Projekt 2: Markovkedjor och skattning av övergångsmatris

25 mars 2025

1 Bakgrund:

En Markovkedja är en följd $\{X_0, X_1, X_2, \ldots\}$ av slumpvariabler som tar sina värden i $T = \{1, 2, \ldots, m\}$ för något känt m. Mängden T kallas för Markovkedjans tillståndsrum och de m olika tillstånden kan representera en mängd olika saker.

Ex 1: T kan representera antal kunder i en kö allteftersom kunder kommer och går. m spelar här rollen som maximal kölängd och händelsen $X_5 = 7$ innebär att vid tiden 5 finns det 7 kunder i kön.

Ex 2: Om vi betraktar en kortlek finns det $m = 52!$ olika sätt som leken kan vara ordnad på. Tillstånden i mängden T representerar vart och ett av dessa 52! olika ordningar. Markovkedjan kan beskriva tillstånden när man utför någon typ av blandning. Händelsen $X_7 = 1245$ innebär att vid tiden 7 så var ordningen 1245, och om man sedan blandar får man en ny ordning så $X_8 = 312$ innebär att efter nästa blandning fick man tillståndet 312.

För att kallas en Markovkedja måste X_i:na uppfylla att

$$\mathbb{P}(X_{n+1} = t_{n+1} | X_n = t_n, \ldots, X_0 = t_0) = \mathbb{P}(X_{n+1} = t_{n+1} | X_n = t_n)$$

för alla $n, t_0, t_1, \ldots, t_{n+1}$, dvs fördelningen för nästa tillstånd beror bara var man står just nu (kolla att detta är intuitivt rimligt i de två exemplena ovan). Här kan X_0 vara vald på vilket sätt som helst.

Skriv nu $p_{jk} = \mathbb{P}(X_{n+1} = k | X_n = j)$, $j, k = 1, \ldots m$. Den $m \times m$-matris P som har p_{jk}:na som element kallas för Markovkedjans *övergångsmatris*.

Antag nu att vi har en Markovkedja där övergångsmatrisen är okänd och vi bara får se observerade värden $X_0 = x_0, \ldots, X_n = x_n$. En naturlig skattning av p_{jk} är $\hat{p}_{jk} = n_{jk}/n_j$ där n_{jk} är antalet gånger vi ser ett hopp från j till k och n_j är totala antalet gånger kedjan besöker j. Ett problem kan dock uppstå; vi har m^2 olika okända koefficienter och mycket ofta är antalet observationer n betydligt mindre än m^2. Detta betyder att väldigt många p_{jk} skattas med 0. Detta är olyckligt eftersom vi knappast tror att ett sådant hopp är *omöjligt*.

En vanlig lösning är att införa s.k. *pseudo counts*, vilket i klartext betyder att man inför en parameter $b > 0$ och räknar som om man för varje j och k observerat hopp från j till k, b ”gånger” redan innan kedjan startat och använder skattningen

$$\hat{p}_{jk} = \frac{n_{jk} + b}{n_j + mb}.$$

2 Era uppgifter:

I bifogad Python-datavektor (trajectory.npy, vilket är en numpy vektor) är $m = 100$ och $n = 1000$. Använd de 500 första observationerna till att skatta p_{jk}:na enligt ovan med ett b som ni väljer själva. Ni får då en skattad övergångsmatris $\hat{P} = [\hat{p}_{jk}]$. Betrakta sannolikheten för att få just de 500 sista hoppen för en Markovkedja med övergångsmatris $\hat{P}$, dvs betrakta $\prod_{l=501}^{1000} \hat{p}_{x_{l-1}x_l}$. Denna sannolikhet kommer att bero av hur b är vald, så pröva att använda en stor mängd av möjliga b, t.ex. $b = 0.01, 0.02, \ldots, 2$ och ju högre den betraktade sannolikheten blir, desto bättre anser vi att valet av b är. Vilket blir bästa värdet på b? "Zooma inrunt den punkten och kör proceduren igen fast med högre upplösning (dvs med mindre steglängd i b).

Obs: Om man beräknar $\prod_{l=501}^{1000} \hat{p}_{x_{l-1}x_l}$ kommer resultatet bli så litet att man riskerar att bara få svaret 0. Beräkna i sådana fall istället logaritmen och jämför för olika b.

Vid presentationen ger ni era medstudenter en mycket kort introduktion till Markovkedjor och visar era resultat, gärna i en tabell över olika b mot motsvarande logaritmerade sannolikhet.

Om ni känner er mer bekväma med Matlab kan ni istället använda er av trajectory.mat