Projekt 5: Finn den dolda processen

2 april 2019

I det här projektet ska ni arbeta med ett problem från ämnesområdet *Dolda Markovkedjor*. Låt $p(1,1) + p(1,2) = p(2,1) + p(2,2) = q(1,1) + q(1,2) = q(2,1) + q(2,2) = 1$, med alla dessa storheter mellan 0 och 1. Antag att $X = (X_0, X_1, \ldots, X_n)$ är en följd av stokastiska variabler som antar värdena 1 eller 2 enligt $X_0 = 1$ och sedan $X_t = i$ med sannolikhet $p(1,i)$ om $X_{t-1} = 1$ och med sannolikhet $p(2,i)$ om $X_{t-1} = 2$, $i = 1, 2$, $t = 1, 2, \ldots, n+1$. Låt sedan $Y = (Y_0, Y_1, \ldots, Y_n)$ också ta värden i $\{1, 2\}$, vara betingat oberoende givet X och ges av att $Y_t = i$ med sannolikhet $q(1,i)$ då $X_t = 1$ och med sannolikhet $q(2,i)$ då $X_t = 2$.

Antag nu vidare att vi får se att $Y = y$, men inte X. Vi vill göra en bra skattning av vad X var och naturligt är att gissa att X är den mest sannolika vägen betingat med $Y = y$, dvs det x som maximerar

$$\mathbb{P}(X = x | Y = y).$$

Enligt Bayes formel är $\mathbb{P}(X = x|Y = y)$ proportionell mot $\mathbb{P}(X = x, Y = y)$, så vi söker det x som maximerar $\mathbb{P}(X = x, Y = y)$. Att finna detta x kan göras med hjälp av *Viterbis algoritm*. Låt

$$d_i(t) = \max_{x:x_t=i} \mathbb{P}(Y_0 = y_0, ..., Y_t = y_t, X_0 = x_0, \ldots, X_t = x_t)$$

för $t = 0, 1, \ldots, n$ och $i = 1, 2$. Då får man $d_1(0) = q(1, y_0)$ och $d_2(0) = q_2(y_0)$ och sedan rekursivt för $t = 1, \ldots, n$,

$$d_i(t) = \max_{j\in\{1,2\}} d_j(t-1)p(j,i)q(i,y_t)$$

(fundera gärna över varför denna rekursion är sann) och låter $m_i(t)$ vara det j för vilket detta maximum antogs. Slutligen får vi $\max_x \mathbb{P}(X = x|Y = y))$ ges av $\max_i d_i(n)$. Låt x_n^* vara det i för vilket detta sista maximum uppnås

Det gäller nu att återskapa det x som maximerar detta. Detta görs genom en *bakåtrekursion*:

$$x_t^* = m_{x_{t+1}^*}(t+1),$$

$t = n-1, n-2, \ldots, 0$. Observera att för att kunna genomföra bakåtrekursionen krävs att man har sparat alla $m_i(t)$.

Implementera Viterbis algoritm för att finna den betingat mest sannolika vägen x betingad med $Y = y$ för det y som ges i filen **hidden.dat** på kurshemsidan. I det här fallet är $p(1,1) = 0.9$, $p(2,2) = 0.8$, $q(1,1) = 0.6$ och $q(2,2) = 0.7$. Eftersom $d_i(t)$:na kan bli oerhört små då t stort, är det bättre att arbeta med $\log d_i(t)$.

Med Viterbis algoritm finner man alltså på ett effektivt sätt den mest sannolika vägen $x = (x_0, x_1, \ldots, x_n)$ givet $Y = y$. Detta är som sagt en viktig bit av information om fördelningen av

X givet $Y = y$, men helst skulle man förstås vilja veta hela fördelningen. Det är i princip inte så svårt (men en aning bökigt) att använda Bayes formel till att skriva

$$\mathbb{P}(X = x|Y = y) = \frac{\mathbb{P}(Y = y|X = x)\mathbb{P}(X = x)}{\sum_z \mathbb{P}(Y = y|X = z)\mathbb{P}(X = z)}$$

och att skriva ner sannolikheterna i detta uttryck. Problemet är dock att summan i nämnaren innehåller så många termer så att den är i praktiken omöjlig att beräkna. I vårt enkla exempel med $n = 100$ innehåller den 2^{100} termer! Men eftersom nämnaren bara är en proportionalitetskonstant, gäller att $\mathbb{P}(X = x|Y = y) \propto \mathbb{P}(X = x)\mathbb{P}(Y = y|X = x)$, vilket kan utnyttjas (och vilket utnyttjades ovan också) till att konstruera en *Gibbs sampler* som väljer ett X enligt den sökta fördelningen. Detta fungerar på följande sätt.

Låt $X^0 = (x_0^0, x_1^0, \ldots, x_n^0)$ vara en godtycklig vektor i $\{1, 2\}^{n+1}$, t.ex. $X^0 = (1, 1, \ldots, 1)$. Definiera sedan $X^1, X^2, X^3, \ldots$ rekursivt genom att för $s = 1, 2, \ldots$,

1. välja $k \in \{1, 2, \ldots, n\}$ likformigt på måfå,
2. låta $w(i) = p(X_{k-1}^{s-1}, i)p(i, X_{k+1}^{s-1})q(i, y_k)$, $i = 1, 2$ (där vi betraktar $p(i, X_{k+1}^{s-1})$ som 1 om $k = n$),
3. låta $X_l^s = X_l^{s-1}$ för alla $l \neq k$ och låta $X_k^s = 1$ med sannolikhet $w(1)/(w(1) + w(2))$ och annars $X_k^s = 2$.

Steg 2 och 3 innebär att X_k^s väljs så att $\mathbb{P}(X_k^s = i|X_l^s = x_l, l \neq k) = \mathbb{P}(X_k = i|Y = y, X_l = x_l, l \neq k)$. Man kan då visa att det för alla x gäller att $\mathbb{P}(X^s = x) \to \mathbb{P}(X = x|Y = y)$ då $s \to \infty$. Med andra ord gäller att då s är stort så har X^s en fördelning som är mycket nära fördelningen för X givet $Y = y$. Hur stort s behöver vara är inte helt enkelt att reda ut, men i det här projektet kan man tryggt nöja sig med $s = n^2$ och kommer att få en mycket god approximation.

Uppgiften är nu att implementera denna Gibbs sampler och tillämpa den i samma situation som för Viterbis algoritm ovanför. Vad blir resultatet? Kommer man nära den mest sannolika vägen som man fick med Viterbi? Upprepa många gånger.

Ni har nu sett två metoder att göra inferens om den dolda processen: Viterbi och Gibbs sampling. Viterbi svarar mot att se X som en okänd parametervektor och att ML-skatta denna. Gibbs sampling svarar mot att se X som stokastisk (vilket den ju faktiskt är i modellen). Viterbi har fördelen att den ger det mest sannolika x:et givet $Y = y$, medan Gibbs har fördelen att den fångar hela den betingade fördelningen för X genom att generera observationer från den.

Uppgiften nu är att jämföra de två metoderna på följande sätt. Låt

$$l(x) = \log \mathbb{P}(Y = y|X = x),\ x \in \{1, 2\}^{n+1}$$

där y fortfarande är observationen av Y given i **hidden.dat**. Låt $\hat{x}$ vara den mest sannolika vägen som gavs av Viterbi och låt $\tilde{x}$ vara en observation enligt Gibbs samplern. Ett vanligt kriterium för att jämföra $\hat{x}$ och $\tilde{x}$ är att säga att $\hat{x}$ är bättre"än $\tilde{x}$ om $l(\hat{x}) > l(\tilde{x})$ och vice versa. Generera nu ett stort antal oberoende observationer $\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_m$ och jämför var och en av dem med $\hat{x}$ enligt detta kriterium. I hur stor andel av fallen blir $\tilde{x}_k$ bättre"än $\hat{x}$.

Data i filen **hidden.dat** har genererats genom att simulera en process X enligt p:na ovanför och givet den sedan simulerat Y enligt a:na. Detta kan ni förstås upprepa själva; simulera X och sedan Y, glöm"X och utför Viterbis algoritm on Gibbs sampling på er simulerade Y-process för att återfinna X.