Hidden Markov Model

במודל מרקוב חבוי נתונות לנו קבוצה של מצבים חבויים $S = \{s_1, .., s_K\}$ וקבוצה של תצפיות אותן (בשונה מהמצבים החבויים) ניתן לראות בשטח $O = \{o_1, .., o_N\}$ וכמו כן נתונות מטריצות מעברים: A כאשר $A_{ij} = P(s_j|s_i)$ וB כאשר $B_{ij} = P(o_j|s_i)$.

בהינתן קבוצת תצפיות $Y = \{y_1, .., y_T\}$ לאורך זמן T אנו צריכים לענות על שתי שאלות:

1. מה ההסתברות לקבלת רצף תצפיות שכזה $P(y_1, .., y_T)$?
2. מה קבוצת המצבים החבויים $X = \{x_1, .., x_T\}$ הסבירה ביותר בהתחשב בתצפיות $argmax_X P(X|Y)$?

ע"מ לענות על השאלה הראשונה נשתמש ב***forward algorithm*** ובשביל השאלה השנייה נשתמש ב***Viterbi algorithm***.

לפני זה חשוב לציין כי במודל מרקוב מתקיים $P(s_{t+1}|s_t, s_{t-1}, s_{t-2} \dots) = P(s_{t+1}|s_t)$ כלומר, המצב הבא ניתן לחיזוי רק על סמך המצב האחרון ללא קשר לשאר המצבים.

כמו כן ניתן לגלות את ההסתברות הכללית $P(s_i)$ ע"י התהליך האיטרטיבי הבא:

- אתחל וקטור π בגודל k שסכום האברים שלו הוא 1 (בדרך כלל מאתחלים בהתפלגות אחידה).
- עדכן $\pi = \pi A$ מספר פעמים עד להתכנסות (כלומר, שערך הווקטור אינו משתנה או משתנה מעט מאוד).
- כעת $\pi_i = P(s_i)$.

Forward algorithm

אלגוריתם זה משתמש בתכנות דינמי, במקום לחשב מ0 כל צירוף אפשרי של X אלגוריתם זה בונה טבלה T בגודל $T\ x\ K$ כאשר $T_{ij} = a_i(x_j) = P(x_i = s_j|Y = \{y_1..y_i\})$, כלומר, מה ההסתברות שביום i (אנו מתייחסים אליו כיום האחרון) נהיה במצב חבוי s_j כאשר אנו מסתכלים על התצפיות עד יום i. האלגוריתם מתבצע כך (מומלץ לוודא שהשורות הבאות מובנות לך אינטואיטיבית):

- אתחל את השורה הראשונה בטבלה $T_{1j} = \pi_j \cdot P(y_1|x_j)$.
- עבור כל שורה בטבלה עדכן: $T_{ti} = \sum_{j=1}^{K} T_{t-1,j} P(y_t|x_i) P(x_i|x_j)$.
- לבסוף, ההסתברות ל Y היא $P(Y) = \sum_{i=1}^{K} T_{Ti}$.

Viterbi algorithm

גם פה התכנות הדינמי משחק תפקיד מרכזי, אנו בונים שתי טבלאות T_1, T_2 בגודל $K\ x\ T$, כאשר $T_{1ij} = \max_{x_1,..,x_j} P(x_1..x_j \cap x_j = s_i|y_1..y_j)$, כלומר ההסתברות המקסימלית שמביא לנו מסלול כלשהו של j מצבים חבויים הנגמר בs_i.

ובנוסף $T_{2ij} = the\ x_{j-1}\ of \operatorname{argmax}_{x_1,..,x_j} P(x_1.., x_{j-1}, x_j \cap x_j = s_i|y_1..y_j)$, כלומר, בשונה מ$T_{1ij}$ ששומר את ההסתברות של המסלול המקסימלי, T_{2ij} שומר את האיבר האחד לפני האחרון של המסלול המקסימלי (כאן המקום לציין שלמען הנוחות הקבוצות O, S מקודדות למספרים שלמים במידה ועד כה הם היו מילים או כל ייצוג אחר לא שלם).

האלגוריתם מתבצע כך:

- אתחל את השורה הראשונה בטבלה הראשונה $T_{1i1} = \pi_i \cdot P(y_1|x_i)$. את כל הטבלה השנייה אתחל ב0.
- עדכן את הטבלאות בצורה הבאה:

$$T_{1i} = \max_k (T_{1k,j-1} \cdot P(s_i|s_k)) \cdot P(y_j|s_i)$$
$$T_{2ij} = \operatorname*{argmax}_k (T_{1k,j-1} \cdot P(s_i|s_k))$$

לאחר שעדכנו את הטבלאות ניתן לחלץ מהם את המסלול הטוב ביותר X הסתברותית כך:

- אתחל $x_T = argmax_k(T_{2k}\ )$.
- עדכן $x_i = T_{2\,x_{i+1},i+1}$ עבור $i = T-1,..,1$.
- x_T הוא המסלול הגבוה ביותר הסתברותית.

Code

בקוד ישנה סימולציה די קטנה למקרה ספציפי אבל בקלות ניתן להגדיר מטריצות מעבר חדשות בגודל שונה ווקטור תצפיות שונה ללא צורך לשנות שום דבר בקוד, כמו כן הספרייה היחידה בשימוש היא numpy.