רקורסיה: לכל מצב j בזמן t מחשבים בהתבסס על כל המצבים האפשריים בזמן הקודם: $\alpha_t(j) = \sum_{i=1}^N \alpha_{t-1}(i) A_{ij} b_j(o_t) \forall j$.
- c. סיום: הסיכוי המבוקש בסופו של תהליך מחושב בקלות לפי משתני העזר:

$$P(O|\lambda) = \sum_{i=1}^N \alpha_T(i)$$

השימוש בתכנון דינמי מקצר את זמן החישוב ל- $O(N^2T)$.

.2 Decoding – Viterbi Algorithm

רעיון הביצוע דומה ל-1: שוב נחזיק משתנה עזר שאותו נחשב רקורסיבית בזמן (אין קשר לקבוצת התצפיות):

$$v_t(j) = \max_{q_1, \dots, q_{t-1}} P(q_1, \dots, q_{t-1}, o_1, \dots, o_t, q_t = j | \lambda)$$

הוא יעזור לנו לחישוב הסיכויים הסופיים. בכל שלב נשמור לא רק את הערך $v_t(j)$, אלא גם את המסלול שהביא אותנו לערך שלו, $bt_t(j)$, כי בסוף המסלול (רצף המצבים) הוא מה שחשוב.

אלגוריתם:

- a. אתחול: כמו קודם, $v_1(j) = \pi_j b_j(o_1) \forall j$. כאמור, נחזיק גם $bt_1(j) = 0$ (סדרה ריקה של המצבים עד זמן זה מהם הגענו לכאן).
- b. רקורסיה: דומה לאלגוריתם Forward, רק שמחפשים מקסימום על המסלולים האפשריים. חישוב המקסימום מתבצע בצורה חמדנית (הסיכוי הגבוה ביותר להגיע למצב הקודם, כפול סיכויי המעבר ממנו למצב הנוכחי, ומקסימום על כל אחד מהמצבים הקודמים האפשריים):

$$v_t(j) = \max_{i \in \{1, \dots, N\}} v_{t-1}(i) A_{ij} b_j(o_t) \forall j$$

$$bt_t(j) = \arg \max_{i \in \{1, \dots, N\}} v_{t-1}(i) A_{ij} b_j(o_t) \forall j$$

- c. סיום: לבסוף, מקבלים את המסלול האופטימלי בתור מה שנותן את המקסימום על השלב האחרון:

$$v_t(j) = \max_{i \in \{1, \dots, N\}} v_{t-1}(i) A_{ij} b_j(o_t) \forall j$$

$$bt_t(j) = \arg \max_{i \in \{1, \dots, N\}} v_{t-1}(i) A_{ij} b_j(o_t) \forall j$$

3. Training – Baum-Welch (Forward-Backward) Algorithm

עד כה הנחנו קיום של המודל מבלי לדעת איך מחשבים אותו. כאן נבין איך החישוב הזה קורה. כזכור, אנחנו מחפשים את A, B (וגם π , פחות קריטי), וידועה לנו רק סדרת התצפיות O . האלגוריתם הזה משתמש בשני רעיונות – תכנון דינמי ואלגוריתם Expectation-Maximization: בשלב ה-Expectation משתמשים ב- A, B שאתחלנו או שיערכנו קודם לחישוב הסתברויות חשובות (נראה בהמשך את משתני העזר), ובשלב ה-Maximization משתמשים בערכי משתני העזר האלה לשערוך המטריצות A, B , בתהליך שמתבצע שוב ושוב עד התכנסות.

השם Forward-Backward מגיע מחישוב של שני משתני עזר (שמהם מחשבים את שני משתני העזר העיקריים): הראשון הוא $\alpha_t(j)$ שראינו בחישוב הראשון (Forward), והשני דומה לו ומחושב לאחור (נראה בתוך האלגוריתם כיצד הוא מתבצע):

$$\beta_t(i) = P(o_{t+1}, \dots, o_T | q_t = i, \lambda)$$

החישוב בסופו של דבר מתבסס על ההנחה כי ההערכה הכי טובה שתהיה לנו למטריצות היא:

$$\hat{A}_{ij} = \frac{\text{expected number of transitions from } i \text{ to } j}{\text{expected number of transitions from } i}$$

$$\hat{b}_j(v_k) = \frac{\text{expected number of times in state } j \text{ and observing } v_k}{\text{expected number of times in state } j}$$

ולכן ניעזר במשתנים הבאים:

$$\xi_t(i, j) = P(q_t = i, q_{t+1} = j | O, \lambda) = \frac{P(q_t = i, q_{t+1} = j, O | \lambda)}{P(O | \lambda)} = \frac{\alpha_t(i) A_{ij} b_j(o_{t+1}) \beta_{t+1}(j)}{\sum_{j=1}^N \alpha_t(j) \beta_t(j)}$$

$$\gamma_t(i) = P(q_t = i | O, \lambda) = \frac{P(q_t = i, O | \lambda)}{P(O | \lambda)} = \frac{\alpha_t(i) \beta_t(i)}{\sum_{j=1}^N \alpha_t(j) \beta_t(j)}$$

אלגוריתם:

a. אתחול: נאתחל את המטריצות \hat{A}, \hat{B} עם ערכים כלשהם.

b. נבצע איטרציות עד התכנסות:

i. שלב ה-Expectation: בהתבסס על \hat{A}, \hat{B} , נחשב את $\alpha_t(i), \beta_t(i)$ כמו

באלגוריתם Forward, β_t כך (כמו Forward רק אחורה):

$$1. \text{ אתחול: } \beta_T(i) = 1 \quad \forall i$$

$$2. \text{ רקורסיה: } \beta_t(i) = \sum_{j=1}^N \hat{A}_{ij} \hat{b}_j(o_{t+1}) \beta_{t+1}(j) \quad \forall i, t$$

שצריך. ניתן גם לכתוב את ההסתברות שחישבנו ב-Forward בעזרת

$$\beta_1(i)$$

מתוך כל מה שאנחנו יודעים, מחשבים לכל t, i, j :

$$\xi_t(i, j) = \frac{\alpha_t(i) \hat{A}_{ij} \hat{b}_j(o_{t+1}) \beta_{t+1}(j)}{\sum_{j=1}^N \alpha_t(j) \beta_t(j)}$$

$$\gamma_t(i) = \frac{\alpha_t(i) \beta_t(i)}{\sum_{j=1}^N \alpha_t(j) \beta_t(j)}$$

ii. שלב ה-Maximization: כאשר ידועים המשתנים $\xi_t(i, j), \gamma_t(i)$, מעריכים

מתוכם את \hat{A}, \hat{B} :

$$\hat{A}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i, j)}{\sum_{t=1}^{T-1} \sum_{k=1}^N \xi_t(i, k)}$$

$$\hat{b}_j(v_k) = \frac{\sum_{t=1}^T \gamma_t(j)}{\sum_{t=1}^T \gamma_t(j)} \quad \text{s.t. } o_t = v_k$$

c. סיום: מוציאים את המטריצות אליהן המודל התכנס. ניתן גם להעריך את ההתפלגות

$$\pi_i = \gamma_1(i)$$

לאחר שהבנו את רעיון הפתרון, מהתצפיות הנתונות ללא ידיעה של מודל ועד למציאת המודל וחיזוי סדרת המצבים הסבירה ביותר בו, נראה כמה שימושים ב-HMM:

- עיבוד וזיהוי דיבור – מודלים קיימים לזיהוי דיבור מורכבים מקבלת קול בתור גל, עיבוד הגל לסדרת פיצ'רים בזמן, המרת הפיצ'רים להברות (פונמות) וחיזוי המילים או המשפטים שנאמרו מתוך רצף הברות הזה. בהמון מודלים (בוויקיפדיה טענו שגם Siri) ההמרה של רצף הפיצ'רים לסדרת הברות ("המודל האקוסטי") מתבצעת בתור HMM: המצבים החבויים הם הפונמות, סדרת התצפיות היא סדרת הפיצ'רים וההמרה לסדרת הברות מתבצעת על ידי חיזוי של המודל שלמדנו.

- תרגום מכונה – מודלים (יחסית לא מודרניים) לתרגום משפט משפה אחת לאחרת הם מבוססי HMM. סדרת התצפיות היא המשפט (רצף מילים), והמצבים החבויים הם המילים בתרגום המשפט (זה כמובן, אפילו קלאסית, יותר מסובך מזה, אבל אכן קיימים מודלים מבוססי HMM לתרגום).

כאמור בהסבר המודל, אנחנו מניחים על סדרות הנתונים שלנו הרבה הנחות. לא תמיד הן נכונות. למשל, תרגום מכונה עובד גרוע אם מניחים הנחה מרקובית, הרי הדקדוק בשפות שונות יכול לגרום לתלות של מילה אחת במילה שאפילו לא נמצאת בסביבה קרובה שלה. לכן, להתמודדות עם סדרות בזמן מצאו פתרונות אחרים, במיוחד בשנים האחרונות ובאמצעות שיטות מתקדמות בלמידה עמוקה.