HMM – Ombrello e Meteo

Hidden Markov Models (HMM) – Ombrello e Meteo

Teoria e pratica applicata con esempi numerici semplici.

Obiettivo del documento:

- capire l'idea generale degli HMM usando un esempio quotidiano (portare l'ombrello);
- vedere passo per passo i calcoli dei principali algoritmi:
  * Forward algorithm (probabilita' di una sequenza di osservazioni);
  * Viterbi algorithm (sequenza di stati nascosti piu' probabile);
  * On-line filtering (probabilita' dello stato attuale dato cio' che ho visto finora);
  * Kalman Filter (versione continua per grandezze come la temperatura).

Nota sui colori (usati negli esempi):

- Blu = stato SOLE;
- Rosso = stato PIOGGIA.

1. HMM – Definizione sull'esempio dell'ombrello

1. Introduzione agli HMM con l'esempio dell'ombrello

Immagina che il 'vero' meteo (sole o pioggia) sia nascosto. Tu non lo osservi direttamente, ma vedi solo indizi: nuvole, pavimento bagnato, gente con l'ombrello.

In un Hidden Markov Model (HMM) abbiamo:

• Stati nascosti (quello che non vedo ma che voglio stimare):
  - stato 1: SOLE;
  - stato 2: PIOGGIA.

• Osservazioni (quello che vedo):
  - osservazione 1: NUVOLE;
  - osservazione 2: PAVIMENTO BAGNATO;
  - osservazione 3: OMBRELLI APERTI.

• Probabilita' di transizione tra stati (come evolve il meteo da un giorno all'altro):
  - P(SOLE domani | SOLE oggi)   = 0.8 (rimane bello con alta probabilita');
  - P(PIOGGIA domani | SOLE oggi) = 0.2;
  - P(SOLE domani | PIOGGIA oggi) = 0.4;
  - P(PIOGGIA domani | PIOGGIA oggi) = 0.6.

Mettiamo queste probabilita' in una tabella mentale (matrice di transizione):
  da SOLE   -> SOLE: 0.8, da SOLE   -> PIOGGIA: 0.2
  da PIOGGIA-> SOLE: 0.4, da PIOGGIA-> PIOGGIA: 0.6

• Probabilita' di osservazione (quanto sono probabili i segnali che vedo in base allo stato):
  Se lo stato nascosto e' SOLE:

- P(NUVOLE | SOLE) = 0.6
- P(PAVIMENTO BAGNATO | SOLE)= 0.2
- P(OMBRELLI APERTI | SOLE) = 0.1

Se lo stato nascosto e' PIOGGIA:

- P(NUVOLE | PIOGGIA) = 0.3
- P(PAVIMENTO BAGNATO | PIOGGIA)= 0.6
- P(OMBRELLI APERTI | PIOGGIA) = 0.8

• Probabilita' iniziali (come inizia la giornata):

- P(SOLE all'inizio) = 0.7;
- P(PIOGGIA all'inizio)= 0.3.

Stato SOLE (blu) = meteo bello / soleggiato.

Stato PIOGGIA (rosso) = meteo piovoso.

2. Forward Algorithm – Teoria

2. Forward Algorithm – Teoria (spiegata semplice)

Domanda a cui risponde il Forward algorithm:

«Quanto e' probabile vedere una certa sequenza di osservazioni,
senza preoccuparmi di quale sia stata la sequenza 'vera' degli stati?»

Nel nostro esempio, consideriamo la sequenza di osservazioni:

1) NUVOLE,
2) PAVIMENTO BAGNATO,
3) OMBRELLI APERTI.

Il Forward algorithm calcola passo passo:

- la probabilita' di aver visto le osservazioni fino ad un certo tempo,
- e di trovarsi in un certo stato in quel momento.

Per ogni istante di tempo t e per ogni stato (SOLE o PIOGGIA) calcoliamo:

```
  forward_prob[t, stato] =
    (somma su tutti gli stati possibili al tempo t-1 di:
       forward_prob[t-1, stato_precedente]
       * probabilita' di passare da stato_precedente a stato
     )
     * probabilita' di osservare l'osservazione corrente dato lo stato.
```

In parole:

- guardo tutte le strade con cui potevo arrivare a quello stato;
- sommo le loro probabilita' (perche' sono modi alternativi);
- moltiplico per quanto e' compatibile l'osservazione con quello stato.

Alla fine, per ottenere la probabilita' totale della sequenza di osservazioni, sommo le probabilita' forward dell'ultimo tempo su tutti gli stati:

```
prob_sequenza = forward_prob[ultimo_tempo, SOLE]
              + forward_prob[ultimo_tempo, PIOGGIA].
```

Nei calcoli pratici useremo nomi tipo:

  forward1_SOLE, forward1_PIOGGIA, forward2_SOLE, ecc.,

cosi' e' piu' chiaro cosa stiamo facendo.

2. Forward Algorithm – Esempio (parte 1)

2.1 Forward Algorithm – Esempio numerico completo

Dati usati nell'esempio:

Probabilita' iniziali:

P(SOLE all'inizio) = 0.7

P(PIOGGIA all'inizio)= 0.3

Probabilita' di osservazione (emissione):

P(NUVOLE | SOLE) = 0.6

P(NUVOLE | PIOGGIA) = 0.3

P(PAVIMENTO BAGNATO | SOLE) = 0.2

P(PAVIMENTO BAGNATO | PIOGGIA) = 0.6

P(OMBRELLI APERTI | SOLE) = 0.1

P(OMBRELLI APERTI | PIOGGIA)= 0.8

Probabilita' di transizione:

P(SOLE domani | SOLE oggi) = 0.8

P(PIOGGIA domani | SOLE oggi)= 0.2

P(SOLE domani | PIOGGIA oggi)= 0.4

P(PIOGGIA domani | PIOGGIA oggi)= 0.6

Sequenza osservata:

tempo 1: NUVOLE

tempo 2: PAVIMENTO BAGNATO

tempo 3: OMBRELLI APERTI

STEP 1 – Inizializzazione (tempo 1, osservazione = NUVOLE) forward1_SOLE   = P(SOLE all'inizio) * P(NUVOLE | SOLE)

= 0.7 * 0.6

= 0.42

forward1_PIOGGIA = P(PIOGGIA all'inizio) * P(NUVOLE | PIOGGIA)

= 0.3 * 0.3

= 0.09

STEP 2 – Tempo 2 (osservazione = PAVIMENTO BAGNATO)

Per SOLE al tempo 2:

somma_per_SOLE =

forward1_SOLE   * P(SOLE al tempo 2 | SOLE al tempo 1)

\+ forward1_PIOGGIA* P(SOLE al tempo 2 | PIOGGIA al tempo 1)

= 0.42 * 0.8 + 0.09 * 0.4

= 0.336 + 0.036 = 0.372

forward2_SOLE = somma_per_SOLE * P(PAVIMENTO BAGNATO | SOLE)

= 0.372 * 0.2

= 0.0744

Per PIOGGIA al tempo 2:

somma_per_PIOGGIA =

forward1_SOLE   * P(PIOGGIA tempo2 | SOLE tempo1)

\+ forward1_PIOGGIA* P(PIOGGIA tempo2 | PIOGGIA tempo1)

= 0.42 * 0.2 + 0.09 * 0.6

= 0.084 + 0.054 = 0.138

forward2_PIOGGIA = somma_per_PIOGGIA * P(PAVIMENTO BAGNATO | PIOGGIA)

$= 0.138 * 0.6$

$= 0.0828$

2. Forward Algorithm – Esempio (parte 2)

2.1 Forward Algorithm – Esempio numerico (parte 2)

STEP 3 – Tempo 3 (osservazione = OMBRELLI APERTI)

```
Per SOLE al tempo 3:
  somma_per_SOLE_t3 =
      forward2_SOLE   * P(SOLE tempo3 | SOLE tempo2)
    + forward2_PIOGGIA* P(SOLE tempo3 | PIOGGIA tempo2)
                     = 0.0744 * 0.8 + 0.0828 * 0.4
                     = 0.05952 + 0.03312
                     = 0.09264

  forward3_SOLE = somma_per_SOLE_t3 * P(OMBRELLI APERTI | SOLE)
                = 0.09264 * 0.1
                = 0.009264

Per PIOGGIA al tempo 3:
  somma_per_PIOGGIA_t3 =
      forward2_SOLE   * P(PIOGGIA tempo3 | SOLE tempo2)
    + forward2_PIOGGIA* P(PIOGGIA tempo3 | PIOGGIA tempo2)
                       = 0.0744 * 0.2 + 0.0828 * 0.6
                       = 0.01488 + 0.04968
                       = 0.06456

  forward3_PIOGGIA = somma_per_PIOGGIA_t3 * P(OMBRELLI APERTI | PIOGGIA)
                   = 0.06456 * 0.8
                   = 0.051648
```

Probabilita' totale della sequenza di osservazioni:

prob_sequenza = forward3_SOLE + forward3_PIOGGIA
= 0.009264 + 0.051648
= 0.060912 (circa 6.1%)

Quindi vedere NUVOLE, poi PAVIMENTO BAGNATO, poi OMBRELLI APERTI in questo piccolo modello di meteo ha una probabilita' di circa 6.1%.

forward3_SOLE (blu) = 0.009264

forward3_PIOGGIA (rosso) = 0.051648

3. Viterbi Algorithm – Teoria + Esempio (parte 1)

3. Viterbi Algorithm – Teoria + Esempio

Domanda del Viterbi algorithm:

«Qual e' la sequenza di stati nascosti piu' probabile che ha generato
le osservazioni che ho visto?»

Differenza rispetto al Forward:

- il Forward somma tutte le possibilita' (tutti i percorsi);
- il Viterbi sceglie il percorso piu' probabile e lo segue.

Nel nostro esempio useremo gli stessi dati del Forward (stesse probabilita').

Definiamo:

viterbi1_SOLE = 'miglior probabilita' di essere in SOLE al tempo 1'
viterbi1_PIOGGIA= idem per PIOGGIA, e cosi' via per i tempi successivi.

STEP 1 – Inizializzazione (tempo 1, NUVOLE):

viterbi1_SOLE = P(SOLE all'inizio) * P(NUVOLE | SOLE)
= 0.7 * 0.6
= 0.42

viterbi1_PIOGGIA = P(PIOGGIA all'inizio) * P(NUVOLE | PIOGGIA)
= 0.3 * 0.3
= 0.09

STEP 2 – Tempo 2 (PAVIMENTO BAGNATO):

Per SOLE al tempo 2 consideriamo i due possibili stati al tempo 1:

- caso 1: SOLE -> SOLE:

  valore1 = viterbi1_SOLE * P(SOLE tempo2 | SOLE tempo1)

  = 0.42 * 0.8 = 0.336

- caso 2: PIOGGIA -> SOLE:

  valore2 = viterbi1_PIOGGIA * P(SOLE tempo2 | PIOGGIA tempo1)

  = 0.09 * 0.4 = 0.036

Prendiamo il massimo:

max_per_SOLE_t2 = max(0.336, 0.036) = 0.336 (viene dal percorso SOLE->SOLE).

Ora moltiplichiamo per l'osservazione:

viterbi2_SOLE = max_per_SOLE_t2 * P(PAVIMENTO BAGNATO | SOLE)

= 0.336 * 0.2

= 0.0672

Per PIOGGIA al tempo 2:

- caso 1: SOLE -> PIOGGIA:

  valore1 = 0.42 * 0.2 = 0.084

- caso 2: PIOGGIA -> PIOGGIA:

  valore2 = 0.09 * 0.6 = 0.054

max_per_PIOGGIA_t2 = max(0.084, 0.054) = 0.084 (percorso SOLE->PIOGGIA).

viterbi2_PIOGGIA = max_per_PIOGGIA_t2 * P(PAVIMENTO BAGNATO | PIOGGIA)

= 0.084 * 0.6

= 0.0504

3. Viterbi Algorithm – Esempio (parte 2)

3. Viterbi Algorithm – Esempio (parte 2)

STEP 3 – Tempo 3 (OMBRELLI APERTI):

Per SOLE al tempo 3:

- caso 1: SOLE -> SOLE -> SOLE:

  valore1 = viterbi2_SOLE * P(SOLE tempo3 | SOLE tempo2)

  = 0.0672 * 0.8 = 0.05376

- caso 2: SOLE -> PIOGGIA -> SOLE:

  valore2 = viterbi2_PIOGGIA * P(SOLE tempo3 | PIOGGIA tempo2)

  = 0.0504 * 0.4 = 0.02016

max_per_SOLE_t3 = max(0.05376, 0.02016) = 0.05376.

viterbi3_SOLE = max_per_SOLE_t3 * P(OMBRELLI APERTI | SOLE)

= 0.05376 * 0.1

= 0.005376

Per PIOGGIA al tempo 3:

- caso 1: SOLE -> SOLE -> PIOGGIA:

  valore1 = viterbi2_SOLE * P(PIOGGIA tempo3 | SOLE tempo2)

  = 0.0672 * 0.2 = 0.01344

- caso 2: SOLE -> PIOGGIA -> PIOGGIA:

  valore2 = viterbi2_PIOGGIA * P(PIOGGIA tempo3 | PIOGGIA tempo2)

  = 0.0504 * 0.6 = 0.03024

max_per_PIOGGIA_t3 = max(0.01344, 0.03024) = 0.03024.

viterbi3_PIOGGIA = max_per_PIOGGIA_t3 * P(OMBRELLI APERTI | PIOGGIA)

= 0.03024 * 0.8

= 0.024192

Ora confrontiamo i valori finali:

viterbi3_SOLE = 0.005376

viterbi3_PIOGGIA= 0.024192

Il valore piu' grande e' quello per PIOGGIA, quindi lo stato piu' probabile al tempo 3 e' PIOGGIA.

Ricostruendo i percorsi migliori passo passo (seguendo i 'max'), otteniamo:

tempo 1: SOLE

tempo 2: PIOGGIA

tempo 3: PIOGGIA

Quindi la sequenza di stati nascosti piu' probabile e':

SOLE -> PIOGGIA -> PIOGGIA.

Stato finale piu' probabile: PIOGGIA (rosso).

4. On-line Filtering – Esempio

4. On-line Filtering – Probabilita' che stia piovendo ORA

Il filtering risponde alla domanda:

«Dato tutto cio' che ho osservato finora, qual e' la probabilita' che
lo stato attuale sia SOLE o PIOGGIA?»

Usiamo i valori forward gia' calcolati ma li normalizziamo in modo che la somma tra SOLE e PIOGGIA sia uguale a 1 (cioe' 100%).

Dopo osservazione 1 (NUVOLE):

forward1_SOLE   = 0.42
forward1_PIOGGIA= 0.09
somma = 0.42 + 0.09 = 0.51

P(SOLE | NUVOLE)   = 0.42 / 0.51 ≈ 0.823 (82.3%)
P(PIOGGIA | NUVOLE)= 0.09 / 0.51 ≈ 0.177 (17.7%)

Dopo osservazione 2 (NUVOLE + PAVIMENTO BAGNATO):

forward2_SOLE   = 0.0744
forward2_PIOGGIA= 0.0828
somma = 0.0744 + 0.0828 = 0.1572

P(SOLE | prime 2 osservazioni)   = 0.0744 / 0.1572 ≈ 0.473 (47.3%)
P(PIOGGIA | prime 2 osservazioni)= 0.0828 / 0.1572 ≈ 0.527 (52.7%)

Dopo osservazione 3 (NUVOLE + PAVIMENTO BAGNATO + OMBRELLI APERTI):

forward3_SOLE   = 0.009264
forward3_PIOGGIA= 0.051648

somma = 0.009264 + 0.051648 = 0.060912

P(SOLE | tutte le osservazioni)   = 0.009264 / 0.060912 ≈ 0.152 (15.2%)

P(PIOGGIA | tutte le osservazioni)= 0.051648 / 0.060912 ≈ 0.848 (84.8%)

Interpretazione:

- all'inizio, con solo NUVOLE, siamo portati a credere che ci sia SOLE (magari nuvoloso);
- con il PAVIMENTO BAGNATO la PIOGGIA diventa piu' probabile;
- con gli OMBRELLI APERTI la PIOGGIA diventa molto probabile (84.8%).

Probabilita' finale di PIOGGIA ≈ 84.8% (rosso).

Probabilita' finale di SOLE ≈ 15.2% (blu).

5. Kalman Filter – Teoria + Esempio (parte 1)

5. Kalman Filter – Versione continua (esempio con la temperatura)

Nel Kalman Filter gli stati non sono piu' discreti (SOLE / PIOGGIA), ma continui.

Per esempio: la temperatura reale in gradi Celsius.

Idea di base:

- lo stato nascosto e' un numero reale (es. 20.0 °C);
- abbiamo un modello che dice come cambia nel tempo (es. da un minuto al successivo la temperatura non cambia molto);
- abbiamo misure rumorose (es. termometro non perfetto).

Useremo un esempio semplice:

stato x_t = temperatura reale al tempo t;

misura z_t = lettura del termometro al tempo t;

errore di modello = incertezza su come cambia davvero la temperatura;

errore di misura = incertezza dello strumento di misura.

Dati numerici:

- stima iniziale della temperatura: $x_0 = 20$ °C;
- incertezza iniziale sulla stima: $P_0 = 1$ (varianza) → errore standard ≈ 1 °C;
- il modello dice: temperatura nuova ≈ temperatura vecchia (nessun cambiamento forte);
- incertezza di modello (varianza): $Q = 1$;
- incertezza di misura (varianza): $R = 4$ (errore standard ≈ 2 °C).

Osservazione 1:

il termometro legge $z_1 = 18$ °C (un po' piu' bassa della nostra stima iniziale).

PASSO 1 – Predizione:

stima_predetta_x1 = x_0 = 20 °C (pensiamo che la temperatura non sia cambiata molto);

incertezza_predetta_P1 = P_0 + Q = 1 + 1 = 2.

PASSO 2 – Calcolo del 'Kalman gain' (quanto fidarci della misura rispetto al modello):

kalman_gain_K1 = incertezza_predetta_P1 / (incertezza_predetta_P1 + R)

= 2 / (2 + 4)

= 2 / 6 ≈ 0.333.

PASSO 3 – Aggiornamento con la misura:

differenza_misura = z_1 - stima_predetta_x1 = 18 - 20 = -2.

nuova_stima_x1 = stima_predetta_x1 + kalman_gain_K1 * differenza_misura

= 20 + 0.333 * (-2)

= 20 - 0.666

≈ 19.33 °C.

nuova_incertezza_P1 = (1 - kalman_gain_K1) * incertezza_predetta_P1

= (1 - 0.333) * 2

≈ 1.33.

5. Kalman Filter – Esempio (parte 2)

5. Kalman Filter – Esempio (parte 2)

Osservazione 2:

il termometro legge z_2 = 22 °C (piu' alta di prima).

PASSO 1 – Predizione al tempo 2:

stima_predetta_x2 = nuova_stima_x1 ≈ 19.33 °C;

incertezza_predetta_P2 = nuova_incertezza_P1 + Q

≈ 1.33 + 1

≈ 2.33.

PASSO 2 – Nuovo Kalman gain:

kalman_gain_K2 = incertezza_predetta_P2 / (incertezza_predetta_P2 + R)

= 2.33 / (2.33 + 4)

≈ 2.33 / 6.33

≈ 0.368.

PASSO 3 – Aggiornamento con la nuova misura:

differenza_misura2 = z_2 - stima_predetta_x2

= 22 - 19.33

≈ 2.67.

nuova_stima_x2 = stima_predetta_x2 + kalman_gain_K2 * differenza_misura2

≈ 19.33 + 0.368 * 2.67

≈ 19.33 + 0.983

≈ 20.31 °C.

Il risultato del Kalman Filter e' una stima 'filtrata' della temperatura:

- tiene conto sia del modello (la temperatura non salta troppo);
- sia delle misure (che pero' sono rumorose).

La stima finale (20.31 °C) e' un compromesso intelligente tra
la predizione precedente (circa 19.33 °C) e la nuova misura (22 °C).

6. Riassunto finale

6. Riassunto finale – Cosa devo ricordare

• HMM (Hidden Markov Model):
  - ci sono stati nascosti (SOLE, PIOGGIA) che non vedo direttamente;
  - vedo solo osservazioni (NUVOLE, PAVIMENTO BAGNATO, OMBRELLI APERTI);
  - conosco le probabilita' di transizione tra stati e le probabilita' delle osservazioni.

• Forward algorithm:
  - calcola la probabilita' complessiva di una sequenza di osservazioni;
  - per ogni tempo e ogni stato somma su tutti i modi per arrivarci;
  - alla fine somma sulle probabilita' degli stati finali.

• Viterbi algorithm:
  - cerca la singola sequenza di stati nascosti piu' probabile;
  - usa 'max' invece della somma e ricorda da quale stato viene il percorso migliore.

• On-line filtering:
  - dice quanto e' probabile che lo stato attuale sia SOLE o PIOGGIA;
  - usa i valori forward ma li normalizza per farli sommare a 1.

• Kalman Filter:
  - e' una versione continua (stato = numero reale, es. temperatura);
  - combina predizione del modello e misura rumorosa;
  - riduce progressivamente l'incertezza sulla stima.

Suggerimento per lo studio:
- prova a rifare i conti dell'esempio su carta;
- per ogni passaggio scrivi una riga del tipo:

'nuovo_valore = cosa sto combinando * perche' lo sto moltiplicando',

- se ti perdi, torna a chiederti sempre: 'che domanda sta rispondendo questo algoritmo?'.